

Web aplikacija za vođenje režija za kućanstvo.

Filipović, Željko

Master's thesis / Diplomski rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:040040>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-31**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

**WEB APLIKACIJA ZA VOĐENJE REŽIJA ZA
KUĆANSTVO**

Diplomski rad

Željko Filipović

Osijek, 2016.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada**

Osijek, 25.09.2016.

Odboru za završne i diplomske ispite**Imenovanje Povjerenstva za obranu diplomskog rada**

Ime i prezime studenta:	Željko Filipović
Studij, smjer:	Diplomski sveučilišni studij Elektrotehnika, smjer Komunikacije i informatika
Mat. br. studenta, godina upisa:	D-179, 15.10.2013.
OIB studenta:	97580112777
Mentor:	Prof.dr.sc. Željko Hocenski
Sumentor:	Doc.dr.sc. Ivan Aleksi
Predsjednik Povjerenstva:	Doc.dr.sc. Tomislav Matić
Član Povjerenstva:	Doc.dr.sc. Ivan Aleksi
Naslov diplomskog rada:	Web aplikacija za vođenje režija za kućanstvo.
Znanstvena grana rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak diplomskog rada:	REZ Filipovic baze podataka sa unosima stanja brojila za električnu energiju (visoka tarifa, niska tarifa), plin, vodu - automatski izračun potrošnje + fiksnu naknadu distributera te cijena mjesečnih računa drugih režija kao što su telefon/internet/iptv, rtv pristojba, odvoz smeća, komunalna naknada i ostalo te mogućnost odabira ako korisnik živi u zgradi pa ima još i pričuve, održavanje lifta, čišćenje hodnika i ostalo.. Izveštaji i pregled mi mogli biti zasebno za svaku stavku ili sveukupno na mjesečnoj/godišnjoj bazi i za protekle mjesece/godine.. Bila bi forma za registraciju i logiranje korisnika, mogućnost slanja izvještaja na mail te ispis svih stavki.. Koristio bi HTML, CSS, PHP i SQL..
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 Postignuti rezultati u odnosu na složenost zadatka: 3 Jasnoća pismenog izražavanja: 2 Razina samostalnosti: 3
Datum prijedloga ocjene mentora:	25.09.2016.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O ORIGINALNOSTI RADA**

Osijek, 02.11.2016.

Ime i prezime studenta:

Željko Filipović

Studij:

Diplomski sveučilišni studij Elektrotehnika, smjer Komunikacije i informatika

Mat. br. studenta, godina upisa:

D-179, 15.10.2013.

Ephorus podudaranje [%]:

6%

Ovom izjavom izjavljujem da je rad pod nazivom: **Web aplikacija za vođenje režija za kućanstvo.**

izrađen pod vodstvom mentora Prof.dr.sc. Željko Hocenski

i sumentora Doc.dr.sc. Ivan Aleksi

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

I. UVOD	1
1.1. Zadatak diplomskog rada.....	1
1.2. Poveznica na izrađenu internet stranicu	1
II. PRIMIJENJENE TEHNOLOGIJE I ALATI ZA RAZVOJ APLIKACIJE	2
2.1. Internet (Web) aplikacije	2
2.2. Arhitektura i princip rada web aplikacija	2
2.3. Internetski preglednici	4
2.4. HTML.....	6
2.5. CSS	7
2.6. PHP.....	8
2.7. MySQL.....	9
2.8. Apache web poslužitelj.....	10
2.9. XAMPP	12
III. REALIZACIJA SUSTAVA	14
3.1. Struktura i princip rada aplikacije	14
3.2. Korisničko sučelje aplikacije.....	15
3.3. Razvoj sustava prijave i registracije u aplikaciju	16
3.4. Razvoj funkcija za rad s bazom podataka	20
3.4.1. Spajanje u bazu podataka	22
3.4.2. Upis u bazu podataka	22
3.4.3. Ispis iz baze podataka.....	25
3.4.4. Uređivanje iz baze podataka	27
IV. ZAKLJUČAK	29
LITERATURA.....	30
SAŽETAK.....	31
ABSTRACT	32
ŽIVOTOPIS	33
PRILOZI.....	34

I. UVOD

U ovom diplomskom radu će se razviti web aplikacija za vođenje režija za kućanstvo. Zadatak aplikacije je omogućiti svakom korisniku evidenciju plaćanja režija u svom domaćinstvu putem Interneta. Jedini uvjet za korištenje web aplikacije je imati pristup Internetu i imati osnovni web preglednik. Aplikacija će sadržavati registraciju i prijavu korisnika, pregled troškova za svaku pojedinu režiju po mjesecu i godini, te unos novih vrijednosti režija za protekli mjesec ili promjena vrijednosti za bilo koji mjesec i godinu.

Za programiranje aplikacije koristit će se PHP skriptni programski jezik, te će se opisati izgradnja i konfiguracija razvojne okoline za rad aplikacije. Opisat će se instalacija i konfiguracija Apache web servera i MySQL sustava za upravljanje bazama podataka. U svrhu spremanja podataka, objasniti će se proces kreiranja i upravljanja bazama podataka te tablicama unutar njih.

Opisat će se funkcije za rad sa bazom podataka, koje će spremati vrijednosti u bazu te ispis vrijednosti za pregled u sučelju. Prikazat će se tok izrade od idejnog rješenja do funkcionalne web aplikacije.

1.1. Zadatak diplomskog rada

Baza podataka sa unosima cijena mjesečnih računa režija kao što su električna energija, voda, komunalije, plin, RTV pristojba, pričuvna, čišćenje hodnika u zgradi, telefon/internet. Pregled može biti zasebno za svaku stavku ili sveukupno na mjesečnoj/godišnjoj razini, te za protekle mjesece i godine. Forma za registraciju i prijavu korisnika. Koristit će se HTML, CSS, PHP i MySQL.

1.2. Poveznica na izradenu internet stranicu

<http://www.zeljko Filipovic.com.hr/rezije>

II. PRIMIJENJENE TEHNOLOGIJE I ALATI ZA RAZVOJ APLIKACIJE

Pojam razvoja aplikacija i programiranja često se znaju poistovjećivati iako se zapravo radi o različitim procesima. Kada govorimo o pojmu razvoja aplikacija (engl. *Application Development*) ono obuhvaća sve faze razvoja informatičkog sustava ili računalne aplikacije koji može biti složen od više komplementarnih aplikacija. Susrećemo se s raznim tehnologijama i metodama koje definiraju procese u razvoju aplikacija.

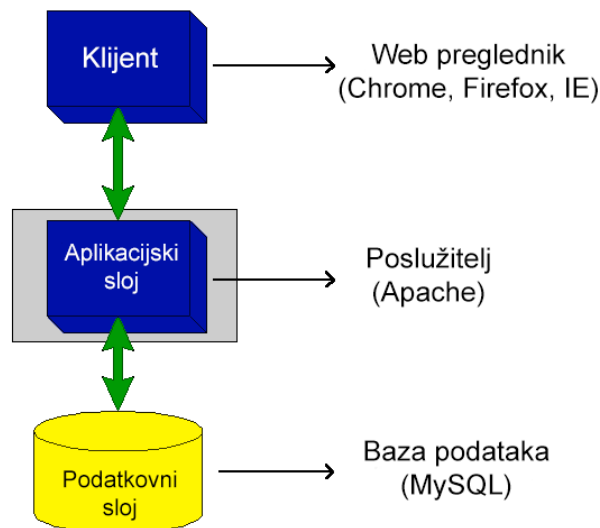
2.1. Internet (Web) aplikacije

Internet aplikacije su programska rješenja kojima se pristupa putem Internet preglednika (engl. *browser*) koristeći Internet ili intranet mrežu. Veliki rast, razvoj i popularizacija web aplikacija može se zahvaliti činjenici da su dostupne u bilo koje vrijeme, s bilo kojeg mjesta i s bilo kojeg uređaja koji ima instaliran Internet preglednik. Visoku zastupljenost u novije vrijeme imaju mobilni uređaji u odnosu na osobna računala.

Ključni razlog popularnosti web aplikacija je mogućnost instalacije na velikom broju klijentskih računala. Ažuriranje i održavanje web aplikacija se odvija bez distribucije i instalacije programske podrške. Ubrzanom širenju Internet aplikacija ponajviše su doprinijele usluge koje omogućavaju korisnicima korištenje, razmjenu, promjenu informacija neovisno o platformi koju koriste. Prijava u web aplikaciju najčešće se obavlja putem korisničkog imena ili e-maila i lozinke, što će se opisati u sljedećem poglavlju.

