



UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET



Aleksandar Sošić

**RAZVOJ VEB PLATFORME I MOBILNIH APLIKACIJA
ZA POTREBE DIGITALIZACIJE I ANALIZE
PODATAKA O INTENZITETU EROZIJE ZEMLJIŠTA**

MAGISTARSKI RAD

Podgorica, 2021. godine



UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET



Aleksandar Sošić

**RAZVOJ VEB PLATFORME I MOBILNIH APLIKACIJA
ZA POTREBE DIGITALIZACIJE I ANALIZE
PODATAKA O INTENZITETU EROZIJE ZEMLJIŠTA**

MAGISTARSKI RAD

Podgorica, 2021. godine

PODACI I INFORMACIJE O MAGISTRANDU

Ime i prezime: Aleksandar Sošić

Datum i mjesto rođenja: 07.09.1990. godine, Bijelo Polje, Crna Gora

Naziv završenog osnovnog studijskog programa i godina završetka studija: Studijski program Primijenjeno računarstvo, Elektrotehnički fakultet, 2012. godine.

INFORMACIJE O MAGISTARSKOM RADU

Naziv postdiplomskih studija: Postdiplomske magistarske studije primijenjenog računarstva

Naslov rada: Razvoj veb platforme i mobilnih aplikacija za potrebe digitalizacije i analize podataka o intenzitetu erozije zemljišta

Fakultet na kojem je rad odbranjen: Elektrotehnički fakultet

UDK, OCJENA I ODBRANA MAGISTARSKOG RADA

Datum prijave magistarskog rada: 23.01.2019. godine

Datum sjednice Vijeća na kojoj je prihvaćena tema: 25.02.2019. godine

Komisija za ocjenu teme i podobnosti magistranda: Prof. dr Nikola Žarić
Doc. dr Velibor Spalević
Prof. dr Slobodan Đukanović

Mentor: Prof. dr Nikola Žarić

Komisija za ocjenu rada: Prof. dr Nikola Žarić
Doc. dr Velibor Spalević
Prof. dr Slobodan Đukanović

Komisija za odbranu rada: Prof. dr Nikola Žarić
Doc. dr Velibor Spalević
Prof. dr Slobodan Đukanović

Lektor: Prof. dr Nikola Žarić

Datum odbrane: _____.____.2021. godine

Datum promocije: _____.____.2021. godine

SADRŽAJ

PREDGOVOR	i
IZVOD RADA	ii
ABSTRACT	iii
1. UVOD	1
1.1 Motiv	2
1.2 Doprinos	2
1.3 Struktura rada	3
2. Analiza metoda za proračun erozije zemljišta.....	4
2.1 Surface and distance measuring (1999)	4
2.2 River basins (2000)	4
2.3 IntErO (2011)	5
3. Razvoj WIntErO veb platforme.....	7
3.1 Tehnologije korištene za izradu veb platforme	8
3.1.1 HTML5 (HyperText Markup Language).....	8
3.1.2 CSS3 (Cascading Style Sheets)	9
3.1.3 PHP 7 (Hypertext Preprocessor).....	10
3.1.4 MySQL (My Structured Query Language).....	10
3.1.5 JavaScript.....	11
3.1.6 jQuery	12
3.1.7 Bootstrap 4.....	12
3.1.8 Microsoft Visual Studio Community 2019.....	13
3.1.9 cPanel.....	13

3.2	Analiza i opis projektovanog rješenja	14
3.2.1	Analiza zahtjeva.....	14
3.2.2	Klijent-server arhitektura	18
3.2.3	Informaciona arhitektura.....	19
3.3	Implementacija veb platforme.....	21
3.3.1	Izrada baze podataka za skladištenje proračuna i podataka o korisnicima	21
3.3.2	Razvoj korisničkog okruženja.....	25
3.3.3	Razvoj javno-dostupnog okruženja.....	42
3.3.4	Razvoj prilagodljivog dizajna za različite uređaje.....	50
4.	Razvoj aplikacija za pametne uređaje.....	54
4.1	Tehnologije korišćene za izradu mobilnih aplikacija.....	54
4.1.1	Xamarin platforma	55
4.1.2	C# programski jezik.....	55
4.1.3	.NET Framework platforma.....	56
4.2	Analiza i opis projektovanog rješenja	57
4.3	Izrada i upotreba mobilnih aplikacija.....	58
4.3.1	Postupak izrade stranice koja prikazuje parametre i rezultate	60
4.3.2	Izrada sekcije koja omogućava zabilježbu novih parametara.....	62
5.	Analiza i diskusija.....	65
6.	Zaključak.....	68
7.	Literatura	70
8.	Dodatak A.....	74
9.	Dodatak B.....	81
10.	Dodatak C.....	85

POPIS SLIKA

- Slika 1: Izgled radnog okruženja „IntErO“ paketa
Slika 2: Dijagram slučajeve korišćenja WIntErO platforme
Slika 3: Prikaz elemenata troslojne arhitekture klijent-server sistema
Slika 4: Prikaz primarne informacione arhitekture WIntErO veb platforme
Slika 5: Prikaz strukture tabele *data*
Slika 6: Prikaz strukture tabele *register*
Slika 7: Prikaz forme za kreiranje naloga
Slika 8: Dio PHP koda stranice za kreiranje naloga
Slika 9: Prikaz menija (a, b i c slike, respektivno)
Slika 10: Prikaz stranice *Profile*
Slika 11: Izgled stranice za unos novih podataka
Slika 12: Izgled dijela stranice za nove unose, element *accordion*
Slika 13: Izgled stranice koja se odnosi na pregled rezultata prije slanja
Slika 14: Prikaz sadržaja *.sli* fajla koji je otvoren pomoću *Notepad* tekstualnog editora
Slika 15: Prikaz stranice koja služi za očitavanje *.sli* datoteka
Slika 16: Prikaz stranice *On Hold*
Slika 17: Dio programskog koda stranice za prikazivanje sadržaja koji je na čekanju
Slika 18: Prikaz glavnog menija i *Google* mape sa tagovima
Slika 19: Prikaz *Google* mape sa tagovima za područje Irana (48 slivova)
Slika 20: Prikaz tabele za sortiranje podataka po državama
Slika 21: Prikaz programskog koda kojim se kreira tabela za sortiranje podataka po državama
Slika 22: Izgled javno-dostupne stranice koja prikazuje parametre rječnog sliva
Slika 23: Izgled javno-dostupne stranice koja prikazuje rezultate rječnog sliva
Slika 24: PHP kôd kojim se definiše formatiranje *.doc* i *.xls* dokumenata pomoću metoda HTTP
Slika 25: PHP kôd kojim se definiše formatiranje PDF dokumenta upotrebom klase FPDF
Slika 26: Prikaz HTML i CSS kodova koji su sastavni dio prilagodljivog dizajna
Slika 27: Snimci ekrana na kojima su prikazane stranice platforme koje su kompatibilne sa rezolucijama koje su zastupljene na mobilnim uređajima
Slika 28: Snimci ekrana na kojima su prikazane stranice platforme koje su kompatibilne sa rezolucijama koje su zastupljene na tablet uređajima
Slika 29: Izgled razvojnog okruženja *Microsoft Visual Studio Community 2019* sa pokrenutim Android emulatorom
Slika 30: Prikaz C# kodova koji su sastavni dio segmenta za prikaz parametara i rezultata
Slika 31: Snimci ekrana iOS i Android uređaja, segment koji prikazuje parametre i rezultate
Slika 32: Prikaz C# kodova koji su sastavni dio sekcije za zabilježbu novih parametara
Slika 33: Snimci ekrana iOS i Android uređaja, prikaz segmenta *notes – parameters*

POPIS TABELA

- Tabela 1: Srednje vrijednosti koeficijenta (Y)
Tabela 2: Srednje vrijednosti koeficijenta (Xa)
Tabela 3: Srednje vrijednosti koeficijenta (φ)
Tabela 4: Intenzitet erozije zemljišta sračunat korišćenjem WIntErO platforme

SKRAĆENICE

API	Application Programming Interface
CDM	Content Delivery Network
CERN	European Organization for Nuclear Research
CLR	Common Language Runtime
CSS	Cascading Style Sheets
CTS	Common Type System
DOM	Document Object Model
FTP	File Transfer Protocol
GIS	Geographic Information System
GPS	Global Positioning System
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IETF	Internet Engineer Task Force
IntErO	Intensity of Erosion and Outflow
JSON	JavaScript Object Notation
MIT	Massachusetts Institute of Technology
MySQL	My Structured Query Language
PHP	Hypertext Preprocessor
RDW	Responsive Web Design
REST	Representational State Transfer
SQL	Structured Query Language
SVG	Scalable Vector Graphics
UWP	Universal Windows Platform
W3C	World Wide Web Consortium
WIntErO	Web Application for Intensity of Erosion and Outflow
WWW	World Wide Web
XML	Extensible Markup Language

PREDGOVOR

Zahvaljujem se profesoru Elektrotehničkog fakulteta Univerziteta Crne Gore, Nikoli Žariću, mentoru, na izdvojenom vremenu i velikoj podršci koju mi je pružio tokom izrade ovog magistarskog rada. Takođe, zahvaljujem se profesoru Biotehničkog fakulteta Univerziteta Crne Gore, Veliboru Spaleviću na odličnim idejama, savjetima i ukazanim smjericama.

Aleksandar Sošić, spec.app

IZVOD RADA

Uticaj informaciono-komunikacionih tehnologija na sve veći broj različitih oblasti u svim aspektima modernog doba je veoma evidentan. Primjena računarskih nauka u sferama geonauke je započeta još devedesetih godina prošlog vijeka. Eksponencijalnim razvojem Interneta, i sami progres geonauke je mnogo lakše ostvarljiv. Prednosti su nemjerljive kada su u pitanju dostupnost informacija i efikasna komunikacija. Stoga, ovaj rad pokazuje važnost primjene informaciono-komunikacionih tehnologija u oblasti geonauke, koja se bavi procesima intenziteta erozije zemljišta, kroz kreiranje veb platforme i mobilnih aplikacija.

U tom kontekstu, sprovedena su neophodna istraživanja i urađene su analize i projekcije. Cilj je bio da se što uspješnije kreiraju pomenuta rješenja, i samim tim se doprinese efikasnijem razvoju geonauke. Implementirana su softverska rješenja čija je glavna svrha obezbjeđivanje digitalizacije i analize podataka o intenzitetu erozije zemljišta. Dat je detaljan opis kompletnog razvoja projektovanih rješenja kroz opširan prikaz dizajnerskog i programerskog razvoja koji sa sobom donosi fleksibilnost i preciznost u radu, a takođe i garanciju postizanja ciljeva koje zahtijevaju stručnjaci. Prikazan je i postupak upotrebe rješenja koja potvrđuju uspješno dinamično funkcionisanje i laku primjenjivost.

Uspostavljenim rješenjima došlo se do utvrđivanja i opravdanosti potrebe primjene savremenih tehnologija u geonauci. Očegledno je da je stalna dostupnost podataka na Internetu od velike koristi za naučnu zajednicu koja se bavi veoma bitnim problemom koji se tiče smanjenja intenziteta erozije zemljišta kod svih rječnih slivova kojih je veoma puno na našoj planeti. Takođe, veoma je korisna i primjena najnovijih markerskih i programskih jezika i drugih računarskih tehnologija i alata, jer se obezbjeđuju najsavremenija softverska rješenja koja su i veoma inovativna.

Glavni doprinosu ovog rada mogu se ogledati u primjeni ovih novih rješenja koja su naprednija u odnosu na prethodno kreirane programske pakete, samim tim jer omogućavaju niz novih progresivnih funkcionalnosti. Kreiran je jedinstveni sistem skladištenja podataka i potpuna digitalizacija ranije dostupnih podataka. Kreirana rješenja su veoma modularna i mogu da se nadograđuju na lak i efikasan način. Kao dodatni doprinos, ovaj rad otvara i veliki broj pitanja na kojima se mogu bazirati i dalja istraživanja.

Ključne riječi: Informaciono-komunikacione tehnologije, veb platforma, mobilne aplikacije, digitalizacija i analiza podataka, intenzitet erozije zemljišta, geonauka

ABSTRACT

The impact of information and communication technologies to an increasing number of different areas in all aspects of the modern age is very obvious. Application of computer sciences in the spheres of geoscience has started back in the 1990s. With the exponential development of the Internet, the progress of geoscience itself is much more to achievable. The benefits are immeasurable when it comes to information availability and effective communication. Therefore, this paper illustrates the importance of the application of information and communication technologies in the field of geoscience, which deals with the processes of soil erosion intensity, through the creation of a web platform and mobile applications.

In that context, the necessary research has been carried out and analysis and projections were made. The objective was to create the above solutions as successfully as possible, and thus contribute to the more efficient development of geoscience. Software solutions whose main purpose is to provide digitalization and analysis of data on the intensity of soil erosion are implemented. A detailed description of the complete development of solutions is provided, through a comprehensive overview of the design and software development that brings flexibility and accuracy in operation and is a guarantee of achieving goals required by the experts. Procedure of using solutions that confirm successful dynamic functioning and easy applicability is also demonstrated.

The develop solutions let to identification and justifiability of the need applying contemporary technologies in geoscience. It is obvious that the constant availability of data on the Internet is of great benefit to the scientific community, which deals with a very important problem of the reduction of soil erosion in all river basins, of which there are many on our planet. Also, the use of the latest marker and programming languages and other computer technologies and tools is very useful, because the most modern, innovative software solutions are provided.

The main contribution of this paper can be reflected in the application of these new – in comparison to the earlier versions – more advanced software packages, enabling a number of new progressive functionalities. An unique data storage system has been developed and full digitization of previously available data has been achieved. These new solutions are very modular and can be upgraded in an easy and efficient way. As an additional contribution, this paper tackles a large number of questions that can serve as basis for further research.

Keywords: Information and communication technologies, web platform, mobile applications, digitization and data analysis, intensity of soil erosion, geoscience

1. UVOD

Informacioni sistemi kakve danas poznajemo počeli su da se razvijaju sedamdesetih godina prošlog vijeka upotrebom prvih personalnih računara i razvojem računarskih mreža. Međusobno povezivanje računara dovelo je do stvaranja globalnog sistema konektovanih mreža, Interneta, koji je postao uobičajena navika koja nam je svima neophodna. Najznačajnija usluga koju pruža Internet i koja je najviše doprinijela njegovom razvoju je veb usluga, poznatija kao WWW (World Wide Web). Veb je razvijen u istraživačkom centru „Evropska organizacija za nuklearno istraživanje“, koji je poznatiji kao CERN (European Organization for Nuclear Research), od strane engleskog naučnika Tim Berlers-Li-a i zvanično je predstavljen 6. avgusta 1991. godine [1]. Od tada počinje konstantno unaprijeđivanje markerskih i raznih programskih jezika i drugih kompjuterskih alata koji omogućavaju programerima da doprinose razvoju veba što svakako utiče i na napredak naše civilizacije. Danas uopšteno, enorman napredak razvoja računarskih nauka i informaciono-komunikacionih tehnologija ima veoma dominantnu ulogu u svim sferama savremenog društva.

Cilj ovoga rada je da se izradom WIntErO (Web Application for Intensity of Erosion and Outflow) veb platforme i aplikacija za pametne uređaje predstavi važnost primjene računarskih nauka u sferi geonauke, kao i da se detaljno istraže i prikažu računarske tehnike i tehnologije potrebne za izradu, predstavi postupak same izrade, a zatim i analizira razvijena veb platforma i aplikacije čijim kreiranjem se teži da se na što jednostavniji način izvršavaju analize procesa erozije zemljišta. U radu je predstavljen problem digitalizacije, obrade, filtriranja i pripreme podataka kroz programerska rješenja koja će učiniti platformu interaktivnom, a sve u namjeri da se stručnjacima iz oblasti geonauke omogući brz, precizan i jednostavan rad. WIntErO platforma je naslednik i predstavlja unaprijeđenu verziju programskog paketa IntErO (Intensity of Erosion and Outflow) koji će se detaljnije razmotriti u narednom poglavlju kada će se analizirati dosadašnje metode koje su korištene za proračune erozije zemljišta. Glavna poboljšanja novog rješenja ogledaće se u omogućavanju jednostavne digitalizacije i objedinjavanja proračuna, obezbjeđivanju stalne dostupnosti istih i stvaranju boljeg komfora u radu kako bi se postigao što učinkovitiji rezultat u raznim sferama ispitivanja oblasti za koju je rješenje i namijenjeno. Pored navedenog, sama platforma je razvijana tako da dodavanje novih funkcionalnosti bude lako čime bi se budući specifični zahtjevi implementirali na djelotvoran način, a programski kôd je pisan na način koji je

lak za razumijevanje i održavanje, pa samim tim i za dalja buduća proširenja. U narednim poglavljima govoriće se detaljno i o računarskim tehnologijama i prikazati programski kodovi za određene djelove veb platforme i aplikacija, čime će se dati opis koji će omogućiti potpuno razumijevanje postupka izrade i načina upotrebe.

1.1 Motiv

Trenutni načini ispitivanja intenziteta erozije zemljišta koji su u primjeni nijesu adekvatni da na pravi način odgovore na stvarne zahtjeve sa kojima se susrijeću stručnjaci, a koji su veoma kompleksni. Postoji nekoliko načina ispitivanja koja će se detaljnije obraditi u narednom poglavlju, a najefikasnije već pomenuto rješenje je programski paket IntErO koji radi lokalno na jednom računaru i to samo sa instaliranim operativnim sistemom *Microsoft Windows*. Nedostupnost na različitim platformama i ograničene mogućnosti programskog paketa IntErO čine ga neadekvatnim za upotrebu u današnjem i budućem vremenu. Prikupljanje podataka o eroziji zemljišta, kao i manipulacija tim podacima nijesu ni malo jednostavan zadatak ukoliko se ne primijenuje efikasno rješenje, pa čak ni za profesionalce sa velikim iskustvom. Motiv ovog rada je baziran na potrebi za objedinjavanjem i unaprijeđivanjem do sada korištenih načina kreiranjem novog međuplatformskog programerskog rješenja koje će znatno unaprijediti rad stručnjaka iz pomenute oblasti. Ideja rješenja ogleda se u kreiranju veb platforme i aplikacija za pametne uređaje, čime će se u potpunosti zamijeniti svi dosadašnji načini rada i samim tim stvoriti progres u geonauci primjenom računarsko-geografskih metodologija. Realizacija ovog rada mogla bi da omogući da rezultati dobijeni korišćenjem WIntErO platforme, za bilo koja područja slivova širom svijeta, budu od pomoći u utvrđivanju trenutnog stanja produkcije nanosa, kao i predloga načina korišćenja zemljišta koje će u budućnosti uticati na smanjenje intenziteta erozije u proučavanim slivovima. Pronalaženjem optimalne strukture načina korišćenja zemljišta za predmetne slivove uvažavaće se biološko-ekološke zakonitosti i principi očuvanja životne sredine. Između ostalog, ovaj rad može da otvori i mogućnosti za dalji progres u geonaukama primjenom informaciono-komunikacionih tehnologija.

1.2 Doprinos

Korišćenjem naprednih markerskih i programskih jezika i računarskih alata za razvoj veb platformi i mobilnih aplikacija razviće se platforma WIntErO koja će u potpunosti da obezbijedi digitalizaciju i verifikaciju svih istraživačkih podataka o intenzitetu erozije zemljišta kao i da omogući jedinstven i „otvoren“ globalni pristup istim, uz stalnu dostupnost naučnoj zajednici i donosiocima odluka, te brži i detaljniji način izvršavanja potrebnih analiza. Doprinos ovog rada može se ogledati u uspješnom uspostavljanju funkcionisanja WIntErO platforme, čime će u mnogome biti vidljivo olakšanje i bolja djelotvornost u radu kako stručnjaka tako i same naučne zajednice. Samim tim, pretpostavka je da će ovo rješenje doprinijeti i cjelokupnom poboljšanju

rada istraživača kako bi se postiglo smanjenje i ublažavanje erozionog procesa i postigli bolji rezultati u drugim bitnim zonama ispitivanja zemljišta i slivova u odnosu na ranije načine rada.

Takođe, namjera je i da upotreba platforme bude široko rasprostranjena i korištena od strane što više pojedinaca i organizacija. Oblasti na kojima rezultati dobijeni upotrebom platforme mogu biti primjenjivani su brojne i različite. Neki od bitnijih područja primjene su u vodoprivrednim organizacijama koje se bave gazdovanjem u slivovima, kod različitih kompleksnih istraživanja iz oblasti erozije zemljišta, kao i kod upotrebe prilikom kreiranja primjera potencijalnog integralnog gazdovanja u slivovima u zemlji i u svijetu.

1.3 Struktura rada

Nastavak rada je ograničavan na sledeći način:

- Poglavlje 2: analizirane su tri metode za proračun erozije zemljišta koje su korištene do sada, a na kojima je WIntErO platforma bazirana;
- Poglavlje 3: dat je detaljan prikaz tehnika, algoritama i metoda koje su korištene za razvoj WIntErO veb platforme. Detaljno su opisani i načini upotrebe platforme, sistemi identifikacije korisnika sa različitim pravima pristupa i kontrole, geomapiranje lokacija na kojima je izvršeno mjerenje intenziteta erozije, moduli za unos, prikaz i modifikaciju podataka, odgovarajuće baze podataka i njihovo povezivanje, kao i alat za digitalizaciju i eksportovanje podataka u odgovarajuće formate radi dalje analize i obrade;
- Poglavlje 4: dat je pregled tehnika, algoritama i metoda koje su korištene za izradu aplikacija za pametne uređaje. Prikazani su i načini povezivanja aplikacija sa jedinstvenom bazom podataka koju koristi i WIntErO veb platforma;
- Poglavlje 5: opisani su i analizirani dobijeni rezultati, a dat je opis i diskutovano je o uspešnosti metoda koje su primijenjene kako bi se testirala uspešnost platforme realizovane u ovom magistarskom radu;
- Poglavlje 6: na kraju je razmatrano i diskutovano o uspostavljenj platformi WIntErO, kao i o budućem razvoju.

2. Analiza metoda za proračun erozije zemljišta

Do sada je za potrebe proučavanja proračuna intenziteta erozije zemljišta u upotrebi bilo nekoliko rješenja, a to su: *Surface and distance measuring* (1999) [2], program *River basins* (2000) [3], kao i softverski paket IntErO (2011) [4]. Glavno svojstvo prvog rješenja bilo je da predstavi površine poligona i rastojanja između željenih makro i mikro objekata, a preostala dva rješenja su trebala da obrađuju ulazne podatke, a zatim i da predstave rezultate o intenzitetu erozije zemljišta, s tim što je posljednje, treće rješenje, pored ove obrade podataka u sebi imalo i integrisane mogućnosti iz prvog rješenja. U upotrebi postoji veliki broj matematičkih modela za proračun, a ova rješenja, kao i nova WIntErO platforma, koriste analitički model prof. Gavrilovića (1972) [5]. U nastavku će se detaljnije govoriti o ovim računarskim programima.

2.1 Surface and distance measuring (1999)

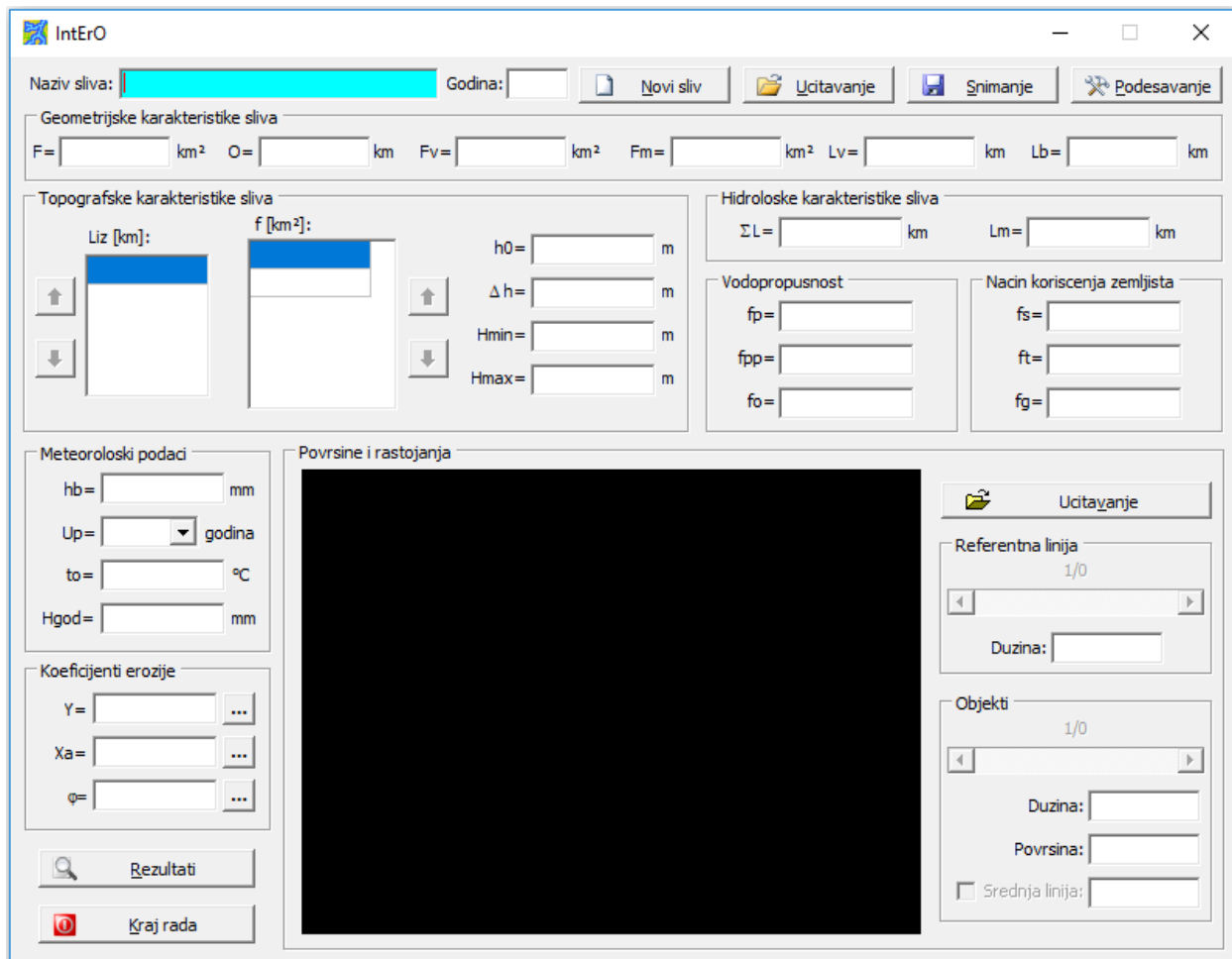
Program *Surface and distance measuring* [2] napravljen je sa ciljem kako bi se unaprijedilo i ubrzalo mjerenje vrijednosti površina poligona analitičkim metodama i takođe postigla veća preciznost dobijenih rezultata u odnosu na način koji je do tada primjenjivan korišćenjem mehaničkih instrumenata planimetra i kurvimetra. Prema ranijim istraživanjima utvrđeno je da je ovakav unaprijeđen način dobijanja rezultata u odnosu na prethodno korištene načine znatno poboljšao rad stručnjacima i otvorio mogućnosti za neka nova unaprijeđivanja koja su dovela i do ideje za razvoj WIntErO platforme. Program je pisan u *Borland Delphi* razvojnom okruženju za *Microsoft Windows* operativne sisteme, kreiran je kao vrlo jednostavan za upotrebu i nije zahtijevao visok nivo znanja rada na računaru pa je za vrlo kratko vrijeme i našao primjenu u oblasti geonauke.

2.2 River basins (2000)

Program *River basins* [3] urađen je sa namjerom da obrađuje proračune dobijene terenskim istraživanjima, koristi matematički model prof. Gavrilovića [5]. Uspostavljanjem algoritma na osnovu ovog matematičkog modela, kreiralo se programersko rješenje koje je u odnosu na klasičan način rada povećalo brzinu dobijanja rezultata i do 1000 puta. Pored prednosti povećanja brzine, program je osmišljen tako da posjeduje mogućnost otkrivanja logičkih grešaka nakon popunjavanja forme sa proračunima. Ovo posebno ima značaja pošto bi se kod klasičnog proračuna mogli dobiti neki rezultati, a prilikom korišćenja ovog programa javila bi se greška i ne bi bilo moguće dobiti rezultate ukoliko se radi o nerealnim unosima. Dakle, ovaj program sadrži formu za unos parametara dobijenih terenskim ili nekim sličnim istraživanjima, mogućnost otkrivanja logičkih grešaka, snimanje unešenih parametara na lokalnom računaru, kao i pregled rezultata.

2.3 IntErO (2011)

Programski paket IntErO [4] predstavlja unaprijeđenu verziju programa *River basins*, a takođe u sebi sadrži i integrisanu unaprijeđenu verziju programa *Surface and distance measuring*. Program je kao i prethodna dva pomenuta pisan u *Borland Delphi* razvojnom okruženju za *Microsoft Windows* operativne sisteme. Mogućnosti programa su znatno unaprijeđene. Program je dvojezičan, pa je tako prilikom rada sa njim moguće izabrati engleski ili srpski jezik za korisnički interfejs i izvještaje, tj. rezultate. Program kao i prethodna verzija ima mogućnosti otkrivanja logičkih grešaka, a dodate su i provjere parametara za proračun koeficijenata i opciona mogućnost automatske preraspodjele procenata po elementima do 100%. Glavni dio programa sadrži ukupno 26 polja za unos i prikazuje ukupno 21 rezultat, dok potprogram za mjerenje površina i rastojanja ima 3 polja za unos i jedno polje za pregled rezultata. Izgled radnog okruženja IntErO paketa prikazan je na slici 1.



Slika 1: Izgled radnog okruženja „IntErO“ paketa

Snimanje parametara i podataka koji se koriste za dobijanje proračuna vrši se upisivanjem istih u *.sli* i *.plt* datoteke, što omogućava dosta bolje praćenje promjena procesa intenziteta erozije

zemljišta na slivovima. Program IntErO može da se koristi i u skraćenoj verziji, korisnik je u mogućnosti da odabere one rezultate koje želi da mu se prikažu. Kod ove skraćene verzije program bira samo one unose koji su mu potrebni za dobijanje izabranih rezultata, tako da ukoliko korisnik ne unese nepotrebne parametre neće dobiti upozorenje. Takođe, postoje i dva opciona unosa koja služe za opis, a koje je poželjno popuniti zbog bolje sistematizacije fajlova. Pored navedenog, postoje i opcije za eksportovanje dobijenih rezultata sa opcionim podešavanjem detaljnog izvještaja čime se stvara mogućnost i eksportovanja kombinacije ulaznih podataka, rezultata i komentara.

Kod dobijanja vrijednosti površina poligona i rastojanja između objekata princip računanja se svodi na izračunavanje po formuli koja je implementirana u programski kôd:

$$P = \left| \sum_{i=1}^n x_i y_{i-1} - x_{i-1} y_i \right| \quad (1)$$

gdje su: n – broj tjemena poligona, a x_i, y_i – koordinate tjemena poligona.

Vrijednosti površina i dužina, kao i svih ostalih rezultata, koje se dobijaju korišćenjem IntErO paketa su identične sa onima koje su vršene analitičkim putem, a samo očitavanje i obračunavanje se ostvaruje veoma brzo.

3. Razvoj WIntErO veb platforme

Istraživanja sprovedena tokom izrade ovog magistarskog rada omogućila su prikupljanje naučnih saznanja i analiziranje računarskih tehnika neophodnih za uspješnu realizaciju zamišljenog rješenja.

Na početku istraživanja bilo je važno fokusirati se na pregled stanja geonauke kako bi se došlo do odgovora koji se tiču potrebe budućeg razvoja, pa se tako i došlo do bitnih činjenica koje i opravdavaju sami razvoj. Istraživanjima je dokazano da je proces erozije zemljišta u prirodi česta pojava koja se javlja pod raznim klimatskim uticajima i ujedno je najrasprostranjeniji oblik degradacije zemljišta širom svijeta što predstavlja globalno-ekološki problem koji ugrožava poljoprivredu, kvalitet vode, infrastrukturu i drugo [6]. Stope održivosti zemljišta kada su u pitanju gubici od erozionih procesa su veoma nepovoljne na svim kontinentima. U Sjedinjenim Američkim Državama je uništeno ili teško oštećeno preko 120 miliona hektara kvalitetnog zemljišta [7], procjena je da će biti uništeno preko 140 miliona u Africi i Aziji ukoliko se ne usvoje napredniji načini upotrebe zemljišta [8], a takođe i u ostalim krajevima svijeta stanje nije znatno povoljnije [9], [10].

