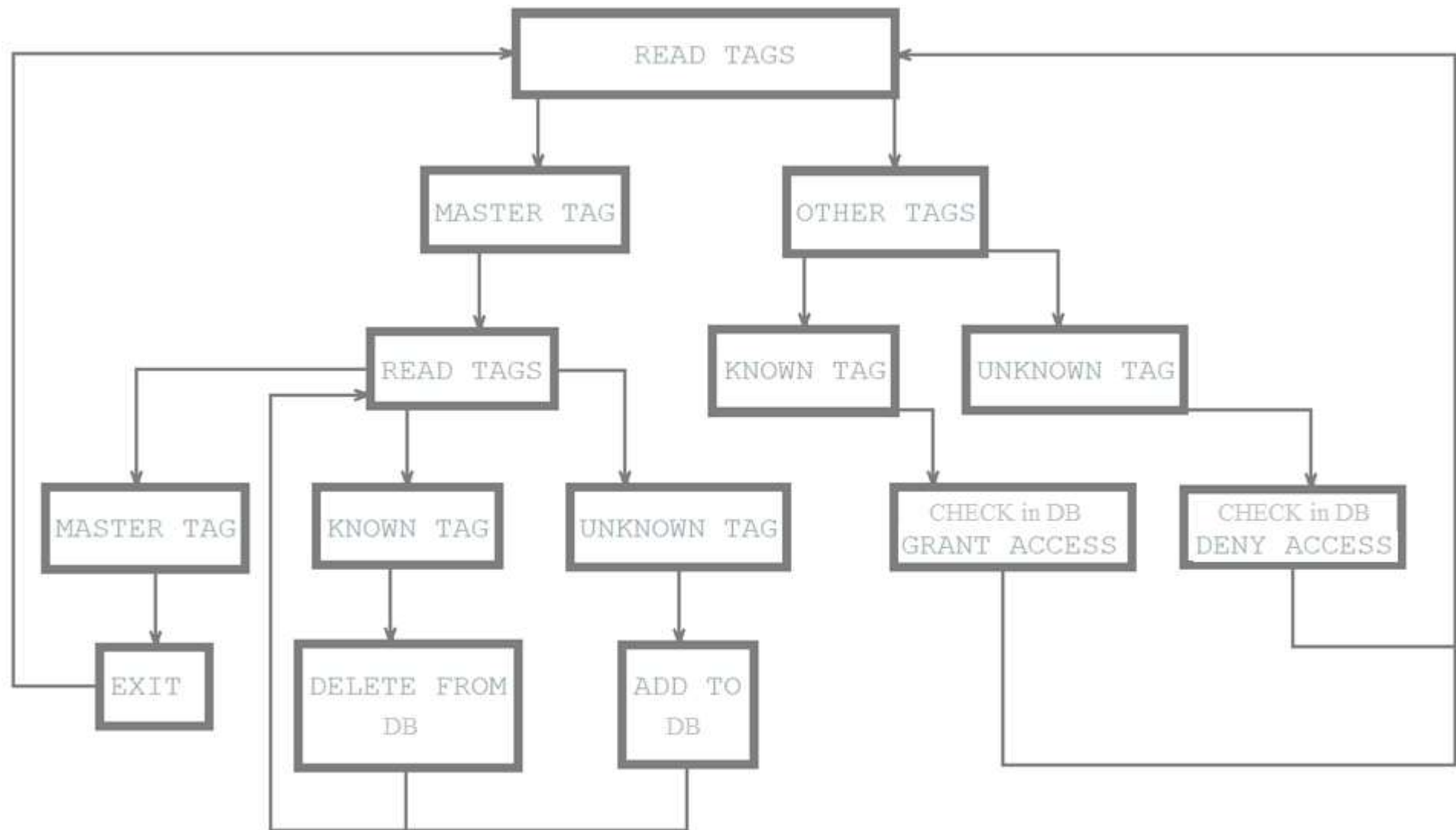


IDENTIFIKACIONI SISTEMI S BAZOM PODATAKA



ID UREĐAJ - DIJAGRAM TOKA



POVEZIVANJE ID UREĐAJA I BAZE PODATAKA



HTTP POST



1 /resource

Body URL Encoded
api_key=API_KEY&field1=30



HTTP Response



2 Status 200 (OK)

HTTP – za komuniciranje između servera i klijenta

- ▶ HTTP – Hypertext Transfer protokol
 - ▶ Dizajniran da omogući komunikaciju između servera i klijenta
 - ▶ Protokol zahtjeva i odgovora
 - ▶ Klijent šalje HTTP zahtjev serveru – server klijentu uzvraća odgovor
 - ▶ Odgovor sadrži status izvršenja zahtjeva, a može sadržati i dodatne podatke.
-
- ▶ U radu sa ThingSpeak platformom Arduino uređaj će imati ulogu klijenta a ThingSpeak platforma ulogu servera.

HTTP zahtjev

HTTP zahtjev generiše klijent, prema imanovanom host-u, lociranom na serveru.

Cilj zahtjeva je pristup resursu na serveru.

Korektno sastavljen HTTP zahtjev sadrži sljedeće elemente:

- Liniju zahtjeva;
- HTTP zaglavlja;
- Tijelo poruke, ako je potrebno.

Nakon svakog HTTP zaglavlja slijedi znak za povratak na početak reda (carriage return) i znak za prelazak u novi red (line feed) (CR-LF). Nakon poslednjeg zaglavlja dodatni CR-LF je dodat (za dobijanje prazne linije), nakon kojeg počinje tijelo poruke.

HTTP zahtjev – Linija zahtjeva

Linija zaglavlja je prva linija u poruci zahtjeva. Sastoji se iz tri dijela:

- ▶ Metod. Metod je jedno-rječna komanda koja govori serveru što da radi sa resursom. Na primjer, server može biti upitan da pošelje resurs klijentu.