2.2. Arhitektura i princip rada web aplikacija

Princip rada Internet aplikacija temelji se na načinu komunikacije između klijenta i poslužitelja (engl. *server*), preko internetske mreže. Zadatak svake Internet aplikacije je prijenos informacija putem Internet protokola od web poslužitelja, do njenog korisničkog sučelja na klijentskoj strani i obrnuto. Prema slici 2.1. prikazana je arhitektura Internet aplikacije.



Sl. 2.1. Arhitektura web aplikacija

Web aplikacije podijeljene su na tri glavna sloja, prvi sloj predstavlja korisnički sloj (engl. *Presentation tier*), sloj koji prikazuje informacije korisniku putem web preglednika. Drugi sloj predstavlja aplikacijski sloj (engl. *Middleware tier*), sloj koji upravlja aktivnostima koje aplikacija treba izvršavati. Drugim riječima ovaj sloj upravlja sadržajem podataka koji se prikazuju korisniku, dok ujedno i obrađuje podatke dobivene od korisnika pretvarajući ih u upite za učitavanje ili upisivanje u bazu podataka. Treći sloj je podatkovni sloj (engl. *Data tier*), sloj koji upravlja pohranjivanjem podataka u bazu i prikazom podataka iz baze na poslužitelju. Svi slojevi međusobno komuniciraju preko standardiziranih Internetskih protokola, kojih se prilikom razvoja aplikacija treba pridržavati. Prema [3] web aplikacija dinamički generira niz HTML dokumenata koji se mogu prikazati Internetskim preglednikom. Skriptni jezici, koji se izvode na klijentskoj strani, koriste se za korisničko sučelje u koje se žele dodati dinamički elementi. Korisniku se pojedinačna stranica prikazuje kao statični dokument, a stranice se interaktivno mogu mijenjati u aplikaciji. Korisnik može podatke unositi s pomoću web formi koje su uključene u stranice.

Web aplikacije se najčešće sastoje od sljedećih aktivnosti:

- Korisnik pokreće web preglednik
- Preko web sučelja postavi određeni upit
- Sučelje za upite šalje zahtjev programu na serveru
- Program na serveru daje odgovor na zahtjev koristeći izvor podataka (bazu podataka)
- Odgovor iz baze prosljeđuje preko poslužitelja na klijentsku stranu
- Odgovor preko web sučelja postaje vidljiv korisniku

Prednosti web aplikacija:

- Web aplikacije ne zahtijevaju složenu proceduru implementacije kod velikih organizacija
- Podaci su sigurno smješteni na serverima čime se isključuje mogućnost slučajnog brisanja od strane korisnika
- Pristupiti je moguće sa bilo koje lokacije gdje ima Internet veza
- Kompletna nadogradnja se vrši na poslužitelju, svi korisnici trenutno dobiju novu verziju
- Mogu se integrirati sa ostalim web aplikacijama
- Rade bez obzira na operacijski sustav koji je instaliran, potreban je samo web preglednik

Nedostaci web aplikacija:

- Nedovoljno usuglašeni standardi za HTML, CSS od strane proizvođača web preglednika
- Korisnik u svom pregledniku može samostalno postaviti parametre prikaza npr. font, veličinu slova i drugo, stoga se narušava konzistentnost prikaza aplikacije
- Brzina rada aplikacije ovisi o brzini poslužitelja i brzini mrežne veze s poslužiteljem na kojem se nalazi web aplikacija
- Mogući sigurnosni problemi u mreži (vatrozid, virusi)

2.3. Internetski preglednici

Svaki uređaj koji pristupa sadržaju na Internetu treba imati instaliran Internetski preglednik (engl. *Web browser*). Internetskim preglednikom se definira instalirani program na uređaju koji podržava HTTP protokol (engl. *HyperText Transfer Protocol*) za prijenos informacija na webu. Program omogućuje korisniku pregled web stranica, čitanje elektroničke pošte (engl. *e-mail*), čavljanje (engl. *chat*), pregled multimedijalnih sadržaja vezanih uz njih itd. Općeniti naziv za to je „surfanje“ internetom.

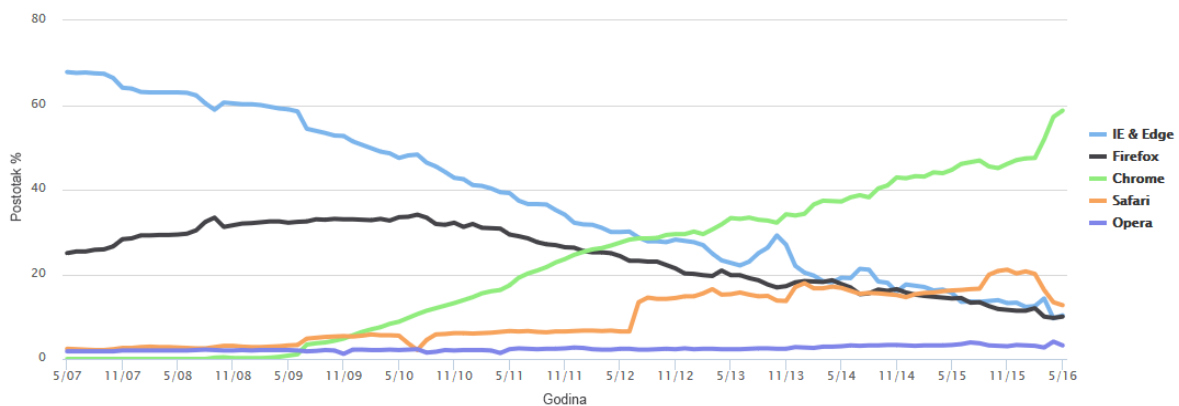
Postoje različiti Internetski preglednici od kojih neki dolaze kao pred instaliran preglednik na operacijskom sustavu kao što je npr. Internet Explorer na Microsoft Windows OS-u.

Prema [4] najkorišteniji Internetski preglednici u 2016. godini su:

- 1) Google Chrome 71.4 %
- 2) Mozilla Firefox 16.9 %
- 3) Internet Explorer i Edge 5.7 %
- 4) Safari 3.6 %
- 5) Opera 1.2 %
- 6) Ostali 1.2 %

Kao što se vidi Google-ov Internetski preglednik Chrome je izdan krajem 2008. godine te je postajao sve popularniji, da bi se krajem 2012. godine izjednačio korištenjem s Microsoft-ovim Internet Explorerom.

Na slici 2.2. prikazan je grafički prikaz postotka korištenja navedenih najkorištenijih Internetskih preglednika od 2007. do 2016. godine.



Sl. 2.2. Grafički prikaz korištenja Internetskih preglednika za razdoblje od 2007. do 2016. godine. [5]

Kako se Internetski preglednici koriste samo za prikazivanje sadržaja, ne sudjeluju u izravnom generiranju sadržaja web aplikacije krajnjem korisniku. Međutim, vrlo je bitan jer je izravan posrednik između dva računala (poslužitelja i klijenta) koji komuniciraju preko Internet protokola.

2.4. HTML

Prema [6] HTML (engl. *HyperText Markup Language*) što znači prezentacijski jezik za izradu web stranica. Hipertekst dokument stvara se pomoću HTML jezika. HTML jezikom oblikuje se sadržaj i stvaraju se hiperveze hipertekst dokumenta. HTML je jednostavan za uporabu i lako se uči, što je jedan od razloga njegove opće prihvaćenosti i popularnosti. Svoju raširenost zahvaljuje jednostavnosti i tome što je od početka bio zamišljen kao besplatan i tako dostupan svima. Prikaz hipertekst dokumenta omogućuje web preglednik. Temeljna zadaća HTML jezika jest uputiti web preglednik kako prikazati hipertekst dokument. Pri tome se nastoji da taj dokument izgleda jednako bez obzira o kojemu je web pregledniku, računalu i operacijskom sustavu riječ. HTML nije programski jezik niti su ljudi koji ga koriste programeri. Njime ne možemo izvršiti nikakvu zadaću, pa čak ni najjednostavniju operaciju zbrajanja ili oduzimanja dvaju cijelih brojeva. On služi samo za opis naših hipertekstualnih dokumenata. HTML datoteke su zapravo obične tekstualne datoteke, ekstenzija im je .html ili .htm. Osnovni građevni element svake stranice su znakovi (engl. *tags*) koji opisuju kako će se nešto prikazati u web pregledniku. Poveznice unutar HTML dokumenata povezuju dokumente u uređenu hijerarhijsku strukturu i time određuju način na koji posjetitelj doživljava sadržaj stranica.