Upotreba trenutno aktuelnih načina za proučavanje zemljišta [2], [3], [4] ne može efikasno pomoći pri rješavanju problema kao što su rješavanje pojačanog procesa erozije koje vremenom napravi zemljište neupotrebljivim ili negativan uticaj koji erozija ima na vodu za piće, proizvodnju hrane, poplave, i smanjenje emisije ugljenika. Zbog navedenih problema, i pošto je erozija zemljišta važan socio-ekonomski problem i pošto povećanje sveukupne javne svijesti igra veoma bitnu ulogu u procesu smanjivanja negativnih uticaja erozije [10], došlo se na ideju za kreiranjem jedinstvenog rješenja u vidu kreiranja veb platforme [11], [12], [13] i aplikacija za pametne uređaje kako bi se izvršavao monitoring zemljišta sa prikazom opširnih informacija i omogućio uvid svim zainteresovanim u iste.

U današnjem vremenu, progres nauke bez upotrebe Interneta je veoma teško ostvarljiv, a naučna istraživanja u mnogome zavise od efikasne saradnje i komunikacije što je i presudno za tempo naučnog napretka. Internet omogućava da se podaci prezentuju i analiziraju na različite načine što u velikoj mjeri povećava vrijednost tih podataka i potencijal za dobijanje korisnih analiza. Na osnovu najnovijih istraživanja u oblasti geonauke, Internet i tehničke metode mogu pomoći u rješavanju teških i multidisciplinskih izazova, novim načinima prikupljanja podataka i inovativnim metodama i modeliranjem [14]. U nastavku ovog poglavlja detaljnije će se diskutovati o WIntErO veb platformi, a u narednom će se govoriti o WIntErO aplikacijama za mobilne uređaje. Poglavlje je podijeljeno na sljedeće cjeline:

- tehnologije korištene za izradu veb platforme;
- analiza i opis projektovanog rješenja;
- implementacija veb platforme.

3.1 Tehnologije korištene za izradu veb platforme

Platforma WIntErO je veb aplikacija koja je kreirana upotrebom savremenih markerskih i programskih jezika i drugih računarskih tehnologija koji su široko rasprostranjeni, dokazano praktični i upotrebljavaju se svakodnevno u širokom opsegu od strane velikog broja stručnjaka za razvoj najsavremenijih sistema. Za izradu veb platforme korištene su sledeće tehnologije i alati:

- HTML5 (HyperText Markup Language),
- CSS3 (Cascading Style Sheets),
- PHP 7 (Hypertext Preprocessor),
- MySQL (My Structured Query Language),
- JavaScript,
- jQuery,
- Bootstrap 4,
- Microsoft Visual Studio Community 2019,
- cPanel.

3.1.1 HTML5 (HyperText Markup Language)

HTML5 je markerski jezik čija je osnovna namjena da se koristi za kreiranje struktura veb stranica, mada njegova primjena može biti i daleko veća pošto je njegovom upotrebom moguće kreirati i aplikacije za pametne uređaje, razne softverske okrive (engl. *software frameworks*¹) koji se koriste za automatizaciju i testiranje aplikacija i drugo. Ove strukture sadrže elemente kao što su zaglavlja, stubci, naslovi, tabele i slično. HTML5 je trenutno poslednja verzija HTML jezika koja je odobrena od strane W3C (World Wide Web Consortium) međunarodne organizacije koja je zadužena za izradu standarda Interneta. Ovom poslednjom verzijom ovaj jezik je dobio i neke druge funkcije pored funkcija markiranja hiperteksta². Dodati su novi elementi za pozicioniranje elemenata, omogućen je rad sa multimedijalnim audio, video, *canvas*³ i SVG (Scalable Vector Graphics) elementima, modifikovani su i dodati tipovi unosa i dodati novi atributi za unose, a dodata je i opcija geolociranja koja omogućuje korisnicima koji nemaju GPS (Global Positioning System) da koriste mogućnosti navigacije i lociranja [15].

Istorija HTML jezika počela je krajem osamdesetih godina prošlog vijeka projektovanjem prvih sistema za dijeljenje dokumenata između istraživača u CERN-u, takođe od strane britanskog naučnika Tim Berners-Li-a koji je i tvorac veba. Početkom devedesetih godina izdata je specifikacija koja je sadržala svega dvadesetak elemenata, ali projekat nije zvanično usvojen sve

¹ *Software framework* – predstavlja programski okvir koji je moguće selektivno mijenjati i samim tim dodavati određene funkcionalnosti nekoj većoj softverskoj platformi olakšavajući tako efikasan razvoj iste.

² Hipertekst (engl. *Hypertext*) – je tekst koji se prikazuje na elektronskom uređaju koji u sebi sadrži referentne linkove koji korisnika neposredno mogu da vode do nekog „drugog“ teksta.

³ *Canvas* - element koji omogućuje postavljanje dinamičkih grafičkih elemenata na veb stranicama.

do 1994. godine kada se desilo osnivanje W3C-a na Institutu za tehnologiju u Masačusetsu (engl. MIT - Massachusetts Institute of Technology). Već krajem 1995. godine izdata je specifikacija HTML jezika od strane IETF (Internet Engineer Task Force) organizacije koja je također zadužena za izradu standarda i protokola interneta. Od 1991. godine do 1999. godine HTML je razvijan od verzije 1 (HTML 1.0) do verzije 4 (HTML 4.0). HTML je postao internacionalni standard 2000. godine. 2008. godine izdat je i zvanično HTML5, a trenutno je i dalje u razvoju [16].

3.1.2 CSS3 (Cascading Style Sheets)

CSS3 je stilski ili opisni jezik pomoću koga se saopštava veb čitaču ili nekom drugom korisničkom agentu na koji način treba da prikazuje sadržaj. Dakle, CSS jezik obezbjeđuje sve vrste informacija o izgledu sadržaja. U ranim danima razvoja veba kada još uvijek nije bio razvijen CSS, HTML je imao zadatak da definiše izgled stranica, pored toga što je obavljao funkcije struktuiranja sadržaja. Vremenom, kako se HTML jezik razvijao, rasle su i potrebe za njegovu pojednostavljivanje i održavanje. Postojali su predlozi različitih rješenja kako bi se ispunile te potrebe, a usvojeno je rješenje koje je dovelo do stvaranja CSS jezika koji je krajem 1996. godine i zvanično standardizovan od strane W3C organizacije, verzija CSS1. Nakon dvije godine, u maju 1998. godine izdata je nova verzija, CSS2, a odmah nakon toga, 1999. godine objavljena je i treća, poslednja zvanična verzija koja se i dalje razvija [17].

CSS3 jezik koristi jednostavnu sintaksu za opisivanje elemenata. Jedan element je moguće opisati sa više različitih opisa, dok se jedan opis može iskoristiti za više različitih elemenata. Svaki opis mora prethodno da bude definisan nazivom koji se direktno povezuje sa određenim sadržajem, a uz naziv potrebno je izvršiti dodavanje svojstava sa željenim vrijednostima. Svojstva su predstavljena jednostavnim riječima koje su definisane W3C standardom, dok vrijednosti svojstava mogu biti zadate preko upisivanja određenih ključnih riječi ili brojeva. Dodavanje CSS stilova određenim dokumentima može da se izvrši na tri načina, a to su:

- umetnuti stilovi (*inline styles*) – su stilovi koji se primjenjuju na HTML elementima pomoću atributa *style*⁴;
- ugrađeni stilovi (*embedded styles*) – su stilovi koji se dodaju web stranici ili dokumentu pomoću elementa *style*⁵;
- spoljni opisni stilovi (*external styles*) – su stilovi koje se obično skladište u datoteci sa ekstenzijom *.css* koju je moguće povezati sa proizvoljnim brojem HTML dokumenata [18].

⁴ atribut *style* – je HTML atribut koji omogućava dodavanje CSS stila direktno u element.

⁵ element *style* – je HTML element koji omogućava dodavanje proizvoljnog broja CSS stilova između početnih i krajnjih oznaka elementa *style*.

3.1.3 PHP 7 (Hypertext Preprocessor)

PHP je programski jezik koji se koristi za pisanje skript fajlova koji se izvršavaju na serverskoj strani, serveru. PHP kôd ili PHP skripta može biti dio HTML stranice i tako omogućiti klijent-server komunikaciju, interaktivnost te stranice sa korisnikom. Klijent ili korisnik veb lokacije može da zahtijeva određenu stranicu sa implementiranim PHP kodom od servera, server će prihvatiti taj zahtjev i obraditi ga, a potom i generisati odgovor i poslati ga klijentskoj strani. Jezik je projekat otvorenog koda (engl. *open-source*) i potpuno je besplatan, a izvorni kôd je javno dostupan. Razvoj ovog jezika počeo je 1994. godine i od tada pa do danas radilo se na njegovom unaprijeđivanju. Zvanično je publikovano 8 verzija ovog jezika, a poslednja stabilna verzija je 8.0.7. Detaljan pregled izdatih verzija i unapređenja dostupan je na zvaničnom sajtu PHP jezika⁶.

PHP jezik je široko rasprostranjen i najviše se primjenjuje na vebu. Prema statistici sajta W3Techs⁷ koja je bila dostupna u vrijeme pisanja ovog rada, PHP je bio u upotrebi na 79.1% veb-sajtova koji koriste neki serverski programski jezik. PHP je svoju zastupljenost u poređenju sa ostalim sličnim jezicima ostvario na osnovu mnogih činjenica, a ključne su: performanse, skalabilnost, niska cijena, veoma je lak za učenje i upotrebu, a ima i veoma kvalitetnu dokumentaciju i podršku. Tvorac PHP jezika, Rasmus Lerdorf, opisuje ovaj jezik kao „*shared-nothing*“ arhitekturu. Da bi bilo jasnije, ovdje se misli na to da se veoma efikasno i jeftino napisane PHP skripte mogu proširiti na mnogobrojne manje servere. PHP omogućava uspostavljanje veze sa velikim brojem sistema za upravljanje bazama podataka, a najzastupljenija veza je sa MySQL sistemom koji će biti opisan u sledećem dijelu ovog poglavlja. Ujedno, ova veza PHP & MySQL je upravo i primijenjena za izradu WIntErO platforme. PHP podržava i mogućnosti objektno orijentisanog programiranja [19].

3.1.4 MySQL (My Structured Query Language)

MySQL predstavlja sistem za upravljanje bazama podataka. MySQL je višekorisnički i višenitni sistem, jer ima mogućnost da pristupanje podacima obezbijedi za više korisnika u isto vrijeme. Kao što se i iz naziva može pretpostaviti, MySQL podržava SQL⁸ (Structured Query Language). MySQL dijeli neke epitete sa PHP jezikom, kao što je projekat otvorenog koda. Može da bude i upotrijebljen u komercijalne svrhe, izuzetno je brz i lak za učenje, a ima i široku dostupnu podršku u vidu dokumentacije, obuka, konsultantskih usluga i sertifikovanja znanja. Administracija MySQL baze podataka može da se izvršava sa grafičkog ili interfejsa u obliku komandne linije [19].

⁶ Verzije PHP jezika (arhiva) - <https://www.php.net/archive/2021.php>

⁷ Statistika upotrebe PHP jezika - <https://w3techs.com/technologies/details/pl-php/all/all>

⁸ SQL – je standardni jezik koji se koristi da bi se izvršavali upiti u baze podataka.

Prvo izdanje ovog sistema datira iz 1995. godine. Od tada pa do 2005. godine razvijane su verzije 3 i 4, a 2005. godine predstavljena je verzija 5 koja je još uvijek aktuelna i za nju postoji podrška do 2023. godine. Postojali su planovi za izdavanje verzije 6, ali usled kupovine MySQL-a od strane kompanije *Sun Microsystems*, koja se desila 2008. godine, ovi planovi nijesu zaživjeli. Dvije godine kasnije, kompanija *Sun Microsystems* je prodana kompaniji *Oracle Corporation* koja je trenutni vlasnik nad MySQL-om. Verzije 6 i 7 ovog sistema ne postoje, a verzija 8 je aktuelna od 19. aprila 2018. godine [20]. Najnovije izdanje nudi niz korisnih unaprijeđenja, kao što su bolja pouzdanost, lakša upravljivost, veća sigurnost, visoka dostupnost, bolje performanse, a dodate su i neke bitne funkcije koje su upotrijebljene kroz izradu ovog rada [21].

3.1.5 JavaScript

JavaScript je skriptni programski jezik čija je osnovna namjena da se koristi za programiranje aplikacija na vebu. Prema pisanju popularnog magazina *Business Insider* [22], JavaScript je najpopularniji programski jezik na svijetu. Najčešća primjena ovog jezika je na strani klijenta, mada pojavom *Node.js*⁹ radnog okruženja ovaj jezik se upotrebljava i na serverskoj strani. JavaScript je počeo da se razvija 1995. godine. U početku je nosio naziv *LiveScript*, ali je kasnije promijenio naziv u JavaScript. Odmah nakon objavljivanja, JavaScript je standardizovan kako bi različite kompanije mogle da ga implementiraju u svojim proizvodima. Njegov standard nosi naziv *ECMAScript*¹⁰. Vremenom je izdato nekoliko verzija JavaScript jezika, a poslednja stabilna verzija je *ECMAScript 2020* [23].

JavaScript je iako jednostavan, veoma moćan jezik pomoću kojeg je moguće kreirati veoma složena rješenja. Za razliku od ostalih jezika, JavaScript ima manju potrebu da vrši neposrednu komunikaciju sa serverom. Moguće je kreirati formu za unos podataka i dodati JavaScript kôd koji će provjeriti ispravnost podataka prije daljeg slanja serveru. Ova funkcionalnost je upotrijebljena kod forme za slanje parametara na WIntErO platformi. Dalje, korisnik može da dobije odgovor bez ponovnog učitavanja stranice što dodatno povećava interaktivnost, takođe upotrijebljeno u realizaciji WIntErO platforme kroz korišćenje mapa sa označenim geolokacijama slivova. Povezivanje JavaScript skriptova je slično kao što je to slučaj sa CSS jezikom. Moguće je izvršiti interne i eksterne skripte, ali takođe i skripte koji su ugrađeni u HTML elemente. JavaScript je objektno baziran, platformski neutralan i višekorisnički jezik, a glavna prednost koja ga je svrstala među najpopularnije programske jezike je njegova integrisanost sa HTML jezikom[24].

⁹ *Node.js* – predstavlja multiplatformsko JavaScript radno okruženje otvorenog koda za izvršavanje JavaScript skriptova na serverskoj strani.

¹⁰ *ECMAScript* – predstavlja specifikaciju koja je standardizovana od strane organizacije *Ecma International* koja je zadužena za uspostavljanje standarda informacionih i komunikacionih sistema.

3.1.6 jQuery

jQuery je multiplatformska JavaScript biblioteka koja je kreirana kako bi olakšala korišćenje funkcija koje su često u upotrebi i koje se ponavljaju. To je najpopularnija JavaScript biblioteka, otvorenog je koda, veoma je skalabilna, a njena zastupljenost je veoma velika. Prema podacima statistike koju ažurira W3Techs¹¹ sajt, u vrijeme pisanja ovog rada, jQuery biblioteka je zastupljena na 77.9% sajtova koji koriste JavaScript jezik. Prva stabilna verzija ove biblioteke izdata je 2006. godine. jQuery biblioteka je licencirana pod MIT licencom¹².

Neke od glavnih funkcija koje jQuery biblioteka može da obavlja efikasno i brzo su navigacija ka dokumentima, selektovanje raznih DOM¹³ (Document Object Model) elemenata, pravljenje animacija, kreiranje različitih vrsta događaja i slično. Trenutno postoji na hiljade kreiranih funkcija koje se koriste svakodnevno za izradu različitih veb aplikacija. Struktura jQuery interfejsa je logično organizovana prema kategorijama pa je samim tim programerima veoma lako prepoznati funkcionalnosti koje su im potrebne u datom trenutku. jQuery biblioteka zahtijeva uključivanje u veb stranicu koje može da se izvrši na dva načina: dodavanjem linka u zaglavlje HTML dokumenta koji vodi do mreže za dostavu sadržaja (engl. CDM¹⁴ – Content Delivery Network) jQuery biblioteke ili preuzimanje biblioteke sa zvaničnog sajta i smještanje iste na lokalni server koji je u upotrebi.

3.1.7 Bootstrap 4

Bootstrap frejmwork¹⁵ je projekat otvorenog koda koji služi za kreiranje složenih veb aplikacija i veb-sajtova. Ovaj frejmwork je zasnovan na strukturi HTML, CSS i JavaScript jezika. Veoma je popularan, jer omogućava programerima uštedu vremena zbog mogućnosti primjene gotovih dizajnerskih rješenja i komponenti. Bootstrap podržava prilagodivi veb dizajn koji omogućava da se tražena veb stranica prilagodi uređajima koji imaju različitu rezoluciju. Ova funkcionalnost je veoma bitna u današnjem vremenu u kojem je upotreba različitih elektronskih uređaja u ekspanziji. Bootstrap zahtijeva uključivanje u veb aplikaciju, a moguće ga je izvršiti na sličan način kao i kod upotrebe jQuery biblioteke.

Bootstrap omogućava korišćenje velikog broja funkcionalnosti i stilova koji mogu da se primijene na različite HTML elemente. Zbog svojih gotovih šablona, upotreba ovog frejmworka

¹¹ Statistika upotrebe jQuery biblioteke - https://w3techs.com/technologies/overview/javascript_library/all

¹² MIT licenca – je softverska licenca koja dozvoljava izmjene i redistribuciju određenih softvera.

¹³ DOM - predstavlja strukturnu prezentaciju HTML dokumenata čiji su elementi organizovani u stablo u kome svaki čvor predstavlja jedan element.

¹⁴ CDM - predstavlja distributivnu mrežu servera čiji je cilj da krajnjim korisnicima dostavi sadržaj uz visoke performanse i maksimalnu dostupnost.

¹⁵ Frejmwork (engl. *Framework*) – predstavlja skup gotovih komponenti čije funkcionalnosti imaju mogućnost izmjena zbog potrebe izgradnje određenih aplikacija.

može da dovede do izgradnje rješenja koje nije originalnog izgleda. Svakako, ovo ne predstavlja veliki problem, jer Bootstrap omogućava izmjene u konfiguraciji svojih komponenata. Pojavio se ne tako davne 2011. godine, a razvila ga je grupa programera radeći u kompaniji *Twitter* [26]. Posljednja stabilna verzija ovog frejmworka je verzija 4. Bootstrap je visokorangirani projekat na GitHub hosting servisu¹⁶¹⁷.

3.1.8 Microsoft Visual Studio Community 2019

Microsoft Visual Studio Community je primarno integrisano razvojno okruženje (engl. IDE – Integrated Development Environment) kompanije *Microsoft*. Ovo radno okruženje se koristi za kreiranje kompjuterskih igara, desktop aplikacija, *cloud*¹⁸ servisa, veb-sajtova, mobilnih i tablet aplikacija i slično. Visual Studio je veoma fleksibilno i produktivno radno okruženje koje podržava veliki broj programskih jezika, a uz to je omogućena i podrška kroz odvojenu nadogradnju onih jezika koji nijesu integrisani uz standardnu instalaciju. Izdanje *Community* je besplatno, a postoje i komercijalna izdanja koja je moguće koristiti uz novčanu nadoknadu. Instalacija je veoma jednostavna, a preuzimanje je moguće izvršiti na zvaničnom veb-sajtu kompanije *Microsoft*. Instalacija se može izvršiti na dva načina, preuzimanjem veb instalacionog fajla ili preuzimanjem *.iso* datoteke. Visual Studio je veoma moćna alatka za kodiranje, posjeduje mogućnost naprednog otklanjanja grešaka, a ima i podršku za Git¹⁹ integraciju koja omogućava manipulaciju sa izvornim kodom u Git repozitorijima koje hostuje bilo koji provajder, uključujući i već pomenuti GitHub. Za potrebe ovog rada korištena je verzija Visual Studio Community 2019, što je ujedno i posljednja publikovana verzija ovog alata [27].

3.1.9 cPanel

cPanel je najpopularnija hosting platforma koja služi da pojednostavi proces hostovanja veb-sajta kroz upotrebu veb pregledača. cPanel je veoma funkcionalna platforma, dostupna je na više jezika, a aktivnosti koje se izvršavaju upotrebom iste su brojne i izuzetno olakšavaju posao programerima. Upotrebom cPanel platforme moguće je izvršiti parkiranje domena, dodavanje datoteka, kreiranje baze podataka, pravljenje rezervne kopije sadržaja koji se nalazi na serveru i još mnogo toga. Platforma posjeduje veliki broj aplikacija koje su podijeljene po kategorijama što ovu platformu čini veoma jednostavnom za rad.

¹⁶ Rezultati repozitorijuma na GitHub-u - <https://github.com/search?q=stars%3A%3E1&type=Repositories>

¹⁷ GitHub – predstavlja servis za kontrolu praćenja promjena verzija izvornog koda.

¹⁸ *Cloud* – predstavlja platformu koja aplikacije i podatke tretira kao uslugu i isporučuje ih preko interneta.

¹⁹ Git – predstavlja sistem za praćenje promjena verzija izvornog koda, a GitHub je jedna od platformi koja funkcioniše na principu ovog sistema.

3.2 Analiza i opis projektovanog rješenja

U ovom dijelu rada analiziraće se zahtjevi i govoriće se o arhitekturi veb platforme WIntErO. Dalje, obrazložiće se i mogućnosti koje će opisana arhitekturna projekcija obezbijediti. Kao što je već rečeno, veb platforma ima za cilj da obezbijedi uspješan rad stručnjacima iz oblasti geonauke koji se bave istraživanjem jako bitnog područja koji se tiče intenziteta erozije zemljišta. Da bi se realizacija uspješno ostvarila prvo je bilo potrebno napraviti prototip platforme koji se kroz testiranje pokazao kao uspješan [11], [12], [13]. Nakon toga krenulo se u detaljniju projekciju i realizaciju ovog rješenja.

3.2.1 Analiza zahtjeva

Prije nego što se krenulo u projekciju arhitekture i izradu same platforme bilo je potrebno izvršiti analizu zahtjeva koji su neophodni za realizaciju ovog projekta. Analiza zahtjeva ovakve ili slične specifičnosti koja obično može da bude jako kompleksne prirode se obavlja od strane programera i stručnjaka iz oblasti za koju se rješenje razvija. Veoma je bitna i neophodna sposobnost za rješavanje problema kako bi se zahtjevi uspješno razumjeli, a kasnije analizirali i zatim konačno povezali u cjelinu koja je logična i treba da predstavlja rješenje. Na početku je odrađena analiza programskog paketa IntErO koji se koristio u svrhe izvršavanja proračuna i prikupljene su potrebne informacije kroz sagledavanje svih nedostataka koje upotreba tog programa ima u današnjem vremenu. Spalević je kroz izradu doktorske disertacije [4] prikazao upotrebu ovog programa koji je u tom periodu predstavljao uspješno rješenje, a ova disertacija ujedno i predstavlja vrlo koristan izvor informacija koje su neizostavne za izradu WIntErO platforme.

Glavni principi na kojima se bazira kvalitetna analiza su razumijevanje potreba krajnjih korisnika, u ovom slučaju to se najviše odnosi na stručnjake koji će najvećim dijelom upotrebljavati platformu, i kreiranje efikasnog sistema koji će obezbijediti potrebne funkcionalnosti. Kako se krenulo od saznanja da je dosadašnje rješenje IntErO bilo veoma upotrebljavano od strane naučne zajednice [28], [29], [30], [31] i da je funkcionisanje bilo precizno i davalo korisne rezultate, sagledale su se sve činjenice, prednosti i nedostaci. Glavne prednosti su te da je IntErO koristio algoritam za proračun koji je davao tačne rezultate i da je zamijenio metode mjerenja koje su vršene analitičkim putem što je dovelo do ubrzanja izvođenja rezultata. Nedostaci IntErO paketa u danjašnjem vremenu mogu da se sagledaju kroz njegovu primjenu samo na *Microsoft Windows* operativnim sistemima, jer IntErO nije multiplatformska aplikacija, nedostupnost na Internetu i samim tim na pametnim uređajima koji su neizostavni u savremenom društvu, a svakako se može uzeti u obzir i nescalabilan način unaprijeđivanja koji se javlja zbog potreba kompleksnih zahtjeva koji su svakodnevni. Nakon ovog kratkog razmatranja na red je došlo i sagledavanje budućih načina izvršavanja proračuna koji bi trebalo da se uspostave izradom veb platforme koja će biti utemeljena na valjanim osobinama IntErO paketa.

Kao što je već pomenuto u drugom poglavlju, upotrebom IntErO paketa moguće je izvršiti skladištenje parametara i proračuna upisivanjem istih u *.sli* datoteke. Pošto je programski paket korišten skoro jednu deceniju, činjenica je da je kreiran veliki broj *.sli* datoteka i da bi bilo veoma korisno objediniti sve te fajlove u jednu centralizovanu bazu koja će biti uvijek i svima dostupna. Ovakva zamisao je svakako veoma interesantna, jer se stvaranjem jedinstvene baze podataka obezbjeđuje uvid u eventualno veliki broj analiziranih prostornih područja i otvara šansa za laku aktuelizaciju poređenja prethodnih istraživanja sa nekim novim čime bi mogle da se vrše komparacije koje će pomagati stručnjacima pri projekciji budućih procjena. Iz iznijetog se da zaključiti da je potrebno osmisliti metod kojim bi se podaci iz postojećih fajlova isčitavali u programskom interfejsu nove veb platforme, a zatim i upisivali u bazu podataka. Jasno je da izrada forme za unos podataka za ove potrebe za sobom povlači i izradu baze podataka koja je sastavni dio ispravnog funkcionisanja. Neophodno je da baza obezbijedi brzo, sigurno i logično upisivanje, pretraživanje i pristup kolekciji podataka.

Pored kreiranja forme na kojoj bi se isčitavali podaci iz *.sli* datoteka, potrebno je kreirati i formu za upis novih podataka. Moguće rješenje je da se koristi ista forma za oba slučaja, ali to bi imalo priličan broj nedostataka. Podaci iz *.sli* datoteka su stari proračuni koji su dobijani godinama unazad i nije potrebno vršiti korekcije na njima u trenutku unosa, potrebno je samo izvršiti isčitavanje i upis koji bi zbog vremenske efikasnosti trebao da bude veoma brz. U nastavku će se ipak prikazati da je postojeće podatke iz baze moguće mijenjati i upoređivati zbog projektovanja raznih analiza, bilo da su oni unešeni kao novi ili isčitani iz *.sli* datoteka. Dalje, novi podaci ipak zahtijevaju određene asistencije pri unosu zbog lakšeg korišćenja platforme, pa samim tim i forma za unos tih novih podataka je nešto kompleksnija. Forma za unos novih podataka mora da obezbijedi niz funkcija koje će imati zadatak da otkriju postojanje eventualnih logičkih grešaka, da izvrše provjere parametara za proračun koeficijenata i automatski izvrše preraspodjelu procenata po elementima do 100% ukoliko direktno nije unešena cjelokupna vrijednost nekih od koeficijenata, a takođe i da isprate međusoban unos vrijednosti u jednoj od sekcija forme koja se tiče određenih karakteristika. Dizajn interfejsa forme, kao i kompletan dizajn platforme moraju da budu moderni i kreirani tako da omogućuje što jednostavniji i efikasniji rad. Jedinstvena tabela baze podataka je naravno neophodna za oba slučaju unosa.

Pošto je glavna ideja da platforma omogući uvid u ove podatke svim zainteresovanim pojedincima, upisane proračune i rezultate je potrebno prikazati i omogućiti laku pretragu i manipulaciju sa istim. Neophodno je osmisliti kvalitetnu organizaciju koja se zasniva na podjeli po kategorijama i podkategorijama radi lakšeg pregleda i pronalaženja. Pored toga, a pošto se radi o disciplini nauke koja se bavi istraživanjem geografskih lokacija, javila se i ideja da se uz razvrstavanje po kategorijama implementira geografska mapa na kojoj bi se označavale geografske lokacije i sa njima povezivali proračuni koji se vezuju za te lokacije. Uz ove mogućnosti, neophodno je obezbijediti i eksportovanje proračuna i rezultata u neke od najpopularnijih formata dokumenata. Ovi dokumenti treba pored proračuna i rezultata da sadrže opis kada i gdje je rađeno istraživanje, geografske koordinate, kao i neke kratke zaključke u vidu predefinisanih komentara na osnovu dobijenih rezultata. Kroz dio koji se tiče implementacije platforme detaljno će se objasniti primjena i upotreba tehnologija koje su korištene za ispunjavanje svih zahtjeva.

Pristup pomenutim formama za unos ne može biti javan kao što je to slučaj sa pregledom kategorisanog sadržaja. Platforma mora sadržati sistem za identifikaciju korisnika sa različitim pravima pristupa koji bi obezbijedio verifikovanje proračuna od strane administracije kako bi stvaranje sadržaja bilo kvalitetno i provjereno pouzdano. Pored toga bi morale da se obezbijede i neke standardne funkcije koje bi doprinosile sigurnosti i jednostavnosti upotrebe sistema. Iz zaključenog se da primijetiti da moraju postojati najmanje dva stepena pristupnih uloga i to korisnička i administratorska. Korisnička uloga bi imala prava na unos, pregled i izmjenu unosa, a administratorska uloga uključuje korisničku sa dodatnim opcijama za verifikovanje poslatih unosa kako ne bi došlo do upisivanja nesmislenog sadržaja u bazi. Takođe, administrator bi trebao da ima uvid u kompletan sadržaj baze podataka. Pošto se pretpostavlja da će upotreba platforme biti veoma rasprostranjena, ideja da postoje samo dva stepena pristupa i nije baš efikasna zbog činjenice da je vrlo moguće da će potreban broj administratora biti veliki i da je veoma korisno postojanje mogućnosti izmjena i dodavanja administratorskih uloga. Dakle, sistem za identifikaciju mora biti projektovan sa trostepenim ulogama, a to su korisnička, administratorska i uloga super administratora²⁰. Posljednja, najznačajnija uloga bi trebala da ima potpuno iste mogućnosti kao i administratorska, a uz to i opciju za upravljanje administratorskim ulogama.

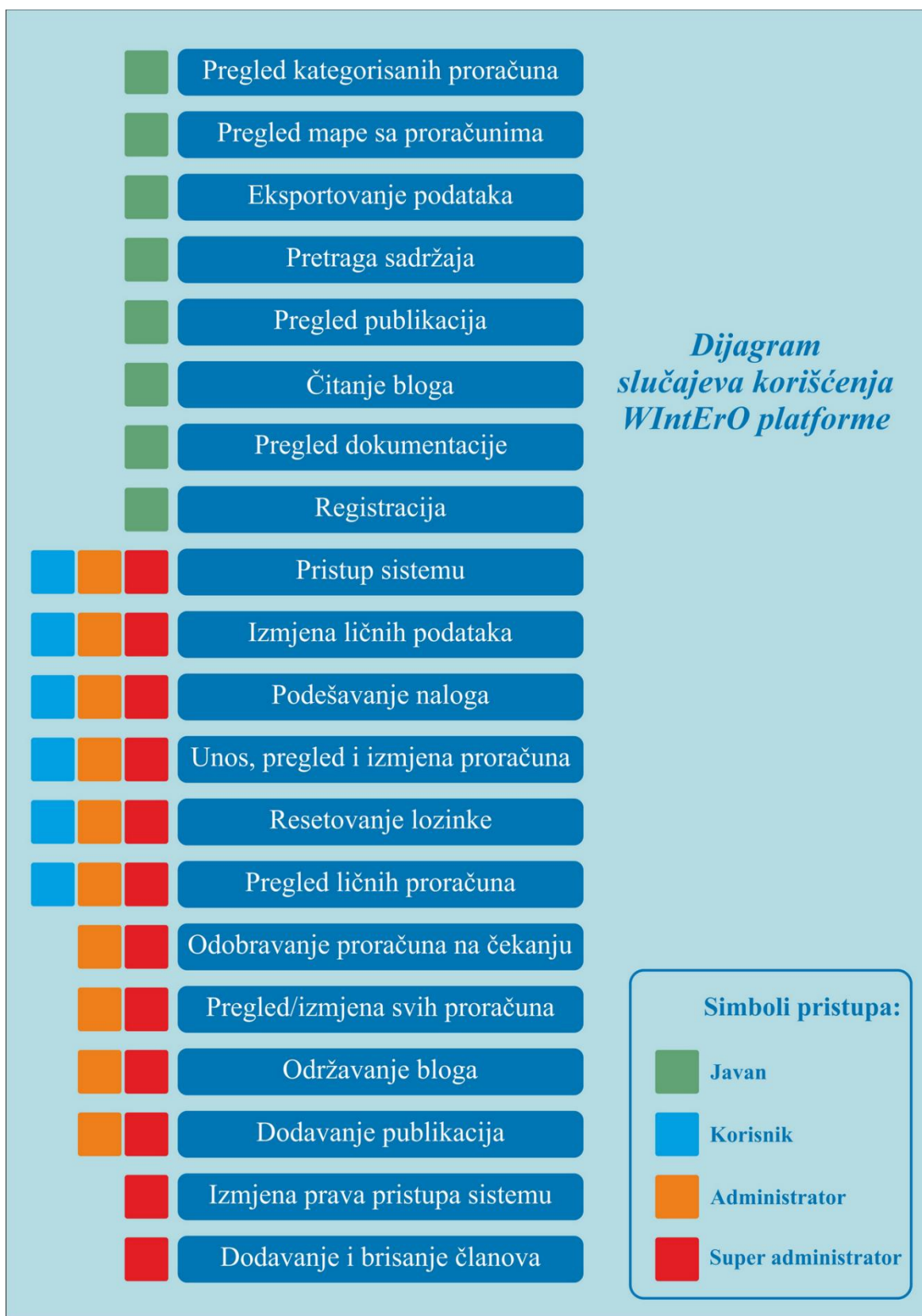
Sistem za identifikaciju korisnika mora posjedovati i neke standardne funkcije koje i svaki drugi sličan sistem ima na raspolaganju. Glavne funkcije koje se moraju implementirati su sigurnost pri identifikaciji korisnika, uvid u lične korisničke podatke i mogućnost izmjena istih, detaljna podešavanja i izmjene podataka za autentifikaciju korisnika, pouzdanost kod slučaja odjavljivanja iz sistema, a uz sve to neophodno je i uspostaviti praćenje promjena odobravanja, izmjena i brisanja unosa. Takođe, sistem mora da funkcioniše u realnom vremenu i da komunikacija u vidu obavještenja ka korisnicima bude uspostavljena u trenutku kada za to postoji potreba.