- ▶ Komponenta staze URL-a za zahtjev. Staza identifikuje resurs na serveru.
- ▶ Broj HTTP verzije, ukazuje na HTTP specifikaciju s kojom je klijent pokušao uskladiti poruku.

Primjer linije zahtjeva:

```
GET /software/htp/cic/indeks.html HTTP/1.1
```

Linija zahtjeva može sadržati i dodatne podatke.

HTTP zahtjev – Zaglavlje (Header)

- Pruža prijemnoj strani informacije o poruci, pošiljaocu i načinu na koji pošiljaoc želi da komunicira sa primaocem.
- Svako HTTP zaglavlje se sastoji od imena i vrijednosti.
- HTTP protokol definiše standardni set HTTP zaglavlja i opisuje kako ih koristiti korektno.
- HTTP zaglavlje zahtjeva klijenta sadrži informacije koje server može upotrijebiti u odlučivanju kako da odgovori na zahtjev. To može biti da klijent čita zahtijevani dokument na francuskom ili njemačkom jeziku i da dokument treba biti poslat jedino ako je mijenjan od naznačenog datuma.

```
Accept-Language: fr, de  
If-Modified-Since: Fri, 10 Dec 2004 11:22:13 GMT
```

HTTP zahtjev – Tijelo poruke

- Može se nazvati i tijelom zahtjeva
- Aktuelni sadržaj poruke.
- Tijelo poruke može biti u originalnom obliku ili može biti kodirano.
- Može se nazvati i tijelom zahtjeva
- Prikladno je za neke metode zahtjeva, dok za druge nije.
- Na primjer, zahtjev sa POST metodom, koji šalje ulazne podatke serveru, ima tijelo poruke, koje sadrži te podatke.
- Zahtjev sa GET metodom, koji od servera traži da pošalje resurs, ne sadrži tijelo poruke.

