Web aplikacija za evidenciju radnog vremena

Bjelčević, Dominik

Undergraduate thesis / Završni rad

2020

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://urn.nsk.hr/urn:nbn:hr:200:105108

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-21**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Stručni studij

WEB APLIKACIJA ZA EVIDENCIJU RADNOG VREMENA

Završni rad

Dominik Bjelčević

Osijek, 2020.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Osijek, 08.10.2020.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Ime i prezime studenta:	Dominik Bjelčević
Studij, smjer:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	AI4531, 24.09.2019.
OIB studenta:	16692874415
Mentor:	Robert Šojo
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Marina Peko
Član Povjerenstva 1:	Robert Šojo
Član Povjerenstva 2:	Dr. sc. Krešimir Romić
Naslov završnog rada:	Web aplikacija za evidenciju radnog vremena
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rada	Web aplikacija omogućava evidenciju radnog vremena zaposlenika, prijava, odjava, ukupan broj odrađenih sati. Na ovaj način se točno evidentira dolazak i odlazak zaposlenika, te njegova lokacija. Analiza radnog vremena - prikaz da li je zaposlenik odradio očekivani broj sati na poslu, koliko ima prekovremenih sati, a u slučaju manjka sati prikazuje koliko je sati korisnik dužan nadoknaditi kroz radni tjedan. Administrator konfigurira tjedno radno vrijeme (npr. 6 ili 8 sati), unosi zaposlenike (ime, prezime, e-mail, korisničko ime i šifru) i dodjeljuje svakom zaposleniku korisničke podatke za prijavu na web portal. Zaposlenik se treba prijaviti na web portalu, te odabrati tipku za prijavu kada dođe na posao, odjavu kada odlazi s posla. Student treba kreirati funkcionalnu web aplikaciju.
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	08.10.2020.

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, 16.10.2020.

Ime i prezime studenta:	Dominik Bjelčević
Studij:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	AI4531, 24.09.2019.
Turnitin podudaranje [%]:	9

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija za evidenciju radnog vremena**

izrađen pod vodstvom mentora Robert Šojo

i sumentora

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

1. UVOD 1
   1.1 Zadatak završnog rada 1
   1.2 Postojeće web aplikacije 2
      1.2.1 Harvest 2
      1.2.2 Toggl 3
      1.2.3 Everhour 3
      1.2.4 Timely 4
2. TEHNOLOGIJE KORIŠTENE PRI IZRADE APLIKACIJE 6
   2.1 XAMPP lokalni server 6
   2.2 HTML – HyperText Markup Language 6
   2.3 CSS – Cascading Style Sheets 7
   2.4 Bootstrap 8
   2.5 JavaScript 8
      2.5.1 jQuery 9
      2.5.2 JSON 10
   2.6 PHP 11
   2.6.1 Smarty 12
   2.7 SQL 13
      2.7.1 MySQL 13
   2.8 Visual Studio Code 14
3. KORIŠTENJE WEB APLIKACIJE 15
   3.1 Pokretanje aplikacije 15
   3.2 Početna stranica 16
   3.3 Uloga Administratora i Zaposlenika 18
   3.4 Struktura aplikacije 19
   3.5 Opis mogućnosti aplikacije 21
   3.5 Odjava 24
4. OPIS FUNKCIONALNOSTI WEB APLIKACIJE 25
   4.1 Autentifikacija zaposlenika 26
   4.2 Administrator 26
   4.3 Zaposlenik 29
5. ZAKLJUČAK 31
LITERATURA 32

SAŽETAK 33
ABSTRACT 34
ŽIVOTOPIS 35
PRILOZI 36

1. UVOD

Programi koji služe za evidenciju radnog vremena postaju sve popularniji. Omogućavaju djelatnicima lakšu organizaciju svog radnog vremena dok administratorima omogućavaju bolji uvid u radno vrijeme zaposlenika. Tema ovog završnog rada je izraditi aplikaciju za evidenciju radnog vremena. Za izradu same aplikacije korištene su razne backend i frontend tehnologije među kojima je najviše korišten PHP. U prvom dijelu aplikacije korisniku je omogućena prijava na portal. U slučaju da se na portal prijavio administrator, bit će preusmjeren na sučelje za administratora, dok u slučaju prijave zaposlenik, preusmjeren na sučelje za zaposlenika. Zaposlenik i administrator imaju različite funkcije. Zaposlenik ima samo mogućnost unosa svog radnog vremena u bazu podataka te odabir projekta na kojem radi. Administrator ima iste mogućnosti kao i zaposlenik, ali mu je omogućen i velik broj ostalih funkcionalnosti. Omogućeno mu je dodavanje projekta na kojem rade zaposlenici, može provjeriti koliko sati je odradio pojedini zaposlenik, može odrediti mjesečnu kvotu radnih sati itd.

U drugom poglavlju opisane su tehnologije koje su korištene u izradi aplikacije, u trećem poglavlju opisuju se funkcionalnosti aplikacije dok je u četvrtom poglavlju prikazano korištenje aplikacije.

1.1 Zadatak završnog rada

Za završni rad kreirana je web aplikacija koja omogućava zaposlenicima evidenciju radnog vremena te omogućava administratoru praćenje radnog vremena zaposlenika. Zaposlenicima je omogućena prijava prilikom dolaska na posao, prilikom pauze daju do znanja da trenutno ne rade, te naposljetku i odjave kada odlaze sa posla. Svakom zaposleniku biti će dodijeljena jedinstvena lozinka pomoću koje se omogućava pristup stranici. Ovakve aplikacije postaju sve učestalije u velikim poduzećima s puno zaposlenika kako bi vlasniku tvrtke ili nekom projekt manageru bilo omogućeno pratiti rad svojih zaposlenika. Aplikacija opisana u ovom radu ne zahtjeva instalaciju ili preuzimanje, potrebno je samo imati lokalni server te se pomoću njega povezati. Administratoru će biti omogućen pregled svih zaposlenika u firmi te mu je omogućeno pregledavanje kada je zaposlenik došao na posao te kada se odjavio s posla. Na kraju mjeseca administrator će biti u mogućnosti provjeriti koliko je koji zaposlenik odradio sati za cjelokupni mjesec te će mu se podaci prikazati pomoću grafa.

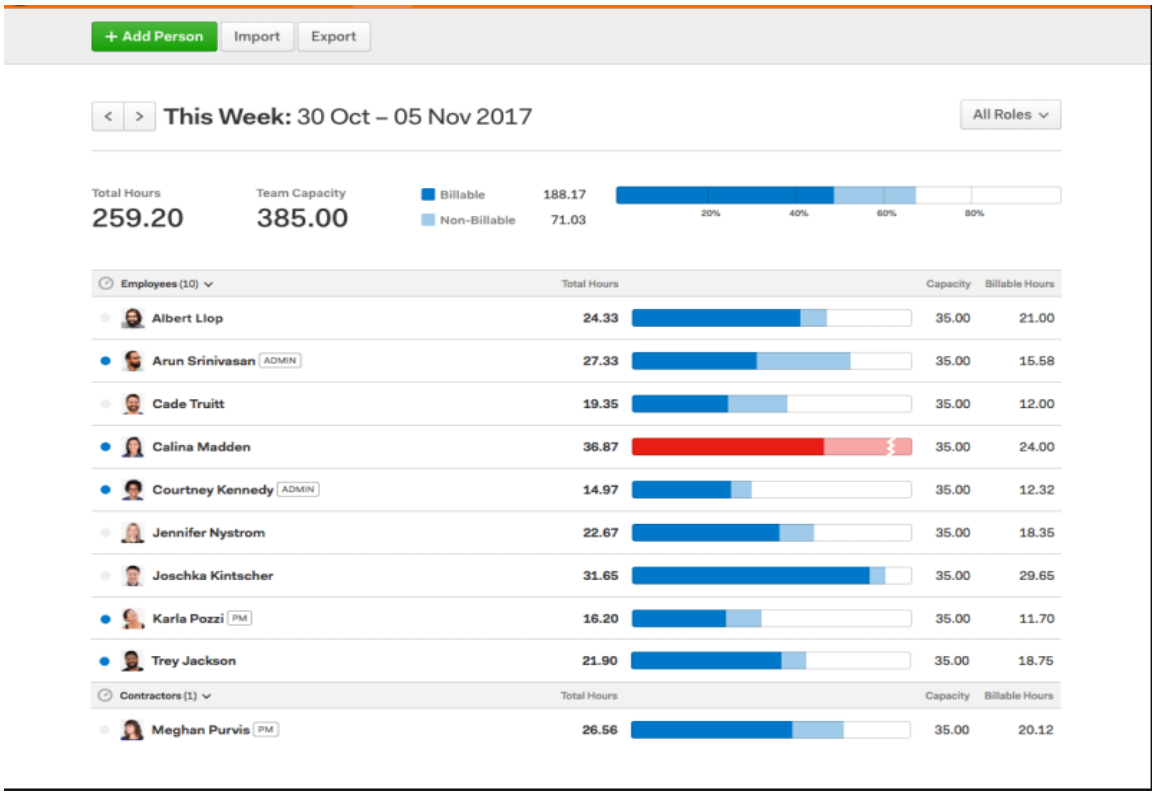
1.2 Postojeće web aplikacije

U ovom poglavlju dan je prikaz najpoznatijih i najboljih aplikacija za mjerenje radnog vremena, a to su:

- Harvest
- Toggl
- Everhour
- Timely

1.2.1 Harvest

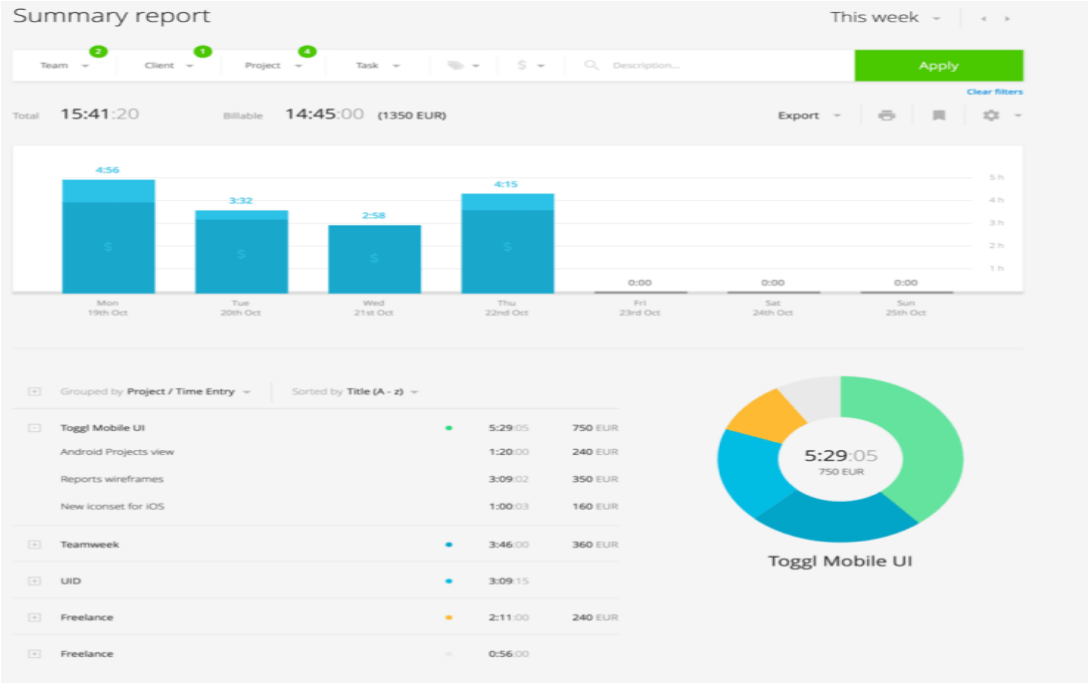
Harvest je web aplikacija za mjerenje radnog vremena zaposlenika koje je napravila tvrtka Iridesco LLC u 2006. godini. Harvest nudi, osim mjerenja radnog vremena i opcije praćenja troškova te slanje automatskih podsjetnika korisniku za plaćanje. Slika 1.1 prikazuje izgled Harvest aplikacije [1].



Slika 1.1 *Prikaz Harvest aplikacije.*

1.2.2 Toggl

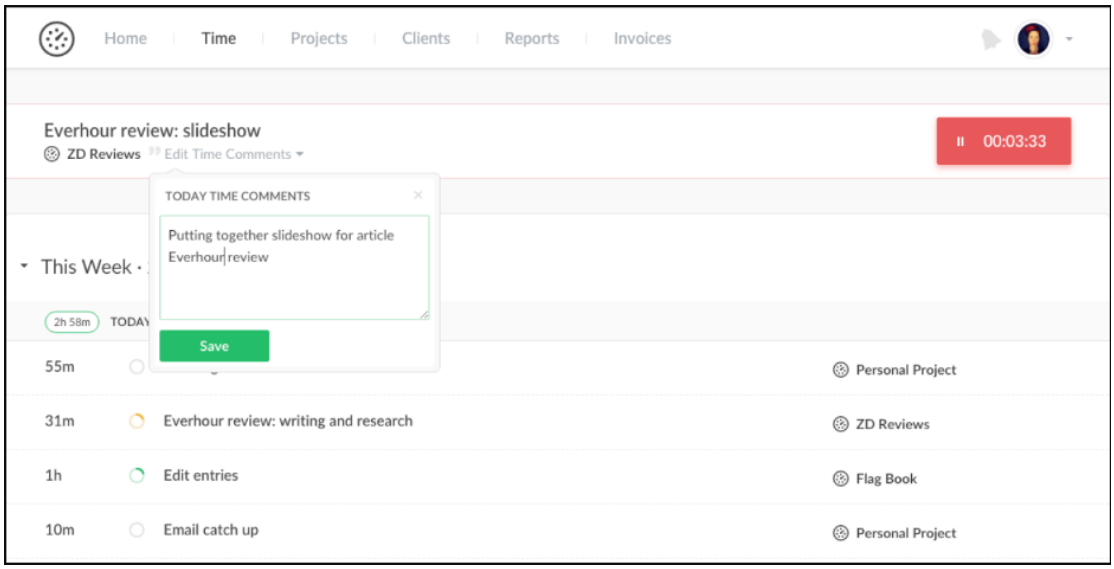
Toggl je aplikacija za mjerenje radnog vremena, te je vrlo pogodna i pristupačna aplikacija zato što svojim korisnicima nudi i besplatne servise. Također omogućen je rad aplikacije na svim platformama (Android, iOS, Linux, macOS, Windows, web). Nedostatak besplatne verzije je taj što je nemoguća suradnja s ostalim zaposlenicima već samo praćenje svog rada. U Toggle-u je moguće dodati projekt, te pratiti vrijeme provedeno na pojedinom projektu. Slika 1.2 prikazuje izgled Toggl aplikacije [2].



Slika 1.2 *Prikaz Toggl aplikacije.*

1.2.3 Everhour

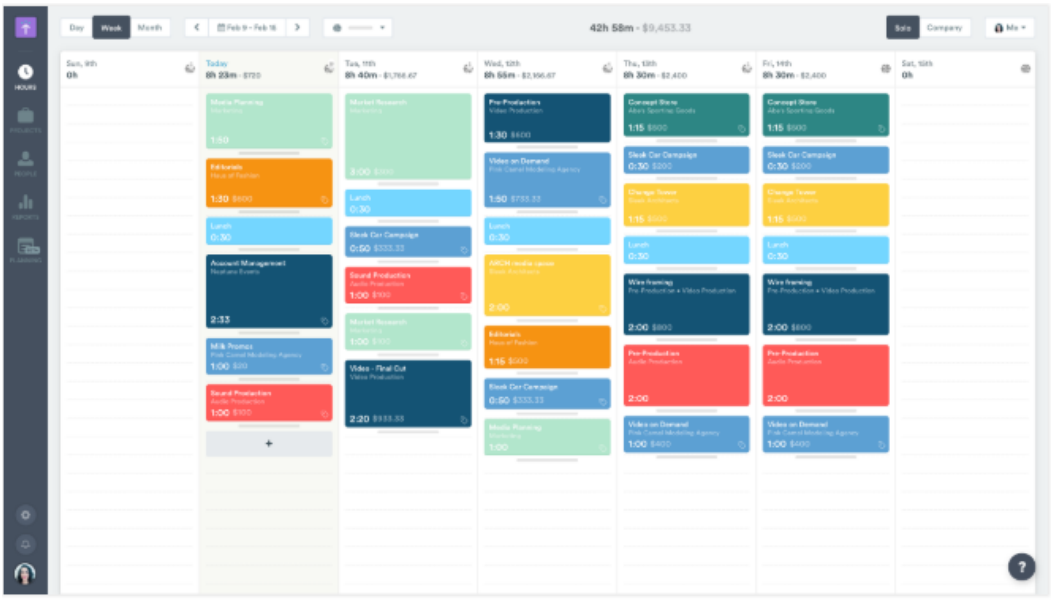
Everhour je aplikacija za praćenje radnog vremena bazirana na webu kojoj je glavna stavka to što je moguće sinkronizirati projekte iz neke druge aplikacije te raditi na njima u Everhouru. Everhour je dizajniran 2015. godine od strane Weavora Consulting LLC koja je bazirana u Bjelorusiji. Također, moguće je uključivanje svog projekta s Githuba ili sličnih Git servisa te pomoću Everhoura pratiti rad na tom projektu. Jedini nedostatak ove aplikacije je to što nema podršku za mobitele. Slika 1.3 prikazuje izgled Everhour aplikacije [3].



