

ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU



WEB TUTORIJAL ZA SQL JEZIK

–Master rad–

Kandidat:

Simon Tadić 2016/3219

Mentor:

vanr.prof. dr Zoran Čića

Beograd, April 2018.

SADRŽAJ

SADRŽAJ.....	2
1. UVOD	3
2. ALATI KORIŠĆENI PRI IZRADI APLIKACIJE.....	5
2.1. XAMPP (CROSS-PLATFORM APACHE MARIADB PHP PERL).....	5
2.2. HTML (HYPERTEXT MARKUP LANGUAGE).....	5
2.3. CSS (CASCADING STYLE SHEETS)	5
2.4. PHP (HYPERTEXT PREPROCESSOR).....	6
2.5. JAVASCRIPT	6
2.6. MySQL (MY STRUCTURED QUERY LANGUAGE)	6
3. UPUTSTVO ZA KORIŠĆENJE APLIKACIJE.....	7
3.1. UPUTSTVO ZA KORISNIKA.....	7
3.1.1. <i>Naslovna strana, logovanje i registracija</i>	7
3.1.2. <i>Navigacioni meni</i>	10
3.1.3. <i>Lekcija</i>	11
3.1.4. <i>Stranica za samostalno sastavljanje SQL upita</i>	12
3.1.5. <i>Kviz</i>	14
3.2. UPUTSTVO ZA ADMINISTRATORA	15
3.2.1. <i>Upravljanje kvizom</i>	15
3.2.2. <i>Upravljanje korisničkim podacima</i>	16
3.2.3. <i>„Moji podaci“</i>	17
4. OPIS KODA I STRUKTURA BAZE PODATAKA	18
4.1. STRUKTURA BAZE PODATAKA	18
4.2. OPIS KODA.....	22
4.2.1. <i>Header i Footer</i>	22
4.2.2. <i>Config.php</i>	23
4.2.3. <i>Stranica za samostalno sastavljanje upita</i>	24
4.2.4. <i>Kviz</i>	29
5. ZAKLJUČAK.....	34
LITERATURA.....	35

1. UVOD

Tutorijal predstavlja jedan od metoda prenošenja znanja i može se koristiti kao deo procesa učenja. Specifičniji je od knjige ili predavanja po tome što je zasnovan na primerima. Tutorijal je strukturno podeljen na više kratkih celina. Svaka celina predstavlja kratku lekciju i sastoji se iz kratkog opisa, primera ili instrukcija za rešavanje predstavljenog problema, kao i samog zadatka za rešavanje. Zadatak je uglavnom modifikacija primera i osmišljen je tako da se lako reši uz pomoć navedenih instrukcija. Tutorijal, kao niz kratkih lekcija, predstavlja efikasan metod za sticanje znanja iz željene oblasti.

Predmet ovog rada je tutorijal namenjen učenju SQL (*Structured Query Language*) jezika. SQL se prvobitno zvao SEQUEL (*Structured English Query Language*) i nastao je u istraživačkoj laboratoriji kompanije IBM u mestu San Hoze u Kaliforniji 1974. godine kao deo projekta „System R“. To je relacioni upitni jezik po ANSI (*American National Standards Institute*) i ISO (*International Standards Organization*) standardima. Osnovne karakteristike SQL-a su:

- Jednostavnost – relacije kreiraju jednom naredbom i odmah su dostupne za korišćenje.
- Uniformnost – svi podaci i rezultati operacija se prikazuju u vidu tabela.
- Neproceduralnost – korisnik određuje koje podatke treba uneti, prikazati ili ukloniti, a ne kako da se izvede operacija nad podacima.

Većina relationalnih baza podataka jedino preko SQL-a dozvoljava pristup podacima. Ovo je i jedan od razloga potrebe da većina korisnika baza podataka upozna komande SQL jezika. SQL omogućava korisnicima pristup podacima u sistemima za upravljanje relacionim bazama podataka. SQL je neproceduralni jezik tj. serveru se samo kaže šta da uradi, ali ne i kako to da uradi. Server baze podataka prevodi komande SQL-a u interne procedure koje obrađuju zahtev. Ovaj jezik je jednostavan zato što sakriva sve detalje obrade podataka. Korisnik dobija pristup podacima iz baze podataka izvršavanjem SQL upita. Upiti SQL jezika se mogu podeliti u četiri grupe. To su upiti za:

- Pretraživanje – upiti omogućavaju samo prikazivanje postojećih podataka u bazi. Ovi upiti uvek počinju ključnom rečju SELECT iza koje slede podaci, iskazi i slično. Ovi upiti ne menjaju podatke u bazi podataka.
- Manipulaciju – jezik za manipulaciju podacima DML (*Data Manipulation Language*). DML upiti se koriste za izmenu podataka u bazi.
- Definisanje – jezik za definisanje podataka DDL (*Data Definition Language*). DDL se koristi za kreiranje, menjanje i brisanje objekata iz baze podataka. Pomoću DDL-a možemo kreirati tabele, odrediti nazive i tipove podataka, definisati primarni ključ tabele i slično.
- Kontrolu – jezik za kontrolu podataka DCL (*Data Control Language*). DCL se koristi za dodelu ili oduzimanje privilegija za pristup bazi podataka.

Pojavom komercijalnih relationalnih sistema ubrzan je proces standardizacije relacionog upitnog jezika. Prvi deo procesa se završava 1986. godine kada se SQL usvaja kao standardni relacioni upitni jezik. Standard iz 1986. godine se naziva još i SQL-86. Zatim slede standardi SQL-

1989 i SQL-1992, dok se najvažnija standardizacija dogodila 1999. godine kada su bili uključeni koncepti objektne tehnologije.

U narednom, drugom poglavlju, biće predstavljeni softverski alati i programski jezici koji su korišćeni u samom radu. U trećem poglavlju će biti dat opis veb tutorijala za stanovišta korisnika, odnosno administratora. U četvrtom poglavlju biće predstavljeni i objašnjeni najznačajniji delovi koda same aplikacije, kao i opis strukture baze podataka. Na kraju rada će biti analizirana veb aplikacija, njena primena, kao i mogući pravci za njeno dalje unapređivanje.

2. ALATI KORIŠĆENI PRI IZRADI APLIKACIJE

Za potrebe kreiranja veb aplikacije korišćeni su alati *XAMPP* i tekstualni editor *Notepad++*. Od programskih jezika korišćeni su *HTML*, *CSS*, *PHP*, *Javascript*. Baza podataka koja je korišćena je *MySQL*.

2.1. XAMPP (Cross-Platform Apache MariaDB PHP Perl)

XAMPP je softverski paket koji omogućava simulaciju veb servera na računaru i namenjen je za sve operativne sisteme (Windows, Linux, Mac OS X, Solaris). Namenjen je za upotrebu u lokalnoj mreži. XAMPP je veb server koji je baš kao i svaki drugi server u svetskoj mreži, samo što je on postavljen lokalno na nekom računaru i nije mu moguće pristupiti od spolja, osim ako nije drugaćije konfigurisano. Svakako, krajnji domet odakle se može pristupiti lokalnom serveru jeste iz lokalne mreže na koju je povezan računar na kome je lokalni server instaliran. XAMPP u sebi sadrži Apache server, MySQL server, PHP i Perl i instalira ih sve zajedno sa konfigurisanim modulima i podešavanjima. Postavljanje lokalnog veb servera je osnovni preduslov za postavljanje sopstvenog razvojnog okruženja u kojem je moguće testirati i razvijati veb aplikacije.

2.2. HTML (HyperText Markup Language)

HTML je opisni jezik i specijalno je namenjen za opisivanje veb stranica. U HTML standard su ugrađeni elementi koji detaljnije opisuju sam dokument kao što su kratak opis dokumenta, ključne reči, podaci o autoru i slično. Pomoću njega se jednostavno odvajaju elementi kao što su naslovi, parografi i drugi. HTML stranice imaju ekstenziju .html ili .htm, a nalaze se u određenom direktorijumu servera koji je vezan na Internet, što ih čini dostupnim na vebu. Pomoću HTML jezika se generišu hipertekst dokumenti. Hipertekst je tekst koji sadrži veze i linkove ka drugim dokumentima, ili na samog sebe. To je skup stranica, međusobno povezanih linkovima koje su umetnute u stranice. Glavni zadatak HTML jezika jeste da uputi veb pregledač kako da prikaže hipertekst dokument. Pri tome se nastoji da taj dokument izgleda jednako bez obzira o kom je veb pregledaču, računaru i operativnom sistemu reč. Osnovni elementi svake HTML stranice su tagovi koji opisuju kako će se nešto prikazati u veb pregledaču. Veze unutar HTML dokumenta povezuju dokumente u uređenu hijerarhijsku strukturu i time određuju način na koji korisnik doživljava sadržaj stranica.

2.3. CSS (Cascading Style Sheets)

Iako je moguće i u samom HTML kodu definisati stil svakog elementa, praksa je da se stil elementa definiše kroz poseban CSS fajl. CSS je dizajniran kako bi se rešio problem sa primenom različitih stilova veb strana, načina na koji će neki elementi biti prikazani itd. Znatno olakšava i ubrzava proces izrade sajta jer se sva pravila i stilovi se nalaze na jednom mestu, pa se sama izmena ili dodavanje novih stilova može izvesti na brz i jednostavan način. Uobičajen način korišćenja ovog alata je takav da se u posebnoj datoteci sa .css ekstenzijom nalazi dokument koji sadrži skup svih definisanih pravila i stilova. Ovaj dokument se u HTML kodu obično priključuje u okviru taga <head>.

2.4. PHP (Hypertext Preprocessor)

PHP je specijalizovan skriptni jezik koji je prvenstveno namenjen za izradu dinamičkog veb sadržaja i realizuje se na strani servera. Može da memoriše podatke i da ih skladišti, pa da ih koristi kasnije u nekom trenutku. Svoju popularnost je stekao zahvaljujući svojoj jednostavnosti, u većoj ili manjoj meri već poznatoj sintaksi baziranoj na sintaksi C jezika, kao i činjenici da je besplatan. Tokom vremena jezik se proširivao i sticao mogućnosti za objektno orijentisano programiranje, naročito od verzije 5.0. Sličan je jeziku C++ u smislu što dozvoljava i čisto proceduralno programiranje, ali istovremeno omogućava i korišćenje klasa i drugih koncepta objektno orijentisanog programiranja. Iako se PHP može koristiti za programiranje konzolnih aplikacija i grafičkih interfejsa, njegova osnovna i glavna upotreba je u programiranju dinamičkih stranica na Internetu.