HTTP odgovor

- HTTP odgovor generiše server i šalje klijentu.
- Cilj odgovora je da obezbijedi klijentu treženi resurs ili da ga informiše o izvršenju zahtjeva ili da dojavu da je došlo do greške.
- HTTP odgovor se sastoji iz:
 - Statusne linije;
 - Zaglavlja;
 - Tijela poruke, koje je obično neophodno.

Nakon svakog HTTP zaglavlja slijedi znak za povratak na početak reda (carriage return) i znak za prelazak u novi red (line feed) (CR-LF). Nakon poslednjeg zaglavlja dodatni CR-LF je dodat (za dobijanje prazne linije), nakon kojeg počinje tijelo poruke.

HTTP odgovor - Statusna linija

- Statusna linija je prva linija u odgovoru. Sasloji se iz tri segmenta:
 - Broj HTTP verzije, koji ukazuje na HTTP specifikaciju po kojoj je server pokušao da usladi odgovor.
 - Statusni kod, koji je trocifarski broj i ukazuje na rezultat izvršenja zahtjeva.
 - Fraza razloga, poznata i kao tekst statusa, koji je čitljiv čovjeku i sažima značenje statusnog koda.

Primjer statusne linije:

```
HTTP/1.1 200 OK
```

HTTP odgovor – Zaglavlja (Headers)

- Sadrži informacije koje klijent koristi da pronađe više podataka o odgovoru, kao i da pronađe podatke o serveru koji je poslao poruku.
- Ove informacije mogu pomoći klijentu u prezentaciji odgovora korisniku.
- Na primjer, prikazana zaglavlja govore klijentu kada je odgovor poslat, od strane kojeg servera je poslat, kao i da je to JPEG slika.

```
Date: Thu, 09 Dec 2004 12:07:48 GMT  
Server: IBM_CICS_Transaction_Server/3.1.0(zOS)  
Content-type: image/jpeg
```

HTTP odgovor – Tijelo poruke

- Naziva se i tijelom odgovora.
- Većina odgovora sadrže tijelo poruke. Izuzeci su kada server odgovara na zahtjev klijenta, koji je koristio HEAD metod (koji koristi zaglavlja ali ne i tijelo odgovora) i gdje server koristi određene statusne kodove.
- U odgovoru na uspješno izvršen zahtjev, tijelo poruke sadrži resurs koji je klijent zahtijevao ili neke informacije o statusu radnje koju je klijent zahtijevao.
- U odgovoru na neuspješno izvršen zahtjev, tijelo poruke može da pruži dodatne informacije o razlozima greške ili o nekoj radnji koju klijent treba da preduzma da bi se zahtjev uspješno izvršio.

HTTP zahtjev - metode

- HTTP definiše set metoda (načina) da indicira akciju koja će biti izvršena na datom resursu.
- Mada mogu biti i imenice, metode zahtjeva se često označavaju kao HTTP glagoli.
- Svaki metod koristi različitu semantiku.

HTTP zahtjev – vrste metoda

- **GET** Get metod se koristi za zahtijevanje podataka iz preciziranog resursa.
- **HEAD** HEAD metod zahtijeva zaglavlja resursa identificiranog datim URL-om, bez vraćanja samog resursa.
- **POST** POST metod se koristi za slanje podataka na obradu određenom resursu, često izazivajući promjene stanja ili druge efekte.
- **PUT** PUT se koristi za ažuriranje ili zamjenu postojećeg resursa na serveru sa obezbijeđenim podacima.
- **DELETE** DELETE metod briše specificirani resurs.
- **CONNECT** Uspostavlja vezu sa serverom, onosno, identificiranim ciljanim resursom.
- **OPTIONS** metod u HTTP protokolu koristi se za traženje informacija o komunikacionim opcijama dostupnim za određeni resurs ili server. Primarna uloga OPTIONS metoda je osigurati metapodatke o mogućnostima i konfiguraciji servera ili resursa.
- **TRACE** metod u HTTP protokolu koristi se u dijagnostičke svrhe i obično se ne koristi u uobičajenim scenarijima web aplikacija. Njegova primarna uloga je osigurati način na koji klijent može zatražiti da server vrati primljenu poruku zahtjeva, dopuštajući klijentu da vidi koje su promjene, ako ih je bilo, usput napravili posrednički serveri.
- **PATCH** PATCH metod obavlja parcijalnu modifikaciju resursa

HTTP zahtjev – GET metod

- GET se koristi za traženje podataka iz specificiranog izvora
- Treba imati na umu da se upitni string (par ime/vrijednost) šalje u URL-u GET zahtjeva.