Svaki HTML dokument sastoji se od osnovnih građevnih blokova - HTML elemenata. Svaki, pak, HTML element sastoji se od para HTML oznaka (engl. *tag*). Također, svaki element može imati i attribute kojim se definiraju svojstva tog elementa. Na samom početku HTML dokumenta preporučljivo je postaviti `<!DOCTYPE>` element, kojim se označava DTD (engl. *Document Type Declaration*), čime se definira točna inačica standarda koja se koristi za izradu HTML dokumenta. Nakon `<!DOCTYPE>` elementa, `<html>` elementom označava se početak HTML dokumenta. Unutar `<html>` elementa nalaze se i `<head>` element te `<body>` element. `<head>` element predstavlja zaglavlje HTML dokumenta u kojemu se najčešće specificiraju jezične značajke HTML dokumenta kao i sam naslov (engl. *title*) stranice. Pomoću određenih HTML elemenata unutar zaglavlja dodaju se i stilska obilježja stranice, bila ona direktno ugrađena (engl. *embedded*) ili dodana kao referenca na vanjsku CSS datoteku.

2.5. CSS

CSS je kratica za (engl. *Cascading Style Sheets*). CSS je jezik za oblikovanje stila koji određuje izgled HTML dokumenta. CSS može određivati vrstu fonta, boje, margine, crte, visinu, širinu, pozadinsku sliku, izgled gumba, napredno pozicioniranje i drugo. HTML također može biti korišten za određivanje izgleda web stranice. CSS nudi puno mogućnosti, točniji je i detaljan. CSS podržavaju svi današnji preglednici. HTML koristimo za određivanje strukture sadržaja. CSS koristimo za oblikovanje strukture sadržaja. Odvajanje stila izgleda web stranice od sadržaja dokumenta, učinio je održavanje mnogo lakšim. Na primjeru dijela CSS koda (Sl. 2.3.) korištenog u aplikaciji, nalaze se elementi *header* i *footer* s vrijednostima pozicija, visine, širine, pozadinske boje, boja fonta i poravnavanje teksta. [7]

```
#header
{
    width:100%;
    height:65px;
    background:#3553a6;
    color:#fff;
    font-family:Verdana, Geneva, sans-serif;
}

#header #left
{
    float:left;
    position:relative;
}

#header #right
{
    float:right;
    position:relative;
}

#header #right #content
{
    position:relative;
    top:25px;
    right:100px;
    color:#fff;
}

#footer
{
    position:fixed;
    bottom:0;
    width:100%;
    height:30px;
    background:#3553a6;
    color:#fff;
    text-align:center;
}
```

Sl. 2.3. Primjer dijela CSS koda korištenog u izradi aplikacije

2.6. PHP

PHP (engl. *Open Source Server Side*) skriptni programski jezik otvorenog koda za dinamičko generiranje HTML koda. PHP je programski jezik pomoću kojeg se može kreirati HTML stranica na poslužitelju, prije nego što je ona poslana klijentu sa potpunim dinamičkim sadržajem. Ovakvim načinom generiranja sadržaja klijent ne može vidjeti kod (skriptu), već ima pristup čistom HTML kodu. Klijent može vidjeti samo HTML, CSS i JavaScript kod. Danas jednu od najiskorištenijih i najnaprednijih server side skriptnih tehnologija, a koja se upotrebljava, predstavlja PHP. Prema statističkim podacima, gotovo 90% web stranica na Internetu je pisano upravo PHP-om.

Osim što PHP ne ovisi o Internetском pregledniku, također ne ovisi ni o operacijskom sustavu posjetitelja web aplikacije. Svaka linija koda u PHP-u mora završiti znakom „ ; “ koji razdvaja naredbe. Postoje dvije osnovne naredbe za ispis teksta: `echo` i `print`. Varijable se koriste za pohranjivanje vrijednosti, brojeva, stringova ili nizova. Kako bi se mogle koristiti više puta u skripti, sve varijable u PHP-u počinju znakom `$`. Vrlo bitna karakteristika PHP-a je razdvajanje server side koda od statičkog HTML koda. Tako u PHP-u korištenjem znakova `<?php ?>` govorimo poslužitelju da se između njih nalazi PHP kod i da je potrebno prvo njega izvršiti, a tek nakon toga poslati HTML izlaz tog koda klijentu. Ovo je osnova poslužitelj-klijent mrežne komunikacije putem TCP/IP protokola.

Za upotrebu PHP-a potrebno je imati zadovoljene ove pretpostavke:

- Instaliran Apache server na Windows ili Unix/Linux poslužitelju
- Instaliran PHP jezik na Windows ili Unix/Linux poslužitelju
- Instaliranu MySQL bazu na Windows ili Unix/Linux poslužitelju

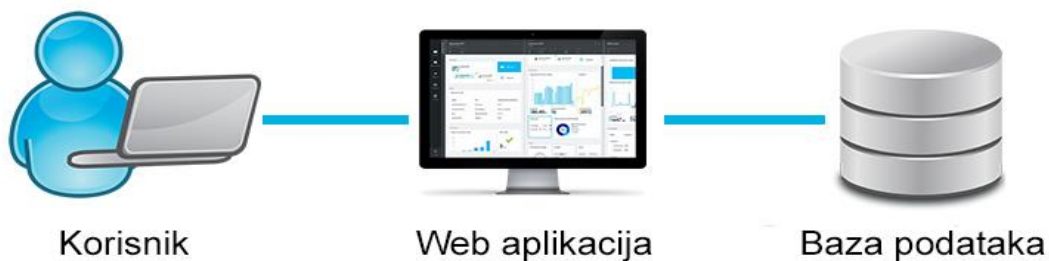
PHP sadrži mnogobrojne instrukcije i podržava rad s različitim tipovima podataka. Podržava kontrolne strukture koje određuju tijek skripti kao što su *if-else*, *switch-case*, *while*, *for* i ostale metode grananja programskog koda podržane u brojnim naprednim i poznatim programskim jezicima. Kao objektno orijentiran programski jezik sadržava i velik broj gotovih funkcija koje omogućavaju stvaranje i korištenje klasa i objekata. Jedne od najkorisnijih funkcija su funkcije za rad s bazama podataka, funkcije za rad s datumom i vremenom, podatkovne (*File* i *FTP*) funkcije, string funkcije, varijable, sesije i mnoge druge. [2]

2.7. MySQL

MySQL je sustav za upravljanje relacijskim bazama podataka. Dostupan je za široku primjenu, a ujedno i sustav koji se najviše koristi zajedno s PHP programskim jezikom pošto je međusobna integracija i podržanost vrlo dobra, ali PHP ne može samostalno pristupiti bazi podataka i tražiti informacije o korisnicima. Upravljanje podacima preko posebnog programskog jezika SQL (engl. *Structured Query Language*) odnosi na njihovo skladištenje, pregled i ažuriranje podataka, dok se pojam relacije odnosi na organizaciju podataka i njihovu međusobnu povezanost.

Baza podataka u MySQL-u može se dizajnirati s pomoću posebnog alata, phpMyAdmin ili direktno programskim kodom SQL jezika i PHP-a. Aplikacija phpMyAdmin je izdana pod licencom otvorenog koda, besplatna je za korištenje svim korisnicima što doprinosi popularnosti.

Ono što je posebno važno kod korištenja baze podataka je mogućnost da na jednom poslužitelju može postojati više različitih baza podataka, koje se sastoje od više tablica. Isto tako moguće je istovremeni pristup bazi podataka od više korisnika. Takvu hijerarhiju moguće je održati iz razloga zato što se u istoj aplikacije može postaviti više korisnika. Naime, svakom se korisničkom računu na poslužitelju mogu dodijeliti različita administracijska prava i ovlasti neovisno za svaku bazu podataka.



Sl. 2.4. Relacija između korisnika i baze podataka odvija se preko web aplikacije

Prije pristupa podacima u bazi, potrebno je napraviti vezu ili konekciju prema bazi. U MySQL jeziku, veza se postavlja preko funkcije `mysql_connect()`. Primjer prikaza `mysql_connect()` sa bazom podataka vidimo na slijedećem primjeru (Sl. 2.5.).

```

<?php
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '');
define('DB_DATABASE', 'rezije');
$db = mysql_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_DATABASE)
or die("Dogodila se greska!");
?>

```

Sl. 2.5. PHP kod za spajanje s bazom korišten pri izradi aplikacije

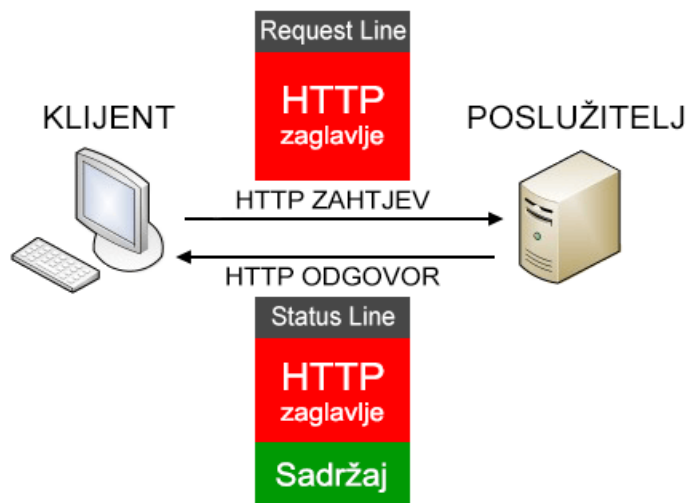
Vidimo primjer *mysql_connect* funkcije za ostvarivanje veze sa MySQL serverom, koja zahtjeva putanju do servera, ime korisnika, lozinku, naziv baze. U ovom slučaju definiran je lokalni server *localhost*, ime korisnika je zadano *root*, lozinka nije postavljena, a naziv baze je '*rezije*'. Funkcija *\$db* određuje postavljanje MySQL instrukcija upita, *die* funkcija označava ispis poruke o pogrešci uslijed greške pri izvršavanju ostalih funkcija. [2]

2.8. Apache web poslužitelj

Kako bi se na internetu moglo pristupiti web aplikacijama, potrebno je na poslužiteljskom računalu imati instaliran web poslužitelj (engl. *server*). Poslužiteljem se naziva fizičko računalo koje na sebi ima instaliran web poslužitelj, to jest program koji obrađuje HTTP zahtjeve od klijenata i vraća natrag odgovor u obliku web stranice.