WIntErO platforma, pored navedenih osnovnih osobina treba da posjeduje i neke dodatne elemente koji bi unaprijedili kvalitet i doprinijeli potpunom korisničkom iskustvu u pozitivnom smislu. Neke od osnovnih mogućnosti koje bi trebalo obezbijediti su pretraživanje sadržaja, publikovanje naučnih radova koji su povezani na bilo koji način sa upotrebom WIntErO platforme, održavanje bloga, a tu je naravno i potreba za izradom dokumentacije koja se tiče upotrebe platforme, uslova korišćenja i pravila politike privatnosti. Na slici 2 dat je dijagram slučajeve korišćenja²¹ koji oslikava opisane zahtjeve.

U softverskom inženjeringu analiziranje zahtjeva prije početka realizacije bilo kojeg projekta je neophodno, a zadaci koji treba da se izvrše moraju biti testirani, djelotvorni, skalabilni i dokumentovani. Nakon analize na red dolazi projekcija arhitekture sistema i projekcija informacione arhitekture koje će se bazirati na osnovu analiziranih zahtjeva, a takođe i na tehnologijama koje je moguće primijeniti. U nastavku će se dati pregled i opis prvo klijent-server arhitekture, a potom i informacione arhitekture.

²⁰ Super administrator – predstavlja korisnika koji ima sva prava u sistemu.

²¹ Dijagram slučajeve korišćenja - vizuelizuje i prikazuje ponašanje ili funkcionalnost sistema.



Slika 2: Dijagram slučajeva korišćenja WIntErO platforme

3.2.2 Klijent-server arhitektura

Kao što je već pomenuto, arhitektura sistema će se bazirati na potrebi za ispunjavanjem analiziranih zahtjeva, ali takođe i na tehnologijama dostupnim za rad. Tehnologije koje će se upotrijebiti za izradu platforme su već istražene i njihovom upotrebom je moguće ispuniti potrebne zahtjeve. Veb platforma WIntErO će biti bazirana na troslojnom višenamjenskom klijent-server modelu arhitekture koji je zasnovan na slanju zahtjeva i primanju odgovora. U praksi postoje i modeli arhitekture sa različitim brojem komponenata čijom implementacijom je moguće uspostaviti višeslojne sisteme koji su veoma skalabilni i imaju visok kapacitet performansi i resursa [32]. Za potrebe funkcionisanja WIntErO platforme troslojna arhitektura je u stanju da ispuni očekivanja. Ova arhitektura u praksi najčešće funkcioniše preko Interneta, mada je moguće i izvršiti projekciju koja može da se primijeni i u nekim drugim lokalnim mrežama i sistemima. Troslojna arhitektura, čiji je prikaz dat na slici 3, predstavlja programski model koji omogućava distribuciju funkcionalnosti aplikacije kroz tri nezavisna podsistema, a to su [33]:

- podsistem koji implementira funkcije korisničkog interfejsa,
- podsistem procesa koji se izvršavaju na udaljenim serverima,
- podsistem za upravljanje podacima (bazama podataka).

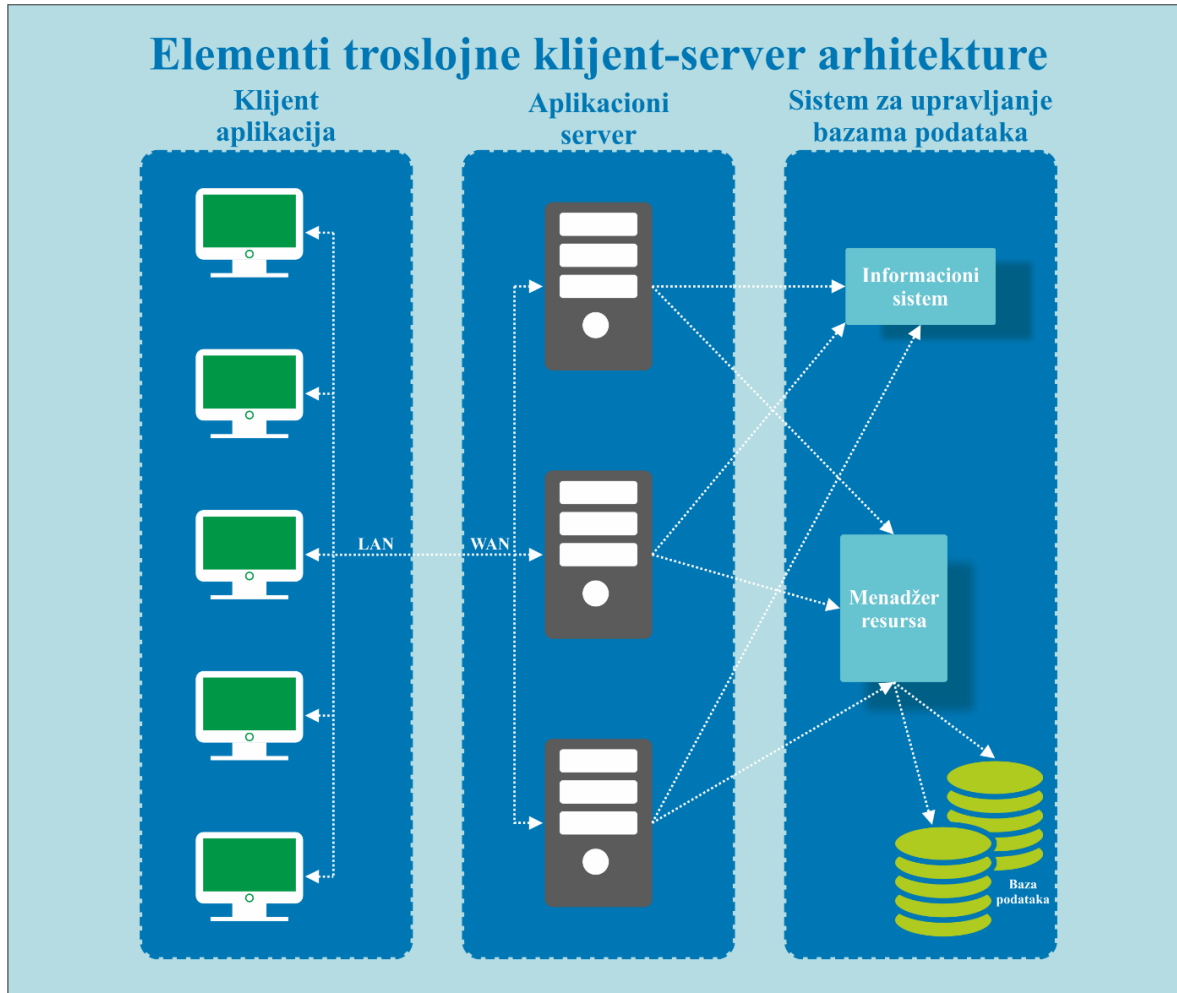
Podsistem prvog reda ima ulogu da omogući prezentovanje sadržaja i interaktivnost prema korisnicima. Takođe, omogućava korisnicima da komuniciraju sa podsistemom drugog reda na siguran i efikasan način. Podsistem prvog reda nema direktnu komunikaciju sa podsistemom trećeg reda, već se komunikacija obavlja indirektno preko podsistema drugog reda.

Podsistem drugog reda ili aplikacioni server je dio sistema koji izvršava programski kôd koji implementira logiku aplikacije. U ovom podsistemu se dešava većina procesa, a moguće mu je istovremeno pristupiti od strane većeg broja klijenata. Razdvajanjem serverske strane na dva podsistema smanjuje se opterećenje, obezbjeđuje efikasnije upravljanje vezama, a poboljšavaju se i ukupne performanse sistema.

Podsistem trećeg reda je, kao što je već rečeno, zaštićen od direktnog pristupa od strane prvog reda. Veza se uspostavlja kroz aplikacioni server što doprinosi očuvanju bezbjednosnih i logičkih prednosti sistema koji je podijeljen na tri nivoa.

Troslojna arhitektura je sistem koji je jednostavan za održavanje, a performanse je moguće unaprijediti na brz i lak način. U praksi je često slučaj da je ovaj sistem implementiran sa većim brojem servera što dovodi do veće pouzdanosti i fleksibilnosti, ali može dovesti i do usporavanja rada sistema u slučaju kompleksnih rješenja. Arhitektura se oslanja na Internet standarde²² čime se omogućava integracija heterogenih sistema.

²² Internet standard – je specifikacija tehnologije ili metodologije koju je moguće primijeniti na Internetu.



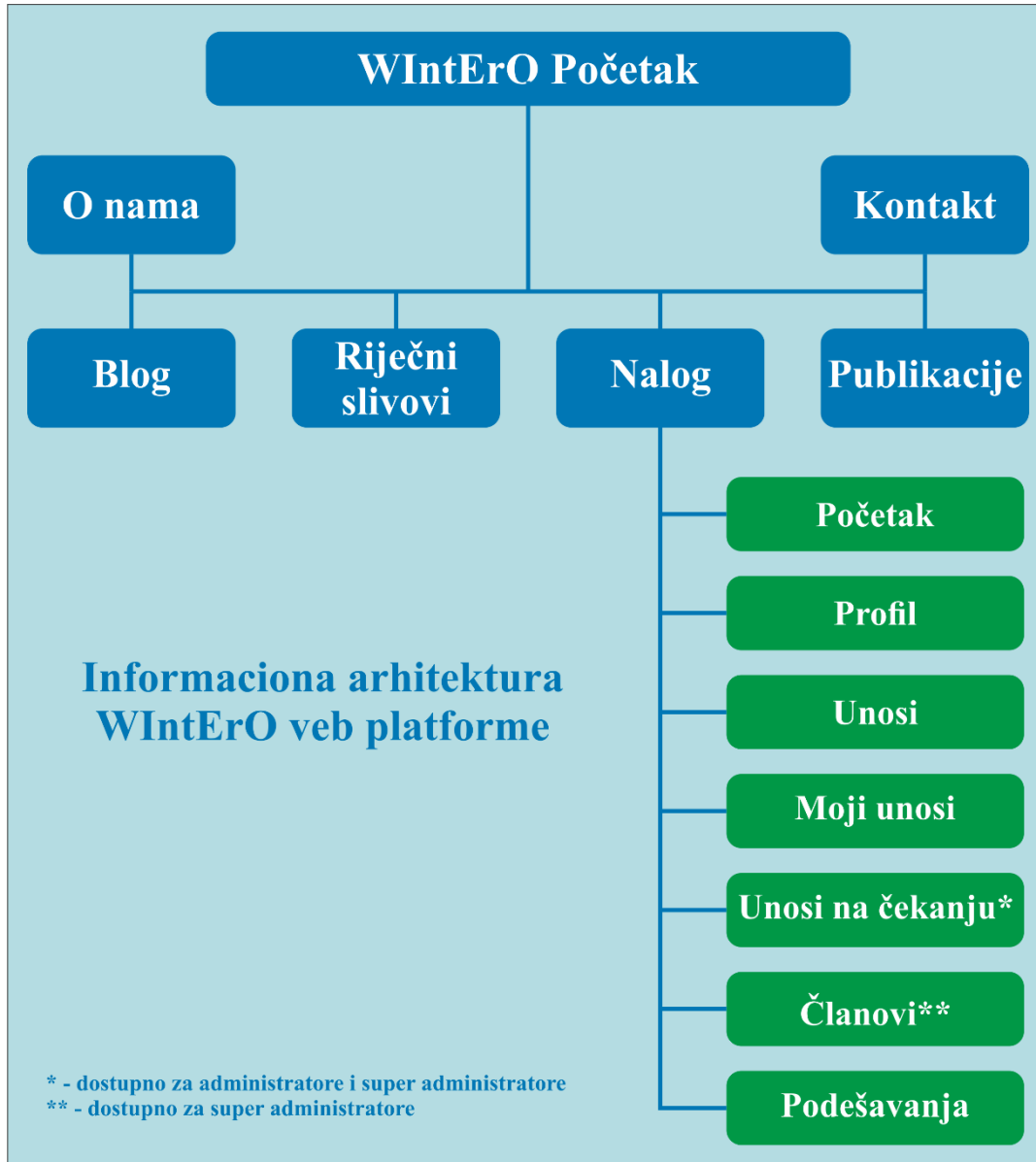
Slika 3: Prikaz elemenata troslojne arhitekture klijent-server sistema

3.2.3 Informaciona arhitektura

Informaciona arhitektura zadužena je za organizaciju, klasifikaciju, struktuiranje i određivanje pozicije podataka koji trebaju da se nalaze na veb platformi i predstavlja veoma važan zadatak koji doprinosi boljem korisničkom iskustvu. Kod izrade kompleksnih projekata, u timu koji je zadužen za razvoj postoji posebna pozicija za informacionog arhitektu, a u slučaju manjih zadataka, kao što je slučaj sa izradom WIntErO platforme, to je posao programera ili dizajnera. Na slici 4 prikazana je primarna informaciona arhitektura WIntErO platforme.

U informacionoj arhitekturi postoje četiri glavne komponente, a to su:

- organizacioni sistem,
- sistem obilježavanja,
- navigacioni sistem,
- sistem pretrage.



Slika 4: Prikaz primarne informacione arhitekture WIntErO veb platforme

Svaka od četiri navedene komponente ima za cilj da poboljša kompletnu arhitekturu. Tako, organizacioni sistem je zadužen za struktuiranje sadržaja kako bi se korisnicima pomoglo u pronalaženju potrebnih podataka. Sistem obilježavanja je sistem koji predstavlja više podataka jednom zajedničkom riječju. Navigacioni sistem omogućuje kretanje korisnicima kroz aplikaciju preko određenih interaktivnih elemenata. Sistem pretrage koristi se za lakše pronalaženje podataka [34]. Postoji nekoliko načina da se kreira informaciona arhitektura, a dva najkorištenija su kreiranjem dijagrama, kao što je slučaj kod WIntErO platforme, ili kreiranjem „umne mape“ (engl. *mind mapping*) kojom se hijerarhijski kreira informaciona arhitektura zajedno sa njenim tokovima. Za kreiranje informacione arhitekture postoji i određen broj kvalitetnih alata, a jedan od najpopularnijih je Microsoft Visio [35].

3.3 Implementacija veb platforme

U ovom poglavlju će se detaljno opisati i pokazati način na koji je implementirana WIntErO veb platforma. Detaljno će se prikazati primjena tehnologija, programskih algoritama i drugih metoda za implementaciju platforme koja omogućava ispunjavanje prethodno analiziranih zahtjeva.

Sama implementacija veb platforme se može podijeliti na dva dijela, i to implementiranje dijela koji je vidljiv i dostupan svima i dijela koji je dostupan samo registrovanim korisnicima, administratorima i super administratorima sistema. Pored navedene podjele po pristupu, veb platforma se može podijeliti po razvoju i postupku razvoja. Što se tiče podjele po razvoju ona se može podijeliti na dva sloja koji se u softverskom inženjerstvu opisuju terminima „prednji kraj“ (engl. *front-end*) i „zadnji kraj“ (engl. *back-end*). Termin „prednji kraj“ se koristi da opiše sve ono što je vidljivo korisniku, dok se terminom „zadnji kraj“ opisuje sve ono što nije vidljivo i što se izvršava u pozadini. Kod klijent-server arhitekture, klijent predstavlja „prednji kraj“, a server „zadnji kraj“. U nastavku ovog poglavlja detaljnije će se prikazati i opisati i korisnički dio, a zatim i dio koji je javno dostupan, dok će se izgradnja „prednjeg“ i „zadnjeg“ kraja određenih djelova veb platforme prikazivati paralelno samom postupku razvoja. Ovo poglavlje je podijeljeno na sledeće cjeline:

- izrada baze podataka za skladištenje proračuna i podataka o korisnicima,
- razvoj korisničkog okruženja,
- razvoj javno-dostupnog okruženja,
- razvoj prilagodljivog dizajna za različite uređaje.

3.3.1 Izrada baze podataka za skladištenje proračuna i podataka o korisnicima

Cilj ovog poglavlja je da se detaljno prikažu strukture tabela koje zajednički čine bazu podataka, kao i sama izrada istih. Ove tabele služe za skladištenje informacija i dobijenih parametara o rječnim slivovima, skladištenje podataka o publikacijama i podataka o korisnicima. U nastavku će takođe biti opisani osnovni pojmovi baza podataka.

Baze podataka se koriste za prikupljanje informacija i čuvanje istih kako bi moglo sa njima da se manipuliše čime se dobijaju neke nove informacije, vrši njihova analiza i slično. Za izradu baze podataka za potrebe platforme WIntErO korišten je MySQL sistem koji je prethodno predstavljen u ovom radu, a koji je sastavni dio cPanel platforme.

Baza podataka mora da bude ispravno organizovana ukoliko se želi postići fleksibilnost i brza manipulacija podacima. Potrebno je napraviti ispravno struktuiranu bazu koja je sposobna da se adaptira na promjene koje se zahtijevaju od strane korisnika, kao i to da bude jednostavna kada su u pitanju proširenja i unaprijeđenja.

Kada su se tek pojavile, baze podataka su bile hijerarhijskog tipa, da bi se nakon toga pojavio mrežni tip istih, a potom su nastale relacione baze podataka koje su danas najzastupljenije. Relacione baze se baziraju na prostim konceptualnim tabelama. Sama riječ relacija ukazuje na to da su tabele međusobno povezane i kao takve djeluju jedinstveno. Sve tabele koje čine jednu bazu podataka imaju neka zajednička svojstva, kao što su primarni ključ (engl. *primary key*), strani ključ (engl. *foreign key*), polje za automatsko povećanje brojača u tabeli (engl. *auto increment*) kako bi svi redovi imali svoj jedinstveni broj i slično. Ova polja su neophodna za ispravno upravljanje podacima. Tabele takođe moraju da imaju i jedinstveno ime. Kako se tabele sastoje od kolona i redova, u redovima tabela se nalaze podaci koji su povezani sa jedinstvenim brojačem, dok svaka kolona ima jedinstven tip podataka za svaki unos [19]. Tipovi podataka mogu da budu različiti, a ukupno devet²³ je podržano kroz MySQL sistem sa kojim je kreirana WIntErO baza podataka. Samo neki od njih su korišteni za izradu, a to su cjelobrojni (engl. *integer*), realni (engl. *real*), simbolni (engl. *string*), znakovni (engl. *char*) i logički (engl. *logical*). U narednom poglavlju, kada bude riječ o izradi aplikacija detaljno će biti objašnjen i JSON (JavaScript Object Notation) tip podataka koji služi da omogući vezu sa istom bazom koju koristi i veb platforma.

U okviru faze osmišljavanja strukture tabela baze podataka uzeti su u obzir svi faktori od značaja. Tabela koja se odnosi na podatke o rječnim slivovima je prva koja je kreirana i na osnovu iste omogućen je razvoj i drugih djelova platforme. Prikaz tabele *data* je dat na slici 5. Tabela, pored polja brojača, sadrži i polja koja se odnose na podatke o rječnim slivovima. Dalje, neophodna polja za ispravno funkcionisanje su i polja *active* i *emailuser*. Polje *active* služi kako da bi administrator/super administrator platforme mogao da vrši odobravanje prikazivanja podataka vezanih za rječni sliv, ukoliko su ti podaci i rezultati logički ispravni. Nakon slanja podataka od strane korisnika u polje *active* podrazumijevana vrijednost je broj 0, dok se nakon odobravanja vrijednost mijenja sa brojem 1. Polje *emailuser* služi za evidenciju korisnika, onih koji šalju podatke. Svi nazivi polja su napisani na engleskom jeziku zbog internacionalizacije platforme.

Već je rečeno da platforma sadrži dvije forme za unos podataka. Prva forma služi za upisivanje potpuno novih podataka, dok druga isčitava postojeće podatke iz *.sli* fajlova. Obje forme koriste istu tabelu. Oni podaci koju budu odobreni od strane administratora su odmah i vidljivi na platformi, dok se u suprotnom podaci iz tabele brišu. Potreba za ovakvim načinom javlja se iz razloga što je potrebno imati bazu koja je efikasna u smislu odziva i jednostavna za održavanje. Nepotrebno gomilanje tabela sa podacima koji nijesu validni imalo bi negativan efekat i kada bi se javila potreba za nadogradnjom baze. Obrada podataka je takođe realizovana na jednostavan način, bez novih upisivanja, a više o tome će se govoriti u narednim podpoglavljima. MySQL sistem obezbjeđuje bezbjedno čuvanje podataka i svakodnevni *backup* (rezervna kopija). Treba napomenuti i to da se konekcija sa bazom podataka vrši tako što se u zaglavlju fajlova koji se povezuju sa bazom, navode putanje fajlova koji sadrže podatke potrebne za verifikaciju. Potencijalna izmjena tih podataka može se izvršiti jedino pristupom serveru kroz cPanel ili pristupom preko nekih od FTP (File Transfer Protocol) programa.

²³ MySQL tipovi podataka - <https://dev.mysql.com/doc/refman/5.7/en/data-types.html>

Server: localhost:3306 » Database: wintpoe_wintero » Table: data

Browse Structure SQL Search Insert Export Import Operations Triggers

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id		int(11)		No	None	AUTO_INCREMENT	
<input type="checkbox"/>	2	riverbasin		varchar(255)	latin1_swedish_ci	No	None		
<input type="checkbox"/>	3	country		varchar(255)	latin1_swedish_ci	No	None		
<input type="checkbox"/>	4	year		varchar(11)	latin1_swedish_ci	No	None		
<input type="checkbox"/>	5	latitude		double		No	None		
<input type="checkbox"/>	6	longitude		double		No	None		
<input type="checkbox"/>	7	active		tinyint(1)		No	None		
<input type="checkbox"/>	8	f		double		No	None		
<input type="checkbox"/>	9	o		double		No	None		
<input type="checkbox"/>	10	fv		double		No	None		
<input type="checkbox"/>	11	fm		double		No	None		
<input type="checkbox"/>	12	lv		double		No	None		
<input type="checkbox"/>	13	lb		double		No	None		
<input type="checkbox"/>	14	lizarray		varchar(255)	latin1_swedish_ci	No	None		
<input type="checkbox"/>	15	farray		varchar(255)	latin1_swedish_ci	No	None		
<input type="checkbox"/>	16	h0		double		No	None		
<input type="checkbox"/>	17	deltah		double		No	None		
<input type="checkbox"/>	18	hmin		double		No	None		
<input type="checkbox"/>	19	hmax		double		No	None		
<input type="checkbox"/>	20	sumal		double		No	None		
<input type="checkbox"/>	21	lm		double		No	None		
<input type="checkbox"/>	22	fp		double		No	None		
<input type="checkbox"/>	23	fpp		double		No	None		
<input type="checkbox"/>	24	fo		double		No	None		
<input type="checkbox"/>	25	fs		varchar(255)	latin1_swedish_ci	No	None		
<input type="checkbox"/>	26	ft		varchar(255)	latin1_swedish_ci	No	None		
<input type="checkbox"/>	27	fg		varchar(255)	latin1_swedish_ci	No	None		
<input type="checkbox"/>	28	hb		double		No	None		
<input type="checkbox"/>	29	temperature		varchar(255)	latin1_swedish_ci	No	None		
<input type="checkbox"/>	30	up		varchar(255)	latin1_swedish_ci	No	None		
<input type="checkbox"/>	31	hgod		double		No	None		
<input type="checkbox"/>	32	y		double		No	None		
<input type="checkbox"/>	43	xa		double		No	None		
<input type="checkbox"/>	51	phi		double		No	None		
<input type="checkbox"/>	62	hsr		double		No	None		
<input type="checkbox"/>	63	isr		double		No	None		
<input type="checkbox"/>	64	emailuser		varchar(255)	latin1_swedish_ci	No	None		

Check all With selected: Browse Change Drop Primary Unique Index Fulltext Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after emailuser Go

Slika 5: Prikaz strukture tabele data

Dalje, preostale dvije tabele se odnose na objavljivanje publikacija i registraciju korisnika. Prva tabela, *publications*, je veoma slična sa tabelom *data* i nije potrebno posebno govoriti o njoj, dok je druga nešto složenija. U nastavku će se govoriti o drugoj, tabeli *register*. Prikaz tabele je dat na slici 6.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	firstname	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
3	lastname	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
4	email	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
5	password	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
6	verification_key	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
7	dateofbirth	date			No	None			Change Drop More
8	city	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
9	countries	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
10	profession	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
11	companyuni	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
12	active	tinyint(1)			No	None			Change Drop More
13	profilepicture	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
14	roles	varchar(100)	latin1_swedish_ci		No	None			Change Drop More

Slika 6: Prikaz strukture tabele *register*

Ova tabela sadrži više bitnih polja, a to su: *password* (polje koje se koristi za smještanje šifre korisničkog naloga), *verification_key* (polje koje služi za smještanje verifikacionog ključa), *profile picture* (polje za smještanje adrese putanje koja vodi do profilne slike korisnika), kao i polje *roles* koje služi za dodjeljivanje prava korisnicima. Pri kreiranju svakog naloga, podrazumijevano pravo korisnika je *guest*, dok je kasnije od strane super administratora moguće promijeniti prava pristupa ili totalno izbrisati kreirani nalog. Više o svim ovim poljima i pravima u narednom potpoglavlju kada se bude govorilo o kreiranju korisničkog okruženja. Tipovi podataka korišteni za izradu ove tabele su isti kao i oni korišteni za prethodno opisanu tabelu.

Iz priloženog se vidi da ove dvije tabele nijesu relaciono uvezane zbog jednostavnosti postupka izrade, ali je to moguće učiniti na lak način, ukoliko bi to bilo potrebno. Obje tabele sadrže polja za unos *e-mail* adresa pa je jednostavnim dodjeljivanjem privatnog i stranog ključa moguće stvoriti njihovu relaciju. *E-mail* adresa je uvijek jedinstvena za svakog korisnika pa ju je moguće i iskoristiti u te svrhe. U ovom slučaju ona služi kao korisničko ime. Dio koji je vezan za upravljanje bazom je omogućen super administratoru, kroz korisnički interfejs. Neke od mogućnosti ipak nijesu dostupne na ovaj način već se mora pristupiti bazi podataka kroz cPanel platformu.

3.3.2 Razvoj korisničkog okruženja

Razvoj korisničkog okruženja neophodan je dio svake platforme ukoliko se želi omogućiti evidencija korisnika, kao i podataka koje korisnici šalju, odnosno upisuju u bazu podataka. Korisnički dio ili lakše administratorski panel kreiran je tako da bude lak i bezbjedan za upotrebu. Ovaj panel sadrži skup od nekoliko jednostavnih i nekoliko naprednih funkcionalnosti, a sve to u cilju lakšeg i efikasnijeg ažuriranja sadržaja. U nastavku će se detaljno opisati neki djelovi administratorskog panela, dok će se ostali samo ukratko spomenuti, zbog obima rada. Uz to, biće prikazan i izgled pojedinih djelova, kao i programski kodovi.

Administratorski panel je hijerarhijski podijeljen na djelove i radilo se na tome da bude što pregledniji. Napravljene su samo one neophodne funkcionalnosti, a ostavljeno je prostora za određena unaprijeđenja. Pristup panelu je jednostavan, kroz javni dio platforme, a meni je kreiran kao bočni stubac na lijevoj strani panela. Dalji opis biće podijeljen na nekoliko djelova:

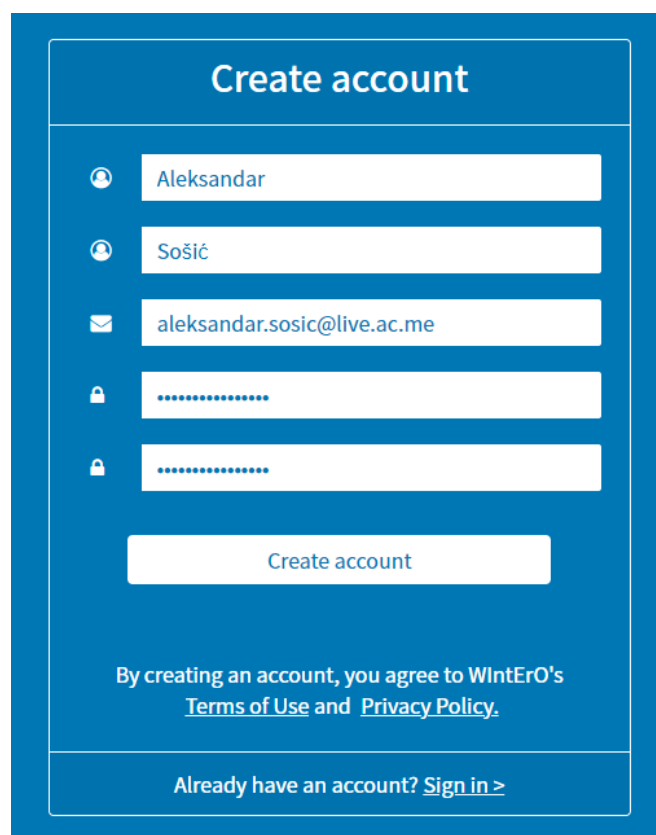
- kreiranje naloga i pregled menija,
- prikaz dijela koji se odnosi na profil korisnika,
- prikaz načina upisivanja sadržaja,
- prikaz dijela sadržaja koji je na čekanju,
- prikaz dijela koji se odnosi na korisnike platforme.

Kreiranje naloga i pregled menija

Kreiranje novog naloga je jednostavno, neophodno je unijeti podatke u nekoliko polja. Korisnik pristupa stranici za kreiranje kroz početnu stranu platforme. Nakon što unese potrebne informacije, sledeći korak je potvrda validnosti *e-mail* adrese, a nakon toga prelazak na stranicu putem koje se vrši prijava i pristup korisničkom panelu. Izgled forme za kreiranje naloga dat je na slici 7. Polja za unos su obavezna, a ukoliko jedno od njih nakon slanja podataka nije popunjeno, stranica se opet učitava, prikazuje već unešene podatke i prikazuje poruku da je potrebno unijeti informacije i u tom polju. HTML5 ima već definisane tipove podataka za određena polja, a ta mogućnost je i iskorištena za kreiranje forme. Tri tipa podataka su upotrijebljena, a to su: *text*, *email* i *password*. Na stranici je ugrađena i *PHPMailer*²⁴ biblioteka koja služi za slanje linka na *e-mail* adresu kako bi se izvršila validacija i potvrda ispravne verifikacije. Ova biblioteka je korištena i na još par mjesta tokom izrade platforme. Radi se o veoma popularnoj biblioteci za slanje *e-mail* poruka kroz PHP programski jezik. Besplatna je za korišćenje, ima dosta integrisanih funkcija, a kompatibilna je i sa nekim starijim verzijama PHP jezika. Upotrijebljene su i neke PHP i MySQL funkcije, a najzanimljivija od njih je *md5()* funkcija koja ima zadatak da kodira unešene podatke potrebne za kreiranje naloga. Dio PHP koda stranice za kreiranje naloga je prikazan na slici 8. Iz

²⁴ PHPMailer biblioteka, izvorni kod i dokumentacija - <https://github.com/PHPMailer/PHPMailer>

priloženog se može vidjeti način na koji se vrši upis podataka u bazu. Funkcija šifrira *e-mail* adresu i *password* kako bi bilo omogućeno ispravno verifikovanje, odnosno čuvanje *password* kombinacije u bazi podataka. Verifikovanje se vrši na način što se *e-mail* adresa kodira sa MD5 algoritmom²⁵, zatim upisuje u bazu podataka, a potom se nakon što korisnik klikne na link dobijen u poruci vrši poređenje i tako utvrđuje autentifikacija. Kada su u pitanju lozinke korisničkih naloga, takođe se vrši enkripcija, i tako enkriptovani podaci se upisuju u bazu podataka. Na taj način nije moguće pristupom podacima u bazi otkriti stvarnu lozinku koja se koristi za pristup. Dekripcija se vrši onda kada vlasnik korisničkog naloga želi da pristupi korisničkom panelu, tako što unosi svoju lozinku koja se opet enkriptuje i upoređuje sa već postojećom upisanom enkriptovanom lozinkom.