`/test/demo_form.php?name1=value1&name2=value2`

- Nekoliko napomena u vezi GET zahtjeva:
 - GET zahtjevi se mogu keširati (spremiti u predmemoriju)
 - GET zahtjevi ostaju u historiji pregledača
 - GET zahtjevi se nikada ne bi trebali koristiti kada se radi o osjetljivim podacima
 - GET zahtjevi imaju ograničenje dužine
 - GET zahtjevi se koriste samo za traženje podataka (ne promjenu)

HTTP zahtjev – POST metod

- POST metod se koristi za slanje podataka serveru za kreiranje/ažuriranje resursa
- Podaci poslani serveru POST metodom smješteni su u tijelu HTTP zahtjeva.

```
POST /test/demo_form.php HTTP/1.1  
Host: w3schools.com
```

```
name1=value1&name2=value2
```

- Nekoliko napomena u vezi POST zahtjeva:
 - POST zahtevi se nikada ne kešuju
 - POST zahtevi ne ostaju u istoriji pregledača
 - POST zahtevi nemaju ograničenja u pogledu dužine podataka

HTTP zahtjev – GET vs. POST metod

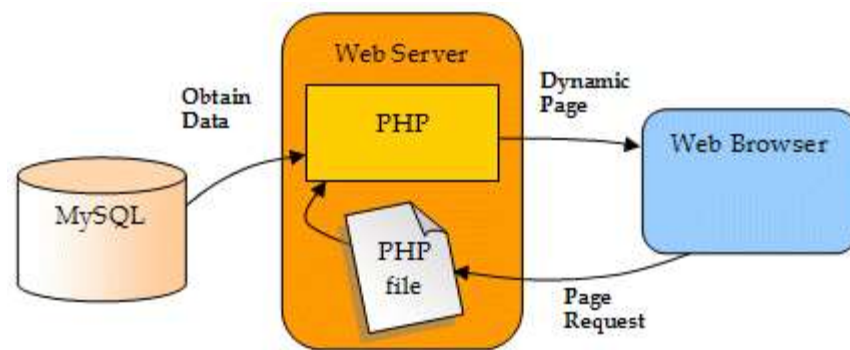
	GET	POST
BACK dugme/Ponovo učitaj	Neškodljivo	Podaci će biti ponovo poslani
Označenost	Može se označiti	Ne može se označiti
Keširanje	Može se keširati	Ne može se keširati
Tip kodiranja	application/x-www-form-urlencoded	application/x-www-form-urlencoded ili multipart/form-data. Upotreba višedjelnog kodiranja za binarne podatke
Istorija	Parametri ostaju u istoriji pregledača	Parametri ne ostaju u istoriji pregledača
Ograničenja u dužini podataka	Da, kod slanja podataka, GET method dodaje podatke na URL; dužina URL-a je ograničena (maximalna URL dužina je 2048 karaktera)	Bez ograničenja
Ograničenja u tipu podataka	Samo ASCII karakteri dozvoljeni	Bez restrikcija. Binarni podaci su takođe dozvoljeni
Bezbjednost	GET manje siguran u poređenju s POST, jer su podaci dio URL-a Ne koristiti GET kada se šalje ložinka ili druge osjetljive informacije	POST je malo sigurniji od GET jer parametri nijesu smješteni u istoriji pregledača ili u web server logu
Vidljivost	Podaci su vidljivi svima u URL-u	Podaci nijesu prikazani u URL-u

- Što je XAMPP?
- XAMPP – pednosti
- Instaliranje XAMPP-a
- XAMPP – Kontrolni panel
- XAMPP – direktorijumi
- Konfigurisanje XAMPP-a
- LAMP



ŠTO JE XAMPP?