Slika 1.3 *Prikaz Everhour aplikacije.*

1.2.4 Timely

Timely je malo drugačiji od ostalih aplikacija jer je prikaz aplikacije u obliku kalendara. Moguće je izdvojiti projekt kao blok te ga staviti u svoj kalendar na aplikaciji. Timely stavlja planiranje i praćenje u jedno te je zbog toga jako popularan. Također, omogućena je suradnja u timovima, tako da je moguć prikaz aktivnosti i drugih korisnika na kalendaru. Nažalost, Timely nema besplatnu opciju već se svaki mjesec korištenja naplaćuje. Slika 1.4 prikazuje izgled Timely aplikacije [4].



Slika 1.4 *Prikaz Timely aplikacije.*

2. TEHNOLOGIJE KORIŠTENE PRI IZRADE APLIKACIJE

Za izradu aplikacije korištena je *backend* tehnologija PHP (engl. *Hypertext Preprocessor*) dok su za izradu *frontend* dijela aplikacije korišteni HTML (engl. *HyperText Markup Langauge*), CSS (engl. *Cascading Style Sheets*), bootstrap te JavaScript. Za bazu podataka korišten je MySQL. Lokalni server je omogućen pomoću XAMPP-a na kojem pokrećemo Apache server te MySQL bazu podataka.

2.1 XAMPP lokalni server

XAMPP je potpuno besplatan alat za stvaranje web poslužitelja i web servera. Pri izradi aplikacije, XAMPP služi kao lokalni server koji se pokreće na način tako da se u pregledniku upiše https://localhost te nakon toga ime projekta koji je lokalno spremljen u mapi htdocs. XAMPP podržava puno tehnologija i programskih jezika poput Apachea, PHP-a, Perl-a i mnogih drugih. Također podržava kreiranje i održavanje baza podataka poput MariaDB i SQL-a. Kada se pokrene XAMPP kontrolna ploča, dovoljno je samo kliknuti START na SQL server i Apache server kako bi se prikazala web aplikacija. Nakon toga dovoljno je samo putem preglednika upisati localhost te putanju do aplikacije koja je prikazana u ovom radu.

2.2 HTML – HyperText Markup Language

HTML je opisni jezik kojim se omogućava prikaz sadržaja na web stranici. HTML opisuje što će se prezentirati korisniku, ali ne i kako će se prezentirati. Prema [5] za kreiranje HTML zaslužan je Tim Berners-Lee krajem 1991. godine. Svaki HTML dokument sastoji se od HTML elemenata. Svaki element ima svoju oznaku ("tag"). HTML oznake daju pregledniku mogućnost da se to što je u oznaci prikaže na stranici. Prema [5] preglednik svaku oznaku interpretira drugačije, te se u oznakama nalazi sadržaj koji se prikazuje na stranici. Svaka oznaka ima svoj početak (<element>) i kraj (</element>) te se one nalaze u šiljastim zagradama. Osnovna građa HTML dokumenta sastoji se od sljedećih oznaka: <!DOCTYPE html>, <html>, <head>, <body>. U <head> dio se spremaju podaci poput skripte za CSS ili skripte za JavaScript te naslov koji nam se prikazuje u kartici na pregledniku. U <body> dio idu svi podaci koji se prikazuju na web stranici. Na slici 2.1 opisana je osnovna građa HTML dokumenta [6].

Slika 2.1 *Osnovna građa HTML-a.*

HTML 5 je najnoviji standard koji donosi brojne nove mogućnosti koje HTML 4 nije imao. Za kreiranje HTML dokumenta dovoljno je samo imati tekstualni uređivač. Za ovaj rad korišten je *Visual Studio Code*.

2.3 CSS – Cascading Style Sheets

CSS je stilski jezik pomoću kojega se određuje kako će izgledati neki HTML dokument. CSS nam opisuje kako će elementi HTML-a izgledati na stranici. HTML i CSS su dva najpopularnija jezika kako bi se korisniku prikazao *frontend* dio web stranice. Svaki takozvani *stylesheet* sastoji se od selektora i deklaracije. Selektor je dio kôda na koji se primjenjuje neki stil dok se deklaracije nalaze u vitičastim zagradama. Prema [7] svaka deklaracija sastoji se od svojstva, dvotočke i vrijednosti. Primjer jednog css bloka možemo zapisati kao: p {*color: red*}. Ovaj kod prikazat će paragraf p, koji je selektor, u crvenoj boji. CSS je kreirala organizacija W3C,a trenutna verzija je CSS 3.

2.4 Bootstrap

Bootstrap je besplatni razvojni okvir za CSS otvorenog koda. U njemu je sadržan CSS te u nekim slučajevima Javascript predložak za izradu različitih formi, tipki i mnogih drugih elemenata. *Bootstrap* su izradili Mark Otto i Jacob Thornton. *Bootstrap* služi za pojednostavljivanje izrade cijelog CSS-a za stranicu. Trenutna inačica *Bootstrapa* je *Bootstrap* 4.3, no *Bootstrap* 5 je najavljen kao zamjena za trenutnu verziju [8].

2.5 JavaScript

JavaScript je prvi puta predstavljen 1995. godine kako bi omogućio interaktivnu funkcionalnost web stranicama. JavaScript je omogućio razvoj modernih web aplikacija kod kojih je moguća interakcija bez da se osvježava stranica za svaku akciju. Trenutno je JavaScript sastavni dio svih internet stranica te skoro svaki moderni preglednik ima JavaScript interpreter kako bi se kôd mogao izvršiti. Iako JavaScript ima u sebi naziv „Java“ on nije ni u kojem obliku povezan sa programskim jezikom Java. U tablici 2.1 prikazana je usporedba programskog jezika Java i programskog jezika JavaScript.

JAVA	JAVASCRIPT
Objektno orijentirani programski jezik	Skriptni programski jezik
Izvršava se u JVM-u ili pregledniku	Izvršava se u pregledniku
Zahtjeva Java Development Kit (JDK)	Pisan u bilo kojem text editoru
Programski kod sakriven od korisnika	Programski kod dostupan korisniku
Android aplikacije, programiranje hardvera, analiza podataka	Koristi se za web aplikacije te za neke serverske tehnologije poput node.js

Tablica 2.1 *Usporedba Jave i JavaScripta.*

JavaScript je dominantan klijentski poslužitelj i prema [9] više od 95% stranica na internetu koristi JavaScript u tu svrhu. Većina web stranica koriste neku biblioteku ili framework kao dio svog klijentskog servera. Najpopularniji među njima je jQuery. jQuery je JavaScript biblioteka pomoću koje se omogućuje lakše upravljanje događajima, CSS animacijama te AJAX (engl. *Asynchronous JavaScript and XML*). AJAX je web tehnika pomoću koga se kreiraju asinkrone web aplikacije. Pomoću AJAX-a server asinkrono prima i šalje podatke bez da se

narušava sam izgled i ponašanje web stranice. Po posljednjim istraživanjima jQuery se koristi u više od 75% web stranica. JavaScript se u HTML stavlja pomoću <script> oznaka. Oznaka se može staviti bilo gdje u HTML dokument, ali u pravilu se stavlja u <head> područje HTML-a.

2.5.1 jQuery

jQuery je JavaScript biblioteka koja je napravljena kako bi HTML učinio jednostavnijim. jQuery je dizajniran kako bi omogućio lakše pretraživanje dokumenata, kako bi se lakše odabrali elementi, kreirale animacije te omogućava lakšu izvedbu aplikacija koje koriste AJAX. jQuery je nastao u siječnju 2006. godine. Od 2016. godine do danas, broj stranica na kojima se izvodi jQuery konstantno raste. Pomoću jQuery AJAX metoda, moguće je zatražiti text, HTML ili XML(engl. *Extensible Markup Language*) sa udaljenog servera koristeći HTTP protokol (engl. Hyperext *Transfer Protocol*). Na primjer, pomoću AJAX-a je moguće promijeniti boju teksta kada se pritisne dugme, moguće je promijeniti sadržaj teksta itd. Na slici 2.2 prikazan je primjer AJAX-a [10].