Slika 7: Prikaz forme za kreiranje naloga

Korisnički dio platforme sadrži i meni, odnosno tri različita menija u zavisnosti od toga sa kojim privilegijama mu pristupa korisnik, a koji su prikazani na slici 9. Pored već navedenih djelova koji će se detaljnije opisati, iz prikaza slike se vidi da postoji još nekoliko interesantnih mogućnosti. Dio *My Entries* se odnosi na one unose koje je korisnik poslao i koji su odobreni od

²⁵ MD5 algoritam, dokumentacija - <http://www.faqs.org/rfcs/rfc1321.html>

strane administratora, dok dio *On Hold* predstavlja sadržaj koji je na čekanju. Razlog što se opisuje ovaj drugi dio je taj što on govori i o prihvatanju, odnosno odbijanju poslatih podataka, a ne prikazuje samo listu kao prethodni. U meniju se može vidjeti i sekcija za objavljivanje publikacija vezanih za IntErO i WIntErO alate. Takođe, postoji i mogućnost dodavanja fotografija i kreiranje albuma sa fotografijama za lokacije na kojima su vršena mjerenja na terenu. Ovo je nova funkcionalnost koja nije postojala u IntErO paketu, a omogućava korisnicima da uvijek imaju pristup fotografijama sa terena i da kasnije mogu da vrše poređenja. U *Settings* dijelu postoji opcija koja dozvoljava promjenu *e-mail* adrese i lozinke. Promjenu lozinke takođe je moguće izvršiti i na stranici koja služi za prijavljivanje, klikom na dugme *Reset it*.

```

if(isset($_POST) & !empty($_POST)){

    $firstname = mysqli_real_escape_string($connection, $_POST['firstname']);
    $lastname = mysqli_real_escape_string($connection, $_POST['lastname']);
    $email = mysqli_real_escape_string($connection, $_POST['emailuser']);

    $verification_key = md5($email);

    $password = md5($_POST['passworduser']);
    $passwordrepeat = md5($_POST['passworduserrepeat']);

    if ($password == $passwordrepeat) {

        $fmsg = "";
        $smsg = "";
        $emailsql = "SELECT * FROM register WHERE email = '$email'";
        $emailresult = mysqli_query($connection, $emailsql);
        $emailcount = mysqli_num_rows($emailresult);

        if ($emailcount == 1) {

            $fmsg .= "Account with that E-mail address already exist! Please
            use different E-mail address!";

        } else {

            $sql = "INSERT INTO register (firstname, lastname, email,
            password, verification_key) VALUES ('$firstname', '$lastname', '$email', '$password',
            '$verification_key')";

            $result = mysqli_query($connection, $sql);

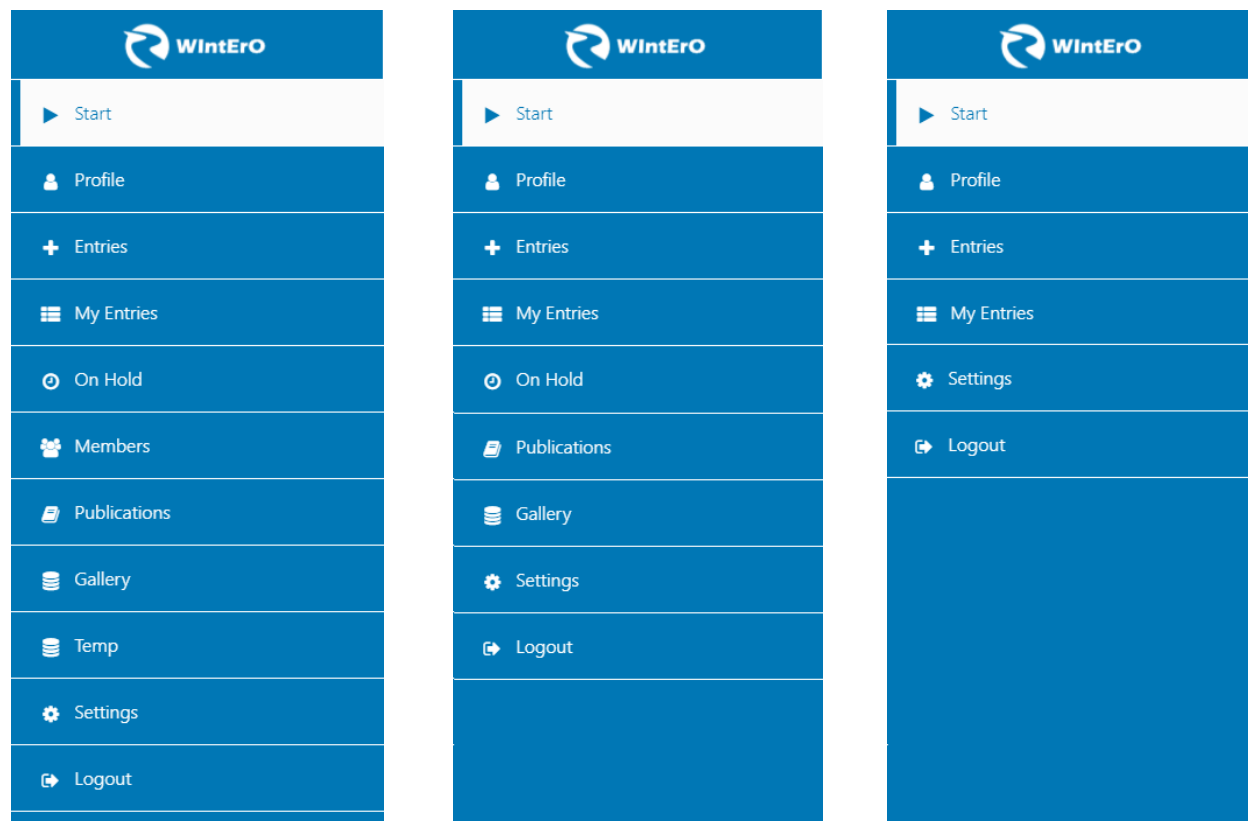
```

Slika 8: Dio PHP koda stranice za kreiranje naloga

Meni je kreiran u jednoj PHP datoteci i uključen (*include*²⁶) u ostale stranice. Na taj način olakšan je postupak eventualnih korekcija. Meni koji je prikazan na slici 9a, je meni kojem super administratori mogu da pristupe. „Obični“ administratori, (slika 9b), nemaju pristup i pregled

²⁶ PHP naredba *include* se koristi kako bi se sadržaj jedne PHP datoteke umetnuo u drugu PHP datoteku.

sekcije koja se odnosi na članove, odnosno korisnike platforme. Korisnici sa privilegijom „Gosti“ (slika 9c) nemaju pristup sekcijama *On Hold*, *Members*, *Publications* i *Gallery*.



Slika 9: Prikaz menija (a, b i c slike, respektivno)

Prikaz dijela koji se odnosi na profil korisnika

Uobičajeno je da korisnički profil bilo da se radi o veb platformi ili nekim drugim platformama, donosi niz mogućnosti krajnjim korisnicima. Kreiranjem profila na platformi koja je tema ovog rada moguće je pristupiti prethodno pomenutom meniju, a dalje i koristiti ostale mogućnosti. Stranica *Profile* sadrži osnovne informacije o korisniku, i njen prikaz je dat na slici 10. Stranica je podijeljena na dva dijela, prvi dio sadrži sliku korisnika i opciju za izmjenu iste, ime i prezime, u zavisnosti od prava pristupa sadržaju prikazana je i uloga korisnika, a na kraju se nalazi i sistemaska poruka. Drugi dio sekcije sadrži informacije o korisniku i to: *first name*, *last name*, *e-mail*, *date of birth*, *city*, *country*, *profession* i *company&university*. Sva ova polja je moguće korigovati, osim polja koje služi za *e-mail* adresu.

Princip upisa podataka u bazu je sličan kao i kada je u pitanju kreiranje naloga. Takođe je upotrijebljena i *PHP Mailer* biblioteka koja ovog puta ima zadatak da obavijesti korisnika da je došlo do izmjena informacija. Tipovi *input* polja su različiti, dakle korištena su tri tipa, a to su: *text*, *email*, i *date*. Ugrađen je i HTML element *select* koji služi za kreiranje padajuće liste.

Slika 10: Prikaz stranice Profile

Korisnički profil je moguće dalje nadograđivati, kako radi samih informacija na stranici, tako i u smislu povezivanja sa drugim djelovima platforme i drugim korisnicima. U ovoj fazi razvoja korisnički profil nije javno vidljiv, osim za administratore i super administratore sistema, i naravno svaki vlasnik profila ima uvid u isti. Tokom daljeg razvoja moglo bi se obezbijediti niz novih mogućnosti. Na primjer, moguće bi bilo kreirati sadržaj koji je javno dostupan na način što bi se evidencija kreatora sadržaja javno prikazivala. To bi dovelo do toga da pored sadržaja stoji *e-mail* adresa ili ime i prezime korisnika koji je postavio određene informacije na platformi. Dalje, klikom na jedan od navedenih podataka otvarao bi se profil korisnika sa informacijama koje bi, recimo, bile javno vidljive. Na ovaj način, došlo bi se i do mogućnosti da korisnici međusobno kontaktiraju, prave grupe ljudi koje povezuju zajednička interesovanja i slično, što bi moglo doprinijeti boljoj povezanosti istraživača međusobno, kao i istraživača i drugih korisnika platforme.

Prikaz načina upisivanja sadržaja

Kao što je već kazano ranije, kada se govorilo o analizi projektovanog rješenja, postoje dvije mogućnosti za upisivanje sadržaja o rječnim slivovima. Obje mogućnosti se vrše kroz kreirane HTML forme. Prva mogućnost je napravljena kako bi se obezbijedilo upisivanje novih podataka, dok druga opcija omogućava isčitavanje postojećih *.sli* datoteka i dopunsko upisivanje nedostajućih informacija, odnosno onih dodatnih informacija koje su novina u odnosu na staro rješenje. Pošto postoje dvije forme koje služe kako bi se podaci uspješno skladištili, izgled obje je približno isti uz neke male modifikacije. Detaljno će se u nastavku opisati obje mogućnosti, prikazaće se i nekoliko slika kako bi se što jasnije prezentovalo sve ono što je bitno, a napisani programski kodovi za obje solucije se nalaze u dodatku A i dodatku B, koji se nalaze na kraju ovoga rada. Radi jasnijeg pregleda kodova dodati su i komentari pored nekih linija koda.

Kada je u pitanju prva solucija koja se odnosi na unos novih podataka, bilo je potrebno kreirati HTML formu koja će obezbijediti sve što je omogućavao i programski paket IntErO, plus i neka dopunska upisivanja. HTML forma je sastavni dio PHP fajla, a naravno isti fajl sadrži i JavaScript kodove, kao i CSS umetnute i ugrađene stilove. U dodatku A se nalazi dio programskog koda koji omogućava funkcionalnost ovog segmenta koji se odnosi na upis novih podataka. Ovdje će se opisati postupak kreiranja stranice za izvršavanje unosa, kao i prikazati njen izgled na slici 11. Dodatno će se navesti formule koje se koriste za dobijanje rezultata, a uz to predstavice se i polja koja se tiču logičkih promjena i upozorenja.

Da bi se što jasnije opisalo kompletno rješenje neophodno je prethodno predstaviti i polja za unos. Iz opisa će se vidjeti i da nijesu sva polja obavezna, već je moguće izostaviti neka od njih ukoliko postoji već dobijena vrijednost za neka druga polja. Inače, ova stranica se izvršava kao i prethodno opisane kada je u pitanju slanje podataka u bazu. I ovdje je upotrijebljena *PHPMailer* biblioteka koja obavlja korisnika o izvršenom slanju.

Ukupan broj vidljivih inputa za unos parametara iznosi 54, mada konačan broj zavisi od toga koliko ima koeficijenata kada su u pitanju dužine izohipsi i površine između susjednih izohipsi. Ukupan broj obaveznih unosa iznosi minimum 27. Postoji mogućnost izračunavanja vrijednosti poslednja tri inputa navedena u listi ispod, pa u tom slučaju postoji još 27 opcionih polja. Sledeći parametri su implementirani na stranici:

- 1) površina sliva (F);
- 2) dužina linije vododjelnice (O);
- 3) prirodna dužina matice glavnog vodotoka (L_v);
- 4) dužina sliva mjerena serijom paralelnih linija (L_b);
- 5) površina većeg dijela sliva (F_v);
- 6) površina manjeg dijela sliva (F_m);
- 7) ukupna dužina glavnog vodotoka sa pritokama I i II reda (ΣL);
- 8) nadmorska visina početne izohipse (h_o);
- 9) ekvidistanca (Δh);
- 10) dužina izohipsi (L_{iz});
- 11) površine između susjednih izohipsi (f_{iz}) – (*dva inputa*);
- 12) najniža kota u slivu (H_{min});
- 13) najviša kota u slivu (H_{max});
- 14) dio površine sliva koji je sastavljen od jako vodopropustljivih tvorevina iz stijena (f_p);
- 15) dio površine sliva koji je sastavljen od stijena srednje vodopropustljivosti (f_{pp});
- 16) dio površine sliva koji je sastavljen od stijena slabe vodopropustljivosti (f_o);
- 17) dio površine sliva pod šumom (f_s);
- 18) dio površine sliva pod travama, livadama, pašnjacima i voćnjacima (f_t);
- 19) dio površine sliva koji je pod goletima, oranicama i zemljištima bez travne vegetacije (f_g);
- 20) najkraće rastojanje između dvorišta i ušća (L_m);
- 21) visina bujične kiše (h_b);
- 22) srednja godišnja temperatura vazduha (t_o);
- 23) srednja godišnja količina padavina (H_{god});

- 24) tipovi zemljišnih tvorevina i srodne vrste (Y);
- 25) koeficijent uređenja sliva (Xa);
- 26) brojni ekvivalent vidljivih i jasno izraženih procesa erozije zemljišta (ϕ).

Svako *input* polje sadrži oznaku kao i mjernu jedinicu pored oznake, ukoliko postoji. Vidljiva širina polja je fiksna sva polja, dok se opis pojavljuje onda kada se na *input* polje postavi kursor miša. Polja su većinom tipa brojne vrijednosti, dok postoji i tekstualni ili opisni tip na dva mjesta. Na stranici su ugrađene i tri padajuće liste. Dodat je i HTML metod *insertRow()* koji se koristi za kreiranje novih redova u tabeli i služi kako bi se unijeli podaci o dužinama izohipsi i njihovim susjednim površinama. Metod *insertRow()* funkcionira zajedno sa skriptnim kodom. Ugrađen je i element *accordion*, koji omogućava kreiranje sklopivog sadržaja. Takođe, i ovaj element je nemoguće kreirati bez JavaScript koda. Na slici 12. je prikazan izgled ovog dijela stranice, a uz to na slici se i nalazi 27 opcioni polja koja služe kako bi se dobile vrijednosti za tri poslednja parametra sa prethodne liste, koeficijente (Y), (Xa) i (ϕ). Sva *input* polja sadrže i opis. U ovom dijelu je kreirana i mogućnost provjere parametara za ove koeficijente i opciono se nudi izvršenje automatske preraspodjele procenata po elementima do 100%. Iz navedenih *input* polja moguće je dobiti sledeća 22 rezultata:

- 1) koeficijent oblika sliva (A);
- 2) koeficijent razvijenosti vododjelnice (m);
- 3) srednja širina sliva (B);
- 4) (A)simetričnost sliva (a);
- 5) gustina rječne mreže sliva (G);
- 6) koeficijent vijugavosti toka (K);
- 7) srednja nadmorska visina sliva (H_{sr});
- 8) srednja visinska razlika sliva (D);
- 9) srednji pad sliva (I_{sr});
- 10) visina lokalne erozije baze sliva (H_{leb});
- 11) koeficijent erozije energije reljefa sliva (E_r);
- 12) koeficijent vodopropusnosti područja (S1);
- 13) koeficijent vegetacionog pokrivača (S2);
- 14) analitički izraz retencije vode u doticaju (W);
- 15) energetski potencijal slivanja vode za vrijeme bujičnih kiša ($2gDF^{1/2}$);
- 16) maksimalno oticanje iz sliva (Q_{max});
- 17) temperaturni koeficijent područja (T);
- 18) koeficijent erozije sliva (Z);
- 19) produkcija erozionog materijala u slivu (W_{god});
- 20) koeficijent retencije nanosa (R_u);
- 21) stvarni gubici zemljišta (G_{god});
- 22) stvarni gubici zemljišta po km^2 (G_{god}/km^2).

Navedeni rezultati prikazaće se samo ukoliko se ispune svi uslovi koji se zahtijevaju prilikom upisa. Obavezna input polja ne smiju biti prazna, a dodatno stranica se neće izvršiti i ako se jave greške, odnosno ukoliko se upišu nelogični ulazni podaci.

Add new entry

Attention: before you send parameters in our database please check validity. Additional: to find precise google coordinates you can use [this](#) tool.

River basin

Country

Year

Latitude

Longitude

Geometric characteristics of the river basin

F (km²)

Fm (km²)

Lv (km)

O (km)

Fv (km²)

Lb (km)

Topographic characteristics of the river basin

Liz [km]:

f [km²]:

h0 (m)

Δh (m)

Hmin (m)

Hmax (m)

Hydrological characteristics of the river basin

ΣL (km)

Lm (km)

Water permeability

fp

fpp

fo

Land use

fs

ft

fg

Meteorological data

hb (mm)

Up (years)

to (°C)

Hgod (mm)

Erosion coefficients

Y = +

Xa = +

φ = +

Slika 11: Izgled stranice za unos novih podataka

Erosion coefficients

$Y = 0.91$

Sand, gravel and incoherent soils: %

Saline soils...: %

Decomposed limestone and marls: %

Serpentines, red sand stones, flishe deposits: %

Podzols and parapodzols, decomposed schist: %

Solid and Schist limestone, Terra Rosa and Humic soil: %

Brown forest soils and Mountain soils: %

Epieugleysol and Marshlands: %

Good structured Chernozems and alluvial well-structured deposits: %

Bare, compact igneous: %

$X_a = 0.46831$

Bare lands: %

Plough-lands: %

Orchards and vineyards: %

Mountain pastures: %

Meadows: %

Degraded forests: %

Well-constituted forests: %

$\phi = 0.32719$

Depth erosion: %

80% of the river basin under rill and gully erosion: %

50% of the river basin under rill and gully erosion: %

100% of the river basin under surface erosion: %

100% of the river basin under surface erosion, without visible furrows, ravines and land slides: %

50% of the river basin under surface erosion: %

20% of the river basin under surface erosion: %

There are smaller slides in the watercourse beds: %

The river basin mostly under plough-land: %

The river basin under forests and perennial vegetation: %

Slika 12: Izgled dijela stranice za nove unose, element accordion

Kada su u pitanju nelogičnosti prilikom unosa, na stranici je implementiran kôd koji obavezuje korisnika na sledeća pridržavanja:

- ukupan broj izohipsi mora da bude za jedan manji od ukupnog broja površina između izohipsi;
- zbir koeficijenata f_p , f_{pp} i f_o mora da iznosi 1, odnosno 100%;

- zbir koeficijenata f_s , f_t i f_g mora da iznosi 1, odnosno 100%;
- veza između f_s , f_t i f_g mora biti uspostavljena sa X_a :
 - *bare lands* (goleti) + *plough-lands* (oranice) = f_g ;
 - *orchards and vineyards* (voćnjaci i vinogradi) + *mountain pastures* (planinski pašnjaci) + *meadows* (livade) = f_t ;
 - *degraded forests* (degradirane šume) + *well constituted forests* (šume dobrog sklopa) = f_s ;
- ukupna površina sliva mora biti jednaka zbiru parametara F_v i F_m ;
- da nadmorska visina najniže izohipse mora biti manja ili jednaka sa najnižom kotom u slivu;
- da ukupna dužina glavnog vodotoka sa pritokama I i II reda, ΣL , mora biti veća ili jednaka dužini sliva po matici glavnog vodotoka, L_v ;
- da dužina sliva po matici glavnog vodotoka, L_v , mora biti veća ili jednaka sa najkraćim rastojanjem između izvorišta i ušća.

Logičko unošenje podataka koje je već postojalo i u prethodnim programskim alatima kreirano je uz upotrebu JavaScript jezika. Korisnik prilikom unosa podataka dobija upozorenje ukoliko postoji nelogičnost da je vrijednost koja je unesena u određenom polju logički neispravna u odnosu na neko drugo određeno polje. Ovakva rješenja u mnogome potpomažu pri unosu parametara i samim tim doprinose kvalitetnijem korisničkom interfejsu.

Sve formule koje su neophodne za dobijanje rezultata su napisane u PHP jeziku. Vrijednosti se pamte u promenljivim i na taj način prikazuju na stranici *Review* koja omogućava korisniku da pregleda rezultate prije konačnog slanja. Na slici 11. se vide dva dugmeta, *Reset* i *Review*. Klikom na *Review* dugme otvara se nova stranica, u novom prozoru, koja nudi tri opcije, *Go back*, *Compare Your results* i *Send parameters*. Opcija *Compare Your results* nudi korisniku da uporedi rezultate koje želi da pošalje sa nekim prethodnim rezultatima koji su već upisani u bazu. Na taj način moguće je imati uvid u eventualna odstupanja i promjene. Izgled stranice dat je na slici 13. Slijedi prikaz svih upotrijebljenih formula [7]:

$$A = 0.195 * \frac{O}{L_v}; \quad (2)$$

$$m = \frac{O}{2 * \sqrt{\pi * F}}; \quad (3)$$

$$B = \frac{F}{L_b}; \quad (4)$$

$$a = \frac{F_v - F_m}{0.5 * (F_v + F_m)}; \quad (5)$$

$$G = \frac{\Sigma L}{F}; \quad (6)$$

Here are results for Ljesnica (Bijelo Polje) river basin. If these results are correct send parameters in our database, if not please go back and check inputs.

Coefficient of the river basin form	$A = 0.46380788079744$
Coefficient of the watershed development	$m = 0.61818121763477$
Average river basin width	$B = 3.1290967768987 \text{ km}$
(A)symmetry of the river basin	$a = 0.29523846285075$
Density of the river network of the basin	$G = 0.42265224585092$
Coefficient of the river basin tortuousness	$K = 1.1372654647404$
Average river basin altitude	$H_{sr} = 955.49673985673 \text{ m}$
Average elevation difference of the river basin	$D = 383.49673985673 \text{ m}$
Average river basin decline	$I_{sr} = 34.565616558272 \%$
The height of the local erosion base of the river basin	$H_{leb} = 931 \text{ m}$
Coefficient of the erosion energy of the river basin's relief	$E_r = 111.44367319391$
Coefficient of the region's permeability	$S_1 = 0.99262$
Coefficient of the vegetation cover	$S_2 = 0.6958875576$
Analytical presentation of the water retention in inflow	$W = 1.6314539156074 \text{ m}$
Energetic potential of water flow during torrent rains	$2gDF^{1/2} = 613.36518311582 \text{ m km s}^{-1}$
Maximal outflow from the river basin	$Q_{max} = 320.59308252475 \text{ m}^3 \text{ s}^{-1}$
Temperature coefficient of the region	$T = 0.99498743710662$
Coefficient of the river basin erosion	$Z = 0.38998746060057$
Production of erosion material in the river basin	$W_{god} = 34003.085683756 \text{ m}^3 \text{ god}^{-1}$
Coefficient of the deposit retention	$R_u = 0.2949181023796$
Real soil losses	$G_{god} = 10028.125504904 \text{ m}^3 \text{ god}^{-1}$
Real soil losses per km ²	$G_{god} \text{ km}^2 = 200.55902037113 \text{ m}^3 \text{ km}^2 \text{ god}^{-1}$

Slika 13: Izgled stranice koja se odnosi na pregled rezultata prije slanja

$$K = \frac{Lv}{Lm}; \quad (7)$$

$$H_{sr} = \frac{\Sigma (h*f)}{F}; \quad (8)$$

$$D = H_{sr} - H_{min}; \quad (9)$$

$$I_{sr} = \frac{\Sigma (L_{sr} * \Delta h)}{F}; \quad (10)$$

$$H_{leb} = H_{max} - H_{min}; \quad (11)$$

$$E_r = \frac{H_{leb}}{\pi * \sqrt[4]{F}}; \quad (12)$$

$$S_1 = 0.4 * f_p + 0.7 * f_{pp} + 1 * f_o; \quad (13)$$

$$S_2 = 0.6 * f_s + 0.8 * f_t + 1 * f_g; \quad (14)$$

$$W = H_b * (15 - 22 * H_b - 0.3 * \sqrt{L}); \quad (15)$$

$$2gDF^{1/2} = \sqrt{2 * \frac{9.81}{1000} * \frac{D}{1000} * F^{0.5} * 1000}; \quad (16)$$

$$Q_{max} = A * S_1 * S_2 * W * \sqrt{2 * g * D * F}; \quad (17)$$

$$T = \sqrt{\frac{t_0}{10}} + 0.1; \quad (18)$$

$$Z = Y * X * a * (\varphi + \sqrt{I_{sr}}); \quad (19)$$

$$W_{god} = T * H_{god} * \pi * \sqrt{Z^3} * F; \quad (20)$$

$$R_u = \frac{(O*D)^{0.5}}{0.25*(L+10)}; \quad (21)$$

$$G_{god} = W_{god} * R_u; \quad (22)$$

$$G_{god} km^{-2} = T * H_{god} * \pi * \sqrt{Z^3} * F * R_u; \quad (23)$$

U nastavku će se prikazati tabele koje sadrže brojne vrijednosti prema kojima se izračunavaju koeficijenti (Y), (Xa) i (φ) [7]:

Tabela 1: Srednje vrijednosti koeficijenta (Y)

Tipovi zemljišnih tvorevina i srodne vrste	Srednja vrijednost
Pijeskovi, šljunak i nevezana zemljišta	2,0
Les, tufovi, slatine i stepska zemljišta	1,6
Raspadnuti krečnjaci i laporci	1,2
Serpentini, crveni peščari i flišne naslage	1,1
Podzoli i parapodzoli, raspadnuti škriljci	1,0
Jedri i škriljasti krečnjaci i crvenice i humosnosilikatna zemljišta	0,9
Gajnjače i planinska zemljišta	0,8
Smonice, ritske oranice i močvarna zemljišta	0,6
Čermozemi i aluvijalni nanosi dobre strukture	0,5
Goli, kompaktni eruptivi	0,25

Tabela 2: Srednje vrijednosti koeficijenta (Xa)

	X	a	Xa
Goleti	1,0	1,0	1,0
Oranice	0,9	1,0	0,9
Voćnjaci i vinogradi	0,7	1,0	0,7
Planinski pašnjaci	0,6	1,0	0,6
Livade	0,4	1,0	0,4
Degradirane šume	0,6	1,0	0,6
Šume dobrog sklopa	0,05	1,0	0,05

Tabela 3: Srednje vrijednosti koeficijenta (ϕ)

Dubinska erozija	1,0
80% sliva pod bradastom i jaružastom erozijom	0,9
50% sliva pod bradastom i jaružastom erozijom	0,8
100% sliva pod površinskom erozijom	0,7
100% sliva pod površinskom erozijom bez vidljivih brazda, jaruga i odrona	0,6
50% sliva pod površinskom erozijom	0,5
20% sliva pod površinskom erozijom	0,3
U koritima vodotoka ima manjih kliženja	0,2
Sliv pretežno pod oranicama	0,15
Slim pod šumama i višegodišnjom vegetacijom	0,1

U prethodno prikazanim tabelama navedene su sve postojeće vrijednosti koje se upotrebljavaju za proračun. Na slici 12. se takođe nalaze opisi koeficijenata, ali su ispisani na engleskom jeziku. Treba napomenuti i to da se sve formule za izračunavanje nalaze u fajlu *review.php* koji se izvršava nakon što korisnik klikne na dugme *Send parameters*. Početna stranica je *entrynew.php*.

Druga solucija za unos parametara u bazu je učitavanjem *.sli* datoteka. Ovaj način je već prethodno pomenut kada se govorilo o IntErO paketu. Dakle, postojeće *.sli* datoteke trebalo je prvo očitati na stranici na kojoj bi se dalje podaci razvrstali na pravilan način, a onda upisali u promljenjive i dalje slali u bazu. Ovo je ujedno bio i najkomplikovaniji dio posla. Svi ostali postupci su gotovo identični kao i kod novih unosa. Zapravo, stranica koja vrši očitavanje *.sli* datoteka ima čak i nešto manje zadataka s obzirom da se ispravni parametri već nalaze u datotekama. Nije potrebno provjeravati logičnosti prilikom upisivanja. U dodatku B se nalazi kôd koji detaljnije prikazuje proceduru isčitavanja i slanja podataka.

Ovaj *.sli* format, inače nestandardni format datoteka, sadrži minimum 27 ulaznih podataka, a u većini slučajeva sadrži i 27 opcionih. Podaci se nalaze na linijama, ali redosled tih linija nije baš jednostavan. Takođe, broj linija na kojima se nalaze podaci neophodni za dobijanje ispravnih rezultata nije fiksna, već je promjenjiv. Na slici 14. dat je prikaz sadržaja *.sli* fajla koji je otvoren pomoću tekstualnog editora. Pregled ovih podataka je bio neophodan za početak kreiranja stranice za očitavanje *.sli* datoteka čiji prikaz je dat na slici 15.

```
Polimlje (2018).sli - Notepad
File Edit Format View Help
IntErO
22
A
m
B
a_
G
K
Hsr
D
Isr
Hleb
Er
S1
S2
W
2gDF
Qmax
T
Z
Hgod
Ru
Ggod
Ggodkm2
Polimlje      2018
2900
346
149
160
1563
1337
21
20
107.818
380.0972
690.2956
835.9832
947.6948
936.9316
989.3216
778.286
<
Ln 41, Col 8  100%  Windows (CRLF)  UTF-8

Polimlje (2018).sli - Notepad
File Edit Format View Help
695.4044
627.1052
519.7708
417.198
364.3988
207.4
122.39
48.68
18.9
5.66
1.16
22
384.914
34.279
114.86
182.351
223.497
244.387
245.143
286.661
272.826
206.699
188.871
156.713
116.831
100.169
66.405
42.341
21.315
7.498
3.349
0.6
0.29
500
100
436
2461
1150
122
Ln 82, Col 4  100%  Windows (CRLF)  UTF-8

Polimlje (2018).sli - Notepad
File Edit Format View Help
0
0.14
0.86
0.52930
0.39540
0.07520
62.3
8.5
873
1.02122
9.48
1.72
1.03
44.48
2.59
9.66
11.72
6.21
10.34
2.76
0.422465
0.17
7.35
2.94
18.93
17.67
22.42
30.51
0.323285
4.02
3.97
3.55
2.97
19.97
1.7
1.05
0.31
7.23
55.23
1500
Ln 122, Col 1  100%  Windows (CRLF)  UTF-8
```

Slika 14: Prikaz sadržaja *.sli* fajla koji je otvoren pomoću Notepad tekstualnog editora

Add entry with .sli file

Instructions: For sending parameters please upload .sli file first. After uploading is done please fill up all fields in red. Attention: before you send parameters in our database please check validity. Additional: to find precise google coordinates you can use [this tool](#).