- XAMPP je besplatna open-source platforma, koja sadrži:
 - Apache HTTP server,
 - MySQL bazu podataka,
 - PHP i
 - Perl programski jezik



Naziv **XAMPP** je skraćenica za:

- **X** (čita se „kros“ i znači kros-platforma)
- **A**pache HTTP server
- **M**ySQL
- **P**HP
- **P**erl



ZAŠTO KORISTITI XAMPP?

- Najpopularniji PHP razvojni paket
- Raspoloživ za Windows, Mac OS X i Linux
- Jednostavna instalacija i konfigurisanje
- Sasvim besplatan

INSTALIRANJE?

- Otići na: <https://www.apachefriends.org/download.html>
- Preuzeti i instalirati (može se zahtijevati pokretanja sa administratorskim pravima)
- Uobičajena windows instalacija **Next->Next-> ... -> Finish**

XAMPP Apache + MariaDB + PHP + Perl

Download

Click here for other versions



XAMPP for **Windows**

7.4.6 (PHP 7.4.6)



XAMPP for **Linux**

7.4.6 (PHP 7.4.6)



XAMPP for **OS X**

7.4.6 (PHP 7.4.6)

KONTROLNI PANEL

XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

XAMPP Control Panel v3.2.4

Modules

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	17516 8964	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	18256	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

Config Netstat Shell Explorer Services Help Quit

```
17:27:42 [main] Control Panel Ready
17:30:51 [mysql] Attempting to start MySQL app...
17:30:51 [mysql] Status change detected: running
17:31:05 [Apache] Attempting to stop Apache (PID: 17248)
17:31:05 [Apache] Attempting to stop Apache (PID: 9292)
17:31:05 [Apache] Status change detected: stopped
17:31:07 [Apache] Attempting to start Apache app...
17:31:07 [Apache] Status change detected: running
```

- ./htdocs – lokacija javnih html fajlova
- ./apache – lokacija konfiguracija
- ./mysql – lokacija MySQL baze podataka



- Apache konfiguracioni fajl (**httpd.conf**):
.\apache\conf\httpd.conf
- PHP konfiguracioni fajl (**php.ini**):
.\apache\bin\php.ini
- MySQL konfiguracioni fajl (**my.cnf**):
.\mysql\bin\mycnf

Više o instaliranju i konfigurisanju:

<https://pureinfotech.com/install-xampp-windows-10/>



LAMP



Linux



Apache



MySQL



Php

- **LAMP** je open-sorce Web razvojna platforma koja koristi **Linux** kao operativni sistem i **Apache** kao Web server.
- **MySQL** je upravljački sistem za relacionu bazu podataka
- **PHP** je objektno orijentisan skriptni jezik

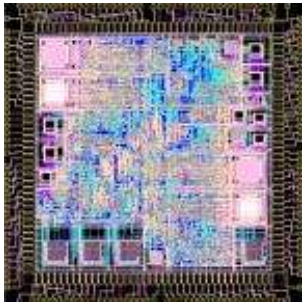
Baza podataka je skup međusobno povezanih podataka, pohranjenih bez nepotrebne redundantnosti, s ciljem da na optimalan način posluže u raznim primjenama.

- Podaci se kreiraju nezavisno od programa koji ih koriste. Zajedničkim pristupom dodaju se novi podaci, te mijenjaju i premještaju postojeći.
- Podaci se pohranjuju u bazu podataka na organizovan način, koristeći odgovarajući model podataka.

- Baza podataka je kolekcija podataka.
- Sistem za upravljanje bazom podataka (DBMS – Database Management System) je softver koji kontroliše te podatke.