2.5. JavaScript

JavaScript je skriptni programski jezik koji se prvenstveno koristi za definisanje funkcionalnosti veb stranica na klijentskoj strani. Takođe se može koristiti i kao skriptni jezik za pristup objektima i drugim aplikacijama. On omogućava dinamički HTML sadržaj. Može da menja vrednosti HTML elemenata i atributa. Koristi se za proveru ispravnosti popunjениh formulara, reaguje na događaje tipa klika na neki element, slanja formulara i slično.

U radu je takođe korišćen jQuery, jedna od najpoznatijih biblioteka JavaScript-a. Ova biblioteka je znatno olakšala upotrebu JavaScript-a na veb stranicama. jQuery obuhvata mnoge zadatke koji zahtevaju veliki broj linija JavaScript koda i omotava ih u metode koje se pozivaju samo jednom linijom. Neke od takvih funkcionalnosti su HTML/DOM manipulacija, CSS manipulacija, AJAX (Asynchronous JavaScript And XML), event metodi, itd...

2.6. MySQL (My Structured Query Language)

MySQL je najkorišćenija besplatna i open sors baza podataka. Zasnovana je na SQL jeziku namenjenom za pristupanje i manipulisanje bazama podataka. Za manipulisanje MySQL bazama podataka postoje definisane funkcije u mnogim programskim jezicima. U okviru ovog rada od značaja će biti funkcije definisane u programskom jeziku PHP.

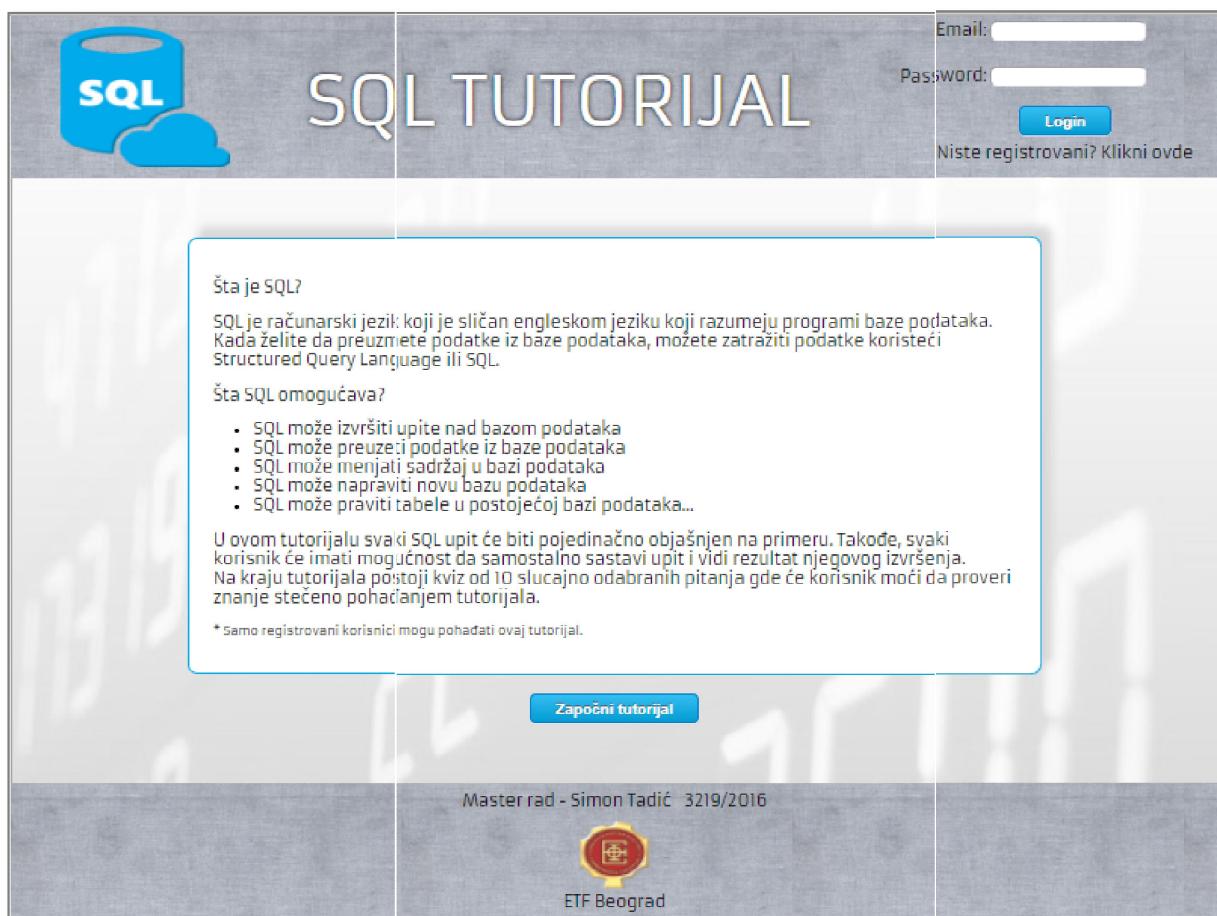
3. UPUTSTVO ZA KORIŠĆENJE APLIKACIJE

Mogućnost i funkcionalnost aplikacije potrebno je sagledati iz ugla dve vrste korisnika. Aplikacija je prvenstveno namenjena korisnicima tutorijala, ali takođe pruža i određene mogućnosti administratoru. U skladu sa tim u ovom delu biće predstavljena dva uputstva za korišćenje aplikacije.

3.1. Uputstvo za korisnika

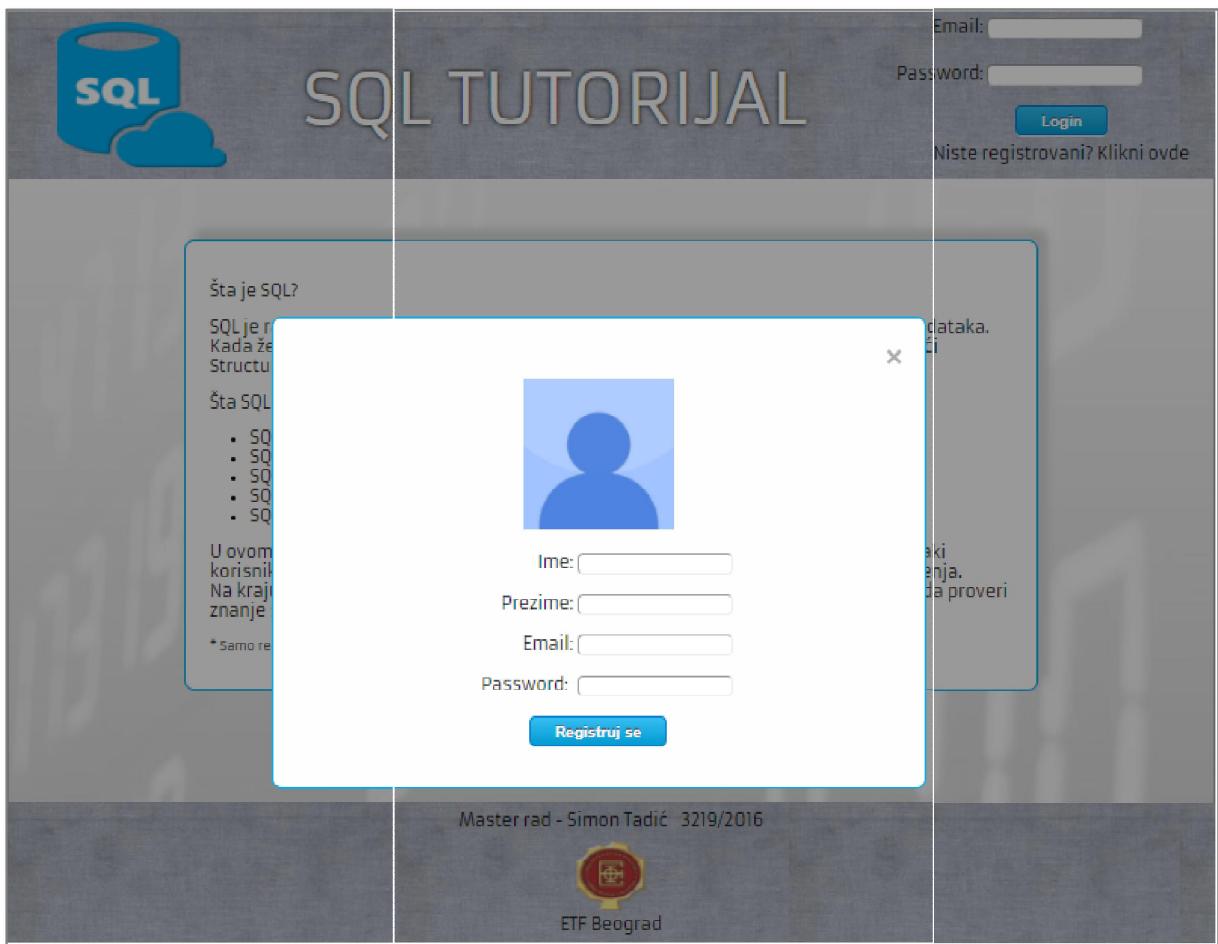
3.1.1. Naslovna strana, logovanje i registracija

Do naslovne strane dolazi se unošenjem adrese u internet pretraživač, u ovom slučaju to je: http://localhost/sql_tutorial/pages/index.php. Naslovna strana je ista za oba tipa korisnika i na njoj su date osnovne informacije o SQL jeziku, kao i kratak pregled samog tutorijala (slika 3.1.1.1).



Slika 3.1.1.1. Naslovna strana sajta.

Prvi korak za svakog korisnika je registracija. Na slici 3.1.1.2 prikazan je modalni prozor na kome se nalazi formular za registraciju korisnika.



Slika 3.1.1.2. Formular za registraciju korisnika.

Nakon završene registracije, potrebno je da se korisnik uloguje sa svojom e-mail adresom i kreiranom lozinkom prilikom registracije. Samo registrovani, a zatim i ulogovani korisnici mogu pohađati ovaj tutorijal. Logovanje se vrši u formi u gornjem desnom uglu. Nakon ispravno unetih podataka, umesto forme za logovanje, prikazuje se silueta korisnika sa njegovim imenom i prezimenom (slika 3.1.1.3).

The screenshot shows a web application interface for a SQL tutorial. At the top left is a logo featuring a blue cylinder with the letters 'SQL' and a blue cloud-like shape below it. To the right of the logo, the text 'SQL TUTORIJAL' is displayed in large, stylized letters. In the top right corner, there is a user profile icon with the name 'Simon Tadić' and a 'Logout' button. A modal window is open in the center of the page. The title of the modal is 'Šta je SQL?'. The content area contains text explaining what SQL is and what it can do. Below this, there is a bulleted list of SQL capabilities. At the bottom of the modal, there is a note about a quiz and a small note about registered users. A blue button labeled 'Započni tutorijal' is located at the bottom of the modal. The footer of the page includes the text 'Master rad - Simon Tadić 3219/2016' and the logo of the Faculty of Technology and Metallurgy (ETF) Belgrade.