Danas je Apache web poslužitelj instaliran na većini poslužiteljskih računala koja služe za distribuiranje web sadržaja. Kao jedan od ključnih razloga je to što je kompatibilnost s Linux operacijskim sustavom i PHP programskim jezikom vrlo dobra i kombinacija ta 3 servisa se pokazala kao jedno od najčešće korištenih rješenja kada su u pitanju web aplikacije. Popularnosti Apache-a pridonosi i činjenica da je besplatan za instalaciju na svim operacijskim sustavima, otvorenog je koda i dostatan je za sve oblike web stranica od manjih do velikih i kompleksnih CMS (engl. *Content management system*) rješenja. Apache se uvijek izvršava u pozadini, podržava više zadatkovni način rada i istovremeno može obrađivati više zahtjeva. Ovisno o potrebama web aplikacije koja će se izvršavati Apache podržava veliki broj modula koji se mogu naknadno instalirati i poboljšati ili prilagoditi rad web aplikacije.

Za instalaciju Apache servisa nije potrebno napredno znanje programiranja, sve njegove postavke su u tekstualnim datotekama koje je moguće mijenjati i uređivati. Apache se konfigurira tako da „sluša“ HTTP promet na određenom portu poslužitelja, a većinom je to port 80 za HTTP protokol. Apache obrađuje zaglavlje zahtjeva koje je poslao web preglednik. Na slici 2.6. prikazan je shematski i pojednostavljeni prikaz komunikacije između klijenta i poslužitelja.



Sl. 2.6. Shematski prikaz komunikacije između klijenta i poslužitelja.

Prema slici 2.7. vidimo HTTP zaglavlje kada pristupimo web stranici Elektrotehničkog fakulteta <http://www.etfos.unios.hr>

```

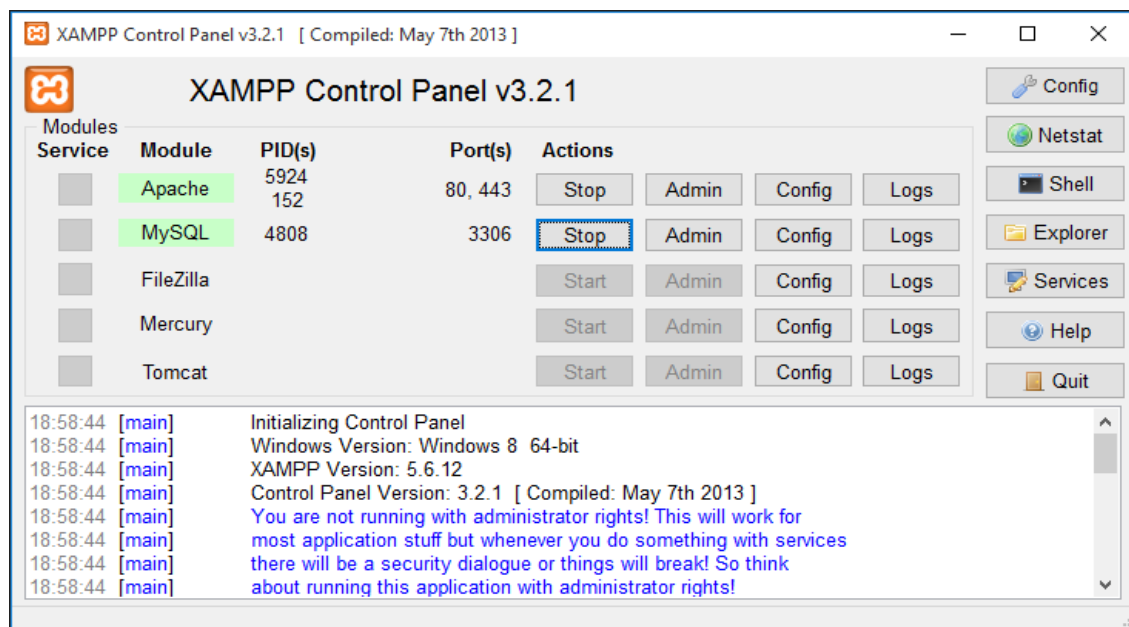
Request URL: http://www.etfos.unios.hr/
Request method: GET
Remote address: 161.53.201.4:80
Status code: ● 200 OK
Version: HTTP/1.1
Edit and Resend Raw headers
Filter headers
Response headers (0,432 KB)
Cache-Control: "no-store, no-cache, must-revalidate, post-check=0, pre-check=0"
Connection: "Keep-Alive"
Content-Type: "text/html; charset=UTF-8"
Date: "Thu, 30 Jun 2016 15:57:33 GMT"
Expires: "Thu, 19 Nov 1981 08:52:00 GMT"
Keep-Alive: "timeout=5, max=100"
Pragma: "no-cache"
Server: "Apache/2.2.22 (Debian)"
Set-Cookie: "PHPSESSID=37sdf9bk051n5lc2s7s1dn1pj1; path=/"
Transfer-Encoding: "chunked"
X-Powered-By: "PHP/5.4.45-0+deb7u4"
Request headers (0,286 KB)
Host: "www.etfos.unios.hr"
User-Agent: "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:47.0) Gecko/20100101 Firefox/47.0"
Accept: "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"
Accept-Language: "en-US,en;q=0.5"
Accept-Encoding: "gzip, deflate"
Connection: "keep-alive"

```

Sl. 2.7. Primjer HTTP zaglavlja učitano s Mozilla Firefox preglednikom

2.9. XAMPP

XAMPP je serverski programski paket koji se temelji se na otvorenom kodu, dostupan je za besplatno preuzimanje u verzijama za rad na različitim operacijskim sustavima, u ovom slučaju Microsoft Windows 8 OS. Namijenjen je za upotrebu na lokalnoj mreži, a ne kao web poslužitelj. Sastoji se od: Apache server, MySQL sustav, PHP. XAMPP instalira sve zajedno s konfiguracijskim modelima i postavkama. Osim toga u instalaciju su uključeni i FTP server, mail server, podrška za FileZilla, PhpMyAdmin korisničko sučelje koje služi za upravljanje bazama podataka i mnoge druge opcije. XAMPP pomoću kontrolne ploče (engl. *Control Panel*) po potrebi može pokretati, konfigurirati i administrirati Apache server, MySQL i mnoge druge programe koji su dio XAMPP paketa.




Sl. 2.8. Upravljačka ploča XAMPP programskog paketa

Nakon pokretanja XAMPP programskog alata, korisničkom sučelju se pristupa putem Internetskog preglednika. Upisom adrese <http://localhost/xampp> u Internetskom pregledniku dolazimo do početne stranice XAMPP korisničkog sučelja. Korisničko sučelje sadrži mogućnost detaljne konfiguracije za napredni rad, mogućnost konfiguracije pojedinih servera, mogućnost sigurnosne postavke kojima se kreiraju ovlasti za pristup XAMPP korisničkom sučelju, bazama podataka te ostalim pojedinim elementima.

Za pravilan rad aplikacije svi direktoriji, stranice i skripte koje se izvode spremaju se u instalacijski direktorij XAMPP-a na lokaciju *C:\xampp\htdocs*.


Pri razvoju aplikacije korišten je XAMPP v5.6.12 programski paket, koji se sastoji od Apache v2.4.16, PHP v5.6.12, MySQL v5.0.11 i PhpMyAdmin v4.4.14. [8]

Primjer pokretanja *phpinfo.php* stranice na adresi *http://localhost/dashboard/phpinfo.php* u Web pregledniku se vidi status, verzija, sustav PHP-a.