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").load("test.txt");
  });
});
</script>
</head>
    <body>

        <div id="div1"><h2>Let jQuery AJAX Change This Text</h2></div>

        <button>Get External Content</button>

    </body>
</html>
```

Slika 2.2 *Primjer AJAX-a*

Na ovom primjeru, klikom na dugme, tekst na stranici promijenit će se u tekst koji je sadržan u datoteci test.txt.

2.5.2 JSON

JSON (engl. *JavaScript Object Notation*) jedan je od tekstualnih standarda koji je dizajniran za razmjenu podataka. Prilično je popularan format te ga puno aplikacija koristi, a prema [11] obično služi kao zamjena za XML u AJAX sustavu. Prvi koji je specificirao JSON bio je Douglas Crockford sredinom 2000-ih godina. JSON tipovi podataka mogu biti kao i u svakom modernom programskom jeziku, a to su: brojevi, stringovi, boolean izrazi, nizovi i objekti. Ekstenzija JSON datoteke je .json dok je njegova meta oznaka, tzv. MIME format, application/json i većina modernih implementacija je to prihvatila. Na slici 2.3 prikazan je primjer JSON-a koji opisuje Firefox preglednik [11].

Slika 2.3 *Primjer JSON-a*

2.6 PHP

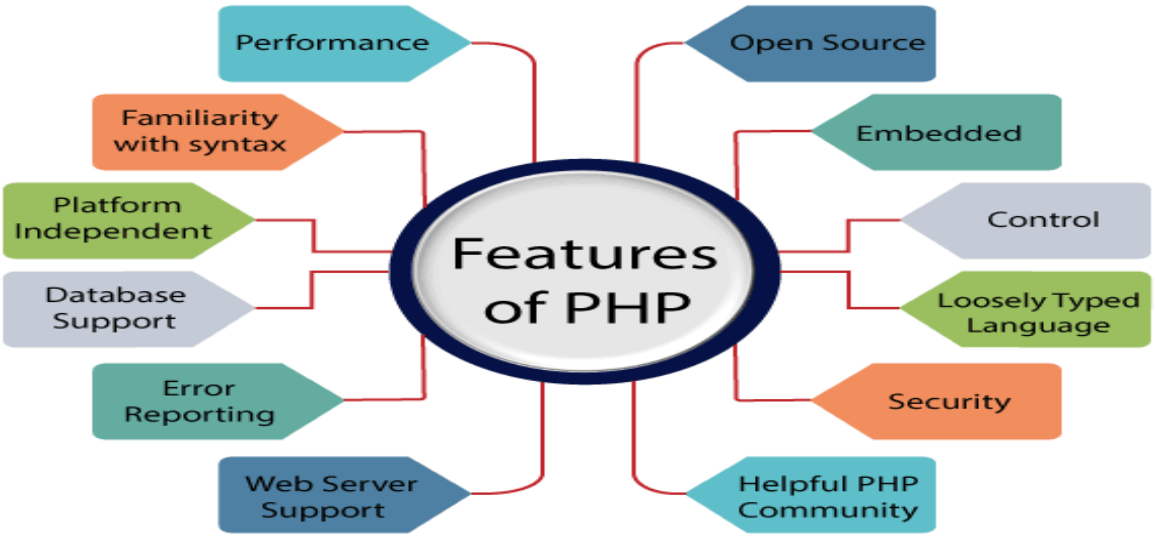
PHP je programski jezik kojemu je glavna upotreba kod web programiranja. PHP je razvijen krajem 1994. godine od strane Dansko-Kanadskog programera Rasmua Lerdorfa. Puno ime PHP-a je PHP: HyperText Preprocessor. PHP je zajedno sa JavaScriptom najpopularniji jezik za web dizajn te se uglavnom koristi za backend pri izradi web stranice. PHP je sintaksu uzeo od C programskog jezika te su neke stvari, što se tiče sintakse, identične kao u C-u. Skripte i kod koji se napiše u PHP-u ne izvršava se direktno u programskom okruženju već se oni izvršavaju putem servera. PHP kod se prevodi putem PHP interpretera. PHP se također može prikazati pomoću HTML-a. Na slici 2.4 prikazan je primjer jednostavnog PHP programa pomoću HTML-a. Prema [12] u verziji PHP5 dodan je objektno-orijentirani pristup programiranja u PHP-u što pomaže da se prave složene web aplikacije.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>PHP "Hello, World!" program</title>
5    </head>
6    <body>
7      <?php
8      echo '<p>Hello, World!</p>';
9      ?>
10   </body>
11 </html>
```

Slika 2.4 *PHP "Hello World" primjer.*

Svaki PHP program počinje sa <?, a završava sa ?>. Glavne karakteristike PHP-a i baš zbog toga su ga mnogi programeri diljem svijeta uzeli pri izradi web aplikacija je to da je efikasan, a jednostavan za korištenje. Iako je originalno zamišljen za izradu web stranica, trenutna primjena PHP-a se uglavnom koristi na serverskoj strani. Također, jedna od mnogih pogodnosti PHP-a je njegov rad s bazama podataka, a najpoznatije među njima su MySQL, PostgreSQL te SQLite. U zadnje vrijeme sve popularnija postaje LAMP arhitektura za izradu web aplikacija. LAMP (Linux, Apache, MySQL, PHP) predstavlja web servis u kojem se PHP koristi kao glavni programski jezik,

dok se za web server koristi Apache, za bazu podataka MySQL te za operacijski sustav Linux. Na slici 2.6 navedene su osnovne značajke PHP-a [13].



Slika 2.5 *Značajke PHP-a.*

Zbog svih gore navedenih značajki po zadnjim istraživanjima PHP koristi 240 milijuna web stranica te je on instaliran na više od 2 milijuna web servera. Od 2020. koristi se PHP 7 kao glavna inačica programa.

2.6.1 Smarty

Smarty je besplatna biblioteka temeljena na PHP-u koja omogućava razdvajanje poslovne logike aplikacije od njenog izgleda. Omogućava korištenje frontend dijela aplikacije neovisno o backend dijelu. Smarty omogućava programerima definiranje funkcije koje se mogu pristupiti pomoću određenih Smarty oznaka. Smarty koristi predloške koji završavaju ekstenzijom .tpl te se onda oni koriste u određenim PHP datotekama. Na slici 2.7 prikazan je primjer Smarty funkcije [14].

```
{config_load file='colors.conf'}

{include file='header.tpl'}
{insert file='banner_ads.tpl' title='Smarty is cool'}

{if $logged_in}
    Welcome, <font color="{#fontColor#}">{$name}!</font>
{else}
    hi, {$name}
{/if}

{include file='footer.tpl' ad=$random_id}
```

Slika 2.6 *Primjer Smarty kôda.*

2.7 SQL

SQL (engl. *Structured Query Language*) je poseban jezik koji se koristi u programiranju kod izrade i pri radu s bazama podataka. SQL ima dvije glavne stavke koje radi bolje od svojih prethodnika, a to su da se podaci mogu dohvatiti pomoću samo jedne naredbe, dok je druga stavka ta da nije potrebno izričito navesti kako će se neki podatak dohvatiti. Pomoću SQL-a moguće je manipulirati s podacima pomoću *INSERT, UPDATE I DELETE* naredbe, moguće je izrađivati shemu baze te je također moguć i rad s podacima koji se nalaze u toj bazi podataka. SQL je razvijen u IBM-u početkom 1970-ih te je danas najpopularniji jezik za rad s bazama podataka. Najvažniji pojam pri izrade baze u SQL okruženju su relacije. Relacije se mogu podijeliti na relacije jedan prema jedan, jedan prema više te više prema više. Sve baze podataka svoje podatke pohranjuju unutar tablica, pa tako i SQL. Tablice se sastoje od stupaca i redova, te se u kolone upisuje atribute pojedinog reda [15].

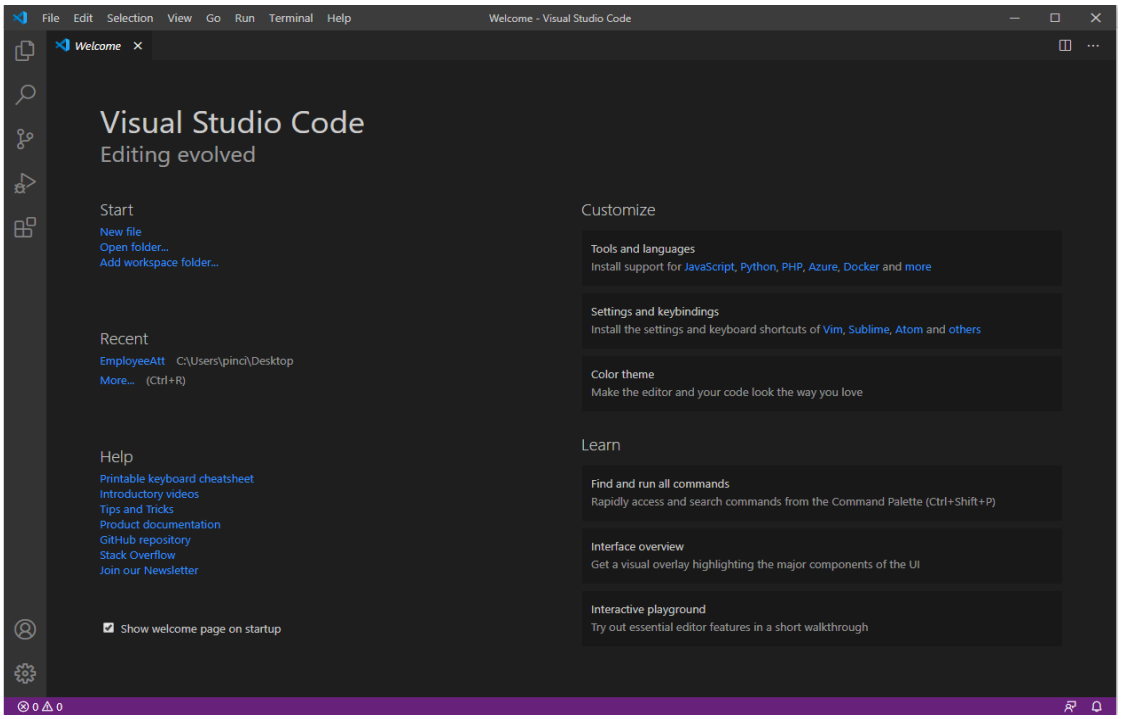
2.7.1 MySQL

U ovom završnom radu, korišten je MySQL kao inačica SQL jezika. MySQL je besplatan, sustav otvorenog kôda (engl. *open source*) za upravljanje bazom podataka. MySQL je trenutno najčešći izbor za izradu baze podataka. Uz PostgreSQL najstabilnija je inačica SQL jezika.