**Aplikacija
dolazi ovdje**

DBMS



Sirovi podaci (goli metal)

DBMS interfejs
omogućuje da aplikacije
i sistem za upravljanje
podacima budu
izvedeni odvojeno

- Web aplikacija
- Olakšava upotrebu MySQL baze podataka (grafički interfejs)
- Pokreće se na:

<http://localhost/phpmyadmin>

- Preuzima se na:

http://www.phpmyadmin.net/home_page/downloads.php

Kako koristiti PHPMyAdmin (u više detalja):



https://www2.slideshare.net/karwanmst/mysql-database-with-phpmyadmin?from_action=save

Naziv baze: **idsys**


Tabele:

- **orgjedinice** – organizacione jedinice korisnika
- **korisnici** – korisnici čiji identifikatori ostvaruju pravo pristupa
- **terminali** – uređaji na kojima se očitavaju identifikatori
- **evidencije** – evidentiranje očitavanja identifikatora


MySQL – SQL – TABELA KORISNICI

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	IDKorisnika 	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	Prezime	varchar(25)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	3	Ime	varchar(25)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	4	TagID 	varchar(16)	utf8mb4_general_ci		No	None		
<input type="checkbox"/>	5	IDOrgJed	int(11)			No	None		
<input type="checkbox"/>	6	Aktivan	tinyint(1)			No	None		



MySQL – SQL – TABELA ORGJEDINICE

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	IDOrgJed 	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	NazivOrgJed	varchar(100)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/> 3	IDRod	int(11)			Yes	NULL		

MySQL – SQL – TABELA EVIDENCIJE

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	
<input type="checkbox"/>	1	ID 			int(11)	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	IDKorisnika			int(11)	No	None		
<input type="checkbox"/>	3	Vrijeme			datetime	No	None		
<input type="checkbox"/>	4	TerminalID			int(11)	No	None		
<input type="checkbox"/>	5	tagID			varchar(16) utf8mb4_general_ci	No	None		