PHP Version 5.6.12


System	Windows NT DESKTOP-ENK4FF8 6.2 build 9200 (Windows 8 Professional Edition) i586
Build Date	Aug 6 2015 11:58:38
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x86
Configure Command	<pre> cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=c:\php-sdk\oracle\86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo" </pre>
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,TS,VC11
PHP Extension Build	API20131226,TS,VC11
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, sslv3, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	convert.iconv.*, mcrypt.*, mdecrypt.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*, bzip2.*

This program makes use of the Zend Scripting Language Engine:
 Zend Engine v2.6.0, Copyright (c) 1998-2015 Zend Technologies



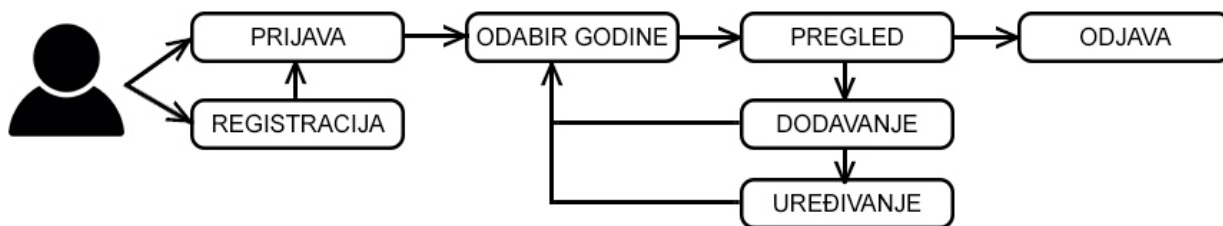
Sl. 2.9. Prikaz informacija o PHP verziji i parametrima

III. REALIZACIJA SUSTAVA

Detaljan razvoj programskog dijela aplikacije bit će objašnjen nakon predstavljanja razvojnog okruženja i alata koje koristimo te predstavljanja osnovnih tehnologija. U samome programskom dijelu aplikacije razlikujemo cjeline od koje svaka ima određene funkcije koje rade sa podacima, a koji se nalaze u bazi podataka. Objasnit će se razvoj dijelova za prijavu, registracijom, odjavu, razvoj korisničkog sučelja, spremanje i uređivanje podataka, te predstaviti kreiranje baze podataka sa pripadajućim tablicama i funkcionalnost aplikacije.

3.1. Struktura i princip rada aplikacije

Struktura i princip rada web aplikacije će se opisati u ovom potpoglavlju. Prilikom pokretanja i pristupanja aplikacije pomoću internetskog preglednika dolazimo do prve početne stranice na kojoj se nalazi forma za upis podataka, odnosno e-mail i lozinka potrebna za prijavu u sustav. Ako se korisnik nije najprije registrirao, odabirom opcije za registraciju otvara se nova forma za unos osnovnih podataka o korisniku te ih je potrebno sve popuniti i registrirati se. Nakon što korisnik odabere opciju *'Registriraj'*, podaci prolaze kroz provjeru i upis u bazu podataka. Ukoliko je netočno ispunjena forma ili postoji korisnik s istim podacima, sustav obavještava korisnika porukom te se upućuje na ponovni upis osnovnih podataka za registraciju. Ako je registracija novog korisnika uspješno registrirana, sustav će obavijestiti korisnika da je registracija uspješna te će ga preusmjeriti na početnu stranicu s prijavom u web aplikaciju. Prijavom na početnoj stranici, korisniku se otvara glavno sučelje. U tom dijelu se nalaze osnovni podaci o korisniku koje je unio prilikom registracije. Glavni dio sučelja je tablica s popisom režija i prikazanim svim mjesecima u tekućoj godini. Odabirom opcije *'Unos novih režija'* otvara se nova stranica s formom za unos novih računa, odnosno cijena režija za određeni mjesec i godinu. Nakon spremanja tablica s popisom režija se automatski popunjava i izračunava se ukupni zbroj troškova koje korisnik treba platiti za tekući, prošli ili bilo koji drugi mjesec u godini. Ukoliko je korisnik pogriješio pri unosu za određenu stavku ili nije u tom trenutku imao cijenu režije, moguće je urediti naknadno odabirom opcije za uređivanje. Spremljene vrijednosti će ispisati u formama da bi korisnik mogao promijeniti samo određenu stavku. Na taj način korisnik će moći pratiti i vidjeti troškove za svaku stavku po mjesecu i kroz cijelu godinu automatskim izračunom u preglednoj tablici.



Sl. 3.1. Prikaz strukture web aplikacije

3.2. Korisničko sučelje aplikacije

Nakon početne prijave u web aplikaciju prikazuje se korisničko sučelje s glavnom tablicom i popisom režija po mjesecima za tekuću godinu. Razvoj korisničkog sučelja zasnovano je na dva dijela, a to je HTML i CSS. Pomoću CSS-a definiramo pravila u stilskom obrascu i povezujemo stilska pravila sa HTML kodom. U aplikaciji se koristi CSS datoteka *style.css* za dizajn web aplikacije.

Pri izradi aplikacije korištena su dva načina za oblikovanje, *class* (.) atribut te *id* (#) atribut. Kod atributa *id* postoji mogućnost samo jednog pojavljivanja na stranici, dok se atribut *class* može pojaviti i više puta. Još jedna razlika je u tome što atribut *id* se može odnositi na samo jednu stavku elementa kao što je npr. oblikovanje gumba, a kod atributa *class* imamo mogućnost navesti, uz pomoć samo jedne reference, cijelu vrstu elementa.

🏠 Moje Režije
Pozdrav Zeljko, [Odjava](#)

Podaci o korisniku:
 Korisnik: Zeljko Filipovic
 Adresa: Kneza Trpimira 2B, Osijek

PREGLED REŽIJA

Prikaz za godinu: 2016.

odabir godine ▾

Režije/Mjesec	1 📅	2 📅	3 📅	4 📅	5 📅	6 📅	7 📅	8 📅	9 📅	10 📅	11 📅	12 📅	Ukupno godišnje
Struja	213.00	123.80	100.00	129.66	129.00	152.90	200.10	199.00	173.00	123.40	1456.00	189.00	3188.86 kn
Voda	74.05	78.00	125.00	66.00	112.00	62.89	66.00	78.00	73.33	67.20	115.00	87.70	1005.17 kn
Komunalije	28.70	46.90	150.00	46.00	67.00	76.00	34.00	55.00	55.70	55.00	63.00	66.50	743.80 kn
Plin	0.00	0.00	99.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.00 kn
HRT	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	960.00 kn
Pričuva	96.25	100.00	99.00	99.00	90.89	90.89	100.00	90.89	90.89	90.89	90.89	100.00	1139.59 kn
Čišćenje	30.00	35.00	46.00	30.00	25.00	30.00	25.00	25.00	30.00	25.00	25.00	30.00	356.00 kn
Telefon	263.64	145.98	193.00	222.00	166.78	245.98	134.80	175.00	224.80	213.10	166.98	234.80	2386.86 kn
= UKUPNO	785.64 kn	609.68 kn	892.00 kn	672.66 kn	670.67 kn	738.66 kn	639.90 kn	702.89 kn	727.72 kn	654.59 kn	1996.87 kn	788.00 kn	9879.28 kn

UNOS NOVIH REŽIJA


© 2016. Režije | by Željko Filipović

Sl. 3.2. Dizajn korisničkog sučelja web aplikacije

Sučelje se sastoji od zaglavlja (engl. *header*) s logotipom i nazivom web aplikacije „Moje Režije“ na lijevoj strani te na desnoj strani ime trenutno prijavljenog korisnika i link za odjavu iz sustava. Ispod zaglavlja se nalazi osnovni podaci prijavljenog korisnika koje je unio prilikom registracije. Ispod osnovnih podataka o korisniku se nalazi informacija o trenutnom prikazu za godinu i izbor prethodne i sljedeće godine. Glavni sadržaj je tablica s popisom režija i cijenama za svaki mjesec, a dolje je automatski izračun sume za svaki mjesec. Desni stupac izračunava ukupnu godišnju sumu za svaku režiju te ukupnu sumu svih režija za cijelu godinu.

3.3. Razvoj sustava prijave i registracije u aplikaciju

Web aplikacija za vođenje režija za kućanstvo je zamišljena kao jednostavna aplikacija kojom se pristupa preko Internetskog preglednika, te kada korisnik otvori početnu stranicu vidi sučelje za prijavu u aplikaciju, a ukoliko korisnik nije registriran, može se registrirati kako bi njegovi podaci bili pohranjeni u bazi podataka za sljedeću prijavu. Na slici 3.3. prikazana je početna stranica web aplikacije s HTML formom za prijavu.