Također, MySQL ima podršku od brojnih programskih jezika, kao što su PHP, Java, Python i mnogi drugi te je baš zbog toga popularna [16].

2.8 Visual Studio Code

U ovom radu kao razvojno okruženje korišten je Visual Studio Code. Visual Studio Code je uređivač teksta za izvorni kôd koji se instalira na korisnikovo računalo. Na slici 2.7 prikazan je izgled početne stranice Visual Studio Code-a.



Slika 2.7 *Početni izgled Visual Studio Code-a.*

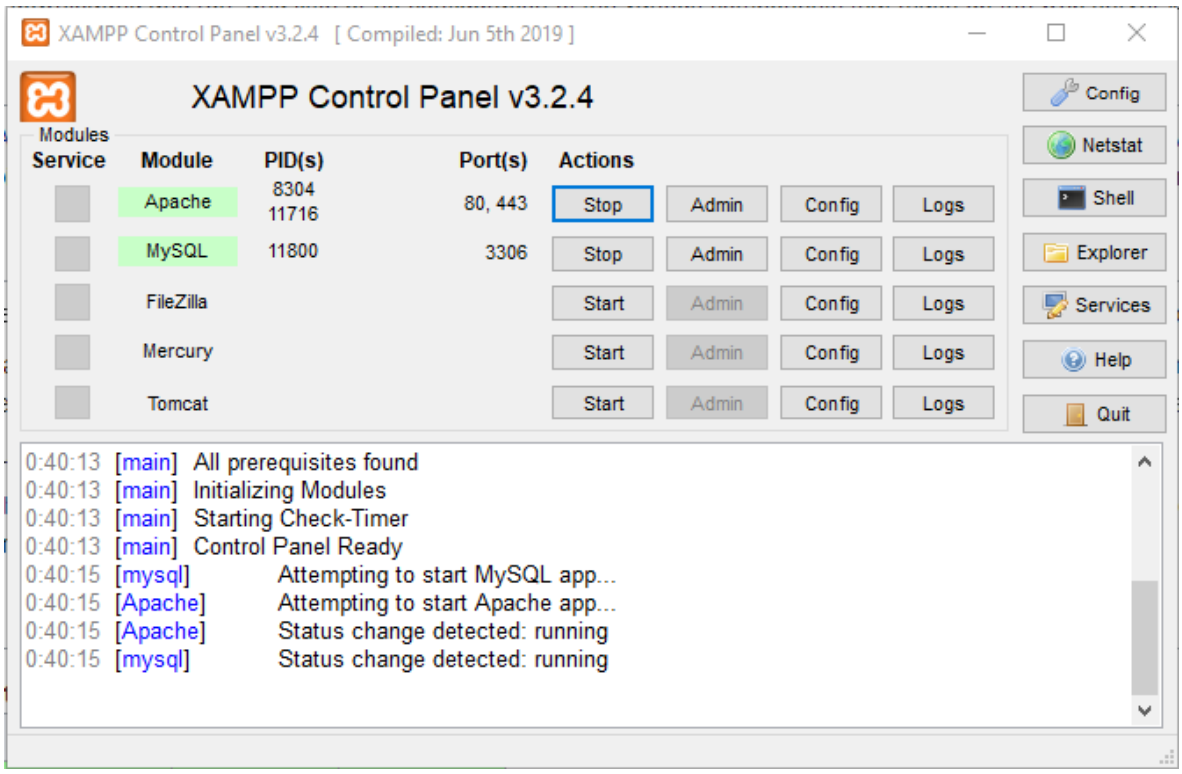
Visual Studio Code je besplatni editor za programere koji je napravio Microsoft te je podržan na sva 3 osnovna operacijska sustava (Windows, Linux, MacOS). Po zadnjoj anketi iz 2019. godine na *Stack Overflowu,* Visual Studio Code je najpopularniji editor [17]. Jedna od njegovih najvećih prednosti u odnosu na ostale editore je to što podržava podršku za mnoštvo programskih jezika. Također omogućava i instaliranje proširenja (engl. *extensions*) za programske jezike koje uvelike olakšavaju rad.

3. KORIŠTENJE WEB APLIKACIJE

Izrađena stranica sastoji se od više stranica i pogleda. Omogućen je pogled od strane administratora te od strane zaposlenika. Administrator ima neke napredne funkcionalnosti u odnosu na običnog zaposlenika, npr. može pregledavati koliko sati je koji zaposlenik odradio, može postavljati koliko sati zaposlenik treba odraditi kroz radni mjesec itd. Zaposlenik je u mogućnosti samo pregledati svoje sate te prijaviti i odjaviti samo svoje radno vrijeme. U nastavku rada opisane su funkcionalnosti same aplikacije.

3.1 Pokretanje aplikacije

Samo pokretanje aplikacije obavlja se pomoću programa XAMPP. XAMPP je besplatni server pomoću kojeg se pokreće Apache Web Server i SQL baza podataka koja se koristi u ovom radu. Da bi se sama aplikacija pokrenula, ona mora biti spremljena u htdocs datoteci te se ona pokreće pomoću web preglednika. Na slici 3.1 prikazana je XAMPP aplikacija na kojoj su pokrenute sve potrebne funkcionalnosti kako bi aplikacija opisana u ovom radu bila u funkciji.

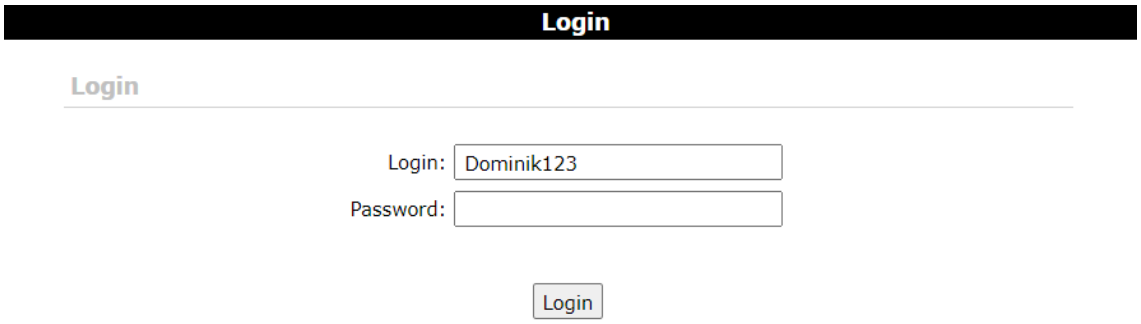


Slika 3.1 *XAMPP aplikacija sa upaljenim Apache serverom i MySQL bazom podataka.*

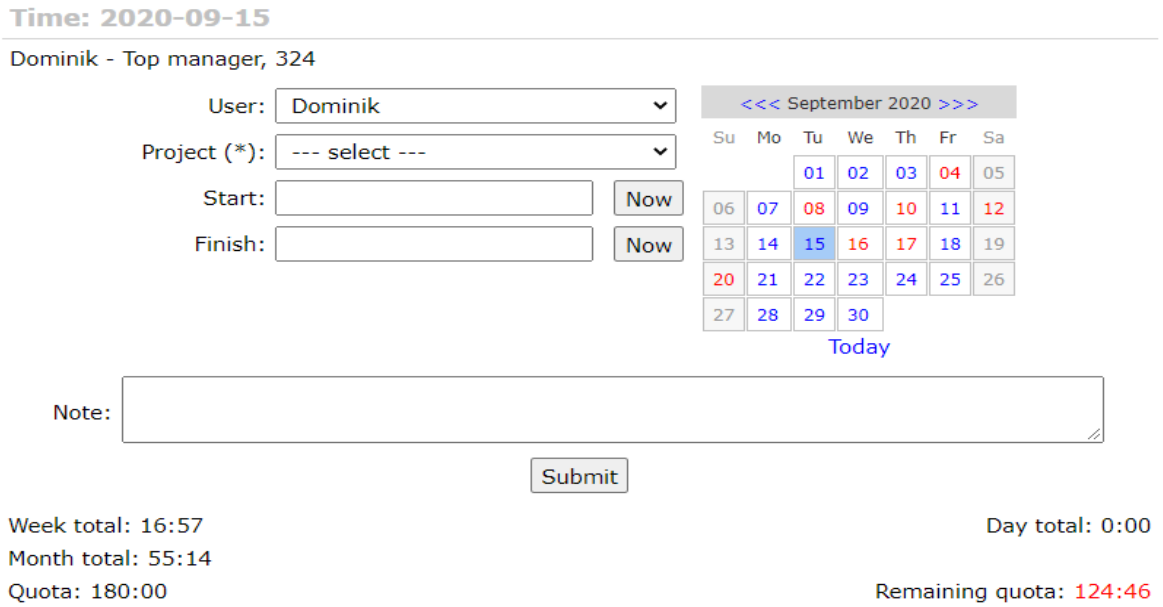
Nakon toga se u web pregledniku pokreće lokalna adresa na kojoj je spremljena aplikacija. Web aplikacija se pokreće tako da u web preglednik upisujemo localhost te nakon toga ime datoteke gdje je spremljena aplikacija (npr. localhost/timetracker).