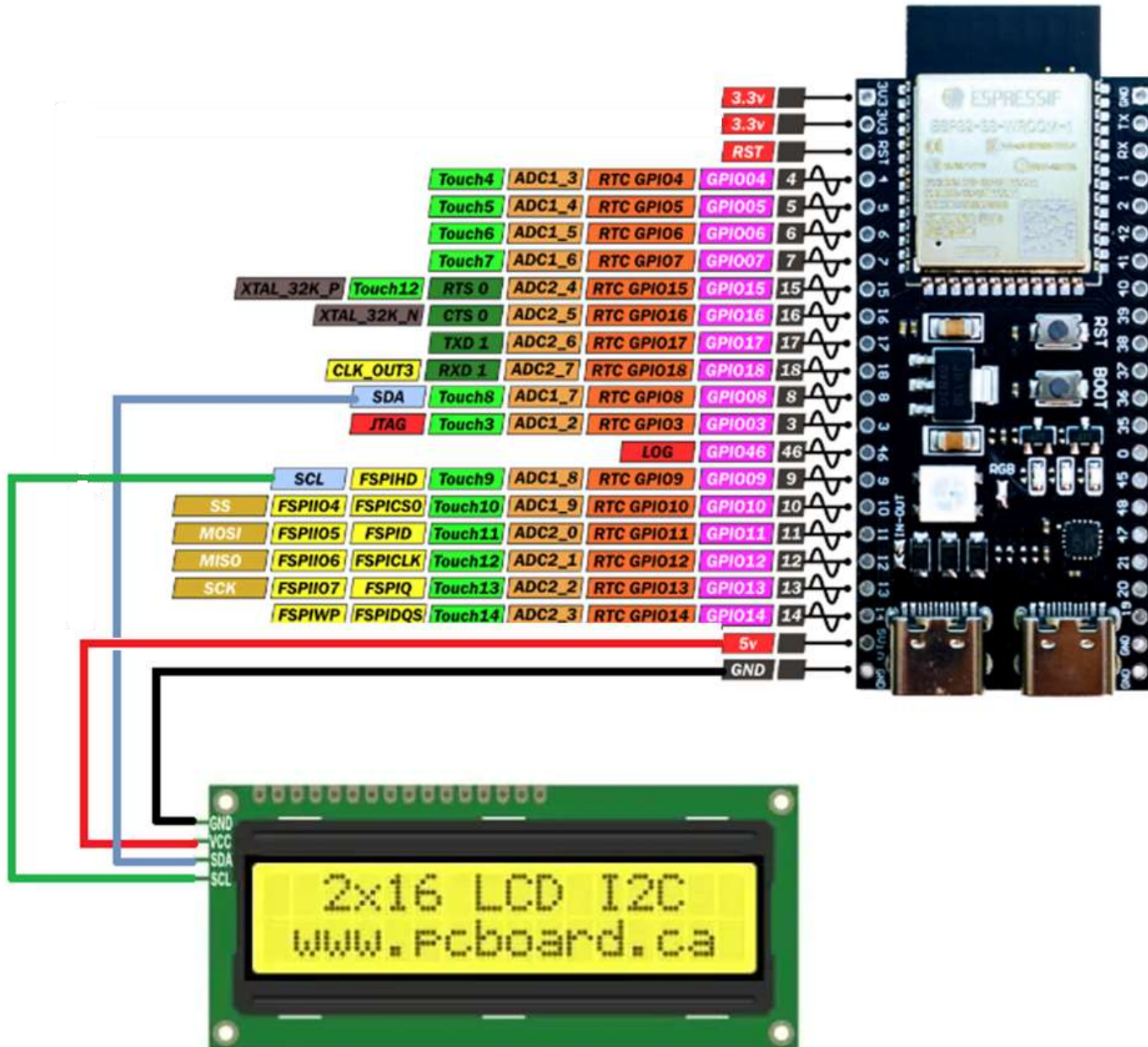
MySQL – SQL – TABELA TERMINALI

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	TerminalID 	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	Adresa 	int(11)			No	None		
<input type="checkbox"/>	3	Naziv	varchar(100)	utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	4	Tip	int(11)			Yes	NULL		



<https://www.w3schools.com/php/>

KAKO POVEZATI LCD



Doraditi arduino skeč i PHP fajl tako da se omogući sljedeće:

- Nakon očitavanja MASTER kartice uređaj ulazi u mod upisivanja i brisanja korisnika.
- Nakon primicanja nepoznatog tag-a treba napraviti da se upiše novi red u tabeli korisnici u kojem će se u koloni TagID upisati ID upravo očitano taga, u koloni TerminalID adresa uređaja, a u koloni Aktivan broj 1. U kolonama Ime, Prezime i IDOrgJed upisati vrijednost NULL. Kolona IDKorisnika je AUTO_INCREMENT i u njoj ne treba upisivati vrijednost kroz PHP program.
(2 -1 bod)
- Nakon primicanja poznatog tag-a treba napraviti da se obriše red iz tabele korisnici u kojem je u koloni TagID upisan ID upravo očitano taga.
(2-1 bod)
- Ponovnim primicanjem MASTER kartice vraća se u mod prepoznavanja.

- Kada je u modu prepoznavanja, treba da radi na sljedeći način:
- Nakon primicanja tag-a treba napraviti da se najprije u tabeli korisnici provjeri da li postoji red u kojem je u koloni TagID upisan ID upravo očitano taga.
 - Ukoliko postoji, uzeti IDKorisnika čiji je to TagID. U tabeli evidencije upisati novi red, u kojem će se u koloni TagID upisati ID upravo očitano taga, u koloni TerminalID adresa uređaja, u koloni IDKorisnika broj preuzet iz tabele korisnici, dok u koloni vrijeme treba upisati trenutni datum i vrijeme.
 - Ukoliko ne postoji, u tabeli evidencije upisati novi red, u kojem će se u koloni TagID upisati ID upravo očitano taga, u koloni TerminalID adresa uređaja, u koloni IDKorisnika broj -1, dok u koloni vrijeme treba upisati trenutni datum i vrijeme.

(2-1 bod)