Šta je SQL?

SQL je računarski jezik koji je sličan engleskom jeziku koji razumeju programi baze podataka. Kada želite da preuzmete podatke iz baze podataka, možete zatražiti podatke koristeći Structured Query Language ili SQL.

Šta SQL omogućava?

- SQL može izvršiti upite nad bazom podataka
- SQL može preuzeti podatke iz baze podataka
- SQL može menjati sadržaj u bazi podataka
- SQL može napraviti novu bazu podataka
- SQL može praviti tabele u postojećoj bazi podataka...

U ovom tutorijalu svaki SQL upit će biti pojedinačno objašnjen na primeru. Takođe, svaki korisnik će imati mogućnost da samostalno sastavi upit i vidi rezultat njegovog izvršenja. Na kraju tutotrijala postoji kviz od 10 slučajno odabralih pitanja gde će korisnik moći da proveri znanje steceno pohađanjem tutorijala.

* Samo registrovani korisnici mogu pohađati ovaj tutorijal.

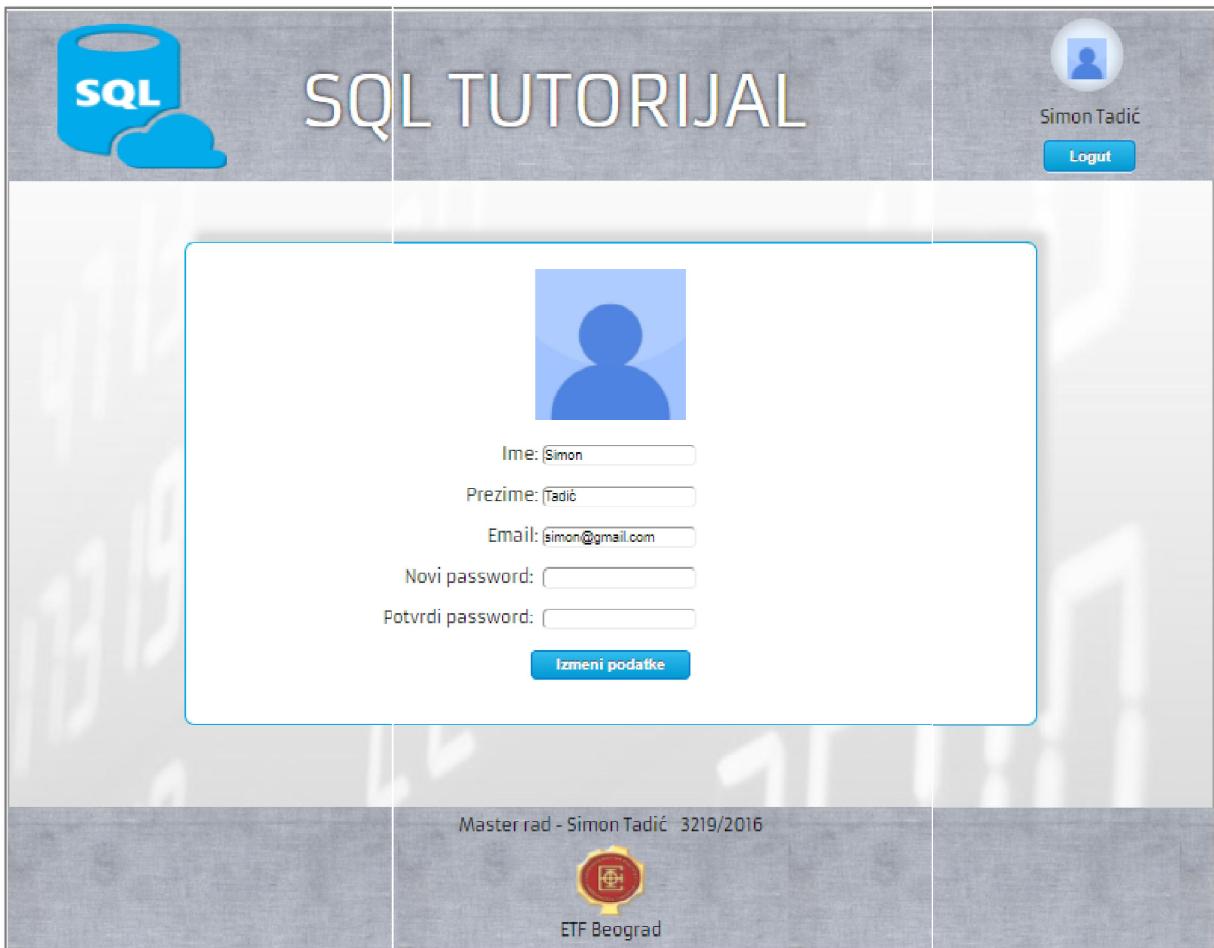
Započni tutorijal

Master rad - Simon Tadić 3219/2016

ETF Beograd

Slika 3.1.1.3. Ulogovan korisnik.

Klikom na siluetu korisnika ili na ime i prezime otvara se modalni prozor sa podacima o korisniku. Korisnik ima mogućnost izmene ličnih podataka kao i mogućnost izmene lozinke (slika 3.1.1.4). Zaključno sa ovim delom završen je administrativni deo korisničkog uputstva.



Slika 3.1.1.4. Izmena ličnih podataka i lozinke.

3.1.2. Navigacioni meni

Klikom na dugme „Započni tutorijal“ otvara se sadržaj samog tutorijala. Sa leve strane nalazi se vertikalni navigacioni meni koji je prisutan na svim stranicama. Navigacioni meni je podeljen u četiri celine: SQL Tutorial, SQL Database, SQL Reference i SQL Kviz.

i) SQL Tutorial

Ovaj deo predstavlja srž samog tutorijala. U njemu se nalaze linkovi ka najvažnijim lekcijama o osnovama SQL jezika. Svaka lekcija posvećena je jednom SQL upitu za pretraživanje ili manipulaciju podataka u bazi.

ii) SQL Database

Nakon stečenih znanja o osnovama SQL-a, ovaj deo je posvećen upitima za definisanje tipova podataka, kreiranje/brisanje baze podataka, definisanje primarnog ključa tabele, itd...

iii) SQL Reference

Ova celina je posvećena osnovnim funkcijama, operatorima i tipovima podataka definisanim za MySQL.

iv) *SQL Quiz*

Kviz se nalazi na kraju svih lekcija i biće opisan u posebnom poglavlju.

3.1.3. Lekcija

Klikom na neku od lekcija iz navigacionog menija otvara se nova stranica sa željenom lekcijom. Svaka lekcija se sastoji od objašnjenja, namene datog SQL upita i priložene tabele (ili više njih u zavisnosti od upita koji se objašnjava) nad kojom se upit izvršava (slika 3.1.3.1).

The screenshot shows a web page titled "SQL TUTORIJAL". On the left, there's a sidebar with a "SQL" logo and a list of topics under "SQLTutorial" such as SQL Syntax, SQL Select, SQL Select Distinct, SQL Where, SQL And,Or,Not, SQL Order By, SQL Insert Into, SQL Null Vrednosti, SQL Update, SQL Delete, SQL Limit, SQL Min & Max, SQL Count, Avg, Sum, SQL Like, SQL In, SQL Between, SQL Aliases, SQL Joins, SQL Inner Join, SQL Left Join, SQL Right Join, SQL Full Join, SQL Self Join, SQL Union, SQL Group By, SQL Having, SQL Exists, SQL Any,All, SQL Select Into, SQL Insert Into,Select, SQL Null Functions, SQL Comments, and a section for "SQL Database" with topics like SQL Create DB and SQL Drop DB. The main content area has a heading "SQL SELECT UPIT" with a brief explanation: "Select upit se koristi za odabir informacija iz određene tabele u nekoj bazi podataka. Select naredba definije listu kolona(jednu ili više kolona iz tabele) koje će biti uključene u rezultujuću tabelu." Below this is a "SINTAKSA UPITA" section with two examples of SQL code. The first example shows selecting specific columns from a table, and the second shows selecting all columns from a table. At the bottom, there's a table titled "IZGLED TABELE: "Studenti" U BAZI PODATAKA" containing student data:

IdStudent	Ime	Prezime	Indeks	GodUpisa	Fakultet	Smer	Prosek
1	Petar	Petrović	111	2013	ETF	Telekomunikacije	8.07
2	Milan	Marković	181	2013	PMF	Informatika	8.87
3	Filip	Jovanović	334	2013	ETF	Signalni i sistemi	8.22
4	Danilo	Nikolić	569	2015	ETF	Signalni i sistemi	9.15
5	Marija	Bogdanović	87	2015	FON	Isit	7.93
6	Sofija	Milić	418	2012	PMF	Informatika	9.33
7	Bojana	Lazić	147	2012	ETF	Rti	9.56
8	Sava	Milenković	333	2012	FON	Isit	7.46

Slika 3.1.3.1. Primer lekcije1.

Pored osnovnih informacija o datom upitu i priložene sintakse za isti, u svakoj lekciji postoji deo gde je napisan primer SQL upita nad priloženom tabelom iz baze podataka. Takođe, radi lakšeg učenja, dat je i rezultat izvršenja pomenutog upita (slika 3.1.3.2).

[SQL Primary Key](#)
[SQL Foreign Key](#)
[SQL Check](#)
[SQL Default](#)
[SQL Index](#)
[SQL Auto Increment](#)
[SQL Dates](#)
[SQL Views](#)
[SQL Injection](#)

SQL Reference

[MySQL Functions](#)
[MySQL Operators](#)
[MySQL Data Types](#)

SQL Quiz

[SQL Quiz](#)

SELECT UPIT PRIMER

U sledećem primeru SELECT upitom odabране su kolone Ime, Prezime i Fakultet iz tabele "Studenti".

```
SELECT Ime, Prezime, Fakultet
FROM Studenti;
```

Rezultat izvršenog upita je:

Ime	Prezime	Fakultet
Petar	Petrović	ETF
Milan	Marković	PMF
Filip	Jovanović	ETF
Danilo	Nikolić	ETF
Marija	Bogdanović	FON
Solija	Milić	PMF
Bojana	Lazić	ETF
Sava	Milenković	FON
Miloš	Filipović	ETF
Anđelija	Pantelić	ETF

Sastavite upit samostalno -->

[< Nazad](#) [Napred >](#)

Master rad - Simon Tadić - 3219/2016

 ETF Beograd

Slika 3.1.3.2. Primer lekcije2.