Sl. 3.3. Početna stranica s formom za prijavu u aplikaciju

Kada se upiše e-mail i lozinka u formu te klikom gumba 'Logiraj se', pokreće se PHP skripta koja služi za provjeru i spremanje podataka u MySQL bazu. Da bi se podaci uspješno spremili u bazu podataka, moraju biti zadovoljeni svi uvjeti. Na slici 3.4. prikazan je primjer koda iz PHP skripte na početnoj stranici na kojima su prikazani postavljeni uvjeti.

Uvjeti koji moraju biti zadovoljeni da bi se podatci spremili u sustav su ispravno unesena e-mail adresa i lozinka. Ti uvjeti su definirani *'if'* petljom. Ukoliko oba polja nisu ispunjena ili su podaci neispravni odnosno ne podudaraju se, pokreće se skripta koja javlja grešku. Greška se otvara u novom prozoru u Internetnom pregledniku, te je potrebno stisnuti gumb *'Ok'* da bi se vratili na početak (Sl. 3.5.).

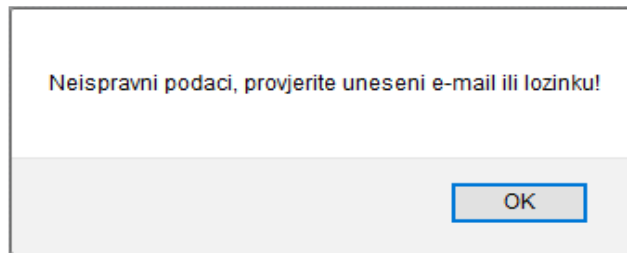
```
<?php
session_start();
include_once 'dbconnect.php';

if(isset($_SESSION['user'])!="")
{
    header("Location: home.php");
}

    if(isset($_POST['button-login']))
    {
        $email = $_POST['email'];
        $upass = $_POST['pass'];
        $res=mysql_query("SELECT * FROM korisnici WHERE email='$email'");
        $row=mysql_fetch_array($res);
        if($row['password']==md5($upass))
        {
            $_SESSION['user'] = $row['user_id'];
            header("Location: home.php");
        }
        else
        {
            ?>
            <script>alert('Neispravni podaci, provjerite uneseni e-mail ili
            lozinku!');
            </script>
            <?php
        }
    }
    ?>
```

Sl. 3.4. PHP kod s postavljenim uvjetima za prijavu u aplikaciju

U kodu se vidi upit koji provjerava postoji li korisnik odnosno e-mail i lozinka pohranjeni u bazi podataka. Ako postoji, pokreće se sesija (engl. *Session*) koja pamti podatke, odnosno *'user_id'* i sprema informaciju o korisniku koji se spaja u aplikaciju. Da bi se nastavilo korištenje aplikacije provjerava se postoji li korisnik, ukoliko je prijavljen nastavlja se s radom, a ukoliko nije otvara se početna stranica s prijavom.



Sl. 3.5. Poruka s informacijom o grešci prilikom prijave u aplikaciju

Klikom na link *'Registriraj se ovdje'* otvara se nova stranica s HTML formom za registraciju. Za uspješnu registraciju je potrebno unijeti sve stavke koje su navedene te se klikom na *'Registriraj'* podaci spremaju u bazu podataka. Nakon registracije ponovno se otvara početna stranica, potrebna je prijava putem e-maila i lozinke koje je korisnik unio pri registraciji. Prema slici 3.6. prikazana je stranica s HTML formom za registraciju na web aplikaciju.

A screenshot of a web page titled "Moje Režije" with a house icon. The page contains a registration form with the following fields: "Ime", "Prezime", "e-mail", "Grad", "Adresa", and "Lozinka". Below the fields is a note: "*Molim popunite svaku stavku." At the bottom, there is a dark grey button labeled "REGISTRIRAJ" and a blue link labeled "Logiraj se ovdje".

Sl. 3.6. Stranica s formom za registraciju

```

<?php
session_start();
if(isset($_SESSION['user'])!="")
{
    header("Location: home.php");
}
include_once 'dbconnect.php';

if(isset($_POST['button-reg']))
{
    $uname = $_POST['uname'];
    $sname = $_POST['sname'];
    $email = $_POST['email'];
    $grad = $_POST['grad'];
    $ulica = $_POST['ulica'];
    $upass = md5($_POST['pass']);

if(mysql_query("INSERT INTO korisnici(ime,prezime,email,grad,ulica,password)
VALUES ('$uname','$sname','$email','$grad','$ulica','$upass)"))
{
    ?>
        <script>alert('Uspješna registracija. Logiraj se za početak rada.');
window.location = 'index.php';</script>
    <?php
}
else
{
    ?>
        <script>alert('Greška prilikom registracije. Molim pokušajte s drugim
podacima.');
    <?php
}
}
?>

```

Sl. 3.7. PHP kod za registraciju novog korisnika u bazu podataka

U kodu se vide varijable, odnosno podaci koji će se spremati u bazu podataka, u tablicu *'korisnici'*. Lozinka će se pohraniti u sigurnom kodiranom MD5 algoritmu. Isto kao i kod prijave u aplikaciju, ukoliko se ne popuni cijela forma ili već ima registrirani korisnik s istim podacima, pokreće se skripta s greškom. Ako je sve ispravno upisano, skripta će se pokrenuti s porukom da je registracija uspješna te će se pokrenuti početna stranica za prijavu u aplikaciju.

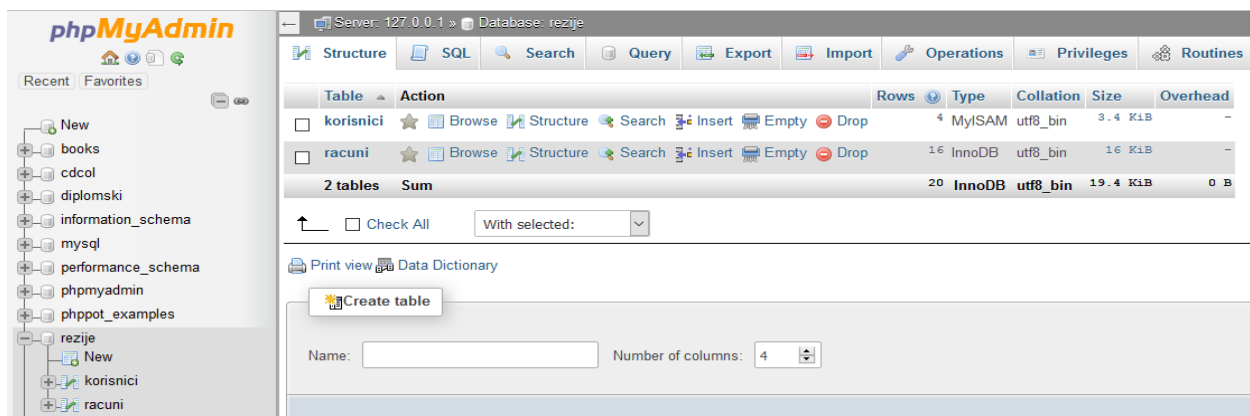
Prije napuštanja aplikacije potrebno je izvršiti odjavu klikom na *'Odjavi se'* u sučelju aplikacije te se sesija s pohranjenom informacijom zaustavlja. Nakon toga se ponovno vraća na početnu stranicu s prijavom.

3.4. Razvoj funkcija za rad s bazom podataka

Kako bi web aplikacija mogla raditi s korisnicima i podacima te primjenjivati funkcije unutar web aplikacije moramo kreirati bazu podataka. Nakon definiranja naziva baze podataka potrebno je kreirati tablice unutar kojih će se pohranjivati podaci. Baza podataka dizajnirana je pomoću phpMyAdmin korisničkog sučelja koji je instaliran s XAMPP serverskim programskim paketom.

Sve unose, promjene i čitanje iz baze će se obavljati pomoću PHP skripti integriranim u web aplikaciji.

Spajanje na phpMyAdmin korisničko sučelje se vrši preko Internetskog preglednika upisom adrese *localhost/phpmyadmin*. Prije toga je potrebno pokrenuti XAMPP serverski programski paket i iz njegove kontrolne ploče pokrenuti Apache server i MySQL. Na slici 3.8. prikazano je phpMyAdmin korisničko sučelje.



Sl. 3.8. phpMyAdmin korisničko sučelje

Za razvoj aplikacije kreirana je baza podataka 'rezije' koja sadrži dvije tablice 'korisnici' i 'racuni'. U tablici 'korisnici' se spremaju korisnički podaci kao što su ime, prezime, e-mail, grad, ulica, lozinka i jedinstveni *user_id*. U tablici 'racuni' nalaze se elementi: *id*, *id_korisnik*, *struja*, *voda*, *komunalije*, *plin*, *hrt*, *pricuva*, *ciscenje*, *telefon*, *mjesec*, *godina*. Tablica 'racuni' je povezana s tablicom 'korisnici' preko *id_korisnik* kako bi svaki korisnik imao uvid u svoje režije.