3.2 Početna stranica

Izgled početne stranice prikazan je na slici 3.2 te omogućava opciju prijave kao zaposlenik i kao administrator. To je omogućeno zbog toga što zaposlenik neće imati sve opcije koje ima i administrator. Administratoru će biti omogućeno dobiti izvještaje svih zaposlenika. Administrator će na temelju izvještaja moći vidjeti koliko je sati zaposlenik odradio i koliko sati još treba odraditi za određeni mjesec. Zaposlenik će imati opciju uračunavanja svojih sati u bazu podataka i da vidi koliko je sati odradio na dnevnoj bazi te će onda od administratora dobiti izvještaj koliko sati je odradio kroz tjedan ili mjesec. Zaposlenika registrira isključivo administrator ili voditelj tvrtke u bazu podataka te mu on i dodjeljuje sve podatke koji su mu potrebni za prijavu. Početni prozor aplikacije nakon prijave administratora prikazan je na slici 3.3. Također, administrator ima mogućnost unijeti koliko je neki zaposlenik odradio sati.

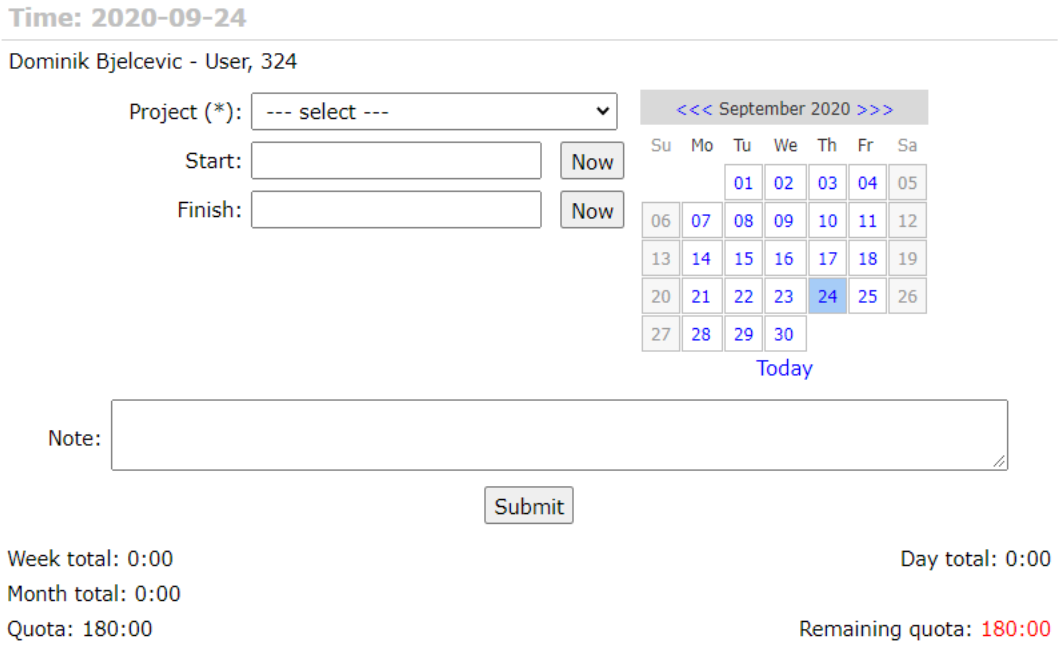


Slika 3.2 *Stranica za prijavu.*



Slika 3.3 *Početni prozor nakon prijave (administrator).*

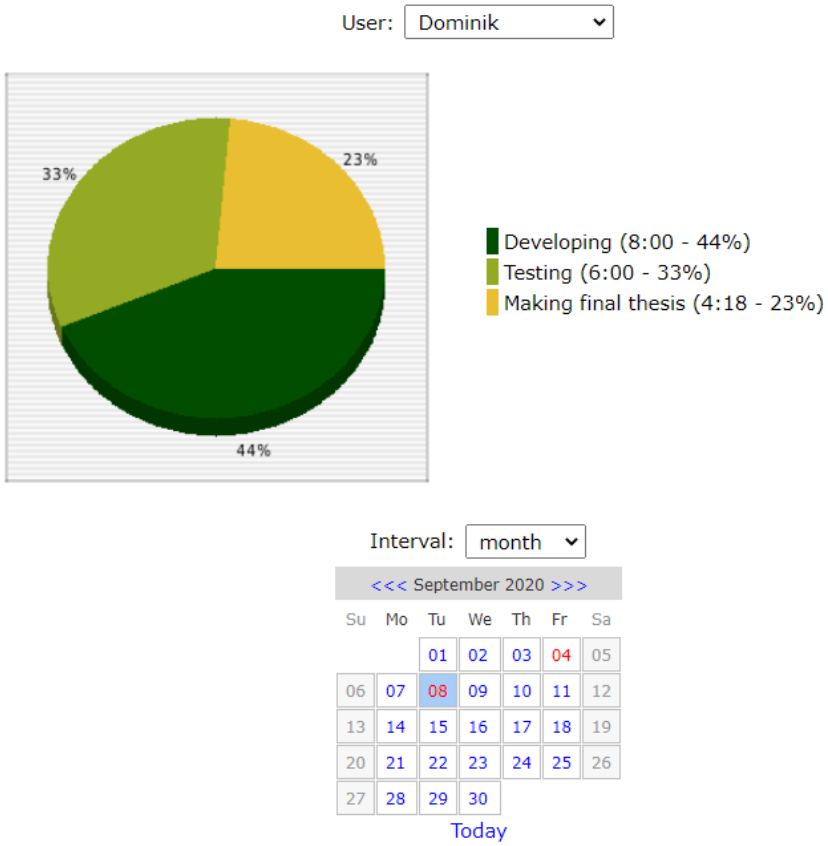
Na slici 3.4 prikazan je početni prozor aplikacije nakon prijave zaposlenika. Zaposlenik nema neke dodatne opcije koje administrator ima.



Slika 3.4 *Početni prozor nakon prijave (zaposlenik).*

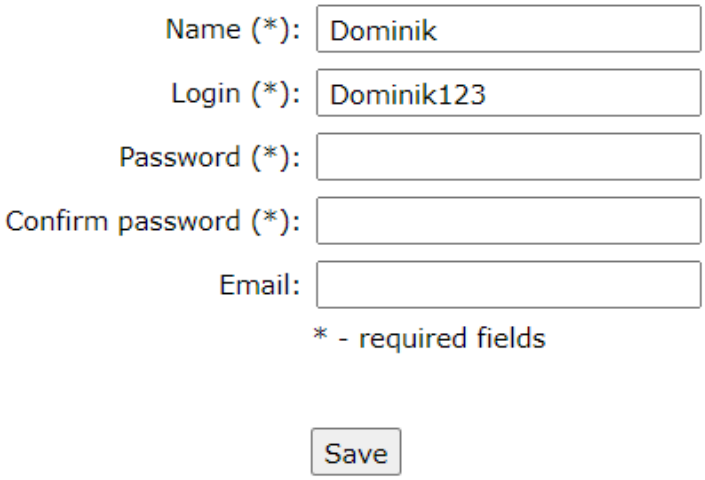
3.3 Uloga Administratora i Zaposlenika

Uloga administratora i zaposlenika ne razlikuje se puno, no administrator će imati nekoliko funkcija više koje može obavljati nad podacima. Administrator će u svakom trenutku moći dobiti listu zaposlenika koji su se prijavili za trenutni dan te će mu na kraju dana, kada se zaposlenik odjavi, biti omogućen pregled koliko je sati odradio taj dan. Također, administrator će moći pregledati koliko je zaposlenik odradio sati kroz svaki tjedan i kroz svaki mjesec. Osim toga što će moći vidjeti koliko je koji zaposlenik obavio sati, administrator će također moći vidjeti tko je sve registriran u bazi neovisno o tome koliko je radnih sati obavio. Nasuprot administratora imamo zaposlenika koji neće imati ove funkcije, no imati će nešto drugo. Zaposlenik nakon prijave na stranicu unosi svoje radno vrijeme kako bi administratoru bilo vidljivo vrijeme kada je počeo raditi. Nakon kraja radnog vremena, zaposlenik unosi radno vrijeme te se šalje u SQL bazu podataka. Na slici 3.5 grafički je prikazano koliko je sati i na kojem projektu radio određeni zaposlenik.