3.1.4. Stranica za samostalno sastavljanje SQL upita

Nakon pročitane lekcije za željeni upit i navedenog primera za isti, na dnu stranice svake od lekcija se nalazi link „Sastavite upit samostalno“ koji vodi na pomenutu stranicu. Ova stranica je najvažnija za korisnika jer pruža mogućnost izvršenja samostalno sastavljenog upita, kao i prikaz rezultata izvršenog upita. Odlaskom na stranicu „Sastavite upit samostalno“ sa neke od lekcija, na pomenutu stranicu učitava se primer vezan baš za lekciju sa koje je korisnik došao na ovu stranicu. Na primer, ukoliko se čita lekcija o selekt upitu, odlaskom na stranicu „Sastavite upit samostalno“ biće učitan primer baš za ovaj upit. Takođe, na vrhu stranice nalazi se tabela (ili više njih, zavisno od primera) iz baze podataka čiji se sadržaj menja upitom iz datog primera (slika 3.1.3.1).



SQL TUTORIJAL

Simon Tadić

[Logout](#)

IZGLED TABELE "Studenti" U BAZI PODATAKA								
IdStudent	Ime	Prezime	Indeks	GodUpisa	Fakultet	Smer	Prosek	
1	Petar	Petrović	111	2013	ETF	Telekomunikacije	8.07	
2	Milan	Marković	181	2013	PMF	Informatika	8.87	
3	Filip	Jovanović	334	2013	ETF	Signalni sistemi	8.22	
4	Danilo	Nikolić	569	2015	ETF	Signalni sistemi	9.15	
5	Marija	Bogdanović	87	2015	FON	Isit	7.93	
6	Sofija	Milić	418	2012	PMF	Informatika	9.33	
7	Bojana	Lazić	147	2012	ETF	Rti	9.56	
8	Sava	Milenković	333	2012	FON	Isit	7.46	
9	Miloš	Filipović	9	2011	ETF	Energetika	9.72	
10	Andrija	Pantelić	545	2012	ETF	Elektronika	8.46	

Slika 3.1.3.1. Tabela na stranici „Sastavite upit samostalno“.

U središnjem delu stranice nalazi se polje u koje se unosi željeni upit. Kao što je već rečeno, na početku je tu učitan upit vezan za primer iz lekcije sa koje se na ovu stranicu dolazi. Korisnik ima mogućnost izmene učitanog upita ili samostalno sastavljanje upita. Klikom na dugme „Izvrši upit“, u polju ispod prikazuje se rezultat izvršenog upita. Takođe, ako upit nije dobro sastavljen ili postoji bilo kakva greška u sintaksi ili nepostojanje datih kolona i tabela, u polju za rezultat biće prikazana greška kako bi korisnik imao uvid šta greši i zbog čega upit nije uspešno izvršen. Pored dugmeta „Izvrši upit“ nalazi se dugme „Resetuj tabelu“ sa nazivom odgovarajuće tabele koja se resetuje. Ovo dugme je namenjeno je za situacije kada korisnik izvršava upite koji menjaju sadržaj tabela u bazi podataka. Korisnik u svakom trenutku u gornjem delu stranice ima pregled tabele(a) u bazi podataka, ali ukoliko poželi da se vrati u početno stanje, to je omogućeno klikom na pomenuto dugme za resetovanje tabele. Sve opisano u prethodnom delu, prikazano je na slici 3.1.3.2.

U ovom delu možete samostalno da sastavite i izvršite upit.
 Klikom na dugme Izvrši upit biće prikazan rezultat upita u bloku ispod.
 Odmah nakon izvršenog upita promene možete videti u tabelama, u bloku iznad.
 Takođe, u svakom trenutku, klikom na dugme Resetuj tabelu Studenti/Ispiti možete vratiti početne vrednosti u tabele nad kojim su upiti izvršavani.

```
SELECT Studenti.Ime, Studenti.Prezime, Studenti.Indeks, Studenti.GodUpisa, Ispiti.ImeIspita, Ispiti.Ocena
FROM Studenti LEFT JOIN Ispiti ON Ispiti.IdStudent=Studenti.IdStudent;
```

[Izvrši upit](#) [Resetuj tabelu Studenti](#) [Resetuj tabelu Ispiti](#)

Ime	Prezime	Indeks	GodUpisa	ImeIspita	Ocena
Petar	Petrović	111	2013	OET1	6
Sofija	Milić	418	2012	OET2	9
Sava	Milenković	333	2012	Matematički	10
Marija	Bogdanović	87	2015	Matematika2	8
Milan	Marković	181	2013	Programiranje1	9
Filip	Jovanović	334	2013	Programiranje2	6
Miloš	Filipović	9	2011	Sociologija	10
Marija	Bogdanović	87	2015	Fizika	8
Daniilu	Nikolić	569	2015		
Bojana	Lazić	147	2012		
Andrija	Pantelić	545	2012		

Slika 3.1.3.2. Polje za unos upita i prikaz rezultata.

3.1.5. Kviz

Na kraju samog tutorijala, nakon svih lekcija nalazi se kviz. Kviz se sastoji od 10 slučajno odabralih pitanja. Na svakom pitanju postoje ponuđeni odgovori od kojih je samo jedan tačan. Nakon odabira željenog odgovora, korisnik može da pređe na sledeće pitanje, pri čemu mu se prikazuje da li je na prethodno pitanje tačno odgovorio. Da bi korisnik mogao da pređe na sledeće pitanje, mora izabrati neki od ponuđenih odgovora na trenutno učitano pitanje. Kviz se smatra uspešno urađenim ukoliko je na minimalno 7 od 10 pitanja odgovoren tačno. Korisnik može neograničen broj puta da radi kviz i time proveri svoje znanje stečeno na tutorijalu. Primer jednog pitanja u kvizu prikazan je na slici 3.1.5.1.

[SQLTutorial](#)

- [SQL Syntax](#)
- [SQL Select](#)
- [SQL Select Distinct](#)
- [SQL Where](#)
- [SQL And,Or,Not](#)
- [SQL Order By](#)
- [SQL Insert Into](#)
- [SQL Null Vrednosti](#)
- [SQL Update](#)
- [SQL Delete](#)
- [SQL Limit](#)
- [SQL Min & Max](#)
- [SQL Count, Avg, Sum](#)
- [SQL Like](#)

Kojom naredbom se iz tabele "Studenti" briše student čije je ime "Petar" ?

- DELETE ROW Ime='Petar' FROM Studenti
- DELETE Ime='Petar' FROM Studenti
- DELETE FROM Studenti WHERE Ime='Petar'
- DELETE FROM Studenti Ime='Petar'

[Sledeće pitanje](#)

Slika 3.1.5.1. Primer pitanja u kviz.

3.2. Uputstvo za administratora

Administrator poseduje sve mogućnosti kao i običan korisnik, ali uz dodatne opcije vezane za upravljanje i održavanje same aplikacije. Kada je administrator ulogovan, za razliku od običnog korisnika, on poseduje dugme „Manage“ sa padajućom listom. Opcije u padajućoj listi su „Kviz“, „Korisnici“ i „Moji podaci“.

3.2.1. Upravljanje kvizom

Izborom opcije kviz iz padajuće liste učitava se stranica na kojoj postoje dva dugmeta: „Dodaj pitanje“ i „Izmena pitanja“.

Klikom na dugme za dodavanje pitanja učitava se forma za dodavanje pitanja (slika 3.2.1.1). Postoji niz definisanih provera za unos novog pitanja tako da administrator ne mora da brine o ispravno definisanom pitanju za kviz. Neke od provera su, na primer: mora biti ponuđeno najmanje dva odgovora, kao tačan odgovor mora biti naveden neki od ponuđenih, itd.

The screenshot shows the 'SQL TUTORIJAL' application interface. At the top right, there is a user profile icon labeled 'admin admin' with 'Logout' and 'Manage' buttons. A dropdown menu is open over the 'Manage' button, showing options: 'Kviz' (selected), 'Korisnici', and 'Moji podaci'. Below the header, a large central box contains a form for adding a question. The form includes fields for 'Tekst pitanja' (Question text), 'Ponuđeni odgovori:' (Offered answers), 'Tačan odgovor:' (Correct answer), and a 'Dodaj pitanje' (Add question) button at the bottom. The 'Ponuđeni odgovori:' section contains four input fields, each labeled 'Ponuđeni odgovor1', 'Ponuđeni odgovor2', 'Ponuđeni odgovor3', and 'Ponuđeni odgovor4' respectively.

Slika 3.2.1.1. Forma za dodavanje pitanja.

Ukoliko administrator primeti neku pravopisnu grešku, želi da promeni ponuđene odgovore ili iz bilo kog razloga poželi da menja sadržaj pitanja, ponuđenih odgovora i tačnog odgovora, to je omogućeno klikom na dugme „Izmena pitanja“. Takođe je omogućeno brisanje pitanja iz baze

podataka. Klikom na pomenuto dugme za izmenu pitanja učitava se stranica sa listom svih pitanja koje se nalaze u bazi podataka. Pitanja su sortirana po redosledu unošenja u bazu i raspoređena radi preglednosti na više stranica (na svakoj stranici nalazi se 5 pitanja). Pomenute mogućnosti za izmenu pitanja prikazane su na slici 3.2.1.2.

The screenshot shows a web application interface titled "SQL TUTORIJAL". In the top right corner, there is a user profile icon labeled "admin admin" with "Logout" and "Manage" buttons. The main content area displays five questions listed vertically, each with "Izmeni" (Edit) and "Obriši" (Delete) buttons:

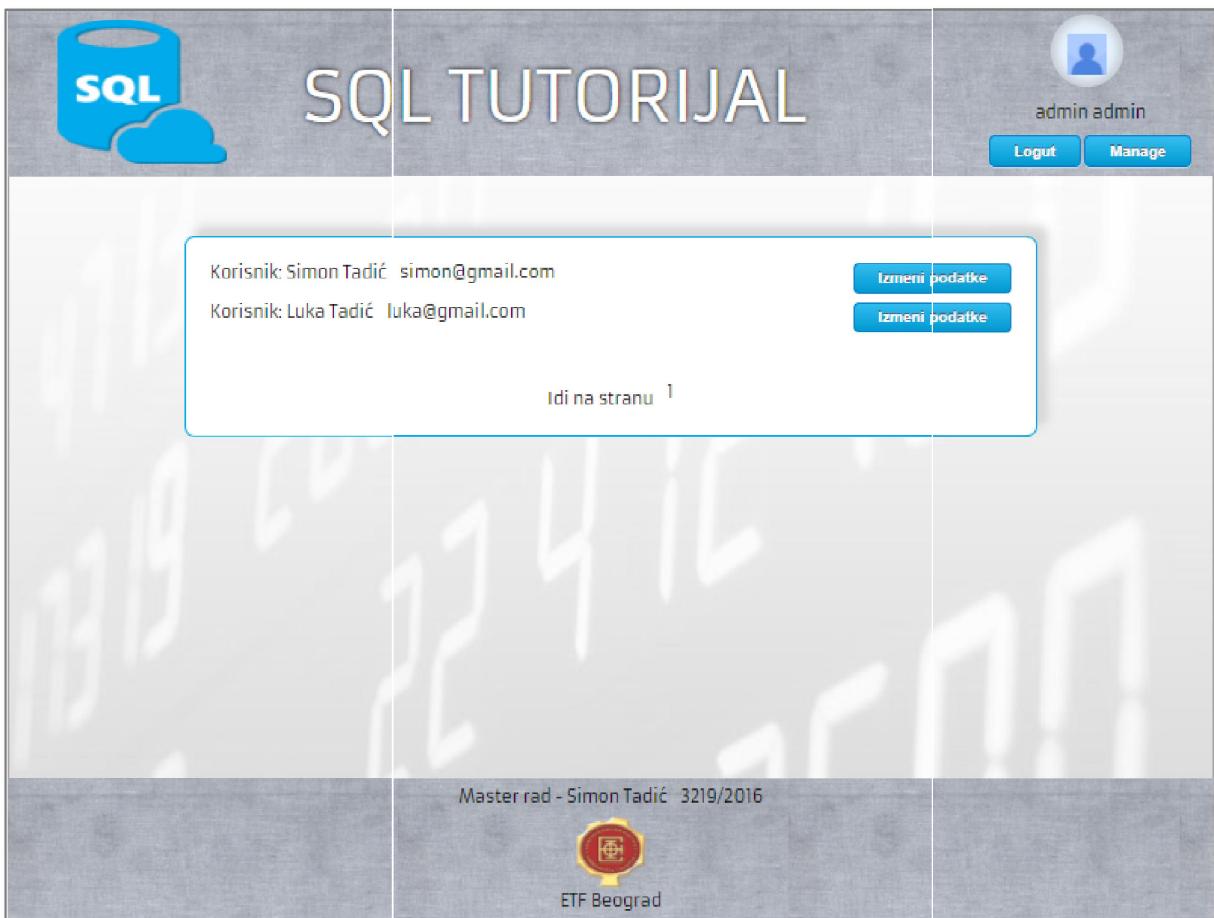
- 1: Kojom naredbom se izvlače podaci iz baze podataka ?
- 2: Koja SQL ključna reč se koristi za sortiranje skupa rezultata?
- 3: Koja naredba vraća sve zapise iz tabele "Studenti" sortirane po indeksu u opadajućem redosledu?
- 4: Kojom naredbom se iz tabele "Studenti" briše student čije je ime "Petar" ?
- 5: Kojom od sledećih SQL naredbi se vraća broj rekorda iz tabele "Studenti" ?

At the bottom of the list, there is a link "Idi na stranu 1 2 3 4 5 6".

Slika 3.2.1.2. Izmena pitanja.

3.2.2. Upravljanje korisničkim podacima

Klikom na dugme „Manage“ i odabirom stavke „Korisnici“ učitava se stranica na kojoj se nalazi spisak svih registrovanih korisnika (slika 3.2.2.1). Pored svakog korisnika nalazi se dugme „Izmeni podatke“ koje vodi na stranicu za izmenu korisničkih podataka. Forma za izmenu korisničkih podataka je identična onoj koja se učitava i samom korisniku kada pregleda svoje podatke. Na ovaj način administrator može da menja podatke za sve korisnike koji su se registrovali na tutorijalu.



Slika 3.2.2.1. Izmena korisničkih podataka.

3.2.3. „Moji podaci“

Klikom na dugme „Manage“ i odabirom stavke „Moji podaci“ učitava se stranica na kojoj se nalaze administratorski podaci. Administrator takođe može videti svoje podatke kao i običan korisnik klikom na svoje ime ili na siluetu korisnika. Opcija „Moji podaci“ je napravljena radi bolje preglednosti administratorskih mogućnosti, ali i zbog boljeg korisničkog iskustva. Naime, kada neki korisnik (u ovom slučaju i administrator) izvrši izmenu podataka aplikacija ga automatski izloguje kako bi se ulogovao sa novim izmenjenim podacima. U slučaju da administrator menja svoje podatke preko opcije „Moji podaci“, nakon promene podataka neće biti izlogovan, već će moći da nastavi sa daljim upravljanjem aplikacije u skladu sa svojim administratorskim pravima.

4. OPIS KODA I STRUKTURA BAZE PODATAKA

U ovom delu biće opisani najvažniji delovi koda aplikacije. To su delovi koji se odnose na izvršavanje upita nad bazom sastavljenih od strane korisnika i deo koda u kom je realizovan kviz znanja koji se nalazi na kraju tutorijala. Takođe, biće dato detaljno objašnjenje vezano za strukturu baze podataka.

4.1. Struktura baze podataka

Koristi se MySQL baza podataka, a za rad sa bazom podataka korišćen je phpMyAdmin. Za ovu veb aplikaciju korišćena je baza podataka po nazivom „sqlTutorial“ (slika 4.1.1). Baza se sastoji od 5 tabela: „studenti“, „studenti_si“, „ispiti“, „kviz“ i „user“.

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** 127.0.0.1
- Database:** sqlTutorial
- Structure:** The selected tab.
- Tables:** A list of five tables: ispiti, kviz, studenti, studenti_si, and user.
- Table Data:** A detailed view of each table's structure, including columns, types, and constraints.
- Operations:** Buttons for actions like Browse, Structure, Search, Insert, Empty, Drop, etc.
- Statistics:** Rows, Type, Collation, Size, Overhead information.
- Total:** 5 tables, Sum.

Slika 4.1.1. Tabele u bazi podataka.

Tabela „studenti“ se koristi u lekcijama i služi za čuvanje podataka o studentima koji su upisani na neki od fakulteta. Njena struktura prikazana je na slici 4.1.2.

The screenshot shows the structure of the studenti table in the sqlTutorial database:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	IdStudent	int(11)	utf8_unicode_ci		No	None		AUTO_INCREMENT	Change Drop Primary Unique Index ▾ More
2	Ime	varchar(1024)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index ▾ More
3	Prezime	varchar(1024)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index ▾ More
4	Indeks	int(11)			No	None			Change Drop Primary Unique Index ▾ More
5	GodUpisa	int(11)			No	None			Change Drop Primary Unique Index ▾ More
6	Fakultet	varchar(1024)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index ▾ More
7	Smer	varchar(1024)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index ▾ More
8	Prosek	float			No	None			Change Drop Primary Unique Index ▾ More

Slika 4.1.2. Tabela „studenti“ u bazi podataka.

Kolona „IdStudent“ predstavlja identifikacioni broj studenta, tipa je **int**, limitirana je na maksimalnih 11 cifara i uključena je opcija AUTO_INCREMENT koja omogućava automatsko uvećanje vrednosti za 1 pri svakom novom unosu u tabelu.

Kolona „Ime“ predstavlja ime studenta, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „Prezime“ predstavlja prezime studenta, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „Indeks“ predstavlja redni broj indeksa studenta, tipa je **int** i limitirana je na maksimalnih 11 cifara.

Kolona „GodUpisa“ predstavlja godinu upisa studenta na fakultet, tipa je **int** i limitirana je na maksimalnih 11 cifara.

Kolona „Fakultet“ predstavlja ime fakulteta na koji je student upisan, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „Smer“ predstavlja smer na fakultetu koji student pohađa, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „Prosek“ predstavlja prosečnu ocenu studenta, tipa je **float**.

Tabela „studenti_si“ služi za čuvanje podataka o studentima koji su upisani na smer „SI“ fakulteta „ETF“. Njena struktura je ista kao i struktura tabele „studenti“ (slika 4.1.2.). Dakle, tabela je identične strukture kao i tabela „studenti“, samo kolona „fakultet“ ima fiksiranu vrednost „ETF“. Ova tabela napravljena je za potrebe lekcija u tutorijalu vezanih za upite gde dve ili više tabela nad kojima se upiti izvršavaju moraju imati istu strukturu i tipove podataka (npr. SQL naredba „Union“).

Tabela „ispiti“ se služi za čuvanje podataka o položenim ispitima. Jednim unosom u ovu tabelu čuvaju se informacije o položenom ispitu, kao i identifikacioni broj studenta koji je dat ispit položio. Struktura tabele prikazana je na slici 4.1.3.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	IdIspit	int(11)			No	None	AUTO_INCREMENT		
2	IdStudent	int(11)			No	None			
3	Imelispita	varchar(1024)	utf8_unicode_ci		No	None			
4	Semestar	varchar(1024)	utf8_unicode_ci		No	None			
5	BrojESPB	int(11)			No	None			
6	Katedra	varchar(1024)	utf8_unicode_ci		No	None			
7	Poeni	float			No	None			
8	Ocena	float			No	None			

Slika 4.1.3. Tabela „ispiti“ u bazi podataka.

Kolona „IdIspit“ predstavlja identifikacioni broj ispita, tipa je **int**, limitirana je na maksimalnih 11 cifara i uključena je opcija AUTO_INCREMENT koja omogućava automatsko uvećanje vrednosti za 1 pri svakom novom unosu u tabelu.

Kolona „IdStudent“ predstavlja identifikacioni broj studenta iz tabele „Studenti“, tipa je **int**, limitirana je na maksimalnih 11 cifara. Ova kolona predstavlja vezu sa tabelom „Studenti“.

Kolona „ImeIspita“ predstavlja ime ispita koji je student položio, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „Semestar“ predstavlja semestar kome dati ispit održava, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „BrojESPB“ predstavlja broj espb poena koji dati ispit nosi, tipa je **int** i limitirana je na maksimalnih 11 cifara.

Kolona „Katedra“ predstavlja ime katedre kojoj dati ispit pripada, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „Poeni“ predstavlja ukupan broj osvojenih poena na celokupnom ispitnu na osnovu kojih se izvodi konačna, tipa je **float**.

Kolona „Ocena“ predstavlja konačnu ocenu koju je student dobio na nekom ispitnu, tipa je **float**.

Tabela „kviz“ služi za čuvanje podataka o pitanjima koja se pojavljuju u kvizu. Struktura tabele prikazana je na slici 4.1.4.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idPitanje	int(11)			No	None	AUTO_INCREMENT		
2	tekstPitanja	varchar(255)	utf8_unicode_ci		No	None			
3	odgovor1	varchar(255)	utf8_unicode_ci		No	None			
4	odgovor2	varchar(255)	utf8_unicode_ci		No	None			
5	odgovor3	varchar(255)	utf8_unicode_ci		Yes	NULL			
6	odgovor4	varchar(255)	utf8_unicode_ci		Yes	NULL			
7	tacanOdgovor	varchar(255)	utf8_unicode_ci		No	None			

Slika 4.1.4. Tabela „kviz“ u bazi podataka.