Choose File No file chosen

River basin Polimlje	Country Serbia	Year 2018	Latitude 43.522647	Longitude 19.576901
-------------------------	-------------------	--------------	-----------------------	------------------------

Geometric characteristics of the river basin

F (km ²) 2900	Fm (km ²) 1337	Lv (km) 149
O (km) 346	Fv (km ²) 1563	Lb (km) 160

Topographic characteristics of the river basin

Liz (km):	f (km ²):	h0 (m) 500
20	384.914	Δh (m) 100
107.818	34.279	Hmin (m) 436
380.0972	114.86	Hmax (m) 2461
690.2956	182.351	
835.9832	223.497	
947.6948	244.387	
936.9316	245.143	
989.3216	286.661	
778.286	272.826	
695.4044	206.699	
627.1052	188.871	
519.7708	156.713	
417.198	116.831	
364.3988	100.169	
207.4	66.405	
122.39	42.341	
48.68	21.315	
18.9	7.498	
5.66	3.249	
1.16	0.6	
	0.29	

Hydrological characteristics of the river basin

ΣL (km) 1150	Lm (km) 122
-----------------	----------------

Water permeability

fp 0	fpp 0.14	fo 0.86
---------	-------------	------------

Land use

fs 0.52930	ft 0.39540	fg 0.07520
---------------	---------------	---------------

Meteorological data

hb (mm) 62.3	Up (years) 100	to (°C) 8.5	Hgod (mm) 873
-----------------	-------------------	----------------	------------------

Erosion coefficients

Y = 1.02122	+
Xa = 0.422465	+
φ = 0.323285	+

Reset
Review

Slika 15: Prikaz stranice koja služi za očitavanje .sli datoteka

Stranica koja je prikazana na slici 15. je veoma slična stranici čiji je izgled dat na slici 11. Vizuelna razlika se odnosi na dvije stvari. Prva je ta što stranica za upisivanje novog sadržaja nema formu koja omogućava otpremanje datoteke (*File Upload*), a druga jer stranica za očitavanje *.sli* datoteka nema HTML metod *insertRow()* već se podaci o dužinama izohipsi i njihovim susjednim površinama prikazuju u tabelama. Na stranici je potrebno unijeti i podatke za pet polja koja se nalaze u prvom redu, isto kao i kada je u pitanju prva opcija za slanje podataka. Nakon što se obavezna polja ispune i klinke se na dugme *Review*, nastavlja se rad i procedura je potpuno identična kao i kod upisivanja novih podataka.

Prikaz dijela sadržaja koji je na čekanju

Podaci koje korisnici platforme šalju prvo moraju proći postupak provjere od strane administratora. Nakon slanja, podaci se upisuju u bazu podataka, ali nijesu javno vidljivi na platformi. Ideja za uvođenjem ovog koraka proistekla je iz analize korisničkog zahtjeva i potrebe da podaci budu dostupni korisniku koji ih je unio radi mogućih izmjena i analiza, kao i da se u slučaju veoma intenzivnog unosa podataka ne bi preopteretili administratori sistema, na način da moraju da reaguju na sve zahtjeve u veoma kratkom roku. Ukoliko administratori odobre prikazivanje, podaci se istog trenutka prikazuju na javnom dijelu platforme i uvid u iste imaju svi posjetioci, dok se u suprotnom podaci brišu iz baze. Administratori i super administratori imaju pristup stranici *On Hold* i na istoj se nalaze svi poslani rječni slivovi koji su na čekanju i koji treba da prođu proceduru odobravanja. Na slici 16. data je stranica sa primjerom koji prikazuje pet slivova koji su na čekanju. Podaci su smješteni u tabeli i uz ime rječnog sliva nalaze se još neki opisni podaci, ime korisnika koji ih je poslao i dugme *Preview* koje služi kako bi se pregledali ulazni parametri i rezultati. Na stranici *Preview* se nalaze i tri dugmeta. *Go back* služi za prelazak na prethodnu stranu, *Decline* dugme vrši brisanje podataka ukoliko nijesu adekvatni, a klikom na *Accept* dugme prihvata se upis i isti postaje javno vidljiv.



The screenshot shows a web page titled "On Hold" with a table containing five rows of data. Each row represents a river basin with columns for ID, River basin, Country, Year, User, and a Preview button.

ID	River basin	Country	Year	User	
1	Djuricka rijeka	Montenegro	2018	aleksandar.sosic@wintero.me	Preview
2	Komaracka rijeka	Montenegro	2018	aleksandar.sosic@wintero.me	Preview
3	Bijeli Potok	Montenegro	2018	aleksandar.sosic@wintero.me	Preview
4	Novsicki potok	Montenegro	2018	aleksandar.sosic@wintero.me	Preview
5	Velicka rijeka	Montenegro	2018	aleksandar.sosic@wintero.me	Preview

Slika 16: Prikaz stranice *On Hold*

Prikazivanje podataka koji su na čekanju se vrši tako što se PHP i MySQL naredbama definiše da se prikažu oni podaci koji ispunjavaju određene kriterijume. Dio programskog koda je

dat na slici 17. U ovom slučaju rješenje se kreiralo tako što se prilikom slanja podataka u bazu u tabeli *data*, u polju *active* upisuje cifra 0. Nakon odobravanja od strane administratora, cifra 0 se zamjenjuje sa cifrom 1. U nastavku, kada se bude govorilo o javno dostupnom okruženju objasniće se i postupak za javno prikazivanje podataka. Izmjene vrijednosti u poljima kada je u pitanju MySQL vrše se naredbom *UPDATE*, dok se brisanje vrši naredbom *DELETE*. Na veoma sličan način kreirana je i stranica *My Entries*.

```

<?php
    session_start();
    require_once('connect.php');

    $i = 1;
    $sql = "Select id, riverbasin, country, year, active, emailuser FROM data WHERE
active = '0'";
    $result = mysqli_query($connection, $sql);

    if ($result-> num_rows > 0) {
        while ($row = $result-> fetch_assoc() ) {
            echo "<tr>";
            echo "<td>".$i."</td>"; $i = $i+1;
            echo "<td>".$row["riverbasin"]."</td>";
            echo "<td>".$row["country"]."</td>";
            echo "<td>".$row["year"]."</td>";
            echo "<td>".$row["emailuser"]."</td>";
            echo "<td><a href='onholdpreview.php?id=".$row['id']."'
>Preview</a></td></td>";
            echo "</tr>";
        }
        echo "</table>";
    }

    else {
        echo "There is no new entries!";
    }
?>

```

Slika 17: Dio programskog koda stranice za prikazivanje sadržaja koji je na čekanju

Prikaz dijela koji se odnosi na korisnike platforme

Kada je u pitanju korisnički dio platforme, ranije je rečeno da je bilo potrebno uspostaviti sistem sa trostepenim ulogama. Pristup ovom dijelu platforme imaju samo super administratori sistema. Njihova uloga je u tome što vrše kontrolu nad svim ostalim korisničkim nalogima. Super administratori imaju prava da korisnike učine administratorima i suprotno. Imaju mogućnost i da izbrišu nalog ukoliko je to potrebno, kao i da izvrše uvid u isti i pregledaju osnovne informacije koje je korisnik naveo pri kreiranju naloga ili kasnije napisao.

Moguća su dodatna unaprijeđenja na ovoj stranici. U zavisnosti od toga kolika bi bila upotreba ove platforme, moguće bi bilo da administratori budu podijeljeni po kontinentima, regionima ili državama, gdje bi recimo nekoliko administratora pregledalo prispele zahtjeve sa jednog kontinenta, dok bi druga grupa pregledala zahtjeve sa drugog i slično. Dodatno, moguće bi bilo napraviti i forum gdje bi se zainteresovani obraćali jedni drugima kako bi razmjenjivali informacije.

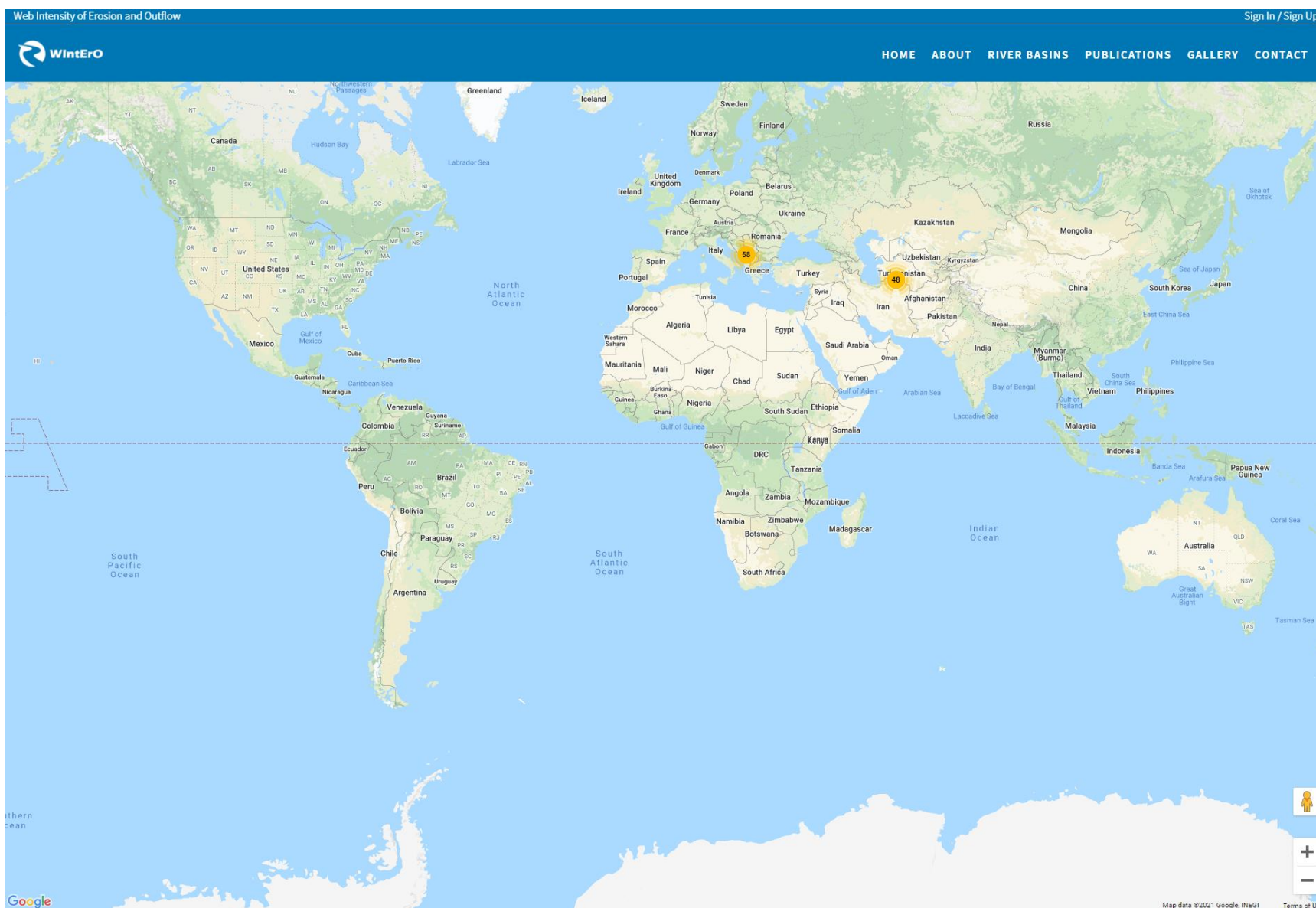
3.3.3 Razvoj javno-dostupnog okruženja

Razvoj javno-dostupnog okruženja omogućava uvid u podatke o rječnim slivovima, kao i u druge interesantne podatke zainteresovanim pojedincima i organizacijama. Pregled sadržaja koji je javno dostupan je kreiran tako da bude jednostavan i jasan za posjetioce platforme. Glavni cilj ovog dijela je bio da se podaci o rječnim slivovima prikažu na razumljiv način. Na platformi su isto tako dostupne i još neke informacije. Dakle, moguće je naći osnovne informacije koje se tiču WIntErO platforme, a uz to dostupne su i publikacije u kojima se govori o WIntErO platformi ili IntErO paketu. Takođe, dostupna je i galerija sa slikama, kao i kontakt informacije. U nastavku će se prikazati postupak izrade djelova koji se tiču javno-dostupnog sadržaja. Uz to, prikazaće se i programski kodovi. Dalji opis će biti podijeljen na sledeće delove, i to:

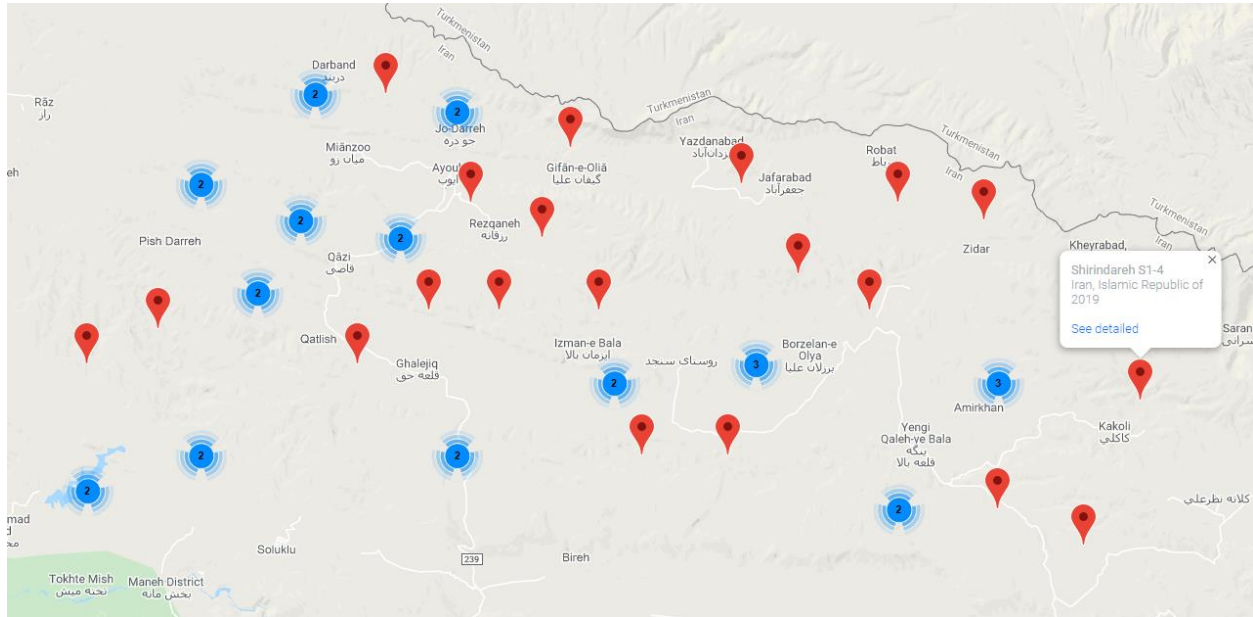
- razvoj dijela koji prikazuje lokacije rječnih slivova na *Google* mapi,
- prikaz rječnih slivova podijeljenih po državama,
- prikaz svih informacija o određenom rječnom slivu,
- eksportovanje podataka u najčešće korištene formate.

Razvoj dijela koji prikazuje lokacije rječnih slivova na *Google* mapi

Prikaz rječnih slivova na *Google* mapi omogućava korisnicima da brže i jednostavnije pronađu određeni sliv. Jednostavnim pregledom i zumiranjem mape moguće je vršiti uvid u lokacije slivova koji su veoma precizno označeni. Klikom na određeni tag koji obilježava odgovarajući sliv otvara se mali prozor u kome se nalaze najosnovnije informacije i dugme za prelazak na detaljni prikaz. *Google* mapa je ugrađena na početnoj stranici platforme i nalazi se odmah ispod glavnog menija. Takođe, mapa se može vidjeti i na stranici *River basins* na kojoj se nalazi lista rječnih slivova. Na slici 18. prikazana je mapa i glavni meni. Ovdje je prikaz mape definisan na način da se vidi kompletna karta svijeta. Zbog same razmjere mape nemoguće je prikazati sve tagove lokacija tako da budu jasno vidljivi, ukoliko je broj istih veliki i samo fizičko rastojanje kratko, onda kada je mapa prikazana u stoprocentnoj širini i visini. Iz tog razloga omogućena je dodatna vrsta tagova koja grupiše tagove lokacija tako što se upravlja prema razmjeri i mogućnosti pregleda istih. Dakle, tagovi koji vrše grupisanje su podijeljeni prema rastojanju i broju tagova lokacija i samim tim doprinose lakšem pregledu. U zavisnosti od broja tagova lokacija, tagovi za grupisanje imaju i drugačiju boju, a takođe se u sredini istih nalazi i tačan broj tagova lokacija. Slika 18. i slika 19. jasno pokazuju ove mogućnosti.



Slika 18: Prikaz glavnog menija i Google mape sa tagovima



Slika 19: Prikaz Google mape sa tagovima za područje Irana (48 slivova)

Kreiranje *Google* mape sa tagovima na WIntErO platformi realizovano je pomoću *Google Maps* platforme. *Google Maps* platforma obezbeđuje niz mogućnosti koje korisnici imaju samim njenim korišćenjem²⁷. Za potrebe izrade tagova lokacija na *Google* mapi upotrijebljena je *Maps JavaScript API* (*Application Programming Interface*) funkcionalnost²⁸. *Maps JavaScript API* omogućava uparivanje sopstvenog sadržaja sa *Google* mapom i implementaciju mape na veb stranicama i mobilnim i tablet uređajima.

U dodatku C se nalaze programski kodovi koji su korišteni za izradu. Naime, radi se o skriptnom kodu koji je ugrađen na stranici na kojoj se i prikazuje mapa i drugom PHP fajlu u kome se nalazi kôd kojim se izvršava preuzimanje potrebnih podataka iz baze. Skriptni dio koji je ugrađen na PHP stranici se povezuje sa PHP fajlom *outputxml.php*. I u ovom dodatku se nalaze komentari pored svih linija koda koje zahtijevaju dodatno obrazloženje.

Outputxml.php fajl zapravo služi kako bi se kreirao XML (*Extensible Markup Language*) fajl. XML je markerski jezik koji se koristi za definisanje formata podataka koji su nezavisni od platforme. Ovaj fajl je neophodno povezati sa bazom podataka kako bi se moglo pristupiti svim potrebnim informacijama. Nakon povezivanja potrebno je i definisati promjenjivu kojoj se zadaje naredba *DOMDocument*²⁹ koja omogućava kreiranje XML sadržaja. Zatim se nastavlja procedura upotrebom dva metoda, *createElement()* kojim se kreira element i *appendChild()* kojim se element upisuje u dokument. Na kraju se izvršava upisivanje svih elemenata u dokument upotrebom petlje *while*. Podaci koji se preuzimaju iz baze su podaci koji se odnose na identifikacioni broj – *id*, ime rječnog sliva, državu, godinu zabilježbe parametara i geografske koordinate.

²⁷ *Google Maps* platforma, dokumentacija - <https://developers.google.com/maps/documentation>

²⁸ *Maps JavaScript API*, dokumentacija - <https://developers.google.com/maps/documentation/javascript/overview>

²⁹ *DOMDocument* - <https://www.php.net/manual/en/class.domdocument.php>

Skriptni dio koda je u potpunosti prilagođen potrebama WIntErO platforme. Kompletan kôd je ugrađen između HTML *div* elemenata kojima se kreira strukturni blok na stranici. Na osnovu definisane klase za *div* element, u *.css* fajlu se deklariše stilski kôd kojim se uspostavlja odgovarajuća širina i visina bloka i samim tim dobija se željeni prikaz mape. Na samom početku skriptnog koda definisana je glavna funkcija *initMap()*. Izvršavanjem ove funkcije inicijalizuje se i dodaje mapa na stranici. Nakon toga potrebno je učitati podatke koji su dobijeni izvršavanjem *outputxml.php* fajla. U nastavku koda se nalaze naredbe kojima se deklariše prikaz učitanih podataka, geografskih koordinata i informacija. Geografske koordinate se koriste kako bi se izvršilo precizno pozicioniranje tagova lokacija. Informacije se odnose na prikaz podataka u malom prozoru koji se otvara klikom na određeni tag. Na dnu ovog prozora nalazi se i dugme *see detailed* koje omogućava prelazak na detaljni dio sa informacijama.

Grupisanje tagova radi efikasnijeg prikaza velikog broja tagova vrši se upotrebom *MarkerClustererPlus*³⁰ biblioteke u kombinaciji sa Maps JavaScript API platformom i samim tim stvaraju se grupe tagova. Klikom na tag određene grupe vrši se zumiranje i tako broj tagova opada i shodno tome pojavljuju se manje grupe tagova ili pojedinačni tagovi. Najbolji dokaz za navedeno vidljiv je na slici 19. Skriptni kôd sadrži linije koda kojima se izvršava i učitavanje biblioteke. Osim toga, u biblioteci su dostupne i slikovne datoteke različitih boja koje predstavljaju različite grupe tagova. Neophodno je na kraju i sprovesti učitavanje API koda navođenjem linka uz dodavanje atributa *async*. Pravilo je da se skripta koja se izvršava uz navođenje *async* atributa mora navesti nakon svih DOM elemenata³¹. Ovim atributom dozvoljava se da pretraživač učitava preostali dio stranice dok se vrši učitavanje API koda. Link koji se navodi je jedinstven jer u sebi sadrži takozvani *API key*. *API key* je jedinstveni identifikator kojim se potvrđuje autentičnost zahtjeva koji se povezuju sa WIntErO platformom. *API key* dobija se kreiranjem projekta na *Google Cloud Console* dijelu, koji je sastavni dio *Google Maps* platforme. Neophodno je imati projekat sa računom za obračun i navesti ime domena platforme kojoj je potreban *API key*³².

Prikaz rječnih slivova podijeljenih po državama

Stranica *River basins* na samom početku prikazuje tabelu koja sadrži sortirane informacije prema srodnosti podataka o rječnim slivovima, u ovom slučaju podijeljene po državama. Tabela ima tri kolone, prva se odnosi na naziv države, druga prikazuje broj slivova za istu državu, dok se u trećoj koloni nalaze dugmad koja omogućavaju prelazak na novu stranicu. Na slici 20. se nalazi pomenuta tabela. Prelaskom na novu stranicu, prikazuje se još jedna tabela koja sadrži spisak svih rječnih slivova za odabranu državu, a pored toga i podatke o godini zabilježbe parametara i na kraju dugmad koja služe za detaljan uvid u parametre, odnosno rezultate.

Ideja za ovakvo rješenje kada su u pitanju liste rječnih slivova proizilazi iz potrebe da se podaci pronalaze što brže i jednostavnije. Ovakav koncept tabela je veoma praktičan i ukoliko je potrebno vršiti sortiranje podataka prema nekim kriterijumima. Dalja podjela bi mogla da se

³⁰ *MarkerClustererPlus* biblioteka - <https://github.com/googlemaps/js-markerclustererplus>

³¹ *Async* atribut - <https://developers.google.com/maps/documentation/javascript/adding-a-google-map>

³² *Google Cloud Console* - <https://developers.google.com/maps/documentation/javascript/cloud-setup>

sprovede prema regijama većih država. Dalje, na primjer, Sjedinjene Američke Države u svom sastavu imaju 50 saveznih država. Podjela bi bila veoma značajna ukoliko bi došlo do unosa parametara za veći broj rječnih slivova. Ove dodatne podjele je moguće izvršiti modifikacijom postojećeg koda.

Country	Number of basins	Detailed
Iran, Islamic Republic of	48	OPEN ALL BASINS
Montenegro	57	OPEN ALL BASINS
Serbia	1	OPEN ALL BASINS

Slika 20: Prikaz tabele za sortiranje podataka po državama

Programski kôd koji je pisan za uspostavljanje ove dvije tabele je gotovo identičan kao i oni kodovi koji se koriste za prikaz lista koje se odnose na razvoj korisničkog okruženja. Naravno, postoje neke manje razlike kada su u pitanju kriterijumi prikazivanja pojedinih informacija. Na slici 21. prikazan je kôd koji se vezuje za prvu tabelu koja sortira podatke po državama. MySQL funkcija *COUNT()* je upotrijebljena kako bi se izvršilo prebrojavanje redova iz tabele *data*, gdje je jedini kriterijum polje *country*.

```

...
<table class="table table-bordered table-hover">
<tr class="headertables">
<th scope="col">Country</th>
<th scope="col">Number of basins</th>
<th scope="col">Detailed</th>

<?php
session_start();
require_once('connect.php');

$sql = "SELECT country, COUNT(*) FROM data GROUP BY country;";
$result = mysqli_query($connection, $sql);

if ($result-> num_rows > 0) {
    while ($row = $result-> fetch_assoc() ) {
        echo "<tr class='bodytables'>";
        echo "<td>".$row["country"]."</td>";
        echo "<td>".$row["COUNT(*)"]."</td>";
        echo "<td class='tablelink' ><a class='tablelinka'
href='basinscountriesall.php?country=".$row['country']."#riverbasins'>OPEN ALL
BASINS</a></td>";
        echo "</tr>";
    }
    echo "</table>";
}
else {
echo "No data!"; } $connection-> close();
?>

```

Slika 21: Prikaz programskog koda kojim se kreira tabela za sortiranje podataka po državama

Prikaz svih informacija o određenom rječnom slivu

Kako je već i opisano u prethodnim djelovima koji govore o javno dostupnom sadržaju, uvid u parametre i rezultate o određenom rječnom slivu moguće je izvršiti pronalaskom traženog sliva na *Google mapi*, kao i uvidom u tabelu koja je dostupna na stranici *River basins*. Otvaranjem stranice koja sadrži podatke o nekom rječnom slivu može se zapaziti da su na samom početku predstavljeni parametri, a nakon toga i rezultati. Na slici 22. prikazan je izgled dijela koji se odnosi na parametre, a koji je podijeljen u sedam djelova. Na samom početku, prije prikaza sadržaja nalaze se i osnovni podaci. Prikaz parametara je grupisan i obuhvaćen je elementom *accordion*, isto onako kako je i urađeno kod djelova platforme koji se tiču unosa parametara.

Geometric characteristics of the river basins			
F	2900	km ²	Surface area of the drainage basin
Fm	1337	km ²	Surface area of smaller portion of the drainage basin
Fv	1563	km ²	Surface area of greater portion of the drainage basin
O	346	km	The length of the watershed
Lv	149	km	Natural length of the main watercourse
Lb	160	km	Surface area of greater portion of the drainage basin

Slika 22: Izgled javno-dostupne stranice koja prikazuje parametre rječnog sliva

Kada je u pitanju prikazivanje rezultata na stranici, sadržaj je kreiran i dizajniran kao tabelarni, podaci su podijeljeni u redove i kolone, baš kao što je to slučaj i za prikaz parametara. Na slici 23. dat je izgled dijela stranice koji prikazuje rezultate odabranog rječnog sliva. Koncept programskog koda za izradu stranice na kojoj se nalaze podaci o rječnom slivu je veoma sličan sa kodovima koji su upotrijebljeni za izradu prethodno opisanih stranica i nije ga potrebno opet posebno predstavljati. Dovoljno je samo reći da se podaci preuzimaju iz postojeće baze podataka i dalje obrađuju. Ono za šta se može reći da je razlika u odnosu na prethodne stranice jeste to što se malo više truda uložilo na pisanje *css* koda kojim se težilo stvaranje ljepše dizajnirane stranice kako bi se postigao što bolji efekat kada je u pitanju prikazivanje svih parametara. Pri samom dnu stranice nalaze se i tri dugmeta koja obezbjeđuju eksportovanje parametara i rezultata u neke od popularnih formata dokumenata o čemu će biti više riječi u sledećem dijelu.

A	0.45281879194631	/	Coefficient of the river basin form
m	0.78051694757089	/	Coefficient of the watershed development
B	18.125	km	Average river basin width
a	0.15586206896552	/	(A)symmetry of the river basin
G	0.39655172413793	/	Density of the river network of the basin
K	1.2213114754098	/	Coefficient of the river basin tortuousness
H _{sr}	1103.2440506897	m	Average river basin altitude
D _{sr}	667.2440506897	m	Average elevation difference of the river basin
I _{sr}	30.049983448276	%	Average river basin decline
H _{leb}	2025	m	The height of the local erosion base of the river basin
E _r	87.836628057213	/	Coefficient of the erosion energy of the river basin's relief
W	0.62097109554194	m	Analytical presentation of the water retention in inflow
S ₁	0.958	/	Coefficient of the region's permeability
S ₂	0.7091	/	Coefficient of the vegetation cover
2gDF ^{1/2}	6161.5624638676	m km s ⁻¹	Energetic potential of water flow during torrent rains
Q _{max}	1176.9545094685	m ³ s ⁻¹	Maximal outflow from the river basin
T	0.9746794344809	/	Temperature coefficient of the region
Z	0.37597530829481	/	Coefficient of the river basin erosion
W _{god}	1787157.8090074	m ³ god ⁻¹	Production of erosion material in the river basin
R _d	0.19112314172951	/	Coefficient of the deposit retention
G _{god}	341567.21522393	m ³ god ⁻¹	Real soil losses
G _{god km⁻²}	117.78179835308	m ³ km ⁻² god ⁻¹	Real soil losses per km ²

Slika 23: Izgled javno-dostupne stranice koja prikazuje rezultate rječnog sliva

Eksportovanje podataka u najčešće korištene formate

Javno-dostupni parametri i rezultati koji su prethodno prikazani mogu se i eksportovati u tri najčešće korištena formata dokumenata. U pitanju su formati *.doc*, *.xls* i *.pdf*. Eksportovanje se realizuje, u zavisnosti od potrebe za formatom, klikom na dugme koje se nalazi na dnu stranice. Ova mogućnost je veoma pogodna za različite analize, a takođe i vrlo korisna ukoliko se javi potreba za grafičkim predstavljanjem podataka ili dijela podataka. Sve tri vrste dokumenata su pravilne strukture i jasnog redosleda. Kao i na stranicama veb platforme, i u dokumentima su realizovani svi detalji koji se tiču podataka.

Eksportovanje podataka u prva dva navedena formata ostvaruje se, upotrebom jednostavnog rješenja. U pitanju je HTTP (Hypertext Transfer Protocol) metod. Ovim metodom se u PHP fajlu definiše odgovarajuće zaglavlje kojim se veb pretraživaču zadaju naredbe. Naredbe se odnose na to da je u pitanju sadržaj koji pretraživač zbog svoje standardne konfiguracije otvora pomoću računarskog programa koji podržava *.doc* i *.xls* formate. Na slici 24. nalazi se kôd koji je potreban za realizaciju ovog metoda. Kôd se odnosi na *.doc* dokument, a jedina korekcija koju treba napraviti kako bi se izvršavao kôd koji kreira *.xls* dokument je promjena u zaglavlju. Dakle, na četvrtoj liniji, gledano sliku, umjesto *.doc* treba upisati *.xls*. Nakon definisanja zaglavlja, sva dalja procedura i pisanje koda nastavlja se tako što se piše uobičajeni HTML kôd. Mogućnosti formatiranja dokumenata su ograničene i nije moguće iskoristiti sve benefite koje HTML jezik nudi. Ovdje je upotrijebljena i funkcija *ob_clean()* koja ima ulogu da izbriše sav sadržaj prije nego što se on pošalje pretraživaču.

```
ob_clean();

header("Content-type: application/vnd.ms-word");
header("Content-Disposition: attachment;Filename=".$row['riverbasin'].".doc");
header("Pragma: no-cache");
header("Expires: 0");

echo "<html>";
echo "<meta http-equiv=\"Content-Type\" content=\"text/html; charset=Windows-1252\">";
echo "<body>";

...
```

Slika 24: PHP kôd kojim se definiše formatiranje *.doc* i *.xls* dokumenata pomoću metoda HTTP

Kada je u pitanju *.pdf* format dokumenta, isti se realizuje tako što se koristi PHP klasa *FPDF* koja generiše PDF datoteke kroz jasan PHP kôd, bez upotrebe bilo koje biblioteke. *FPDF* klasa sadrži dosta bitnih funkcija. Neke od njih su veoma značajne i odnose se na formatiranje stranica i teksta, prikazivanje linkova, dostupna je mogućnost upotrebe različitih boja, a tu je i podrška za nekoliko formata slika. Generisanje dokumenata se odvija veoma brzo. Na slici 25. prikazano je nekoliko linija PHP koda kojima se vrši formatiranje dokumenta i kreiranje nekoliko prvih redova u dokumentu koje se tiču osnovnih informacija. Ostatak PHP koda se bazira na istom principu pa ga iz tog razloga nije potrebno prikazivati.

```

<?php

session_start();
require_once('connect.php');

if(isset($_REQUEST['id'])){

    $space=" ";
    $sql = "SELECT * FROM data WHERE id = ".$_REQUEST['id'].>";";
    $result = mysqli_query($connection, $sql);

    require("library/fpdf.php");

    $pdf = new FPDF('p', 'mm', 'A4');
    $pdf->AddPage();
    $pdf->SetFont('Arial', 'B', 9);
    $pdf->SetTextColor(0, 119, 181);
    $pdf->SetFillColor(255,255,255);

    $pdf->Image('images/logo.png',10,6,40);
    $pdf->cell(0,0, 'Web application for Intensity of Erosion and Outflow', 0, 0,
'R');
    $pdf->Ln(20);
    $pdf->SetFont('Arial', 'B', 13);
    $pdf->cell(0,0, 'WIntErO Report', 0, 0, 'C');
    $pdf->Ln(10);

    while($row = mysqli_fetch_array($result)) {

        $pdf->cell(0,0, 'Name of the River Basin: '.$row['riverbasin'].'",', 0, 0,
'C');
        $pdf->Ln(10);
        $pdf->cell(0,0, 'Country: '.$row['country'].'",', 0, 0, 'C');
        $pdf->Ln(10);
        $pdf->cell(0,0, 'Year: '.$row['year'].'",', 0, 0, 'C');
        $pdf->Ln(10);
        $pdf->cell(0,0, 'GPS coordinates, latitude and longitude with Google
Maps: '.$row['latitude'].','.$row['longitude'].'",', 0, 0, 'C');
        $pdf->Ln(20);
    }
}

```

Slika 25: PHP kôd kojim se definiše formatiranje PDF dokumenta upotrebom klase FPDF

3.3.4 Razvoj prilagodljivog dizajna za različite uređaje

Prilagodljiv dizaj veb platforme ili sajta (RDW – Responsive Web Design) predstavlja dizajn koji obezbeđuje pravilno prikazivanje i jednostavnu upotrebu sajta koji će biti pregledan na različitim uređajima sa različitim rezolucijama. Dakle, sadržaj platforme postaje kaskadno-dinamički sadržaj sposoban da se prilagodi različitim veličinama ekrana. Ukoliko se ne bi primijenile prilagodljive karakteristike, pregledanje i pretraživanje sajta bi bilo prilično zamorno i oduzimalo bi previše vremena posjetiocima. RDW opcija se prvi put pojavila u CSS3 stilskom jeziku [18]. Prije njene dostupnosti, dizajn za različite uređaje mogao je jedino da se ostvari improvizacijom koda. Prilagodljiv dizajn se zasniva na korišćenju tehnika, a tri najvažnije su:

tehnika medija upita (engl. *media queries*), skalabilnosti fotografija (engl. *flexible images*) i fluidne mreže (engl. *fluids grids*). Tehnika medija upita omogućava da stranica koja se prikazuje na uređaju određene rezolucije, koristi upravo CSS svojstva koja su predviđena za tu rezoluciju. Kada su u pitanju skalabilne fotografije, one koriste relativne jedinice mjere zadate u procentima i na taj način se obezbjeđuje prikaz fotografija u okvirima sadržaja stranice. Fluidne mreže takođe koriste relativne jedinice u procentima za kreiranje stranica za različite uređaje. Za razliku od fluidnih mreža, fiksne mreže koriste piksele. U današnjem vremenu je dominantna potražnja za izradom sajtova baziranih na fluidnim rješenjima. RDW je danas veoma razvijem i svakodnevno se unaprijeđuje.