Slika 3.5 *Grafički prikaz radnog vremena zaposlenika.*

Grafički prikaz radnog vremena omogućava administratoru bolji uvid u radno vrijeme zaposlenika te na kojim projektima provodi radno vrijeme. Pri prijavi na portal administrator bira svoju lozinku dok je svakom novom registriranom zaposleniku dodijeljena lozinka za prijavu. To se također može promijeniti pod *Profile* opcijom u izborniku kao što je prikazano na slici 3.6.



Slika 3.6 *Uređivanje profila.*

3.4 Struktura aplikacije

Aplikacija je zamišljena tako da se svaki korak koji se obavi, poput prijave vremena zaposlenika, šalje u bazu podataka. Primjer, kada se zaposlenik prijavi i počne mjeriti svoje radno vrijeme, taj podatak se šalje u SQL bazu te se nakon toga on prikazuje administratoru na njegovom zaslonu. Na slici 3.7 prikazana je SQL tablica svih trenutno registriranih korisnika odnosno zaposlenika.

	id	login	password	name	group_id	org_id	role_id
Edit Copy Delete	1	admin	5ebe2294ecd0e0f08eab7690d2a6ee69	Admin	0	*NULL*	1
Edit Copy Delete	3	Dominik12	79ee79d84c67b9d4a48b0fc758b9b251	Dominik	3	3	2
Edit Copy Delete	4	Dominik123	79ee79d84c67b9d4a48b0fc758b9b251	Dominik	4	4	2

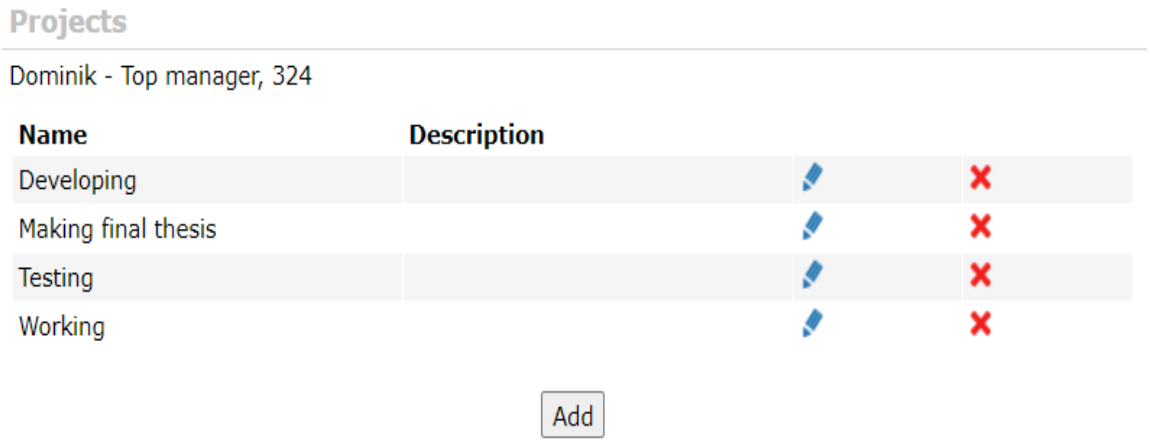
Slika 3.7 *Prikaz svih korisnika u bazi podataka.*

Svi podaci se zapisuju u tablice koje su napravljene u bazi podataka i ti podaci omogućavaju manipuliranje web stranicom. Primjer je, kada se zaposlenik prijavi i počne mjeriti svoje radno vrijeme, taj podatak se šalje u SQL bazu te se nakon toga on prikazuje administratoru na njegovom zaslonu. Na slici 3.8 prikazana je SQL tablica nakon unosa radnog vremena nekog zaposlenika. Same tablice su napravljene u aplikaciji phpMyAdmin.

	id	user_id	group_id	org_id	date	start	duration
Edit Copy Delete	3	3	3	3	2020-09-03	03:17:00	05:00:00
Edit Copy Delete	4	3	3	3	2020-09-06	07:40:00	08:00:00
Edit Copy Delete	5	3	3	3	2020-09-04	07:56:00	08:00:00

Slika 3.8 *Uneseno radno vrijeme prikazano u SQL bazi podataka.*

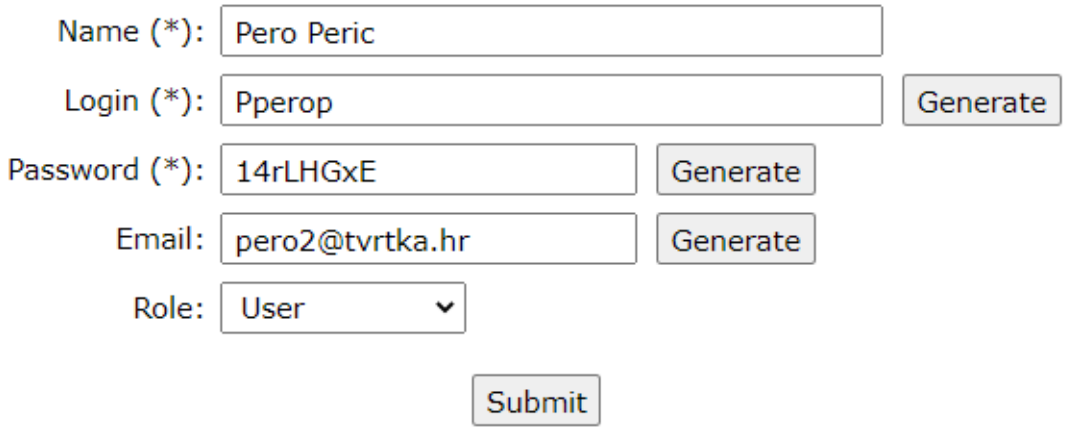
Administrator ima mogućnost dodavanja novih projekta i novih zaposlenika dok je zaposleniku omogućeno samo biranje pojedinog projekta. Dodavanje novih projekta prikazano je na slici 3.9. Administratoru je kroz projekte omogućeno dodavanje novih projekta kao i brisanje samih projekta pri njegovom završetku kao i uređivanju svakog projekta.



Slika 3.9 *Dodavanje novog projekta.*

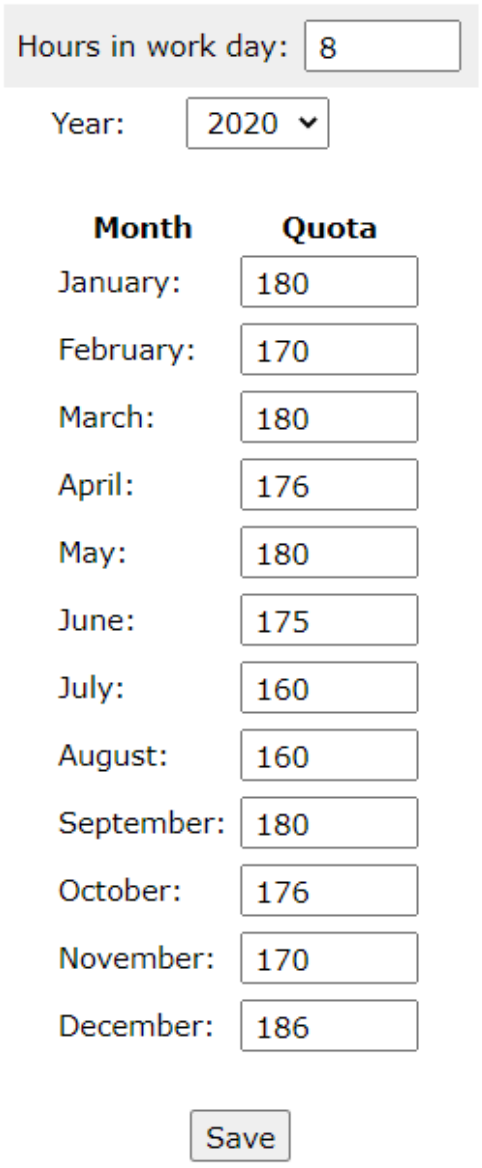
3.5 Opis mogućnosti aplikacije

Sama aplikacija opisana u ovom radu ima puno mogućnosti te će biti opisane u ovom poglavlju. Administratoru je omogućeno dodavanje novih zaposlenika te prati njihov rad. Forma za dodavanje novih zaposlenika prikazan je na slici 3.10.