	#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1	user_id	int(5)			No	None	AUTO_INCREMENT
<input type="checkbox"/>	2	ime	text	utf8_bin		No	None	
<input type="checkbox"/>	3	prezime	text	utf8_bin		No	None	
<input type="checkbox"/>	4	email	varchar(35)	utf8_bin		No	None	
<input type="checkbox"/>	5	grad	text	utf8_bin		No	None	
<input type="checkbox"/>	6	ulica	text	utf8_bin		No	None	
<input type="checkbox"/>	7	password	varchar(50)	utf8_bin		No	None	

Sl. 3.9. Prikaz tablice 'korisnici'

U tablici 'racuni' na slici 3.10. se vidi kako su stavke režija ograničene na dvije decimale, tako da maksimalna vrijednost ne može biti veća od 9999,99.

	#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1	id	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/>	2	id_korisnik	int(11)			No	None	
<input type="checkbox"/>	3	struja	decimal(6,2)			No	None	
<input type="checkbox"/>	4	voda	decimal(6,2)			No	None	
<input type="checkbox"/>	5	komunalije	decimal(6,2)			No	None	
<input type="checkbox"/>	6	plin	decimal(6,2)			No	None	
<input type="checkbox"/>	7	hrt	decimal(6,2)			No	None	
<input type="checkbox"/>	8	pricuva	decimal(6,2)			No	None	
<input type="checkbox"/>	9	ciscenje	decimal(6,2)			No	None	
<input type="checkbox"/>	10	telefon	decimal(6,2)			No	None	
<input type="checkbox"/>	11	mjesec	int(11)			No	None	
<input type="checkbox"/>	12	godina	int(11)			No	None	

Sl. 3.10. Prikaz tablice 'racuni'

3.4.1. Spajanje u bazu podataka

PHP skriptnim jezikom možemo pristupiti bazi podataka te mijenjati sadržaj baze podataka, odnosno u ovom slučaju upisivati podatke. Da bi se podaci mogli spremati u MySQL bazu potrebno je napraviti spajanje na bazu podataka pomoću PHP skripte koja se nalazi u datoteci *dbconnect.php*. Svako spajanje na bazu poziva se ista datoteka koja sadrži potrebnu skriptu i podatke za spajanje na bazu podataka. Naredba koja se koristi u svim datotekama za spajanje na bazu je *include dbconnect.php*. Na slici 3.11. je prikazana PHP skripta za spajanje na MySQL server.

```
<?php
if(!mysql_connect("localhost","root",""))
{
    die('DB connection problem ! --> '.mysql_error());
}
if(!mysql_select_db("rezije"))
{
    die('DB selection problem ! --> '.mysql_error());
}
?>
```









Sl. 3.11. PHP skripta za spajanje na MySQL server

Povezivanje na MySQL server koristi se funkcija *mysql_connect()*, koja ima oblik *mysql_connect(host, user, password)*, a *'host'* se odnosi na adresu servera koji je u ovom slučaju lokalan, odnosno adresa je *'localhost'*, *'user'* se odnosi na MySQL korisničko ime koji je u ovom slučaju *root* jer je na lokalnom poslužitelju, *'password'* se odnosi na lozinku vezanu uz korisničko ime MySQL baze koja nije postavljena, pa stoga je ostavljen prazan prostor. Nakon povezivanja s MySQL bazom definiranim podacima određuje se naziv baze podataka koja se koristi i ima ime *'rezije'*. Ukoliko podaci nisu točno uneseni ili dođe do greške prilikom spajanja na bazu, ispisat će se poruka greške.

3.4.2. Upis u bazu podataka

Kada u aplikaciji stisnemo gumb 'unos novih režija', otvorit će se nova stranica s HTML formom za unos iznosa računa za svaku režiju. Korisnik mora unijeti svaku stavku i to isključivo znamenke s mogućnošću dvije decimale, a ukoliko nema neku od stavki treba unijeti nulu (0). Na dnu forme će se automatski prikazati prethodni mjesec od trenutnog u tekućoj godini i tako za svaki mjesec ukoliko vodi evidenciju redovito svaki mjesec. Korisnik može odabrati isto tako mjesec koji želi i tekuću, prethodnu ili sljedeću godinu. Prema slici 3.12. se vidi forma za unos režija.

UNOS REŽIJA

	struja (kn)
	voda (kn)
	komunalije (kn)
	plin (kn)
	hrt (kn)
	pričuva (kn)
	čišćenje (kn)
	telefon/internet (kn)

Odaberi mjesec: Godina:

**Molim popunite svaku stavku. Ukoliko nemate određenu režiju upišite 0 (nula).*

SPREMI

Sl. 3.12. HTML forma za upis režija

Nakon unosa svih stavki podaci se pohranjuju klikom na gumb 'spremi'.

Na primjeru (Sl. 3.13.) možemo vidjeti prikaz forme upisivanje režija u bazi podataka.

```
<form method="post">
<table width="30%" border="0">
<tr>
<td><input type="number" id="struja" step="any" min="0" lang="nb"
name="struja" placeholder="struja (kn)" required/></td>
</tr>
<tr>
<td><input type="number" id="voda" step="any" min="0" lang="nb" name="voda"
placeholder="voda (kn)" required/></td>
</tr>
<tr>
<td><input type="number" id="komunalije" step="any" min="0" lang="nb"
name="komunalije" placeholder="komunalije (kn)" required/></td>
</tr>
<tr>
<td><input type="number" id="plin" step="any" min="0" lang="nb" name="plin"
placeholder="plin (kn)" required/></td>
</tr>
<tr>
<td><input type="number" id="hrt" step="any" min="0" lang="nb" name="hrt"
placeholder="hrt (kn)" required/></td>
</tr>
.
.
.
<tr>
<td><button type="submit" name="button">Spremi</button></td>
</tr>
</table>
</form>
```

Sl. 3.13. Dio koda HTML forme za upis režija

Da bi sva polja u formi bila popunjena vrijednostima, u svakom *input*-u se stavlja *required*.

Prvo se odabire metoda koja će koristiti pri upisu u bazu iz HTML forme, u ovom slučaju je to POST. Forma se nalazi na istoj stranici *unos.php* kao i PHP skripta koja se nalazi na početku datoteke. Na slici 3.14. može se vidjeti dio PHP skripte s varijablama.

```
$struja = $_POST['struja'];
$voda = $_POST['voda'];
$komunalije = $_POST['komunalije'];
$plin = $_POST['plin'];
$hrt = $_POST['hrt'];
$pricuva = $_POST['pricuva'];
$ciscenje = $_POST['ciscenje'];
$telefon = $_POST['telefon'];
$user = $_POST['user'];
$mjesec = $_POST['mjesec'];
$godina = $_POST['godina'];
```

Sl. 3.14. Dio PHP skripte s varijablama

Prva linija *\$struja* dobiva vrijednost polja *struja* iz forme u dijelu *name="struja"*, a tako redom i za ostale varijable. Kada se sve vrijednosti unesu počinje spremanje vrijednosti u bazu podataka upitom *INSERT* kao što je prikazano na slici 3.15.

```
        if(isset($_POST['button']))
    {
        mysql_query("INSERT INTO racuni
(id_korisnik, struja, voda, komunalije, plin, hrt, pricuva, ciscenje, telefon,
mjesec, godina)
VALUES ('$user', '$struja', '$voda', '$komunalije', '$plin', '$hrt', '$pricuva',
'$ciscenje', '$telefon', '$mjesec', '$godina')");

<script>alert('Uspješan unos! Klikni OK za vraćanje na pregled sadržaja.');
```

Sl. 3.15. MySQL upit za pohranu podataka u bazu podataka

Nakon naredbe *INSERT* koristi se naredba *INTO* koja se odnosi na tablicu u koju se spremaju podaci u ovom slučaju *racuni* s vrijednostima *VALUES*. Upit se vrši *if* petljom koja provjerava je li upit izvršen, a ako je vrijednosti se upisuju u bazu i pokreće se skripta s porukom da je unos uspješan te se preusmjerava na glavnu stranicu s prikazom tablice *home.php*. Ako upit nije izvršen pokreće se skripta s porukom greške.

3.4.3. Ispis iz baze podataka

Nakon prijave u web aplikaciju prikazuje se početna stranica s glavnom tablicom, svim režijama i njihovim vrijednostima po mjesecima u tekućoj godini. Pregled vrijednosti je moguće vidjeti ukoliko su podaci prethodno spremljeni u bazi podataka, a ako nema podataka u pojedinim poljima, ta polja ostat će prazna.