Prilagodljivost sadržaja neke stranice različitim uređajima sprovodi se na način što se u prvom koraku mora navesti HTML naredba na samoj stranici. Naredba se odnosi na to da se u zaglavlju stranice definiše uključivanje objekta *viewport*³³, a zatim se i podešava širina stranice na način što se širina određuje na osnovu rezolucije ekrana uređaja. Naredba se nalazi na slici 26., u HTML dijelu. U samoj naredbi se može vidjeti i da *initialscale*³⁴ ima vrijednost 1 što znači da se pri otvaranju stranice zum faktor nalazi na prvom nivou.

Kada je u pitanju samo pisanje koda za prilagodljivi dizajn, on se izvodi regularno, isto onako kao i standardni CSS kôd. Na slici 26., CSS dio, dat je kôd koji koristi tehniku medija upita. Ovim kodom sa slike definišu se tekstualni naslovi na stranici koja se prikazuje na uređajima koji podržavaju maksimalnu rezoluciju *1280px* u širini i *800px* u visini. Suština pisanja koda za različite uređaje kada je u pitanju ova tehnika bazira se na korišćenju *@media*³⁵ pravila.

```

*****HTML KOD*****
<meta name="viewport" content="width=device-width, initialscale=1" />
*****CSS KOD*****

@media (max-width: 1280px) and (max-height: 800px) {
  .caption h1 {
    font-size: 2.5rem;
    font-weight: 500;
    letter-spacing: .3rem;
    text-shadow: .11rem .11rem .6rem white;
    padding-bottom: .5rem;
  }
  .caption h3 {
    font-size: .8rem;
    text-shadow: .06rem .06rem .18rem white;
    padding-bottom: 1rem;
  }
}

```

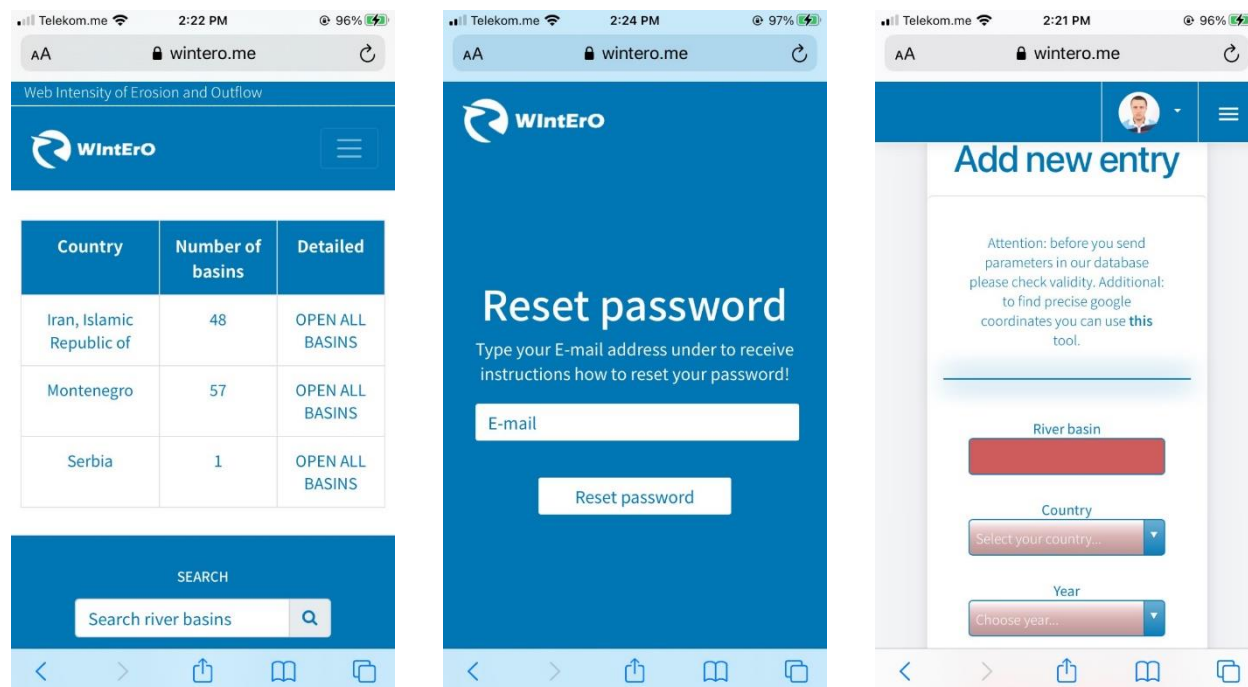
Slika 26: Prikaz HTML i CSS kodova koji su sastavni dio prilagodljivog dizajna

³³ *Viewport* – površina veb stranice koja je vidljiva za korisnika.

³⁴ *Initialscale* – definiše vrijednost početnog zuma.

³⁵ *@media* – pravilo tehnike medija upita koje se koristi za primjenu različitih CSS stilova na različitim uređajima.

Razvoj prilagodljivog dizajna bio je neizostavan dio kada se pravio plan za izradu WIntErO platforme. Imajući u obzir da se u današnjem vremenu sve više koriste prenosivi uređaji bilo je neophodno i prilagoditi stranice platforme da svoj sadržaj prikazuju i na prenosivim laptopovima manjih ekrana, mobilnim i tablet uređajima. Na slici 27. nalaze se snimci ekrana na kojima su prikazane stranice platforme koje su kompatibilne sa rezolucijama koje su zastupljene na mobilnim uređajima. Slika 28. pokazuje snimke ekrana koji prikazuju sadržaj kreiran za tablet uređaje.



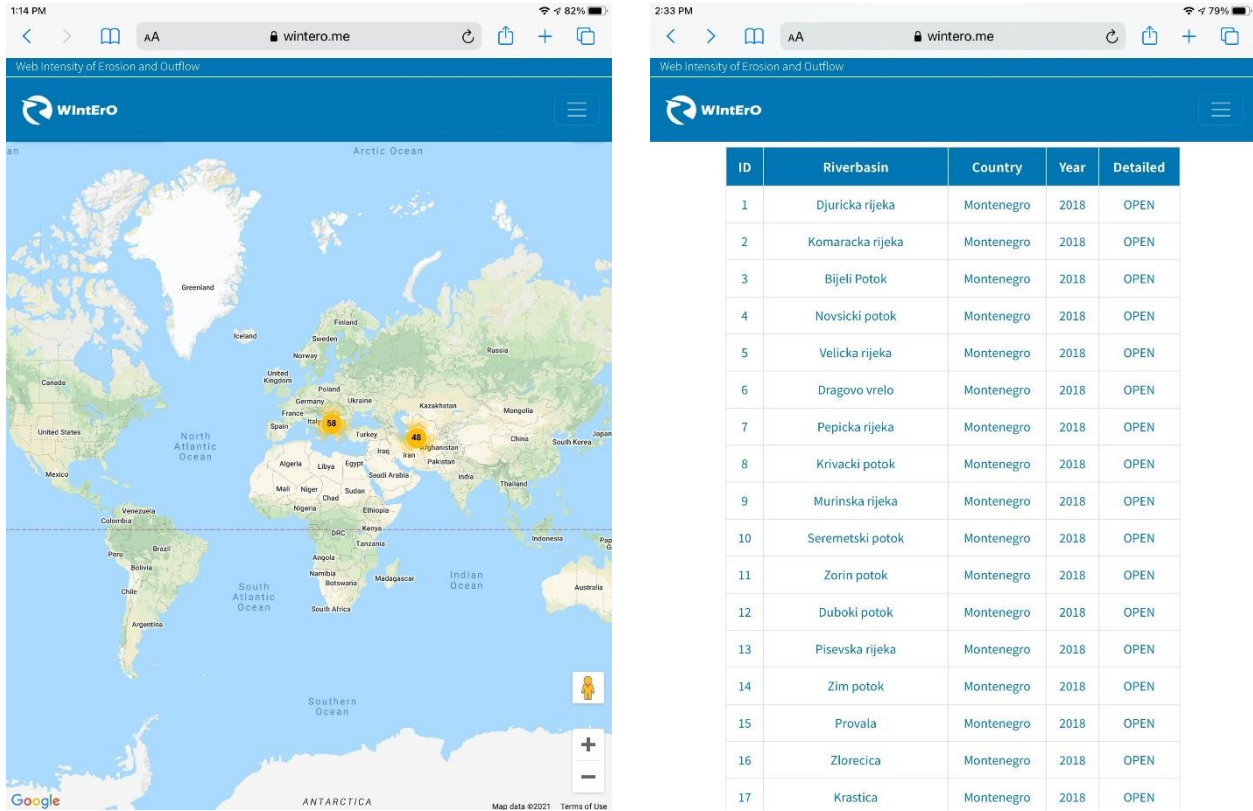
Slika 27: Snimci ekrana na kojima su prikazane stranice platforme koje su kompatibilne sa rezolucijama koje su zastupljene na mobilnim uređajima

Za kreiranje dizajna stranica, u većini slučajeva, a posebno pri korišćenju *@media* pravila, su korištene relativne jedinice mjere, dok su na nekim mjestima upotrijebljene i apsolutne i specijalne mjerne jedinice. Kada su u pitanju relativne jedinice, pretežno su upotrijebljene *em*³⁶ i *rem*³⁷, dok se kod apsolutnih jedinica koristio piksel kao mjera. Specijalne mjerne jedinice su upotrijebljene za vremensko trajanje, uglove i slično. Najbolji primjer za upotrebu specijalnih jedinica je prozirni prozor sa naslovom koji se pojavljuje na početnoj strani WIntErO platforme i na istoj se zadržava 5 sekundi.

Neki djelovi stranica su morali da ostanu fiksni i da se upare sa fluidnim rasporedom kako bi se došlo do korektnog rješenja. Ovakav pristup izradi stranica se u svijetu veb dizajna naziva hibridni pristup i nije rijetka pojava među rješenjima dizajnera.

³⁶ *em* – mjera koja se odnosi na veličinu fonta elementa (*2em* prikazuje dva puta veću veličinu trenutnog fonta)

³⁷ *rem* – relativna mjera koja važi za veličinu fonta koja važi za osnovni element



Slika 28: Snimci ekrana na kojima su prikazane stranice platforme koje su kompatibilne sa rezolucijama koje su zastupljene na tablet uređajima

WIntErO platforma zahtijeva malo kompleksnije procedure kada su u pitanju određeni segmenti. Na primjer, postupak koji se odnosi na upisivanje novih informacija u bazu nije potpuno praktičan kada se realizuje pomoću mobilnog uređaja ili tableta. Razlog je prost, postoji veliki broj polja za unos, pa je samim tim daleko praktičnije koristiti računar, bez obzira na činjenicu što su primijenjene tehnike prilagodljivog dizajna. Mogućnost za prevazilaženje ovog ne potpuno praktičnog postojećeg rješenja, i rješavanje istog obezbjeđivanjem efikasnog načina ogleđa se u kreiranju aplikacija za različite mobilne operativne sisteme. U narednom poglavlju će se više govoriti o WIntErO aplikacijama za dva najpopularnija mobilna operativna sistema i prikazaće se postupak izrade istih.

4. Razvoj aplikacija za pametne uređaje

Mobilni uređaji, kao i mobilne aplikacije postali su zadnjih godina dio naše svakodnevnice i u mnogome nam pomažu da razne zadatke obavljamo na što jednostavniji način. Resursi i sistemi mnogih veb platformi koje obezbjeđuju komunikaciju, rad, učenje i ostalo sve više su dostupni i kroz upotrebu mobilnih aplikacija. Mobilne aplikacije olakšavaju upotrebu korisnicima i pristup sadržaju, a posebno kada su u pokretu i nijesu u mogućnosti koristiti stacionirani računar. Poglavlje je podijeljeno na sledeće cjeline:

- tehnologije korišćene za izradu mobilnih aplikacija,
- analiza i opis projektovanog rješenja,
- izrada i upotreba mobilnih aplikacija.

4.1 Tehnologije korišćene za izradu mobilnih aplikacija

Mobilni operativni sistemi su sistemi koji omogućavaju brojne funkcionalnosti na mobilnim uređajima, tabletima i drugim prenosivim uređajima, a koje već postoje na standardnim operativnim sistemima koji se koriste na desktop i prenosivim računarima. Dva najpopularnija mobilna operativna sistema za mobilne uređaje su iOS i Android OS³⁸.

Arhitekture ovih sistema se razlikuje u samoj osnovi. iOS operativni sistem se bazira na *Juniks* operativnom sistemu, dok se Android OS bazira na *Linuks* jezgru [36]. Način izrade mobilnih aplikacija koji je najzastupljeniji ogleda se u korišćenju *Swift* programskog jezika kroz razvojno okruženje *Xcode* za iOS³⁹, kao i upotrebom *Java* programskog jezika u razvojnom okruženju *Android Studio* za Android OS⁴⁰.

Za izradu WIntErO mobilnih aplikacija korišćeno je drugačije, jedinstveno rješenje kojim se aplikacije kreiraju na nešto brži način. U pitanju je *Xamarin*⁴¹ platforma koja omogućava kreiranje modernih aplikacija koje mogu da rade na različitim operativnim sistemima. Brži razvoj aplikacija pomoću ove platforme je moguć zbog pogodnosti da se upotrijebljava jedan programski jezik, a to je C#. U nastavku će se dati bliži opis najbitnijih tehnologija koje su korišćene za izradu, a to su:

- *Xamarin* platforma,
- C# programski jezik,
- .NET Framework platforma.

³⁸ Statistika upotrebe mobilnih operativnih sistema - <https://gs.statcounter.com/os-market-share/mobile/worldwide>

³⁹ Xcode okruženje, alati za razvoj i dokumentacija - <https://developer.apple.com/xcode/>

⁴⁰ Android Studio, alati za razvoj i dokumentacija - <https://developer.android.com/studio>

⁴¹ Xamarin platforma, alati za razvoj i dokumentacija - <https://dotnet.microsoft.com/apps/xamarin>

4.1.1 Xamarin platforma

Xamarin platforma je razvojna platforma koja služi za kreiranje aplikacija za različite operativne sisteme kao što su iOS, Android OS i WindowsPhone. Platforma je u vlasništvu *Microsoft* kompanije i bazira se na postojećoj .NET Framework platformi, na način što istu proširuje dodavanjem alatki i biblioteka koje potpomažu razvoj aplikacija [37].

Xamarin je projekat otvorenog koda (engl. *open-source*) koji obezbeđuje da se programski kodovi kojima se kreiranju aplikacije, a koji se pišu pomoću C# jezika prevode u izvršne kodove koji su odgovarajući za određene operativne sisteme. Osnovna biblioteka ove platforme je *Xamarin.Forms* i ona je upravo najvažniji dio pomoću kojeg se stvaraju prilagodljivi korisnički interfejsi. Integrirano razvojno okruženje koje se koristi za razvoj aplikacija pomoću ove tehnologije je Microsoft Visual Studio Community 2019.

Xamarin.Forms biblioteka podržava razvoj pet različitih tipova aplikacija za različite operativne sisteme, a to su [37]:

- iOS aplikacije koje se pokreću na *Apple* uređajima *iPhone*, *iPad* i *iPod*;
- Android aplikacije koje se pokreću na Android pametnim telefonima i tabletima;
- Universal Windows Platform (UWP) aplikacije koje se pokreću na *Windows 10* i *Windows 10 Phone* sistemima;
- Windows Runtime API aplikacije namijenjene za *Windows 8.1* sisteme;
- Windows Runtime API aplikacije namijenjene za *Windows Phone 8.1* sisteme.

Xamarin.Forms biblioteka upotrebom kroz Microsoft Visual Studio Community 2019 sadrži pet odvojenih projekata za svaki od prethodno navedenih pet tipova, kao i šesti projekat koji sadrži zajednički kôd za sve ove opcije. U nastavku, u dijelu koji se odnosi na izradu i upotrebu aplikacija biće još nešto više riječi o Xamarin platformi.

Takođe, treba naglasiti da pokretanje i testiranje aplikacija koje se koriste na iOS uređajima nije moguće izvršiti na *Windows* operativnim sistemima bez dodatka *Remoted iOS Simulator for Windows*⁴², koji mora da bude uparen sa *Mac* uređajem. Android i *Windows* aplikacije se mogu testirati na *Microsoft Windows* sistemima bez većih zahtjeva.

4.1.2 C# programski jezik

C# je moderan objektno orijentisani programski jezik koji se odlikuje osobinom bezbjednih tipova (engl. *type-safe*) što znači da se instance tipova mogu koristiti isključivo u skladu sa protokolima definisanim za dati tip. Ovaj jezik je nezavistan od bilo koje platforme, a takođe je kompatibilan sa više različitih kompajlera. C# je veoma složen jezik i bazira se na složenom obliku implementacije objektno orijentisanog modela programiranja. C# zavisi od izvršnog okruženja (Common Language Runtime - CLR) koje izvršava veliki broj zadataka. Jedan od glavnih zadataka

⁴² Remoted iOS Simulator for Windows - <https://docs.microsoft.com/en-us/xamarin/tools/ios-simulator/>

koje izvršno okruženje obavlja je automatsko upravljanje memorijom. Automatsko upravljanje memorijom djeluje kao sastavni dio aplikacije i oslobađa memoriju koja je dodijeljena objektima koje aplikacija više ne upotrijebli. CLR je i glavna komponenta .NET Framework platforme i radi nezavisno od jezika na kojem je aplikacija napisana [38], [39].

C# je prvi put predstavljen 2002. godine i njegova najveća prednost u odnosu na ranije programske jezike poput C i C++ jezika je u tome što je prilično lakši za upotrebu, a također je ispred svoje konkurencije kada su performanse u pitanju. Nastao je kao sastavni dio .NET Framework platforme i glavni cilj je bio poboljšavanje produktivnosti programera. Postoji nekoliko verzija ovog jezika, a posljednja stabilna je verzija broj 9⁴³. Jezik ima veoma široku primjenu i veoma je pogodan za razvoj softverskih rješenja. Kompletna dokumentacija vezana za ovaj jezik dostupna je i na zvaničnom sajtu⁴⁴.

4.1.3 .NET Framework platforma

.NET Framework je platforma koja se sastoji od CLR-a i velikog broja raznovrsnih biblioteka. Postoje dvije vrste biblioteka koje su sastavni dio ove platforme, a to su osnovne i aplikativne biblioteke. Aplikativne biblioteke zavise od osnovnih. Ove biblioteke se koriste od strane programera, koji ih kombinuju sa sopstvenim kodom i tako razvijaju softverska rješenja. Sve mogućnosti koje pruža ova platforma dostupne su kroz obimni skup upravljanih tipova koji su organizovani u hijerarhijske imenske prostore i grupisani su u skup sklopova. Zajedno sa CLR-om oni čine .NET Framework platformu.

.NET Framework obezbjeđuje jezičku nezavisnost, odnosno zajednički tip sistema (Common Type System - CTS). Ovim se definišu svi mogući tipovi podataka i samim tim omogućava razmjenu tipova i objekata između biblioteka i aplikacija napisanih u nekom od .NET programskih jezika kao što su C#, F# i Visual Basic. Kada je u pitanju bezbjednost, ova platforma nudi opšti sigurnosni model za sve aplikacije⁴⁵ [38], [39].

Tvorac i vlasnik ove platforme je kompanija *Microsoft* i predstavljena je prvi put 2002. godine, ali se dosta godina ranije počelo raditi na razvijanju iste. Posljednja stabilna verzija je 4.8 koja je objavljena sredinom 2019. godine⁴⁶. Primjena NET Framework platforme je veoma velika i moguće je njenom upotrebom izraditi aplikacije koje se koriste na veb, mobilnim, desktop, *cloud* i sličnim platformama, a također veliku primjenu ima kod razvoja igrica i raznih mikroservisa. Ova platforma je također i besplatna, baš kao i ostatak .NET platforme. Ne postoje dodatne naknade ili troškovi licenciranja ni onda kada je svrha komercijalna upotreba⁴⁷.

⁴³ C# programski jezik, posljednja stabilna verzija - <https://devblogs.microsoft.com/dotnet/announcing-net-5-0/>

⁴⁴ Dokumentacija za C# programski jezik - <https://docs.microsoft.com/en-us/dotnet/csharp/>

⁴⁵ Bezbjednost .NET Framework platforme - <https://docs.microsoft.com/en-us/dotnet/standard/security/>

⁴⁶ Posljednja verzija .NET Framework platforme - <https://dotnet.microsoft.com/download/dotnet-framework/net48>

⁴⁷ Detaljnije o platformi, kao i o licenci - <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet-framework>

4.2 Analiza i opis projektovanog rješenja

Rješenje koje se odnosi na aplikacije je daleko jednostavnije nego što je to slučaj kod rješenja koje se osmislilo za potrebe izgradnje veb platforme. Postoji više razloga zašto se veći fokus usmjerio na razvijanje veb platforme, ali između ostalog i dizajn platforme je prilagodjen za prikazivanje na mobilnim uređajima, pa donekle i to umanjuje potrebu za kreiranjem složenih aplikacija u ovom trenutku. Aplikacije koje su projektovane za dva najpopularnija mobilna operativna sistema obezbjeđuju najosnovniji pregled i rad sa podacima. Dakle, aplikacije se mogu podijeliti na tri segmenta koji su projektovani, a to su:

- pregled informacija koje se odnose na parametre i rezultate rječnih slivova,
- mogućnost upisivanja privremenih parametara vezanih za rječne slivove,
- najvažnije osnovne informacije o platformi.

Aplikacije omogućavaju da korisnici mogu da pregledaju parametre i rezultate za rječne slivove, ali ne i da šalju nove unose. Ne postoji korisničko okruženje, pa samim tim i izostaju mogućnosti koje su dostupne registrovanim korisnicima. Ovaj nedostatak je moguće prevazići kreiranjem korisničkog okruženja za aplikacije, pri čemu bi se koristila postojeća baza podataka pod nazivom *register*.

Drugi segment, koji je ujedno i novo, korisno rješenje, koje nije obezbijedeno kroz kreiranje veb platforme, je mogućnost upisivanja privremenih parametara vezanih za rječne slivove. Ideja za ovim rješenjem javila se iz razloga što istraživači vrlo često posjećuju lokacije koje možda trenutno i nijesu na najbolji način pokrivena kvalitetnim signalom, pa samim tim im i nije dostupna stabilna internet konekcija kako bi uspješno i bez poteškoća upisivali parametre i slali ih u WIntErO bazu podataka. Upotrebom aplikacija, ovaj segment je moguće koristiti bez internet konekcije. Još važnije opravdanje zašto je projektovana mogućnost upisivanja privremenih podataka je i to što je mnogo jednostavnije koristiti mobilni uređaj za sakupljanje podataka na terenu, nego recimo imati potrebu nositi sa sobom prenosivi računar ili još teže, pisati podatke na papir. Ovaj segment je projektovan da prikazuje sva polja za unose parametara, baš kao i forma na stranici veb platforme, premda se neki od njih dobijaju i na drugačiji način, projekcijom određenih podataka. Dakle, nakon unošenja parametara, oni se čuvaju u lokalnoj memoriji. Nakon toga, istraživači mogu pristupiti podacima u bilo kom momentu i koristiti ih kako bi izvršili unos u bazu.

Posljednji segment aplikacija tiče se osnovnih informacija o WIntErO platformi, a kroz ovaj dio je moguće i kontaktirati administratore platforme.

Projektovane aplikacije su, kao što je već rečeno, veoma proste. Postoje tri dugmeta koja služe za prelazak na opisane segmente. Sama ideja rada je bila da se uspostavi veb platforma koja će obezbijediti sve ono što je omogućavao i programski paket IntErO uz par noviteta. Aplikacije su dodatak koji će u budućem razvoju poslužiti kao osnova za dalji napredak. Svjedoci smo da u današnjem vremenu velika posjećenost određenim platformama iziskuje od tvoraca i obezbjeđivanje kvalitetnih aplikacija za sve pametne uređaje, a ne samo za mobilne. U zavisnosti od upotrebe platforme, veoma važno bi bilo i kreirati aplikacije koje su namijenjene za tablet uređaje. Iz dosadašnjeg, evidentno je da su forme koje su kreirane za upisavanje parametara prilično obimne u dizajnerskom smislu, pa je stoga dosta pogodnije pregledati ih na računaru.

Adekvatnih dizajnom formi za mobilne uređaje, u potpunosti bi bilo moguće prikazati sva polja na veoma pregledan i upotrijebljiv način, a posebno kada su u pitanju tablet uređaji.

Xamarin platforma je jedinstveno rješenje za kreiranje aplikacija koje mogu biti instalirane na različitim uređajima, mada za razvoj i iOS i Android aplikacija postoje drugačije preporuke od strane kompanija koje su njihovi kreatori. Konkretno, za kreiranje iOS aplikacija kompanija *Apple* preporučuje za upotrebu alat⁴⁸ koji su osmislili njihovi inženjeri, dok se za kreiranje Android OS aplikacija koristi *Android Studio*⁴⁹ čiji tvorcima su inženjeri kompanije *Google*. Popularnost je takođe veća kada su u pitanju alati koji su stvoreni od strane matičnih kompanija. Xamarin platforma je veoma dobro rješenje kada se radi o jednostavnijim zadacima, i kada je neophodno kreirati proizvod u bržem vremenskom intervalu. Jedinstveni kôd, samim tim i upotreba jednog programskog jezika zahtijeva jednog programera, a takođe je i razlika u cijeni proizvoda očigledna.

U nastavku će se prikazati izrada i upotreba mobilnih aplikacija, čime će se još bliže opisati i upotrijebljene tehnologije za izradu.

4.3 Izrada i upotreba mobilnih aplikacija

Ovo poglavlje se odnosi na postupak izrade aplikacija, kao i na prikaz upotrebe istih. Kroz dalji postupak opisivanja i prikazivanja osnovnih dijelova koji se tiču aplikacija detaljnije će se objasniti primjena tehnologija kojima se ostvarilo kreiranje, kao i predstaviti neki dijelovi određenih programskih kodova, a uz sve to i dati prikaz izgleda grafičkog interfejsa aplikacija, tačnije prikaz dijelova segmenata koji su dio istih.

Upotreba aplikacija je prilagodjena za skoro sve iOS i Android OS uređaje. U nastavku će se prikazivati i slike ekrana koji su u stvari ekrani prikazani pomoću emulatora. Emulatori su softveri koji omogućavaju da se određene aplikacije ili procesi izvršavaju na platformi koja je drugačija u odnosu na onu platformu za koju je aplikacija i namijenjena. Ova mogućnost obezbjeđuje da se aplikacije testiraju i nadograđuju na lak način, a nakon završenog kreiranja ostaje da se pošalju na konačnu provjeru timovima koji su zaduženi za kontrolu aplikacija prije njihovog konačnog objavljivanja u *online* prodavnicama .

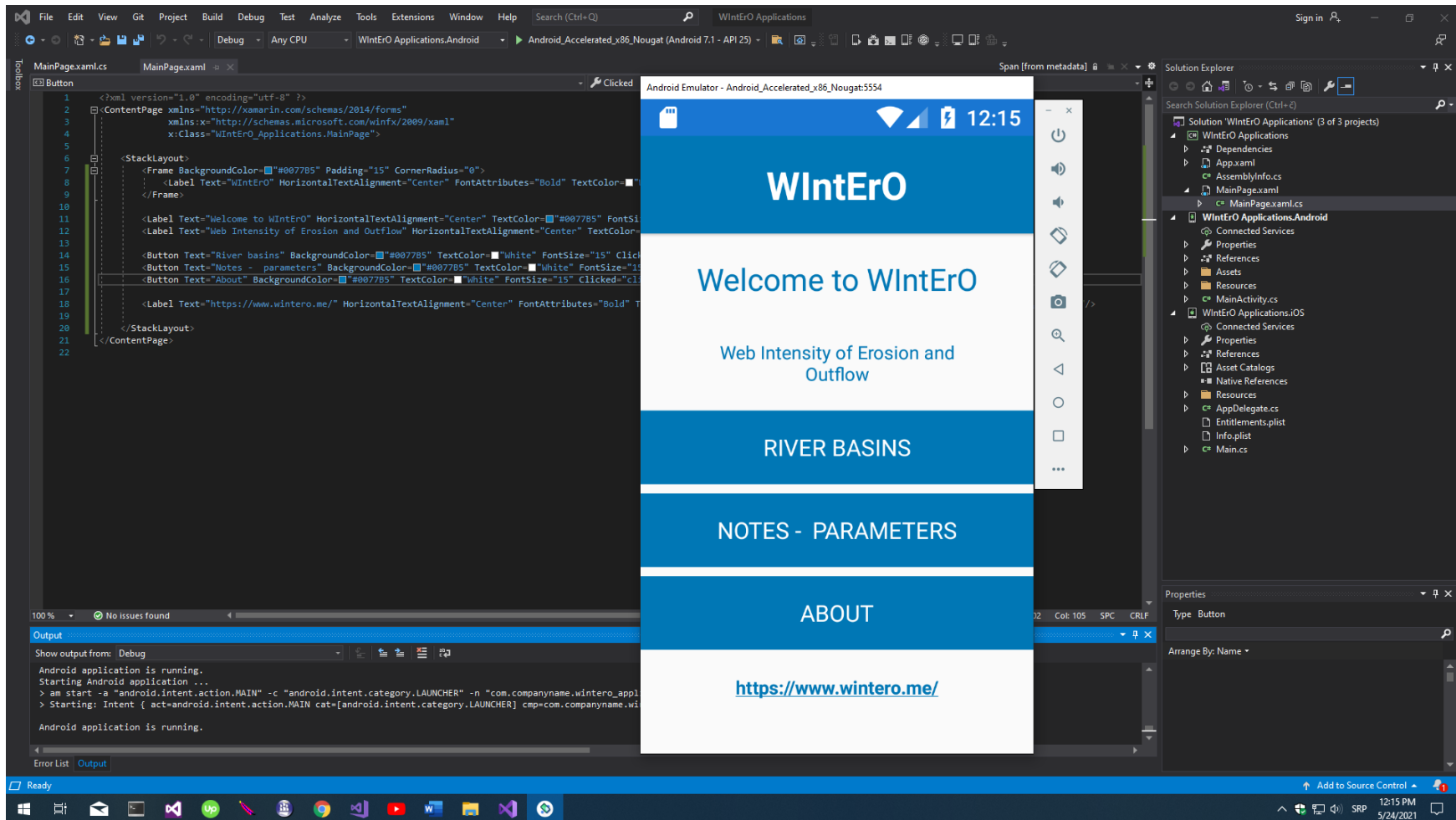
Slika 29. prikazuje izgled razvojnog okruženja Microsoft Visual Studio Community 2019, pomoću kojeg su kreirane aplikacije, a u desnom dijelu vidljiva je sekcija koja sadrži tri projekta, jedan zajednički i dva koja se odnose na aplikacije. Jedan veoma koristan dodatak ovom razvojnem okruženju je i *XAML Hot Reload*. Ovaj dodatak obezbjeđuje da sve izmjene koje su napravljene u dijelu programskog koda momentalno reflektuju u aplikaciji koja je pokrenuta pomoću emulatora. Ova opcija značajno doprinosi bržem kreiranju i testiranju aplikacija.