Kolona „IdPitanje“ predstavlja identifikacioni broj pitanja, tipa je **int**, limitirana je na maksimalnih 11 cifara i uključena je opcija AUTO_INCREMENT koja omogućava automatsko uvećanje vrednosti za 1 pri svakom novom unosu u tabelu.

Kolona „tekstPitanja“ predstavlja tekst pitanja koje se pojavljuje u kvizu, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „odgovor1“ predstavlja tekst prvog ponuđenog odgovora na dano pitanje, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „odgovor2“ predstavlja tekst drugog ponuđenog odgovora na dano pitanje, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „odgovor3“ predstavlja tekst trećeg ponuđenog odgovora na dano pitanje, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „odgovor4“ predstavlja tekst četvrтog ponuđenog odgovora na dano pitanje, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „tacanOdgovor“ predstavlja tekst tačnog odgovora na dano pitanje, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Tabela „user“ služi za čuvanje podataka o korisnicima koji su se registrovali, odnosno kreirali nalog na veb aplikaciji. Struktura tabele prikazana je na slici 4.1.5.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idUser	int(11)			No	None			Change Drop Primary Unique Index Spatial More
2	firstNameUser	varchar(1024)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial More
3	lastNameUser	varchar(1024)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial More
4	emailUser	varchar(1024)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial More
5	passUser	varchar(1024)	utf8_unicode_ci		No	None			Change Drop Primary Unique Index Spatial More
6	typeUser	varchar(255)	utf8_unicode_ci		No	standard			Change Drop Primary Unique Index Spatial More

Slika 4.1.5. Tabela „kviz“ u bazi podataka.

Kolona „idUser“ predstavlja identifikacioni broj registrovanog korisnika, tipa je **int**, limitirana je na maksimalnih 11 cifara i uključena je opcija AUTO_INCREMENT koja omogućava automatsko uvećanje vrednosti za 1 pri svakom novom unosu u tabelu.

Kolona „firstNameUser“ predstavlja ime registrovanog korinsika, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „lastNameUser“ predstavlja prezime registrovanog korinsika, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „emailUser“ predstavlja email sa kojim se korisnik registrovao, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera.

Kolona „passUser“ predstavlja lozinku koju je korisnik uneo prilikom registracije, tipa je **varchar** i limitirana je na maksimalnih 1024 karaktera. U ovoj koloni čuvaju se kriptovane lozinke.

Kolona „typeUser“ predstavlja tip korisnika („standard“ ili „admin“) koji određuje prava prilikom logovanja na veb aplikaciju, tipa je **varchar** i limitirana je na maksimalnih 255 karaktera.

Pored gore navedenih tabela koje su smeštene u bazi podataka, u bazu se takođe smeštaju i „privremene“ tabele koje su kopije gore opisanih tabela. To su tabele nad kojima se izvršavaju upiti koje korisnici unose na stranici za samostalno sastavljanje upita. Tabele ovog tipa su: „studenti“, „studenti_si“ i „ispiti“. Kako bi svakom korisniku na početku njegove sesije u tutorijalu bile na raspolaganju tabele popunjene istim podacima, onemogućeno je izvršavanje upita nad tabelama „studenti“, „studenti_si“ i „ispiti“. Ove tri tabele služe samo kao templejt za replikaciju. Naime, kada se neki korisnik uloguje u aplikaciju, u bazi se kreiraju kopije gore pomenutih tabela koje u nazivu sadrže identifikacioni broj korisnika. Na slici 4.1.6. crvenom bojom su označene kopije pomenutih tabela.

Table	Action	Rows	Type	Collation	Size	Overhead
ispiti		8	InnoDB	utf8_unicode_ci	16 KiB	-
ispit1		8	InnoDB	utf8_unicode_ci	16 KiB	-
kviz		28	InnoDB	utf8_unicode_ci	16 KiB	-
studenti		10	InnoDB	utf8_unicode_ci	16 KiB	-
studenti1		10	InnoDB	utf8_unicode_ci	16 KiB	-
studenti1_si		5	InnoDB	utf8_unicode_ci	16 KiB	-
studenti_si		5	InnoDB	utf8_unicode_ci	16 KiB	-
user		3	InnoDB	utf8mb4_unicode_ci	16 KiB	-
8 tables	Sum	77	InnoDB	utf8_unicode_ci	128 KiB	0 B

Slika 4.1.6. Replikacija tabela prilikom logovanja korisnika koji ima id=1.

Iz razloga uštade memorijskih resursa, replikacija pomenutih tabela se vrši prilikom logovanja korisnika u aplikaciju, dok se njihovo brisanje vrši kada se korisnik izloguje.

4.2. Opis koda

U ovom odeljku biće opisan kod kojim su realizovane najvažnije funkcionalnosti tutorijala. To su stranica za samostalno izvršavanje upita i kviz na kraju tutorijala. Takođe, opisane su i stranice koje su neophodne za izvršavanje koda na pomenutim stranicama.

4.2.1. Header i Footer

Unutar fajla header.php sadržan je kod koji je isti za svaku stranicu koja se prikazuje korisniku. Pošto se ovaj deo stranice nalazi na svim stranama on je izdvojen u poseban fajl header.php. Unutar taga <head> pozivaju se linkovi neophodni za funkcionisanje stranice. Tu između ostalog spada link ka css fajlu „theme.css“ koji se nalazi u folderu „styles“ i link za fajl script.js koji se nalazi u folderu „js“. Ova skripta je zadužena za „hvatanje događaja“, proveru unesenih podataka u formu i njeno slanje na server, AJAX zahteve, itd. Pored navedenog, u okviru tagova <script> poziva se funkcija *doesCookieExists()*; koja ispituje da li postoji kuki koji se setuje prilikom logovanja i u slučaju da ne postoji šalje korisnika na naslovnu stranicu kako bi se ulogovao. Na ovaj način onemogućen je pristup tutorijalu korisnicima koji nisu ulogovani. Kod fajla header.php prikazan je u nastavku.

```
<!DOCTYPE html>
<html>
    <head>
        <title>SQL Tutorijal</title>
        <meta charset="UTF-8" />
        <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
        <link rel="stylesheet" href="../../styles/theme.css">
        <link href="https://fonts.googleapis.com/css?family=Armata" rel="stylesheet">
        <script src="../../js/jquery-3.2.1.min.js"></script>
```

```

<script src="../js/jquery_cookie.js"></script>
<script src="../js/script.js"></script>
<script src="../js/modal.js"></script>
<script>
    doesCookieExists();
</script>
</head>
<body>
<div id="wrapper">
    <div id="header">
        <div id="leftHeader" class="dveTrecineW">
            <div id="logo_div" class="headerLeft1W cetvrtinaW">
                <a href="index.php"></a>
            </div>
            <div class="headerLeftW2 triCetvrtineW">
                <label id="textTutorial">SQL TUTORIJAL</label>
            </div>
        </div>
        <div id="rightHeader" class="trecinaW">
            <?php include("login_form.php"); ?>
        </div>
    </div>
</div>

```

4.2.2. Config.php

Fajl config.php sadrži funkcije koje su pozivane na serverskim stranicama za povezivanje i raskidanje veze sa bazom podataka.

```

function connectToDatabase () {
    $servername = "localhost";
    $username = "root";
    $password = "";
    $database = "sql tutorial";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $database);
    $conn->set_charset("utf8");

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    return $conn;
}

function disconnectFromDatabase ($conn) {
    $conn->close();
}

```

Takođe, sadrži funkciju *respondJson* koja je korišćena za vraćanje statusa o izvršavanju upita. Ovoj funkciji se prosleđuju status o uspešnosti izvršavanja upita i poruka, na osnovu kojih se formira JSON objekat. U nastavku je prikazan kod pomenute funkcije.

```

function respondJson ($status, $message) {
    $obj = new StdClass();
    $obj->status = $status;

```

```

    $obj->message = $message;
    echo(json_encode($obj));
}

```

4.2.3. Stranica za samostalno sastavljanje upita

Stranica za samostalno izvršavanje upita ima naziv tryYourSelf.php. Ona se sastoji iz dve celine. To su deo stranice gde su prikazane tabele iz baze podataka nad kojima korisnik može da izvršava upite i deo stranice za unos željenog upita. Ova stranica se razlikuje u zavisnosti sa koje stranice se na nju dolazi. Na primer, ukoliko je korisnik čitao lekciju vezanu za join upit i sa nje preko odgovarajućeg linka došao do ove stranice, biće učitane tabele koje su korišćene u primeru, nad kojima korisnik zatim može samostalno da sastavi upit.

Najpre se iz kukija uzima id ulogovanog korisnika kako bi se iz baze učitale tabele kopije, namenjene za korisnika sa datim id-em.

```

if(isset($_COOKIE['loggedUser']['idUser'])){
    $idUser = $_COOKIE['loggedUser']['idUser'];
    $tableStudenti = "studenti" . $idUser;
    echo "
<script>
executeQueryInDiv('#userTableStudenti', 'SELECT * FROM $tableStudenti');
</script>
";
}

```

Zatim se uzima vrednost *idTheme* koja se smešta u istoimenu promenljivu. Ova vrednost je na stranicu prosleđena GET metodom.

```

if(isset($_GET['idTheme'])){
    $idTheme = intval($_GET['idTheme']);
}

```

Na osnovu promenljive *idTheme* zna se sa koje stranice je korisnik došao, pa se na osnovu toga učitavaju tabele koje su korišćene u lekciji sa izvorišne stranice. U delu koda gde je uzet id korisnika učitana je tabela „studenti“ jer se ona nalazi na stranici bez obzira sa koje stranice korisnik na nju dolazi. U nastavku je prikazan deo koda koji u zavisnosti od promenljive *idTheme* na stranicu učitava i dodatne tabele.

```

if($idTheme > 13 && $idTheme != 20 ){

    $tableIspiti = "ispiti" . $idUser;
    echo "
<script>
$('#naslovTabeleIspiti').append('<h4> IZGLED TABELE \"Ispiti\" U BAZI PODATAKA
</h4>');
</script>
";
    echo "
<script>
executeQueryInDiv('#userTableIspiti', 'SELECT * FROM $tableIspiti');
</script>
";
}
if($idTheme == 20 ){

    $tableStudenti_si = "studenti" . $idUser . "_si";
}

```

```

echo "
<script>
$( '#naslovTabeleIspiti' ).append('<h4> IZGLED TABELE \"Studenti_si\" U BAZI
PODATAKA </h4>');
</script>
";
echo "
<script> executeQueryInDiv('#userTableIspiti', 'SELECT * FROM
$stableStudenti_si');
</script>
";
}

```

Izgled drugog dela stranice, gde se nalazi polje za unos upita, takođe je određen vrednošću promenljive *idTheme*. U nastavku je prikazan kod sa AJAX zahtevom za upis primera upita.

```

<div id="tryYourSelf" class="box_patern_try">
</div>
<?php
if(isset($_GET['idTheme'])){
$idTheme = intval($_GET['idTheme']);

echo "
<script>
$( '#tryYourSelf' ).load('../server_pages/loadTheme.php?idTheme=$idTheme&idUser=$idUser');
</script>
";
}
?>

```

Gore prikazanim zahtevom se u donji deo stranice učitava polje za unos upita, kao i dugmići za izvršavanje upita i resetovanje tabele.