Na primjeru (Sl. 3.16.) se vidi ispis vrijednosti za prvi i drugi mjesec u 2016. godini, te automatski izračun za sve stavke mjesečno i godišnje. Za ispis vrijednosti koristi se PHP skripta sa *sql* upitom *SELECT * FROM*. Naredba *SELECT* označava da želimo odabrati podatke iz tablice, *** označava da želimo odabrati sve podatke iz tablice, *FROM* označava iz koje tablice želimo ispisati vrijednosti.

Prikaz za godinu: **2016.**

odabir godine ▾

Režije/Mjesec	1	2	Ukupno godišnje
Struja	100.00	145.78	245.78 kn
Voda	124.00	66.80	190.80 kn
Komunalije	65.00	56.00	121.00 kn
Plin	0.00	0.00	0.00 kn
HRT	80.00	80.00	160.00 kn
Pričuva	100.00	100.05	200.05 kn
Čišćenje	25.00	30.00	55.00 kn
Telefon	144.87	222.00	366.87 kn
= UKUPNO	638.87 kn	700.63 kn	1339.50 kn

Sl. 3.16. Ispis vrijednosti režija koje se nalaze u bazi

Na slici 3.17. se vidi upit za ispis vrijednosti iz baze podataka, odnosno iz tablice 'racuni'. Vidi se da se prvi upit odnosi na prvi mjesec i godinu koja je odabrana na početku. Drugim upitom se ispisuje zbroj svih režija, upitom 'SELECT sum' za prvi mjesec u godini koja je odabrana na početku.

```
$moj=mysql_query("SELECT * FROM racuni WHERE mjesec=$mj1 and godina='$god'
and id_korisnik=".$SESSION['user']);
$Row10=mysql_fetch_array($moj);
```

```
$moj=mysql_query("SELECT
sum(struja+voda+komunalije+plin+hrt+pricuva+ciscenje+telefon) as
ukupnomjesec1 FROM racuni WHERE mjesec=$mj1 and godina='$god' and
id_korisnik=".$SESSION['user']);
$Row31=mysql_fetch_array($moj);
```

Sl. 3.17. Primjer prikaza upita vrijednosti režija

Na primjeru (Sl. 3.18.) se vidi 'echo' naredba kojom se ispisuju vrijednosti na stranici.

```
<?php echo $Row10['struja']; ?>
<?php echo $Row31['ukupnomjesec1'] . $valuta; ?>
```









Sl. 3.18. Primjer prikaza ispisa vrijednosti režija

3.4.4. Uređivanje iz baze podataka

Kako bi korisniku omogućili uređivanje, odnosno promjenu spremljene vrijednosti, koristit će se naredba *'UPDATE'*. Nakon što korisnik otvori stranicu za uređivanje klikom na zelenu olovku pored broja mjeseca u glavnom sučelju, učitava se stranica s formom, ali ne praznim poljima, nego već popunjenim s vrijednostima koje su pohranjene u bazi podataka. Na primjeru (Sl. 3.19.) vidi se da je odabrana promjena za šesti mjesec 2016. godine te su ispisane vrijednosti koje su već bile spremljene u bazi podataka.

PROMJENA REŽIJA

Za 6. mjesec, godina: 2016.

	152,9
	62,89
	76
	0
	80
	90,89
	30
	245,98

SPREMI

Sl. 3.19. HTML forma za promjenu režija

Klikom na gumb *'spremi'* izmijenjene vrijednosti će biti spremljene u bazu umjesto starih vrijednosti. Koristi će se dva *'sql'* upita od kojih će prvi biti naredba *'SELECT'* s kojom ćemo odabrati režiju po mjesecu i godini kao što je prikazano u primjeru (Sl. 3.20.).

```
$moj=mysql_query("SELECT * FROM racuni WHERE mjesec = $id and godina = $god
and id_korisnik=".$_SESSION['user']);
$Row10=mysql_fetch_array($moj);
if (!$Row10)
{
    die("Greska: Nisu pronadene vrijednosti u bazi, molimo prvo
unesite rezije.");
}
```

Sl. 3.20. PHP funkcija za dohvaćanje vrijednosti

Pri promjeni vrijednosti u formi koristimo istu formu kao za upis režija samo što npr. u *'value'* vrijednosti upisujemo *<?php echo \$struja ?>*. Isto upisujemo za sve ostale režije.

Drugi upit će biti naredba *'UPDATE'*, zatim naziv tablice koja se zove *'racuni'*. *'SET'* naredba postavlja nove promijenjene vrijednosti koje su upisane u formi, a *'WHERE'* naredba označuje uvjet o kojem mjesecu i godini radi kao što je prikazano u primjeru (Sl. 3.21.).

```
if(isset($_POST['save']))
{

    $struja_edit = $_POST['struja'];
    $voda_edit = $_POST['voda'];
    $komunalije_edit = $_POST['komunalije'];
    $plin_edit = $_POST['plin'];
    $hrt_edit = $_POST['hrt'];
    $pricuva_edit = $_POST['pricuva'];
    $ciscenje_edit = $_POST['ciscenje'];
    $telefon_edit = $_POST['telefon'];
    $user_edit = $_POST['user'];
    $mjesec_edit = $_POST['mjesec'];
    $godina_edit = $_POST['godina'];

    mysql_query("UPDATE racuni SET struja='$struja_edit',voda='$voda_edit',
        komunalije='$komunalije_edit',plin='$plin_edit',hrt='$hrt_edit',
        pricuva='$pricuva_edit', ciscenje='$ciscenje_edit',
        telefon='$telefon_edit' WHERE mjesec = $id and godina = $god")
        or die(mysql_error());

    echo "<script>alert('Uspješna promjena.');

```

Sl. 3.21. PHP funkcija za promjenu podataka

IV. ZAKLJUČAK

Ovim diplomskim radom prikazan je postupak izrade i razvoj web aplikacije za vođenje režija za kućanstvo. Aplikacija je vrlo jednostavna i prilagodljiva za različite namjene. Može se vrlo lako prenamijeniti za vođenje drugih troškova.

Web aplikacija može raditi na različitim operacijskim sustavima, jedini uvjet za korištenje web aplikacije je imati pristup Internetu i imati osnovni web preglednik. Opisana je konfiguracija Apache web servera, PHP programskog skriptnog jezika, MySQL sustav za upravljanje bazom podataka te izgradnja razvojne okoline koja omogućuje uspješan rad aplikacije u lokalnom okruženju. Objašnjen je proces kreiranja i upravljanja bazom podataka te tablicama unutar njih i podacima kao što su spremanje, ispis i promjena postojećih podataka.

Omogućen je pristup svakom korisniku praćenje i evidencija troškova režija u svom domaćinstvu s ciljem uštede novca.

LITERATURA

- [1] W. Jason Gilmore, Beginning PHP and MySQL, W. Jason Gilmore, 2008.
- [2] Larry Ullman, PHP 6 and MySQL 5 for Dynamic Web Sites, Larry Ullman, 2008.
- [3] M. Zekić-Sušac, http://www.mathos.unios.hr/wp/wp2009-10/P14_Web_aplikacije.pdf
- [4] W3Schools, http://www.w3schools.com/browsers/browsers_stats.asp
- [5] W3Counter, <http://www.w3counter.com/trends>
- [6] HTML, <http://en.wikipedia.org/wiki/HTML>
- [7] CSS, http://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [8] XAMPP, <http://www.apachefriends.org>

SAŽETAK

Naslov rada: Web aplikacija za vođenje režija za kućanstvo

U ovom diplomskom radu izrađena je web aplikacija za vođenje režija za kućanstvo. Aplikacija omogućuje korisnicima registraciju, prijavu, unos novih podataka i promjenu podataka. Rezultat rada je brza i jednostavna web aplikacija koja korisniku omogućuje evidenciju potrošnje svih režija u bilo kojem trenutku s ciljem uštede. Opisana je izrada aplikacije i razvoj okoline za rad.

Ključne riječi: PHP, HTML, CSS, MySQL, web aplikacija, upis, ispis, promjena, baze podataka

ABSTRACT

Title: Web application for household bills analysis

In this work a web application for household bills analysis is made. The application allow users to register, login, enter new and edit existing data. Resulting application if fast and user-friendly. In order to save money and reduce expenses, users can track their bills in one place. Detailed description of application development is described in this work.

Key words: PHP, HTML, CSS, MySQL, Web application, input, post, update, database

ŽIVOTOPIS

Željko Filipović, rođen 1988. godine u Vukovaru. Završetkom osnovne škole upisuje srednju školu Nikole Tesle u Vukovaru, smjer elektrotehničar. 2006. godine upisuje Sveučilišni preddiplomski studij elektrotehnike na Elektrotehničkom fakultetu u Osijeku. Nakon završetka preddiplomskog studija, 2009. godine upisuje Sveučilišni diplomski studij elektrotehnike, smjer komunikacije i informatika.

Željko Filipović

PRILOZI

Cijeli program Web aplikacije nalazi se na priloženom CD-u.