Na slici je takođe prikazan i izgled emulatora koji je pokrenut, kao i sve opcije koje nudi radi testiranja aplikacije. Radi se o *Android* emulatoru pomoću kojeg je testirana početna stranica

⁴⁸ Apple, dokumentacija i uputstva za programiranje - <https://developer.apple.com/develop/>

⁴⁹ Google, dokumentacija i uputstva za programiranje - <https://developer.android.com/studio>

aplikacije na kojoj se nalaze tri dugmeta koja služe za prelazak na ranije opisane segmente aplikacije. Na lijevom dijelu slike vidljiv je i programski kôd kojim je definisan izgled, a koji se nalazi u fajlu *MainPage.xaml*. Ovaj fajl se vezuje sa fajlom *MainPage.xaml.cs* u kojem se nalaze instrukcije za radnje koje se odvijaju nakon klika na jedan od tri prikazana dugmeta. Dalji opis izrade i upotrebe mobilnih aplikacija će se usredsrijediti na segment koji se tiče prikaza parametara i rezultata rječnih slivova, a nakon toga govoriće se i o izradi sekcije pomoću koje se vrši bilježenje novih parametara.



Slika 29: Izgled razvojnog okruženja Microsoft Visual Studio Community 2019 sa pokrenutim Android emulatorom

4.3.1 Postupak izrade stranice koja prikazuje parametre i rezultate

Kada je riječ o prvom segmentu, postoje dva bitna koraka, a to su povezivanje sa već postojećom WIntErO MySQL bazom podataka, *data*, a zatim i raspoređivanje podataka za prikaz. Direktna konekcija aplikacija sa bazom podataka je nepreporučljiva iz razloga što je ugrožena bezbjednost podataka. Praktično, podaci koji se koriste za autentifikaciju i povezivanje sa bazom se nalaze u samoj aplikaciji i ukoliko bi se desilo da neko izvrši dekompiliranje iste mogao bi da dođe do osjetljivih informacija. Iz ovog razloga, postoje i veb servisi kojima se prevazilazi ovaj problem. Veb servisi obezbjeđuju postojanu i sigurnu konekciju aplikacijama koje im šalju zahtjeve. Dalje, zahtjevi se prosleđuju bazi podataka, u ovom slučaju MySQL bazi, koja šalje odgovor.

Veb servisi (engl. *web services*) su aplikacije koje su smještene na nekom serveru i njihov glavni zadatak je obezbjeđivanje komunikacije između dvije mašine kako bi se nesmetano izvršila razmjena podataka kroz mrežu. Veb servisi koriste standardizovan sistem za komunikaciju. Ovi servisi su smješteni na jedinstvenom serveru i imaju ulogu da vrše usluge koje mogu biti besplatne ili one koje zahtijevaju plaćanje. Može se kazati da veb servis predstavlja sponu između servera i aplikacije, a dva glavna formata koja se koriste za prenos podataka su XML i JSON. JSON format podataka je upotrijebljen za konekciju WIntErO aplikacija sa matičnom bazom podataka. Glavni protokol koji se koristi je HTTP.

Postoji nekoliko tipova veb servisa, a za potrebe izrade ovih aplikacija je korišten servis koji danas ima dominantnu primjenu na internetu, a to je *RESTful*⁵⁰ servis. Odlikuje ga jednostavnost, a podaci koji se prenose su najčešće u JSON formatu, dok je dostupna mogućnost i upotrebe XML formata. Ovaj servis je zasnovan na REST (Representational State Transfer) arhitekturi⁵¹.

Na slici 30. dat je dio koda koji služi za povezivanje aplikacija sa bazom podataka, dio C# - konekcija. Ovaj dio koda je dio samo jednog fajla koji je povezan sa drugim fajlovima koji djeluju kao cjelina. Princip je isti kao i kod konekcija PHP fajlova o kojima je ranije bilo više diskusije. U dijelu C# JSON, dat je primjer naredbi za jedno polje kojima se vrši serializacija i deserializacija informacija koje se dobijaju iz baze podataka. Slika 31. prikazuje snimak ekrana iOS i Android uređaja za koje je aplikacija i namijenjena. Na slici je dat prikaz dijela aplikacije na kojem se nalaze parametri i rezultati o određenom rječnom slivu. Zbog velikog broja polja, neophodno je vršiti sklorovanje kako bi se izvršio kompletan uvid u podatke. Postupak izrade izgleda ovog dijela je sličan kao i kod dijela koji se odnosi na zabilježbu, o čemu će se prodiskutovati u narednom potpoglavlju.

⁵⁰ *RESTful* veb servis - <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/data-cloud/web-services/rest>

⁵¹ REST – Predstavlja arhitektonski stil za izgradnju veb servisa.

Pošto se radi o početnoj verziji aplikacija, izostavljene su neke opcije koje su dostupne kroz upotrebu veb platforme. Dalja nadogradnja aplikacija zavisi isključivo od popularnosti same veb platforme, a isto tako i budućih zahtjeva koji bi mogli da se pojave. Treba naglasiti i to da se klikom na dugme *River Basins*, koje se nalazi na početnom ekranu aplikacije, prelazi prvo na dio na kojem se nalazi spisak rječnih slivova, grupisan po državama radi lakše pretrage.

```

*****C# KONEKCIJA*****
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using SimpleRESTServer.Models;
using MySql.Data;

namespace WIntErOApplication

    public class wintero
    {

        private MySql.Data.MySqlClient.MySqlConnection conn;

        public wintero()
        {
            string myConnectionString;
            myConnectionString = "server=localhost;uid=Local Instance MySQL80;
            pwd=;database=data";

            try
            {
                conn = new MySql.Data.MySqlClient.MySqlConnection();
                conn.ConnectionString = myConnectionString;
                conn.Open();
            }

            catch (MySql.Data.MySqlClient.MySqlException ex)
            {
            }

        }

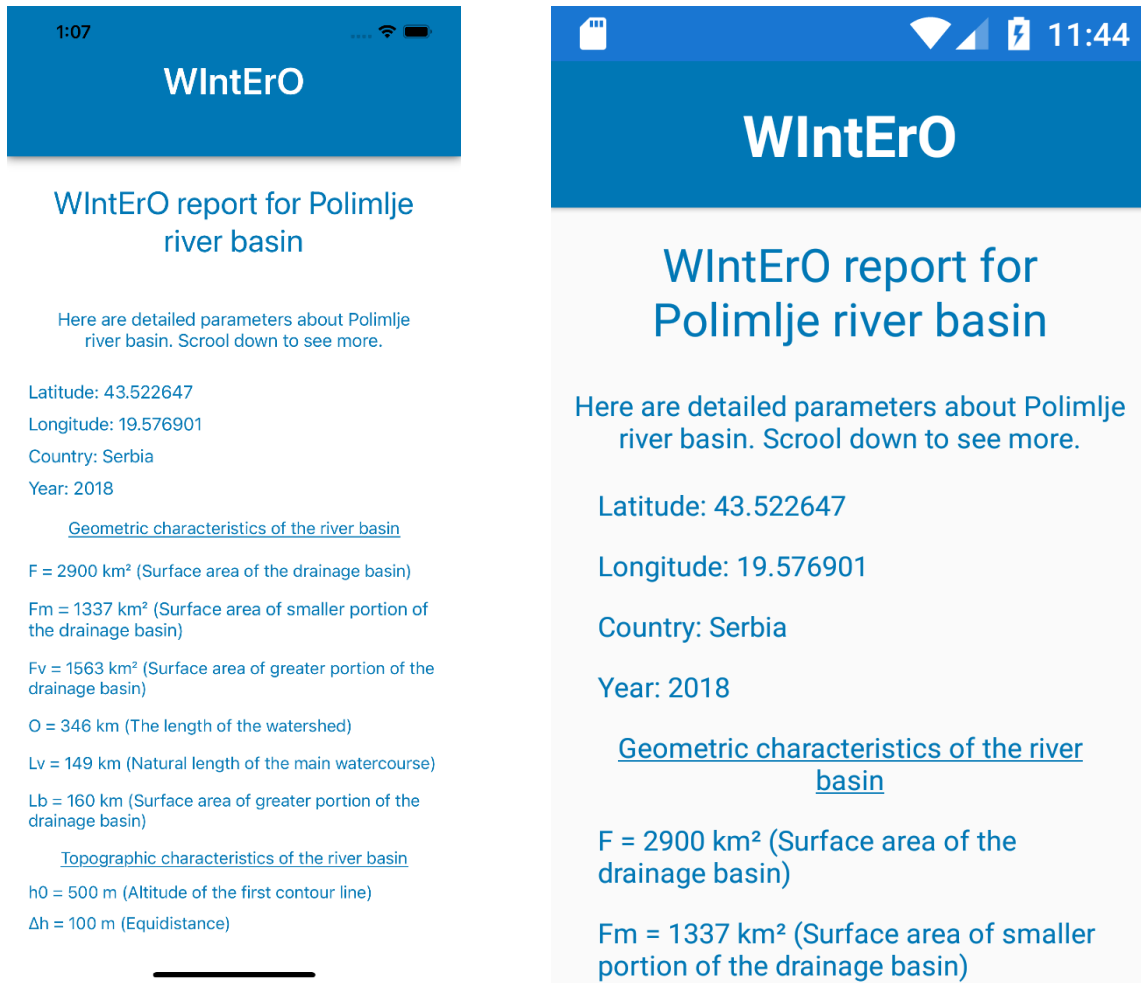
*****C# JSON*****

string jsonString = JsonSerializer.Serialize<RiverBasin>(riverBasin);

RiverBasin riverBasin = JsonSerializer.Deserialize<RiverBasin>(jsonString);

```

Slika 30: Prikaz C# kodova koji su sastavni dio segmenta za prikaz parametara i rezultata



Slika 31: Snimci ekrana iOS i Android uređaja, segment koji prikazuje parametara i rezultate

4.3.2 Izrada sekcije koja omogućava zabilježbu novih parametara

Zabilježba parametara pomoću aplikacije koja ne zahtijeva internet konekciju za ovu svrhu je rješenje koje će biti korisno istraživačima. Aplikacija, suštinski, podsjeća na neke popularne aplikacije koje korisnicima omogućavaju brzu zabilježbu informacija (engl. *notes apps*). Ovaj dio aplikacije sadrži polja za unos parametara (engl. *inputs*), a kojih je ukupno minimum 54, a konačan broj zavisi od broja koeficijenata koji se odnose na dužine izohipsi i površine između susjednih izohipsi. Forma je osmišljena tako da ne postoje obavezna polja, kao što je to slučaj kod forme koja je implementirana na veb platformi. Korisniku se ostavlja mogućnost da napravi izbor kada je u pitanju unos parametara. Logičke operacije nijesu implementirane iz razloga što će podaci da se upisuju naknadno na drugu formu, pa samim tim ova operacija nije neophodna. Unešeni parametri se čuvaju u lokalnoj memoriji. Takođe, dostupna je i lista sa spiskom slivova za koje se izvršilo upisivanje. Podaci se po potrebi mogu i brisati.

Čuvanje podataka se vrši u lokalno kreiranoj tabeli. Za ove potrebe upotrijebljen je SQLite sistem za upravljanje bazama podataka. Rad sa SQLite sistemom je omogućen kroz upotrebu SQLite .NET biblioteke koja je otvorenog koda⁵². Čuvanje podataka bi bilo moguće izvršiti i u XML fajlu. Na slici 32. nalaze se djelovi kodova koji omogućavaju funkcionisanje ovog segmenta. U dijelu C# INPUTS nalazi se dio koda kojim se kreira forma, dok se u dijelu C# SQLite nalazi kôd za formiranje tabele za čuvanje podataka.

```

*****C# INPUTS*****

<Entry Placeholder="Enter name of the River basin" HorizontalTextAlignment="Center"
Margin="10,5,10,5" TextColor="#0077B5" FontSize="10" />
<Entry Placeholder="Enter Latitude" HorizontalTextAlignment="Center"
Margin="10,5,10,5" TextColor="#0077B5" FontSize="10" />
<Entry Placeholder="Enter Longitude" HorizontalTextAlignment="Center"
Margin="10,5,10,5" TextColor="#0077B5" FontSize="10" />.

.
.
.

<Button Text="Send" Clicked="sendparameters"/>

*****C# SQLite*****

using System;
using SQLite;

namespace Notes.WINTEROAPP
{
    public class Wintero
    {
        [PrimaryKey, AutoIncrement]
        public int ID { get; set; }
        public string riverbasin { get; set; }
        public double latitude { get; set; }
        public double longitude { get; set; }
        public double country{ get; set; }
        public int year { get; set; }

        .
        .
        .

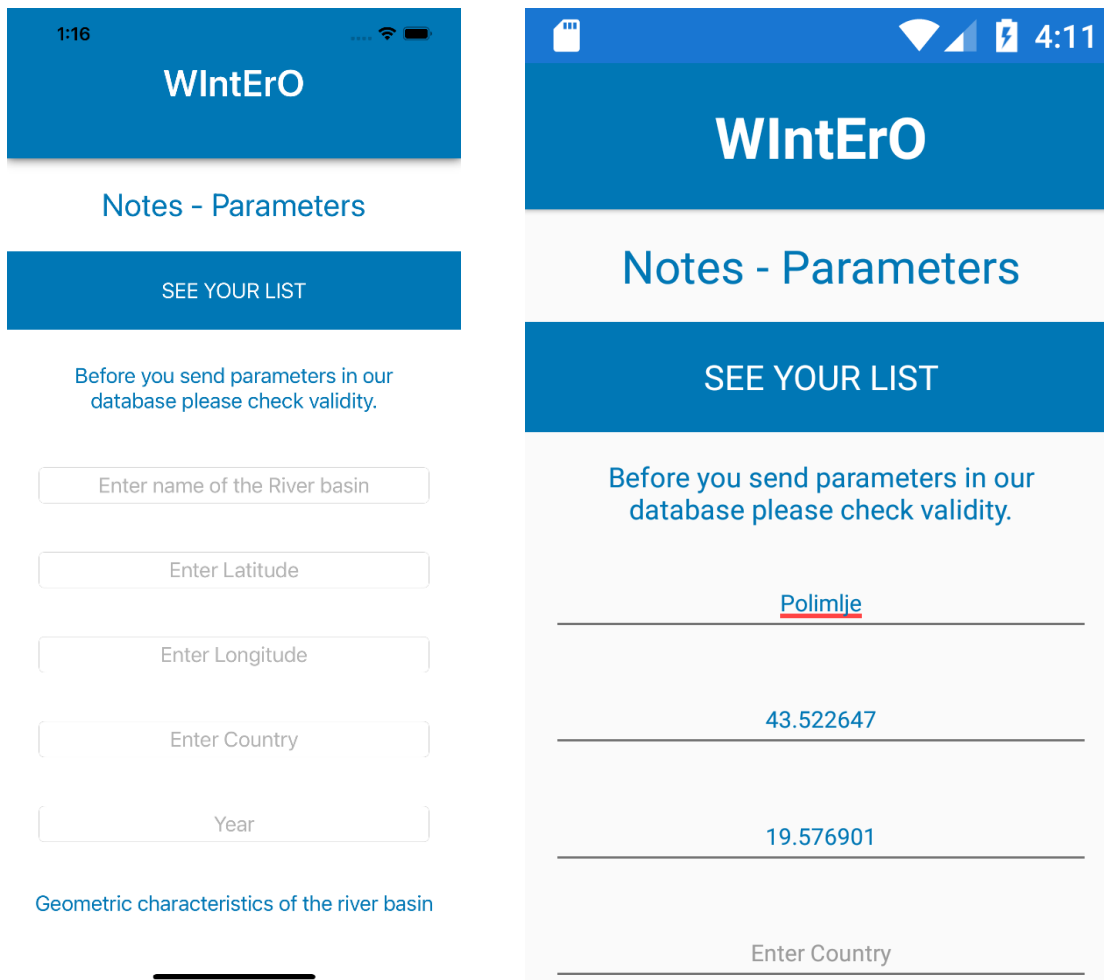
    }
}

```

Slika 32: Prikaz C# kodova koji su sastavni dio sekcije za zabilježbu novih parametara

⁵² SQLite sistem - <https://docs.microsoft.com/en-us/xamarin/get-started/quickstarts/database?pivots=windows>

Polja za unos parametara su tekstualnog i brojnog tipa. Za unos teksta, definisan je tip promjenjive *string* koji se koristi za rad sa tekstualnim podacima. Za sva ostala polja brojnog tipa primijenjeni su tipovi promljenjivih *int* i *double*. Svako polje za unos je označeno sa odgovarajućom oznakom, kao i sa mjernom jedinicom, ukoliko ona postoji. Nakon unosa parametara korisniku aplikacije je omogućeno da izvrši čuvanje podataka, ili eventualno da izbriše unose. Fajl koji sadrži naredbe za kreiranje forme je povezan sa fajlom u kojem se nalaze naredbe za izvršavanje. U ovom slučaju, naredbom *Clicked="sendparameters"*, poziva se procedura za slanje podataka. Na slici 33. nalazi se snimak dijela ekrana koji prikazuje prethodno opisano rješenje. Testiranje je izvršeno za iOS i Android uređaje.



Slika 33: Snimci ekrana iOS i Android uređaja, prikaz segmenta notes – parameters

5. Analiza i diskusija

Analiza i diskusija o realizaciji platforme ima za cilj da sumira postignute rezultate, a takođe i da predstavi metode koji su i koji će primjenom doprinijeti kvalitetnoj provjeri učinkovitosti, a takođe i daljoj evaluaciji.

Izrada i upotreba veb platforme i mobilnih aplikacija je predstavljena kroz treće i četvrto poglavlje ovog rada. Iz priloženog je očigledno da su uspostavljena rješenja ispunila sve zahtjeve koji su planirani i projektovani kroz izradu opisa i analiza projekcija. Zahtjevima je bilo predviđeno da se najprije omogući ispravna funkcionalnost što bi omogućilo preciznost kod dobijanja rezultata koji su od velikog značaja kada je u pitanju rad na postizanju smanjenja intenziteta erozije zemljišta, a takođe i da omogući dodatne funkcionalnosti koje bi obezbijedile jednostavnost pri korišćenju, a samim tim i rasprostranjeniju upotrebu.

WIntErO platforma je, kako je već ranije pomenuto, naslednik IntErO softverskog paketa. Samim tim, platforma je naslijedila dosta funkcionalnosti koje su sada dostupne na mreži i obezbijedila nove. Implementacijom je omogućena obrada parametara, dakle, unos, obračun korišćenjem određenog modela i na kraju analiza. Dodatno, obezbijedeno je objedinjavanje podataka u centralizovanu bazu i javna dostupnost istih, što je bio jedan od nedostataka IntErO paketa. Takođe, stvorena je i mogućnost sakupljanja podataka koji su ranije obrađivani pomoću IntErO paketa, i njihovo upisivanje u jedinstvenu bazu podataka. Samim objedinjavanjem, stvorena je i pogodnost za cjelokupan prikaz lokacija svih rječnih slivova za koje se vrše analize i to na interaktivnoj mapi. Dakle, pregled svih informacija je sada javno dostupan, dok je slanje informacija omogućeno korisnicima koji imaju kreiran nalog na WIntErO platformi. Za korisnike sa nalogom su kreirane i dodatne funkcionalnosti.

Za izradu platforme korištene su veoma zastupljene tehnologije i alati koji se ujedno i koriste za razvoj najsavremenijih softverskih rješenja. Njihova primjena je veoma široka, a odlikuju ih i konstantna unaprijeđenja koja prate zahtjeve najnovijih trendova. Samom njihovom primjenom omogućeno je i implementiranje budućih zahtjeva na platformi koji će se javiti čim se i pojave novi izazovi za istraživače.

Kompletna platforma je testirana i njena funkcionalnost je u potpunosti ispravna. Testiranje je izvršeno i provjerena je tačnost dobijenih rezultata koji se izračunavaju prema primijenjenim formulama modela koji se koristi [5], [7]. U tabeli 4. prikazani su podaci o intenzitetu erozije zemljišta koji su dobijeni upotrebom WIntErO platforme kako bi se prikazala ispravnost funkcionisanja. Kao dokaz validiteta podataka, napravljeno je poređenje sa ranijim istraživanjima za navedene lokacije. Rezultati se poklapaju sa rezultatima koji su dobijeni kroz pređašnja ispitivanja što nesumljivo pokazuje da se WIntErO platforma može koristiti za ostvarivanje ciljeva za koje je i namijenjena.

Tabela 4: Intenzitet erozije zemljišta sračunat korišćenjem WIntErO platforme

Lokacija	$G_{\text{god}} \text{ km}^{-2} (\text{m}^3\text{km}^{-2}\text{god}^{-1})$	Lokacija	$G_{\text{god}} \text{ km}^{-2} (\text{m}^3\text{km}^{-2}\text{god}^{-1})$
Djuricka rijeka	645.40320578739	Lukcka rijeka	266.20864795299
Komaračka rijeka	521.83864522697	Potok Bošnjak	197.98855827916
Bijeli potok	104.30833923355	Tifran	296.44891150156
Novšićki potok	288.96777561491	Seocki potok	255.59773875781
Velička rijeka	562.60324635743	Lješnica	325.19225762077
Dragovo vrelo	399.51743263262	Karličića potok	195.4673628371
Pepićka rijeka	328.96451339684	Biorički potok	264.43225613754
Krivački potok	180.21527753345	Tronoša	286.07459757403
Murinska rijeka	327.68928103057	Radulička rijeka	327.04362723922
Seremetski potok	298.19401494239	Brzava	452.91850117029
Zorin potok	417.67716854135	Rakljanska rijeka	210.3178111765
Duboki potok	278.33377886533	Duboki potok	131.22550391813
Piševska rijeka	427.62791475541	Ramčica	268.0271178115
Zim potok	330.12492972775	Đuren potok	194.75958921455
Provala	597.75647309471	Uvezački potok	293.82038704671
Zlorečića	403.46034148873	Sliv Ljubovide	236.55054719474
Krastica	370.61067677185	Pepića rijeka	262.6148652425
Malski potok	244.32437984527	Nikolin potok	200.42580918965
Trepčanska rijeka	219.39142366023	Rastočki potok	514.60326715077
Šekularska rijeka	286.90356859962	Lješnica (BP)	200.55902037113
Navotinski potok	88.659202765323	Sliv Lipnice	197.29149248141
Vinička rijeka	470.4203741423	Nedakusi	140.64270638328
Rovački potok	324.44709463024	Boljanska rijeka	315.27658284277
Krivaja	212.66772871194	Kisjele vode	216.30378504628
Bistrica (Berane)	385.40922402822	Sutivanska rijeka	250.39242345951
Kaludarska rijeka	492.67897114538	Orahovačka	256.39357539566
Makva	232.46537996851	Miočki potok	413.65867985443
Sušica	305.76130357284	Bistrica	269.25249970202
Dapsicka rijeka	268.08699050901		

U daljem tekstu će se govoriti i o primjeni metoda provjere učinkovitosti. Prije nego je i pokrenuta realizacija platforme, sprovedena je simulacija kako bi se izvršilo testiranje. Metod simulacije je u stvari naučni metod kojim se izvršila verifikacija i validacija. Verifikacijom je utvrđeno da će kreirana platforma u potpunosti odgovarati namjeni, a validacijom se došlo do zaključka da je omogućeno nesmetano funkcionisanje u realnom radu. Ispravnost upotrebe ovog metoda je i uspješno potvrđena nakon što je završeno kreiranje platforme i nakon što se izvršilo testiranje [11], [12], [13].

Metod koji takođe može da pomogne kako u ispravnom radu, tako i u budućim inovacijama je metod prikupljanja podataka. Ovaj metod ima i nekoliko načina, odnosno tehnika pomoću kojih se vrše prikupljanja. Radi se o direktnim, indirektnim i nezavisnim tehnikama kojima se postižu saznanja, tačnije prikupljaju povratne informacije od korisnika, a takođe i ispituje ispravan rad platforme [40].

Direktna tehnika metoda prikupljanja podataka je tehnika pomoću koje se realizuje anketa upitnika koja je veoma efikasna, kako u vremenskom smislu, tako i u smislu prikupljanja podataka od istraživača sa različitih geografskih lokacija. Kreiranjem ankete može se doći do različitih saznanja koja se tiču iskustava koje korisnici imaju u radu sa platformom, pa samim tim i izvršiti dalja unaprijeđenja na osnovu novih saznanja.

Indirektna tehnika ima za cilj da izvrši instrumentaciju različitih alata i dodataka na platformi kako bi se izvršavao monitoring. Monitoringom se može utvrditi način na koji se upotrijebljava platforma, zatim izvršavati analiza korišćenja alata i funkcija, a takođe se mogu i prikupljati podaci o pristupima sadržaju, odnosno o samoj posjećenosti platforme.

Na kraju, tu je i nezavisna tehnika, kojom se vrši kontrola određenih fajlova koji se nalaze na samom serveru. U pitanju su *log* fajlovi, i pristup istim se može izvršiti kroz hosting platformu cPanel koja ujedno i hostuje WIntErO platformu. Uvidom u *log* fajlove mogu se sagledati sve eventualne greške koje nastaju upotrebom veb platforme od strane korisnika.

6. Zaključak

U ovom radu je predstavljen razvoj veb platforme i mobilnih aplikacija upotrebom najsavremenijih dostupnih tehnologija i alata. Na samom početku opisana je problematika koja je trebala da se riješi, a zatim je i izvršena analiza postojećih solucija, a potom i projektovanog rješenja. Prikazana je i arhitekturnalna projekcija na osnovu koje se i bazira sama izrada. Dat je kratak pregled i svih bitnih upotrijebljenih markerskih i programskih jezika, a takođe i ostalih važnih alata koji su bili neophodni za izradu zadataka. Postupak izrade je prikazan na način što su predstavljeni programski kodovi, slike, tabele i formule uz koje je dat i korespondentan opis. Na kraju je izvršena analiza i testiranje uspostavljenog rješenja čime se došlo do utvrđivanja ispravnosti funkcionisanja.

Eksplozivni razvoj informaciono-komunikacionih tehnologija omogućio je razvoj WIntErO platforme koja može u mnogome da poveća produktivnost kada se radi o načinu dobijanja rezultata i samoj dostupnosti istih. Ovi benefiti predstavljaju olakšanje prilikom kreiranja preporuka varijanti načina korišćenja zemljišta koje imaju za cilj da utiču na smanjenje intenziteta erozije kod svih rječnih slivova koji su rasprostranjeni širom naše planete i nalaze se u različitim klimatskim i fizičko-geografskim uslovima [7].

Zemljište se smatra jednim od najvažnijih i najsloženijih resursa koje nam je stvorila priroda. Svakodnevna pojava različitih vrsta procesa u prirodi, a takođe i urbanizacija, sve više doprinose ugrožavanju ovog resursa. Jednim od najvažnijih ekoloških problema u svijetu, smatra se i degradacija zemljišta koja je prouzrokovana erozijom. Pojačan proces erozije utiče kako na poljoprivredu, tako i na rezervoare vode koji služe za piće, turizam, elektroprivredu i slično. Ovi procesi u mnogome mogu i da dovedu do promjena koje se odnose na usluge ekosistema [41].

Dalja sprečavanja erozionih procesa moguća su i kroz razvijanje naučno-istraživačkih projekata koji bi se realizovali primjenom informaciono-komunikacionih tehnologija, i koji bi postignutim rezultatima ukazivali na ekonomske, socijalne i ekološke pozitivne efekte koji su ostvareni. Takođe, može se kazati, da je primjena računarsko-grafičkih metoda, kao i upotreba geografsko-informacionih sistema, veoma bitan strateški pristup za upravljanje zemljištem. U današnjem vremenu, skoro sve vrste erozionih procesa se analiziraju pomoću alata koji omogućavaju daljinsku detekciju, a takođe i pomoću alata geografskog informacionog sistema (GIS – Geographic Information System). Postoji veliki broj modela koji su povezani sa GIS-om, a neki od njih su takođe i modeli koji su dostupni na mreži. Njihov zadatak je da obrađuju i upravljaju ulaznim podacima kako bi se izvršilo modeliranje i analiza rezultata [42]. U bliskoj prošlosti, proračuni su se obavljali klasičnim metodama, ali se danas zbog brzine rada i preciznosti prednost daje računarskim alatima.

Efikasna komunikacija između istraživača, kao i dostupnost korisnih informacija istraživačima, su benefiti koji su od presudnog značaja za dalji naučni napredak uopšte. Uspostavljena platforma WIntErO omogućuje niz pogodnosti i bolju saradnju između istraživača iz Crne Gore i regiona sa istraživačima iz ostalih krajeva svijeta. Do sada, u Crnoj Gori je bilo dostupno nekoliko programskih paketa koji su i predstavljeni u drugom poglavlju.

Dalji razvoj bi mogao da se ogleda u integraciji GIS sistema sa platformom WIntErO, čime bi se dodatno unaprijedilo funkcionisanje. U pozadini bi se nalazila dinamička tabela WIntErO modela. Neophodno bi bilo i obratiti pažnju da se ostvari pravilno razumijevanje i konceptualizacija sistema koji vodi do adaptivnog menadžmenta čime će se dovesti do uspostavljanja stalnog praćenje hidroloških procesa rječnih slivova.

Daljem razvoju, takođe, treba pristupiti studiozno, počev od predstavljanja zahtjeva, izbora tehnologija i alata, kao i procesa održavanja uspostavljenog rješenja. Programski kodovi i ostali dodaci koji su kreirani za potrebe WIntErO platforme su napisani na jasan i razumljiv način, što doprinosi prihvatljivom stepenu sigurnosti, lakšem održavanju i optimizaciji, a takođe i daljoj proširivosti.