Kada je stranica tryYourSelf.php učitana, korisnik može da sastavi upit i pokrene njegovo izvršavanje klikom na odgovarajuće dugme. Za „hvatanje“ događaja klika zadužena je skripta script.js. U kodu ispod prikazan je najpre niz provera, poput toga, da li je korisnik popunio polje za upit, a zatim da li upit sadrži naredbe na koje korisnik nema prava nad tabelama u bazi podataka. Nakon provera, sledi AJAX zahtev ka serverskoj stranici executeQueryJSON.php. Ishod odgovora na zahtev biće analiziran nakon odgovora serverske stranice.

```

$(document).on('click', '#izvrsiJSON', function() {
    $('#queryResult').html('<meta http-equiv="content-type" content="text/html;
charset=UTF-8"/>');
    var queryText = $.trim($("#queryArea").val());
    if(queryText == ''){
        alert('Nije popunjeno polje za upit !');
        return;
    }
    if (queryText.match(/create/i) || queryText.match(/drop/i) ){

```

```

    alert('Nemate create/drop prava nad bazom podataka!');
    $('#queryArea').val('');
    return;
}
var table =
$('<table></table>').addClass('paleBlueRows').attr('border', '1');
$('#queryResult').append(table);

$.ajax({
    url: "../server_pages/executeQueryJSON.php",
    type: "POST",
    data: {
        "queryText" : queryText
    },
    dataType: 'json',

```

Kao što se može videti u kodu, zahtev je kreiran tako da se serverskoj stranici šalje tekst upita, a kao odgovor se očekuje JSON. Sada sledi opis serverske stranice executeQueryJSON.php.

Na početku se proverava da li je do serverske stranice stigao tekst upita, takođe se učitava i fajl config.php odakle se poziva funkcija sa povezivanje i raskidanje veze sa bazom, kao i funkcije za generisanje odgovora serverske stranice u JSON formatu.

```

<?php
include("../server_pages/config.php");
if(isset($_POST['queryText'])){
    $queryText = $_POST['queryText'];
    $conn = connectToDatabase();

```

Zatim se ispituje kog je tipa upit, uzima se prvih 6 karaktera iz teksta upita i ukoliko upit nije u skladu sa korisnikovim pravima, formira se odgovor servera da je došlo do greške. Iako je provera korisnikovih prava vršena i na klijentskoj strani, potrebno je izvršiti je i na strani servera zbog raznih vrsta SQL napada.

```

$queryText = trim($queryText);
$queryText = mb_convert_encoding($queryText, "UTF-8");
$typeQuery = substr($queryText, 0, 6);

if (strcasecmp($typeQuery, 'select') != 0 && strcasecmp($typeQuery, 'insert') != 0 && strcasecmp($typeQuery, 'update') != 0 && strcasecmp($typeQuery, 'delete') != 0) {
    respondJson("Error", "Error in query!");
}

```

Nakon izvršenih provera i utvrđivanja kog tipa je upit sledi grananje na dve grane. Prva je ako je u pitanju upit koji ne menja sadržaj tabele tj. selekt upit, a druga je za upite koji menjaju sadržaj tabela (insert into, update i delete). Najpre će biti opisan slučaj za prvu granu.

Kao što je u opisu strukture baze podataka objašnjeno, pri svakom logovanju za korisnika se kreiraju tabele kopije nad kojima on izvršava upite. Te tabele pored svojih naziva sadrže i

identifikacioni broj korisnika. Kako bi bilo obezbeđeno da se upit koji korisnik unosi izvršava nad „njegovim“ tabelama, imena tabela koje je korisnik uneo u polje za upit menjaju se sa imenima tabela koje su namenjene za izvršavanje upita ulogovanog korisnika. Identifikacioni broj korisnika koji se nalazi u nazivu tabela preuzet je iz kukija. Zatim sledi smeštanje rezultata upita u niz koji se kasnije konvertuje u JSON format i kao takav predstavlja odgovor servera na AJAX zahtev. Kod koji je opisan je prikazan u nastavku.

```

if (strcasecmp($typeQuery, 'select') == 0) {
    $queryText =
        str_ireplace("studenti", "studenti".$_COOKIE['loggedUser']['idUser'],
        $queryText);

    $queryText = str_ireplace("ispiti", "ispiti".$_COOKIE['loggedUser']['idUser'],
        $queryText);

    $queryText = str_ireplace("studenti_si",
        "studenti".$_COOKIE['loggedUser']['idUser']."_si", $queryText);

    if( !($stmt = $conn->prepare($queryText)) ) {
        respondJson("Error", "Error in query: ".$conn->error);
    }

    if(!$stmt->execute()) {
        respondJson("Error", "Error in query: ".$conn->error);
    }

    $nizObjekata = array();
    $br = 0;
    $result = $stmt->get_result();

    while($row = $result->fetch_assoc()) {
        $imenaKolona = array_keys($row);
        $obj = new stdClass();
        for($i = 0; $i < count($imenaKolona); $i++) {
            $imeKol = $imenaKolona[$i]."";
            $obj->$imeKol = $row[$imeKol]."";
        }
        array_push($nizObjekata, $obj);
    }

    die(json_encode($nizObjekata));
}

```

Drugi slučaj je za upite koji menjaju sadržaj tabela u bazi podaka. Kao i u prvom slučaju podešeno je da se upit izvršava nad tabelama ulogovanog korisnika. Nakon toga sledi izvršavanje upita i formiranje odgovora serverske strane pozivom funkcije *respondJson* kojoj se prosleđuje status i broj vrsta koje su izvršavanjem upita pretrpele izmene.

```

if(strcasecmp($typeQuery, 'insert') == 0 || strcasecmp($typeQuery, 'update') ==
0 || strcasecmp($typeQuery, 'delete') == 0) {

```

```

$queryText = str_ireplace("studenti", "studenti".$_COOKIE['loggedUser']['idUser'],
$queryText);

$queryText = str_ireplace("ispiti", "ispiti".$_COOKIE['loggedUser']['idUser'],
$queryText);

$queryText=str_ireplace("studenti_si", "studenti".$_COOKIE['loggedUser']['idUser']
]."_si", $queryText);

if( !($stmt = $conn->prepare($queryText)) ) {
    respondJson("Error", "Error in query: ".$conn->error);
}

if(!$stmt->execute()) {
    respondJson("Error", "Error in query: ".$conn->error);
}

respondJson("SuccessCounting", "Number of rows affected: ".
$conn->affected_rows);

}

}

```

Nakon što je serverska strana izvršila prosleđeni upit i formirala odgovor na AJAX zahtev, sledi povratak na klijentsku stranu. Na klijentskoj strani postoje dve vrste prihvatanja odgovora na zahtev, uspeh i neuspeh. U slučaju uspeha, tj. dobijenog JSON fajla od servera, sledi prikaz rezultata upita u tabeli i osvežavanje tabele „studenti“ koja se nalazi na strani za samostalno sastavljanje upita. Na ovaj način, korisniku su u odmah vidljive promene u tabeli nakon izvršenog upita. Sa druge strane, kao neuspeh se smatra svaki odgovor servera na zahtev koji nije u JSON formatu. Ova dva slučaja prikazana su u kodu u nastavku.

```

success: function(objekti){

    if( objekti.constructor === Array){
        var $thead = $('<thead>');
        var $tr = $('<tr>');
        $.each(objekti[0], function(key, value) {

            $tr.append($('<th>').text(key)) ;
        });
        table.append($thead);
        $thead.append($tr);

        for(var i=0; i<objekti.length; i++){
            var $tr = $('<tr>');
            $.each(objekti[i], function(key, value) {

                $tr.append($('<td>').text(value)) ;
            });
            table.append($tr);
        }
    }else if(objekti.status == "SuccessCounting"){
        console.log(objekti.message);
    }
}

```

```

$( '#queryResult' ).html( objekti.message );

executeQueryInDiv( '#userTableStudenti' , 'SELECT * FROM
studenti '+$.cookie("loggedUser[idUser]"));

} else if( objekti.status == 'Error'){
    $('#queryResult').html( objekti.message );
    console.log("ovo je " + objekti.message );
}

} ,
error: function(objekat){
    var error = objekat.responseText;
    var firstLine = error.split('<')[0];
    $('#queryResult').text(firstLine);
    console.log("Something is wrong or JSON did not arrive.");
}

```

4.2.4. Kviz

U ovom delu biće objašnjeno kako je realizovan kviz koji se nalazi na kraju tutorijala. Kada korisnik preko odgovarajućeg linka dođe do strane kviza, u brauzer se učitava strana quiz.php. Na njoj se nalazi dugme za početak kviza (id dugmeta je „startQuiz“).

```

<?php
    include("header.php");
?>

<div id="content">
<?php include("navigation.php"); ?>
<div id="content_right">
<div id="quiz" class="box_patern">

<p id="quiz_about">
U ovom kvizu možete proveriti vaše znanje stečeno na ovom tutorijalu.<br/>
Kviz se sastoji od 10 pitanja.<br/> Smatra se da ste uspešno savladali osnove
SQL-a u koliko odgovotite tačno na minimum 70% pitanja.
</p><br/>
<center>
<br/><br/>
<button id='startQuiz' class="myButton" >Započni Kviz</button>
</center><br/>
</div>
</div>
</div>

<?php include("footer.php"); ?>

```

Kada korisnik klikne na dugme za početak kviza, kao i u prethodnim slučajevima za taj događaj je zadužena skripta script.js. U pomenutoj skripti za klik na dugme sa id-em „startQuiz“ definišu se promenljive za broj tačno i netačno odgovorenih pitanja i upućuje se AJAX zahtev za učitavanje pitanja ka serverskoj stranici load_question.php.

```

$(document).on("click", "#startQuiz", function(event) {
    noAnswers=0;
    noCorrAns=0;
    $('#quiz').load('../server_pages/load_question.php');
    $('html, body').animate({ scrollTop: 0 }, 'fast');
});