7. Literatura

- [1] Sir Tim Berners-Lee Biography — Academy of Achievement. (October 18, 2018). Retrieved March 3, 2019, from <https://www.achievement.org/achiever/sir-timothy-berners-lee/>
- [2] Spalevic V., Dlabac A., Jovovic Z., Rakocevic J., Radunovic M., Spalevic B., Fustic B. (1999): The "Surface and distance Measuring" Program. *Acta Agriculture Serbica*, Vol IV, 8, 63-71, Serbia
- [3] Spalevic V., Dlabac A., Spalevic B., Fustic B., Popovic V. (2000): Application of computer - graphic methods in the studies of draining out and intensity of ground erosion, 1: program "River Basins", Vol 46(1-2): 19-36, Podgorica, Montenegro
- [4] Spalevic V. (2011): Impact of land use on runoff and soil erosion in Polimlje. Doctoral thesis, University of Belgrade, Faculty of Agriculture, 80-87, Serbia
- [5] Gavrilović S. (1972): Engineering on torrential streams and erosion. *Izgradnja*, Special Issue: 1–292 (in Serbian)
- [6] Efthimiou N. (2018): The importance of soil data availability on erosion modelling. *Catena* 165 (2018) 551–566.
- [7] Spalevic V. (2011): Impact of land use on runoff and soil erosion in Polimlje. Doctoral thesis, University of Belgrade, Faculty of Agriculture, 1-260, Serbia
- [8] Ikponmwosa D. Ighodaro, Francois S. Lategan & Shehu F. G. Yusuf (2013): The Impact of Soil Erosion on Agricultural Potential and Performance of Sheshegu Community Farmers in the Eastern Cape of South Africa
- [9] Panagos P., Borrelli P. (2017): All That Soil Erosion: the Global Task to Conserve Our Soil Resources p. 20-21.
- [10] L.D. Chen, Z. Huang, J. Gong, B.J. Fu, Y.L. Huang (2007): The effect of land cover / vegetation on soil water dynamic in the hilly area of the loess plateau, China
- [11] Aleksandar Susic, Dusko Vujacic, Velibor Spalevic, Nikola Zaric, Mladjen Jovanovic, Devraj Chalise, Goran Skataric, Patrick Ken Kalonde (2018): Application of new WIntErO model for calculation of soil erosion intensity of the Ljesnica river basin, Montenegro and the ability to use for plant production, 22nd International Eco-conference@ and 10th Eco-conference of Safe food, Novi Sad, Serbia
- [12] Velibor Spalevic, Aleksandar Susic, Dusko Vujacic, Nikola Zaric, Mladjen Jovanovic, Morteza Behzadfar, Goran Skataric Zeinab Hazbavi (2018): An integrated assessment of soil erosion intensity by using a new WIntErO model: Case studies of the Bistrica River Basin,

Montenegro and the S2-2 Watershed, Shirindareh Watershed, Iran, Savetovanje Nove metodologije i poslovni izazovi u melioracijama povodom jubileja – 70 godina katedre za Melioracije zemljišta. Poljoprivredni fakultet, Univerzitet u Beogradu, Srbija

[13] Velibor Spalevic, Milic Curovic, Vojislav Djekovic, Borislav Railic, Aleksandar Susic (2018): Application of new WIntErO model for calculation of soil erosion intensity: Case studies of the Zlorečica River Basin, Montenegro and the S1-1 Watershed, Shirindareh Watershed, Iran, 17th International Symposium “Prospects for the 3rd Millennium Agriculture” University of Agricultural Sciences and Veterinary Medicine, Cluj-Napoca, Romania

[14] Biswajeet Pradhan, Hyung-Sup Jung, Ghassan Beydoun (2018): Systems and Sensors in Geoscience Applications DOI: 10.1155/2018/7242495 License CC BY

[15] HTML 5.2 - W3C Recommendation. (December 14, 2017). Retrieved March 9, 2019, from <https://www.w3.org/TR/html52/>

[16] HTML Living Standard - WHATWG community. Retrieved March 9, 2019, from <https://html.spec.whatwg.org/multipage/introduction.html#history-2>

[17] A brief history of CSS until 2016 - World Wide Web Consortium (W3C). Retrieved March 10, 2019, from <https://www.w3.org/Style/CSS20/history.html>

[18] Rachel Andrew (2012): The CSS3 Anthology: Take Your Sites to New Heights Fourth Edition, SitePoint

[19] Luke Welling, Laura Thomson (2009): PHP and My SQL: Web Development, 4th Edition, Person Education, Inc.

[20] History of MySQL - Wikipedia. Retrieved March 10, 2019, from <https://en.wikipedia.org/wiki/MySQL#History>

[21] Announcing General Availability of MySQL 8.0 - The Oracle MySQL Blog. (April 19, 2018):, Retrieved March 10, 2019, from <https://blogs.oracle.com/mysql/announcing-general-availability-of-mysql-80>

[22] The 10 most popular programming languages, according to the 'Facebook for programmers' - Business Insider (Januar 22, 2019). Retrieved March 11, 2019, from <https://www.businessinsider.com/the-10-most-popular-programming-languages-according-to-github-2018-10>

[23] ECMAScript® (2020): Language Specification - Ecma International, 11th Edition, June 2020

[24] Jeremy McPeak (2011): JavaScript 24-Hour Trainer, 1st Edition, Wrox

[25] jQuery Community Experts (2011): jQuery Cookbook: Solutions & Examples for jQuery Developers, 1st Edition, O'Reilly Media

-
- [26] Tanja Ivanović (2017): Elektronske lekcije o radnom okruženju Bootstrap, Master rad, Matematički fakultet, Univerzitet u Beogradu
- [27] Welcome to the Visual Studio IDE - Microsoft. Retrieved March 14, 2019, from <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019>
- [28] Spalevic, V., Al-Turki, A. M., Barovic, G., Silva, M. L. N., Djurovic, N., Souza, W.S., Batista, P.V.G, and Curovic, M. (2016): Modeling of Soil Erosion by IntErO model: The Case Study of the Novsicki Potok Watershed of the Prokletije high mountains of Montenegro. Geophysical Research Abstracts. Vol. 18, EGU2016-13864, 2016. EGU General Assembly 2016.
- [29] Darvishan, A.K., Derikvandi, M., Aliramaee, R., Spalevic, V., Behzadfar, M (2016): Efficiency of IntErO Model to Predict Soil Erosion Intensity and Sediment Yield in Khamsan Representative Watershed (West of Iran). The 5th EUROSIL International Congress, Istanbul, Turkey.
- [30] Draganic, J., Drobnjak, B., Campar, J., Bulajic, B., Zajovic, V., Behzadfar, M. and Spalevic, V. (2015): Calculation of Sediment yield using the IntErO Model in the S1-3 Watershed of the ShirinDareh River Basin, Iran. 9th Congress of the Soil Science Society of Bosnia and Herzegovina. 23rd – 25th of November 2015, Mostar, Bosnia and Herzegovina.
- [31] Spalevic, V., Curovic, M., Vujacic, D., Barovic, G., Frankl, A., and Nyssen, J. (2015): Assessment of soil erosion at the Brzava small watershed of Montenegro using the IntErO model. Geophysical Research Abstracts. Vol. 17, EGU2015-15007, 2015. EGU General Assembly
- [32] Heiko Schuldt (2008): Multi-Tier Architecture, Database and Information Systems Group, Department of Computer Science, University of Basel, Basel, Switzerland
- [33] Three-tier architectures - IBM Knowledge Center (March 19, 2020). Retrieved March 22, 2021, from <https://www.ibm.com/docs/en/was/9.0.5?topic=overview-three-tier-architectures>
- [34] Peter Morville (1998): Information Architecture on the World Wide Web, First Edition, O'Reilly
- [35] Marta Lichaj - Information Architecture Definition, Roles, Strategies and Resources. Retrieved May 29, 2020, from <https://brainhub.eu/blog/information-architecture/>
- [36] Shaun Lewis, Mike Dunn (2019): Native Mobile Development: A Cross-Reference for iOS and Android, 1st edition, O'Reilly Media
- [37] Charles Petzold (2016): Creating Mobile Apps with Xamarin.Forms: Cross-platform C# programming for iOS, Android, and Windows, Microsoft Press
- [38] Mark J. Price (2017): C# 7.1 and .NET Core 2.0 – Modern Cross-Platform Development - Third Edition: Create powerful applications with .NET Standard 2.0, ASP.NET Core 2.0, and Entity Framework Core 2.0, using Visual Studio 2017 or Visual Studio Code, Packt Publishing; 3rd edition
-

[39] Joseph Albahari, Ben Albahari (2017): *C# 7.0 in a Nutshell: The Definitive Reference*, 1st Edition, O'Reilly Media

[40] Jim Jeffers James Reinders (2015): *High Performance Parallelism Pearls Volume Two*, 1st Edition, Morgan Kaufmann

[41] Dragičević N, Karleuša B, Ožanić N (2019): Different Approaches to Estimation of Drainage Density and Their Effect on the Erosion Potential Method. *Water* 11: 593.

[42] Park Y, Kim J, Engel B (2009): Development of web-based SWAT System. *Proceedings of ASABE*, Reno, Nevada, June 2009

8. Dodatak A

Dio PHP koda kojim se vrši prihvatanje podataka, slanje parametara u bazu i slanje obavještenja korisniku

```
<?php

session_start();
require_once('../connect.php'); //autentifikacija sa bazom
include('../config.php'); //smtp

// uključivanje PHPMailer biblioteke
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'PHPMailer/src/Exception.php';
require 'PHPMailer/src/PHPMailer.php';
require 'PHPMailer/src/SMTP.php';

if(!isset($_SESSION['emaillogin']) & empty($_SESSION['emaillogin'])) {
//autentifikacija korisnika

    header('location: ../index.php');

}

$email = $_SESSION['emaillogin'];

if(isset($_POST['send']) & !empty($_POST['send'])){ //"prihvatanje" i slanje
parametara

    $riverbasin = $_POST['riverbasin'];
    $country = $_POST['country'];
    $year = mysqli_real_escape_string($connection, $_POST['year']);
    $latitude = $_POST['latitude'];
    $longitude = $_POST['longitude'];

    $f = $_POST['f'];
    $o = $_POST['o'];
    $fv = $_POST['fv'];
    $fm = $_POST['fm'];
    $lv = $_POST['lv'];
    $lb = $_POST['lb'];

    $lizarray55 = $_POST['lizarray'];
    $farray55 = $_POST['farray'];

    $h0 = $_POST['h0'];
    $deltah = $_POST['deltah'];
    $hmin = $_POST['hmin'];
    $hmax = $_POST['hmax'];
```

```
$sumal = $_POST['sumal'];
$l1m = $_POST['l1m'];

$fp = $_POST['fp'];
$fpp = $_POST['fpp'];
$fo = $_POST['fo'];

$fs = $_POST['fs'];
$ft = $_POST['ft'];
$fg = $_POST['fg'];

$hb = $_POST['hb'];
$temperature = $_POST['temperature'];
$up = mysqli_real_escape_string($connection, $_POST['up']);
$hgod = $_POST['hgod'];

$y = $_POST['y'];
$y1 = $_POST['y1'];
$y2 = $_POST['y2'];
$y3 = $_POST['y3'];
$y4 = $_POST['y4'];
$y5 = $_POST['y5'];
$y6 = $_POST['y6'];
$y7 = $_POST['y7'];
$y8 = $_POST['y8'];
$y9 = $_POST['y9'];
$y10 = $_POST['y10'];

$xa = $_POST['xa'];
$xa1 = $_POST['xa1'];
$xa2 = $_POST['xa2'];
$xa3 = $_POST['xa3'];
$xa4 = $_POST['xa4'];
$xa5 = $_POST['xa5'];
$xa6 = $_POST['xa6'];
$xa7 = $_POST['xa7'];

$phi = $_POST['phi'];
$phi1 = $_POST['phi1'];
$phi2 = $_POST['phi2'];
$phi3 = $_POST['phi3'];
$phi4 = $_POST['phi4'];
$phi5 = $_POST['phi5'];
$phi6 = $_POST['phi6'];
$phi7 = $_POST['phi7'];
$phi8 = $_POST['phi8'];
$phi9 = $_POST['phi9'];
$phi10 = $_POST['phi10'];

$hsr0 = $_POST['hsrvalue'];
$ISR0 = $_POST['ISRvalue'];

//upis u bazu podataka
```

```

$sql = "INSERT INTO data (riverbasin, country, year, latitude, longitude, f, o, fv,
fm, lv, lb, lizarray, farray, h0, deltah, hmin, hmax, sumal, lm, fp, fpp, fo, fs, ft,
fg, hb, temperature, up, hgod, y, y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, xa, xa1,
xa2, xa3, xa4, xa5, xa6, xa7, phi, phi1, phi2, phi3, phi4, phi5, phi6, phi7, phi8,
phi9, phi10, hsr, isr, emailuser) VALUES ('$riverbasin', '$country', '$year',
'$latitude', '$longitude', '$f', '$o', '$fv', '$fm', '$lv', '$lb', '$lizarray55',
'$farray55', '$h0', '$deltah', '$hmin', '$hmax', '$sumal', '$lm', '$fp', '$fpp',
'$fo', '$fs', '$ft', '$fg', '$hb', '$temperature', '$up', '$hgod', '$y', '$y1', '$y2',
'$y3', '$y4', '$y5', '$y6', '$y7', '$y8', '$y9', '$y10', '$xa', '$xa1', '$xa2',
'$xa3', '$xa4', '$xa5', '$xa6', '$xa7', '$phi', '$phi1', '$phi2', '$phi3', '$phi4',
'$phi5', '$phi6', '$phi7', '$phi8', '$phi9', '$phi10', '$hsr0', '$isr0', '$email');"

$result = mysqli_query($connection, $sql);

if($result) {

    $id = mysqli_insert_id($connection);

}

$mail = new PHPMailer(true); //slanje e-mail poruke
try {
    //podesavanja
    $mail->SMTPDebug = false;
    $mail->isSMTP(); //smtp
    $mail->Host = $smtphost;
    $mail->SMTPAuth = true; //SMTP autentifikacija
    $mail->Username = $smtpusername;
    $mail->Password = $smtppassword;
    $mail->SMTPSecure = 'ssl'; //TLS enkripcija
    $mail->Port = 465; //port za konektovanje

    //primalac
    $mail->setFrom('admin@wintero.me', 'WIntErO');
    $mail->addAddress("$email", 'Wintero');

    //Content
    $mail->isHTML(true); //podesavanje HTML formata
    $mail->Subject = 'WIntErO - .sli entry';
    $mail->Body = "Your parameters for ".$riverbasin." river basin was
succesfully sent! Thank You!";

    $mail->AltBody = '';

    $mail->send();
    $message = "Parameters succesfully sent! Thank You!";
}
catch (Exception $e) {
    echo 'Message could not be sent. Mailer Error: ', $mail->ErrorInfo;
}

}

?>

```

Dio koda kojim se kreira jedna od sekcija u kojoj se upisuju parametri

```

<div class="col-12">
  <h5 class="sectionname">Geometric characteristics of the river basin
    <div class="heading-underline-input"></div>
  </h5>
</div>

<div class="form-row">
  <div class="form-group col-lg-4 " title="Surface area of the drainage basin">
    <label>F (km<sup>2</sup></label><input type="number" name="f" required class="form-control"/></div>
  <div class="form-group col-lg-4 " title="Surface area of smaller portion of the drainage basin">
    <label>Fm (km<sup>2</sup></label><input type="number" name="fm" required class="form-control"/></div>
  <div class="form-group col-lg-4 " title="Natural length of main water course">
    <label>Lv (km)<input type="number" name="lv" required class="form-control"/></div>
</div>

<div class="form-row">
  <div class="form-group col-lg-4 " title="Length of the watershed">
    <label>O (km)<input type="number" name="o" required class="form-control"/></div>
  <div class="form-group col-lg-4 " title="Surface area of greater portion of the drainage basin">
    <label>Fv (km<sup>2</sup></label><input type="number" name="fv" required class="form-control"/></div>
  <div class="form-group col-lg-4 " title="Length of the drainage basin measured by a series of parallel lines">
    <label>Lb (km)<input type="number" name="lb" required class="form-control"/></div>
</div>

```

Dio koda koji se upotrebljava za dobijanje rezultata

```

<table class='table table-bordered table-hover'>
<tr><td><p>Coefficient of the river basin form</p></td><td>A =
<?php
    echo $a = 0.195 * $a1;
?>
</td>
<tr><td><p>Coefficient of the watershed development</p></td><td>m =
<?php

```

```

$m1 = pi() * $f;
    $m2 = sqrt($m1);
    $m3 = 2 * $m2;
    echo $m = $lv / $m3;
?>
</td></tr>
<tr><td><p>Average river basin width</p></td><td>B =
<?php
    echo $b = $f / $lb;
?> km
</td></tr>
<tr><td><p>(A)symmetry of the river basin</p></td><td>a =
<?php
    $as1 = $fv - $fm;
    $as2 = $fv + $fm;
    $as3 = 0.5 * $as2;
    echo $as = $as1 / $as3;
?>
</td></tr>
<tr><td><p>Density of the river network of the basin</p></td><td>G =
<?php
    echo $g = $sumal / $f;
?>
</td></tr>
<tr><td><p>Coefficient of the river basin tortuousness</p></td><td>K =
<?php
    echo $k = $lv / $lm;
?>
</td></tr>
<tr><td><p>Average river basin altitude</p></td><td>H<sub>sr</sub> =
<?php
    echo $hsr0;
?> m
</td></tr>
<tr><td><p>Average elevation difference of the river basin</p></td><td>D =
<?php
    echo $d = $hsr0 - $hmin;
?> m
</td></tr>
<tr><td><p>Average river basin decline</p></td><td>I<sub>sr</sub> =
<?php
    echo $isr0;
?> %
</td></tr>
<tr><td><p>The height of the local erosion base of the river
basin</p></td><td>H<sub>le</sub> =
<?php
    echo $hle<sub>b</sub> = $hmax - $hmin;
?> m
</td></tr>
<tr><td><p>Coefficient of the erosion energy of the river basin's
relief</p></td><td>E<sub>r</sub> =
<?php
    $her1 = pow($f, 0.25);

```



```

        $her2 = pi() * $her1;
        echo $her = $hleb / $her2;
    ?>
</td></tr>
<tr><td><p>Coefficient of the region's permeability</p></td><td>S<sub>1</sub> =
<?php
        echo $s1 = 0.4 * $fp + 0.7 * $fpp + 1.0 * $fo;
    ?>
</td></tr>
<tr><td><p>Coefficient of the vegetation cover</p></td><td>S<sub>2</sub> =
<?php
        echo $s2 = 0.6 * $fs + 0.8 * $ft + 1.0 * $fg;
    ?>
</td></tr>
<tr><td><p>Analytical presentation of the water retention in inflow</p></td><td>W =
<?php
        echo $w = $hb/1000 * (15 - 22 * $hb/1000 - 0.3 * sqrt($lv));
    ?> m
</td></tr>
<tr><td><p>Energetic potential of water flow during torrent
rains</p></td><td>2gDF<sup>1/2</sup> =
<?php
        $ep1 = $d/1000;
        $ep2 = 2 * 9.81/1000 * $ep1 * $f;
        echo $ep = pow($ep2, 0.5)*1000;
    ?> m km s<sup>-1</sup>
</td></tr>
<tr><td><p>Maximal outflow from the river basin</p></td><td>Q<sub>max</sub> =
<?php
        $qmax1 = ($a * $s1 * $s2 * $w/1000);
        $qmax2 = sqrt($ep2);
        echo $qmax = $qmax1 * $qmax2 *1000000;
    ?> m<sup>3</sup> s<sup>-1</sup>
</td></tr>
<tr><td><p>Temperature coefficient of the region</p></td><td>T =
<?php
        echo $t = sqrt($temperature / 10 + 0.1);
    ?>
</td></tr>
<tr><td><p>Coefficient of the river basin erosion</p></td><td>Z =
<?php
        $z1 = $y * $xa * sqrt($isr0 / 100);
        echo $z = $y * $xa * $phi + $z1;
    ?>
</td></tr>
<tr><td><p>Production of erosion material in the river
basin</p></td><td>W<sub>god</sub> =
<?php
        $wgod1 = pow($z, 3);
        echo $wgod = $t * $hgod * pi() * $f * sqrt($wgod1);
    ?> m<sup>3</sup> god<sup>-1</sup>
</td></tr>
<tr><td><p>Coefficient of the deposit retention</p></td><td>R<sub>u</sub> =
<?php

```

```

$ru1 = $o * ($d/1000);
    $ru2 = pow($ru1, 0.5);
    $ru3 = 0.5 * ($lv + 10);
    echo $ru = $ru2 / $ru3;
?>
</td></tr>
<tr><td><p>Real soil losses</p></td><td>G<sub>god</sub> =
<?php
    echo $ggod = $wgod * $ru;
?> m<sup>3</sup> god<sup>-1</sup>
</td></tr>
<tr><td><p>Real soil losses per km<sup>2</sup></p></td><td>G<sub>god</sub> km<sup>-
2</sup> =
<?php
    echo $ggodkm2 = $ggod / $f;
?> m<sup>3</sup> km<sup>-2</sup> god<sup>-1</sup>
</td></tr>
</table>

```

Dio JavaScript koda kojim se stilizuju polja koja omogućavaju padajuću listu

```

<script>
    $(document).ready(function(){
        $("input.latlog").change(function(){
            if ($(this).val()=="") $(this).css({color: "white"});
            else $(this).css({color: "#3059A4"});
        });
    });
</script>
<script>
    $(document).ready(function(){
        $("input.latlog").change(function () {
            if ($(this).val()=="") $(this).css({background: "indianred"});
            else $(this).css({background: "white"});
        });
    });
</script>

```

9. Dodatak B

PHP kodovi kojima se izvršava očitavanje .sli fajlova i parametara

```

<?php

if(isset($_POST['submit'])) {
    $target = "../uploads/"; //folder na serveru u kojem se smjestaju .sli fajlovi
    $file_name = $_FILES['file']['name'];
    $tmp_dir = $_FILES['file']['tmp_name'];

    if(!preg_match('/(sli)$/i', $file_name)) { //dozvoljeni tipovi podataka
        echo "That file type is not allowed";
    } else {
        move_uploaded_file($tmp_dir, $target . $file_name);
        $target_path = $_SERVER['DOCUMENT_ROOT'] . "/uploads/" .
basename($_FILES['file']['name']);
    }
}
?>

<?php
$order = file($target_path);
$number_of_orders = count($order); //prebrojavanje
if ($number_of_orders == 0) {
    echo '<p><strong>Please try with different file.</strong></p>';
}
for ($i=0; $i < $number_of_orders; $i++) {
    $found["found".$i] = $order[$i];
    extract($found);
}
?>

<?php
$lizarray = array(); //duzine izohipsi
for ($i=32; $i < 32+$found31-1; $i++) {
    $lizarray[] = $order[$i];
}
for ($i = 0; $i < count($lizarray); $i++) {
    $key=key($lizarray);
    $val=$lizarray[$key];
    if ($val<> ' ') {
        $key ." = " . $val ." <br> ";
    }
    next($lizarray);
}
$lizarrayj = json_encode($lizarray, JSON_UNESCAPED_UNICODE |
JSON_UNESCAPED_SLASHES); //json kodiranje
$fhsr = $found25;
$deltahsr = $order[31+$found31+$found31+4]/1000;
for ($i = 0; $i < count($lizarray); $i++) {

```

```

if ($i == 0) {
    $isr1 = ($lizarray[$i]/2) * $deltahsr;
} elseif ($i > 0 && $i < count($lizarray)-1) {
    $isr2 = (($lizarray[$i-1] +
$lizarray[$i])/2)*$deltahsr;
    $isr4 += $isr2;
} elseif ($i == count($lizarray)-1) {
    $isr3 = (($lizarray[$i]/2) * $deltahsr) +
((( $lizarray[$i-1] + $lizarray[$i])/2)*$deltahsr);
}
}
$isrmain = $isr1 + $isr4 + $isr3;
$isr = $isrmain / $fhsr * 100;
?>

<?php
$farray = array(); //povrsine izmedju susjednih izohipsi
for ($i=31+$found31+2; $i < 32+$found31+$found31+1; $i++) {
    $farray[] = $orders[$i];
}
for ($i = 0; $i < count($farray); $i++) {
    $key=key($farray);
    $val=$farray[$key];
    if ($val<> ' ') {
        $key ." = ". $val ." <br> ";
    }
    next($farray);
}
$farrayj = json_encode($farray, JSON_UNESCAPED_UNICODE | JSON_UNESCAPED_SLASHES);
//json kodiranje
$fhsr = $found25;
$h0hsr = $orders[31+$found31+$found31+3];
$deltahsr = $orders[31+$found31+$found31+4];
$hminhsr = $orders[31+$found31+$found31+5];
$hmaxhsr = $orders[31+$found31+$found31+6];
$hsrh2 = ($h0hsr+$h0hsr+$deltahsr)/2;
for ($i = 0; $i < count($farray); $i++) {
    if ($i == 0) {
        $hsrh1 = (($h0hsr+$hminhsr)/2);
        $hsr1 = ($hsrh1/1000) * $farray[$i];
    } elseif ($i > 0 && $i < count($farray)-1) {
        $hsr2 = ($hsrh2/1000) * $farray[$i];
        $hsrh2 = ($hsrh2 + $deltahsr);
        $hsr4 += $hsr2;
    } elseif ($i == count($farray)-1) {
        $hsr3 = ((( $hmaxhsr+$hsrh2-$deltahsr)/2)/2)/1000) *
$farray[$i];
    }
}
$hsrmain = $hsr1+$hsr4+$hsr3;
$fhsr;
$hsr = $hsrmain / $fhsr * 1000;
?>

```

Programski kôd kojim se parametri o dužinama izohipsi i površinama između susjednih izohipsi prikazuju u tabelama

```

<table>
<tr><td>
  <?php
    echo "<table class=\"table1\" title=\"Length of the contours\"
border=\"1\">\n";
    echo "<tr><th class=\"firstt\" bgcolor=\"white\">Liz [km]:</th><tr>";
    for ($i=32; $i < 32+$found31-1; $i++) {
      $line = explode("\t", $orders[$i]);
      echo "<tr><td><input required style=\"border: 0px solid; color:
#0077B5;\" type=\"text\" name=\"inputf\" value=\"\$line[0]\"/></td></tr>";
    } echo '</table>' ?>
  </td><td>
    <?php
      echo "<table class=\"table2\" title=\"Surface between the two neighboring
contours\" border=\"1\">\n";
      echo "<tr><th class=\"firstt\" bgcolor=\"white\">f
[km<sup>2</sup>]:</th><tr>";
      for ($i=31+$found31+2; $i < 32+$found31+$found31+1; $i++) {
        $line = explode("\t", $orders[$i]);
        echo "<tr><td><input required style=\"border: 0px solid; color:
#0077B5;\" type=\"text\" name=\"inputf\" value=\"\$line[0]\"/></td></tr>";
      } echo '</table>';
    } ?>
  </td></tr>
</table>

```

Dio koda kojim se kreira jedna od sekcija u kojoj se upisuju parametri koji se očitavaju iz *.sli* fajlova i JavaScript kôd kojim se ova sekcija realizuje sa elementom *accordion* (sklopivi sadržaj)

```

<div class="form-group col-lg-12 " title="Numerical coefficient of visible and clearly
pointed processes of soil erosion">
  <button class="accordion" title="Numerical coefficient of visible and clearly
pointed processes of soil erosion" type="button"><label class="smalllabel"> $\phi$  = <input
class="small" type="number" name="phi" required value=<?php echo
$orders[31+$found31+$found31+37]?> /></label></button>
  <div class="panel">
</div>
</br>
<h5>Depth erosion: <input class="coefficientcolor" type="number" name="phi1" required
value=<?php echo $orders[31+$found31+$found31+38]?> /> %</h5>
  <h5>80% of the river basin under rill and gully erosion: <input
class="coefficientcolor" type="number" name="phi2" required value=<?php echo
$orders[31+$found31+$found31+39]?> /> %</h5>
  <h5>50% of the river basin under rill and gully erosion: <input
class="coefficientcolor" type="number" name="phi3" required value=<?php echo
$orders[31+$found31+$found31+40]?> /> %</h5>
  <h5>100% of the river basin under surface erosion: <input class="coefficientcolor"
type="number" name="phi4" required value=<?php echo $orders[31+$found31+$found31+41]?>
/> %</h5>

```

```

<h5>100% of the river basin under surface erosion, without visible furrows, ravines
and land slides: <input class="coefficientcolor" type="number" name="phi5" required
value=<?php echo $orders[31+$found31+$found31+42]?>/> %</h5>
  <h5>50% of the river basin under surface erosion: <input class="coefficientcolor"
type="number" name="phi6" required value=<?php echo $orders[31+$found31+$found31+43]?>
/> %</h5>
  <h5>20% of the river basin under surface erosion: <input class="coefficientcolor"
type="number" name="phi7" required value=<?php echo $orders[31+$found31+$found31+44]?>
/> %</h5>
  <h5>There are smaller slides in the watercourse beds: <input
class="coefficientcolor" type="number" name="phi8" required value=<?php echo
$orders[31+$found31+$found31+45]?> /> %</h5>
  <h5>The river basin mostly under plough-land: <input class="coefficientcolor"
type="number" name="phi9" required value=<?php echo $orders[31+$found31+$found31+46]?>
/> %</h5>
  <h5>The river basin under forests and perennial vegetation: <input
class="coefficientcolor" type="number" name="phi10" required value=<?php echo
$orders[31+$found31+$found31+47]?> /> %</h5>
</div>
</div>

<script>
  var acc = document.getElementsByClassName("accordion");
  var i;

  for (i = 0; i < acc.length; i++) {
    acc[i].addEventListener("click", function () {
      this.classList.toggle("active");
      var panel = this.nextElementSibling;
      if (panel.style.maxHeight) {
        panel.style.maxHeight = null;
      } else {
        panel.style.maxHeight = panel.scrollHeight + "px";
      }
    });
  }
</script>

```

Dio CSS koda za element *accordion*

```

.accordion:hover {
  background-color: white;
  color: #0077B5;
}

.accordion:after {
  content: '\002B';
  color: white;
  font-weight: bold;
  float: right;
  margin-left: 5px;
}

```

10. Dodatak C

PHP kôd kojim se kreira XML fajl koji ima ulogu da prikupi neophodne podatke iz baze podataka, a koji su potrebni za prikazivanje lokacija rječnih slivova na *Google mapi*

```
<?php
require("connectxml.php");

// kreiranje XML fajla
$dom = new DOMDocument("1.0");
$node = $dom->createElement("markers");
$parnode = $dom->appendChild($node);

// Ouspostavljanje konekcije sa MySQL serverom
$connection=mysql_connect ('localhost', $username, $password);
if (!$connection) { die('Not connected : ' . mysql_error());}

// odabiranje odgovarajuće baze
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}

// selektovanje redova u tabeli
$query = "SELECT * FROM data";
$result = mysql_query($query);
if (!$result) {
    die('Invalid query: ' . mysql_error());
}

header("Content-type: text/xml");

// iteracija, dodavanje
while ($row = @mysql_fetch_assoc($result)){
    // upisivanje podataka u XML fajl
    $node = $dom->createElement("marker");
    $newnode = $parnode->appendChild($node);
    $newnode->setAttribute("id",$row['id']);
    $newnode->setAttribute("riverbasin",$row['riverbasin']);
    $newnode->setAttribute("country", $row['country']);
    $newnode->setAttribute("year", $row['year']);
    $newnode->setAttribute("latitude", $row['latitude']);
    $newnode->setAttribute("longitude", $row['longitude']);
}

echo $dom->saveXML();

?>
```

JavaScript kôd kojim se izvršava prikazivanje Google mape sa tagovima

```
<script>

function initMap() {

    var cluster = [];
    var map = new google.maps.Map(document.getElementById('map'), {
        center: new google.maps.LatLng(20, 0),
        zoom: 3
    });
    var infoWindow = new google.maps.InfoWindow;

    // pozivanje XML fajla
    downloadUrl('outputxml.php', function (data) {
        var xml = data.responseXML;
        var markers = xml.documentElement.getElementsByTagName('marker');
        Array.prototype.forEach.call(markers, function (markerElem) {
            var id = markerElem.getAttribute('id');
            var riverbasin = markerElem.getAttribute('riverbasin');
            var country = markerElem.getAttribute('country');
            var year = markerElem.getAttribute('year');
            var link = "https://www.wintero.me/display.php?id=" + id;

            var point = new google.maps.LatLng(
                parseFloat(markerElem.getAttribute('latitude')),
                parseFloat(markerElem.getAttribute('longitude')));

            var infowincontent = document.createElement('div');
            var strong = document.createElement('strong');
            strong.textContent = riverbasin
            infowincontent.appendChild(strong);
            infowincontent.appendChild(document.createElement('br'));

            var text = document.createElement('text');
            text.textContent = country
            infowincontent.appendChild(text);
            infowincontent.appendChild(document.createElement('br'));

            var textyear = document.createElement('textyear');
            textyear.textContent = year
            infowincontent.appendChild(textyear);
            infowincontent.appendChild(document.createElement('br'));

            infowincontent.appendChild(document.createElement('br'));

            var textmore = document.createElement('a');
            textmore.textContent = link

            textmore.innerHTML = "See detailed";
            textmore.setAttribute('href', "https://www.wintero.me/display.php?id="
+ id);
```



```
textmore.setAttribute('target', '_blank');

    infowincontent.appendChild(textmore);
    infowincontent.appendChild(document.createElement('br'));

    var marker = new google.maps.Marker({
        map: map,
        position: point,
        draggable: true,
        animation: google.maps.Animation.DROP,

    });

    marker.addListener('click', function () {
        infoWindow.setContent(infowincontent);
        infoWindow.open(map, marker);
    });

    cluster.push(marker);

});

var options = { imagePath:
'https://developers.google.com/maps/documentation/javascript/examples/markerclusterer/
m' };
    var mc = new MarkerClusterer(map,cluster,options);

});
}
function downloadUrl(url, callback) {
    var request = window.ActiveXObject ?
    new ActiveXObject('Microsoft.XMLHTTP') :
    new XMLHttpRequest;

    request.onreadystatechange = function () {
        if (request.readyState == 4) {
            request.onreadystatechange = doNothing;
            callback(request, request.status);
        }
    };

    request.open('GET', url, true);
    request.send(null);
}

function doNothing() { }
```

```
</script>

<script src="markerclusterer.js"></script>

<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBRs26Z51Xn3gxojp2ADtrmS2i6CChA6
-0&callback=initMap"></script>
```