```

Na serverskoj stranici load_question.php najpre se učitava fajl config.php i poziva funkcija za kreiranje konekcije ka bazi podataka u kojoj se nalaze pitanja koja se pojavljaju u kvizu. Kako se ova stranica poziva svaki put kada je potrebno učitati novo pitanje, ona je realizovana preko „do while“ petlje. U „do“ delu petlje formira se upit koji iz tabele sa pitanjima u bazi podataka izvlači jedno slučajno odabранo pitanje. Ovim upitom dobijeni su podaci koji su potrebni za dalji prikaz pitanja kao i proveru tačnog odgovora. U odgovarajuće promenljive smeštaju se tekst pitanja, id pitanja, ponuđeni odgovori, kao i tačan odgovor na dato pitanje. Pored toga vrši se provera koliko na dato pitanje postoji ponuđenih odgovora zbog kasnjeg prikaza. Postavlja se i kuki koji u sebi nosi tačan odgovor, zbog provere da li je odgovor koji je korisnik izabrao tačan. Ova provera je sprovedena na klijentskoj strani.

```

<?php
    include ("config.php");
    $conn = connectToDatabase ();

    do{
        $query = "SELECT * FROM kviz ORDER BY RAND() LIMIT 1";
        $stmt = $conn->prepare($query);
        $stmt->execute();
        $result = $stmt->get_result();
        if($result->num_rows > 0){

            $row = $result->fetch_assoc();

            $pitanje = $row['tekstPitanja'];
            $idPitanje = $row['idPitanje'];
            $odgovor1 = $row['odgovor1'];
            $odgovor2 = $row['odgovor2'];
            $odgovor3 = $row['odgovor3'];
            $odgovor4 = $row['odgovor4'];
            $isOdgovor4NULL = empty($odgovor4);
            $isOdgovor3NULL = empty($odgovor3);
            $tacanOdgovor = $row['tacanOdgovor'];
            setcookie("tacanOdgovor", $tacanOdgovor, time() + 60*60 , "/");

        }else{

            disconnectFromDatabase($conn);
            die("Greška pri učitavanju pitanja: ".$conn->error);
        }

        while(thereWasQuestion($idPitanje));
    }

```

U „while“ delu petlje poziva se funkcija koja proverava da li je dato pitanje već bilo postavljeno u kvizu datom korisniku. Ova funkcija obezbeđuje da jedno pitanje ne može biti postavljeno dva puta u jednom učešću u kvizu. Funkciji se prosleđuje id pitanja, a ona id-eve postavljenih pitanja smešta u niz i pri svakom pokušaju da se učita novo pitanje proverava da li se

id novog slučajno izvučenog pitanja nalazi u nizu postavljenih pitanja. Ukoliko se novoizvučeno pitanje nalazi u nizu postavljenih pitanja opet se izvršava „do“ deo petlje za učitavanje novog pitanja. Kod opisane funkcije prikazan je nastavku.

```

function thereWasQuestion ($idQuestion) {
    // ovde ulazi ako je prvo pitanje
    if (!isset ($_COOKIE['passedQuestions'])) {
        setcookie ("passedQuestions[0]", $idQuestion, time ()+60*60 , "/");
        return false;
    }
    //ovde ulazi ako nije prvo pitanje
    else{
        $n = count ($_COOKIE['passedQuestions']);
        // ovde ulazi ako je poslednje pitanje
        if ($n == 10){
            for ($i=0; $i<$n; $i++){
                setcookie ("passedQuestions[$i]","", time ()-60*60 , "/");
            }
            return false;
        }
    }
    //ispitivanje da li je pitanje vec bilo
    for ($i=0; $i<$n; $i++){
        $br = intval ($_COOKIE['passedQuestions'][$i]);
        if ($br == $idQuestion) return true;
    }
    setcookie ("passedQuestions[$n]", $idQuestion, time ()+60*60 , "/");
    return false;
}

```

Nakon što je izvršena provera da pitanje već nije bilo postavljeno, prekida se veza sa bazom podataka i sledi prikaz pitanja i ponuđenih odgovora koji je realizovan sledećim kodom.

```

disconnectFromDatabase ($conn);

echo

<div id='question_text'>$pitanje<br/>
</div><br/>
<div>
<form id='quiz_form'>

<input type='radio' name='answer'
value="" .htmlentities ($odgovor1) ."\">$odgovor1<br/>
<input type='radio' name='answer'
value="" .htmlentities ($odgovor2) ."\">$odgovor2<br/>
";
if (!$isOdgovor3NULL) {

echo
<input type='radio' name='answer'
value="" .htmlentities ($odgovor3) ."\">$odgovor3<br/>
";
}

```

```

if ($isOdgovor4NULL) {echo"<br/>";}

if (!$isOdgovor4NULL) {

echo"
<input type='radio' name='answer'
value="" . htmlentities($odgovor4) . "\">$odgovor4<br/><br/>
";
}

echo"
<input type='submit' id='next_question' class='myButton' value='Sledeće pitanje'
/>
</form>
</div>
";

```

Nakon što korisnik klikne na dugme za prikaz sledećeg pitanja opet se poziva skripta script.js. Najpre se vrši provera da li je korisnik izabrao neki od ponuđenih odgovora. Ukoliko nije pojavljuje se alarmni prozor da mora biti odabran jedan od ponuđenih odgovora. Kada je korisnik izabrao jedan odgovor, promenljiva u kojoj se čuva broj odgovorenih pitanja se inkrementira. Zatim sledi provera da li je korisniku postavljeno ukupno 10 pitanja i ukoliko nije, opet se poziva stranica load_question.php za učitavanje novog pitanja. Ukoliko je broj odgovorenih pitanja jednak 10, korisnik je došao do kraja kviza i nije potrebno dalje učitavanje pitanja. U tom slučaju poziva se serverska stranica quiz_result.php. Dati opis realizovan je sledećim kodom.

```

$(document).on('submit', '#quiz_form', function(event) {

event.preventDefault();

var trueAnswer = $.cookie("tacanOdgovor");

var selected =
$("input[type='radio'][name='answer']:checked").val();

if(typeof selected === "undefined") {

    alert("Morate izabrati neki odgovor!");
    return;
}

console.log("selektovan odgovor je " + selected);

if( trueAnswer == selected ) {
    alert("Tačan Odgovor!");
    noCorrAns++;
} else{
    alert("Netačan Odgovor. Tačan odgovor je "+trueAnswer);
}
noAnswers++;

if(noAnswers == 10) {

```

```

$( '#quiz' ).load( '../server_pages/quiz_result.php?noCorrAns=' + noCorrAns + '&noAnswers=' + noAnswers );
    focusOnQuizResult();
} else{
    $( '#quiz' ).load( '../server_pages/load_question.php' );
}

);

```

Stranici quiz_result.php se GET metodom prosleđuje broj odgovorenih pitanja i broj tačno odgovorenih pitanja. Ona računa procenat tačno odgovorenih pitanja i u zavisnosti od toga prikazuje jednu od dve animacije sa rezultatom učešća u kvizu. U nastavku je prikazan kod date stranice.

```

<?php
if(isset($_GET['noCorrAns']) && isset($_GET['noAnswers'])) {
    $noCorrAns = intval($_GET['noCorrAns']);
    $noAnswers = intval($_GET['noAnswers']);
    $quizResult = round(( $noCorrAns / $noAnswers ) * 100, 2);
    if($quizResult >= 70) {

        echo "
<div id='question_text1'>

<center><img src='../images/congrats.gif' ></center>
<p>Tačno ste odgovorili na <span id='percentageSuccess'>$quizResult%</span>
pitanja !</p>
<p>Čestitamo !</p>
</div>
";
    } else{

        echo "
<div id='question_text1'>
<center><img src='../images/failure.gif' ></center>
<p id='p'>
Tačno ste odgovorili samo na <span id='percentageFailure'>$quizResult%</span>
pitanja, da biste prošli kviz morate imati minimum 70% tačnih odgovora.
</p>
</div>
<br/>
<div>
<center><button id='retryQuiz' class='myButton'>Ponovi kviz</button></center>
</div>
";
    }
}
?>

```

Ovim delom je završen opis koda kojim su realizovane najvažnije funkcionalnosti veb tutorijala za učenje SQL jezika.

5.ZAKLJUČAK

Cilj teze je realizacija veb tutorijala za SQL jezik. Korisnik preko navigacionog menija bira lekcije za osnovne SQL naredbe. U okviru svake lekcije dat je primer na kome je lekcija objašnjena. Takođe, korisnik ima mogućnost da sa svake lekcije ode na stranu za samostalno sastavljanje upita, gde mu se u zavisnosti od lekcije koju je prethodno čitao izlaže primer. Korisnik na ovoj strani može da izvršava upite i analizira rezultate njihovog izvršenja. Na kraju tutorijala postoji kviz u kome korisnik može da proveri znanje stečeno pohađanjem tutorijala. Pored dela gde korisnik ima mogućnost interakcije sa aplikacijom, postoje i stranice za bolje upoznavanje sa bazama podataka. Pored korisnika, postoji i administrator sa pravima za izmenu korisničkih podataka i upravljanje kvizom. Ovaj tutorijal je namenjen svim korisnicima koji žele da nauče osnove SQL jezika i baza podataka. Lekcije kreću od osnova SQL jezika i prilagođene su za korisnike bez prethodnog znanja u ovoj oblasti.

Aplikacija je podložna mnogim unapređenjima. Što se tiče vizuelnog dela unapređenje može biti futer. U futer se mogu smestiti linkovi ka naprednijim materijalima iz date oblasti. Takođe, moguće je uvesti ocenjivanje tutorijala i ostavljanje komentara od strane korisnika. Od funkcionalnosti, unapređenje bi moglo biti da korisnici pored osnovnih upita za manipulisanje podacima u bazi podataka, imaju mogućnost kreiranja i brisanja novih tabela. Na ovaj način bi se znatno proširio spektar mogućnosti koje korisnik može da isproba tokom tutorijala pa bi i nivo stečenog znanja mogao biti viši. U svakom slučaju, najbolje je osluškivati zahteve korisnika i uz pomoć ocena i komentara korisnika krenuti u dalje unapređenje početne verzije tutorijala.

LITERATURA

- [1] Aleksandra Smiljanić „Internet programiranje“, 2017.
- [2] E. Castro, *HTML for the World Wide Web with XHTML and CSS*, Fifth Edition, Peachpit Press, 2003.
- [3] D. Falangan, *JavaScript: The Definitive Guide*, O'Reilly Media, Inc.; 5th edition, August 2006.
- [4] Jason Gilmore, *Beginning PHP and MySQL: From Novice to Professional*, Third Edition (Beginning from Novice to Professional), Apress, March 2008.
- [5] <https://sr.wikipedia.org/sr-el/PHP>
- [6] <https://php.net/>
- [7] <https://mysql.com/>
- [8] <https://sr.wikipedia.org/wiki/MySQL>
- [9] <http://www.w3schools.com/>
- [10] <http://www.stackoverflow.com/>