Slika 3.10 *Forma za dodavanje novog zaposlenika.*

Klikom na *Generate* dugme, zaposleniku se automatski generira njegovo korisničko ime za prijavu na portal, lozinka od 8 znakova te mu se dodjeljuje jedinstvena email adresa. Pomoću ovoga administratoru je omogućeno dodavanje novoga zaposlenika te spremanje podataka jedinstvenih za svakog zaposlenika u SQL bazu. Sam zaposlenik nije u mogućnosti dodavati nove zaposlenika. Također, administratoru je omogućeno određivanje koliko sati zaposlenici trebaju odraditi kroz navedeni mjesec. Primjer toga prikazan je na slici 3.11. Na ovaj način administrator može vidjeti koliko sati zaposlenici još trebaju odraditi kroz određeni mjesec te koliko sati imaju prekovremeno.



Slika 3.11 *Određivanje radnih sati.*

Iz slike 3.11 možemo vidjeti kako je ono što je potrebno za određivanje radnih sati svaki mjesec odabir godine u kojoj želimo odrediti kvotu. Nakon toga za svaki mjesec unosimo kvotu koliko zaposlenik mora obaviti te tekuće godine. Pri evidenciji sati zaposleniku je ispisano koliko je sati odradio u danu, tjednu i mjesecu te mu je prikazano koliko sati još mora odraditi kroz tekući mjesec kako bi zadovoljio zadanu kvotu od strane administratora. To je prikazano na slici 3.12.

Slika 3.12 *Prikaz odrađenih i preostalih sati.*

Nakon što je zaposlenik evidentirao svoje vrijeme, administratoru je omogućen pregled izvješća za određeni mjesec svakog zaposlenika. To mu je omogućeno tako što stisne na opciju *Reports* u izborniku aplikacije te nakon toga bira ime zaposlenika za kojeg želi izvješće te za koji mjesec želi izvješće. Također, omogućen mu je i odabir projekta za koji želi vidjeti provedeno radno vrijeme svakog zaposlenika. Pomoću ovoga administrator ima uvid u stanje radnog vremena svih zaposlenika te mu omogućava lakše rukovanje projektima i samim rasporedom sati svih zaposlenika. Generirano izvješće jednog zaposlenika prikazano je na slici 3.13.

Report: 2020-09-01 - 2020-09-30

Dominik - Top manager, 324

Date	User	Project	Start	Finish	Duration
2020-09-04	Dominik	Testing	8:41	14:41	6:00
2020-09-08	Dominik	Making final thesis			4:18
2020-09-08	Dominik	Developing	6:18	14:18	8:00
2020-09-10	Dominik	Making final thesis	3:17	5:10	1:53
2020-09-12	Dominik	Making final thesis	2:05	6:05	4:00
2020-09-16	Dominik	Developing	7:56	20:53	12:57
2020-09-17	Dominik	Making final thesis	12:37	16:37	4:00
2020-09-20	Dominik	Developing	3:17	17:23	14:06
Total					**55:14**

Slika 3.13 *Generirano izvješće jednog zaposlenika.*

3.5 Odjava

Nakon što zaposlenik završi s radom odjavljuje se sa web stranice i svi njegovi podaci biti će poslani u bazu podataka i vidljivi administratoru. Nakon što se zaposlenik odjavi, trenutna sesija se zatvara jednostavnom php funkcijom *session_destroy()*. Na slici 3.14 prikazana je datoteka *logout.php* koja odjavljuje zaposlenika sa portala te ga preusmjerava na stranicu za prijavu.

logout.php

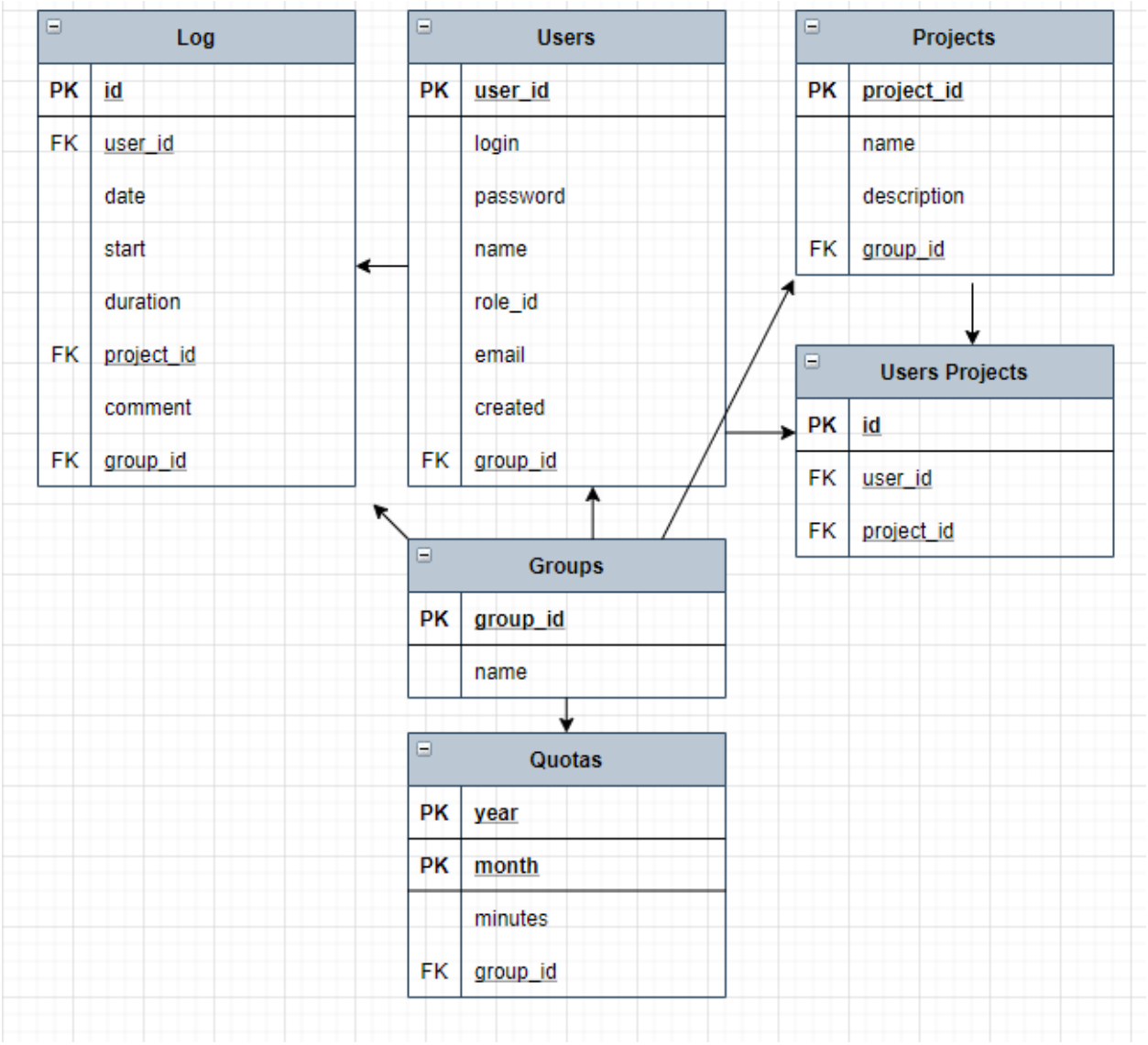
```php
<?php


require_once('initialize.php');
$auth->doLogout();
session_unset();

header('Location: login.php');

```

Slika 3.14 *Prikaz datoteke logout.php koja nam koristi za odjavu sa portala.*

4. OPIS FUNKCIONALNOSTI WEB APLIKACIJE

U ovom poglavlju biti će opisane funkcije koje su se koristile u aplikaciji. Funkcije i primjeri koji su navedeni u trećem poglavlju biti će prikazani pomoću programskog kôda koji ih opisuje.

Na slici 4.1 prikazan je ER dijagram tablica iz baze podataka i njihova povezanost. PK označava primarni ključ tablice (engl. *primary key)* dok FK označava strani ključ tablice (engl. *foreign key).*



Slika 4.1 *ER dijagram baze podataka.*

4.1 Autentifikacija zaposlenika

Pri samoj prijavi na portal, omogućena je prijava zaposlenika i administratora. Kod prijave se provjerava da li se prijavio zaposlenik ili administrator te se nakon toga preusmjerava na odgovarajuću stranicu. Programski kôd za autentifikaciju prikazan je slikom 4.2.

```
if ($request->isPost()) {
  // Validate user input.
  if (!ttValidString($cl_login)) $err->add($i18n->get('error.field'), $i18n->get('label.login'));
  if (!ttValidString($cl_password)) $err->add($i18n->get('error.field'), $i18n->get('label.password'));

  if ($err->no()) {



    if ($auth->doLogin($cl_login, $cl_password)) {

      $current_user_date = $request->getParameter('browser_today', null);
      if ($current_user_date)
        $_SESSION['date'] = $current_user_date;

      // Remember user login in a cookie.
      setcookie('tt_login', $cl_login, time() + COOKIE_EXPIRE, '/');

      $user = new ttUser(null, $auth->getUserId());
      // Redirect, depending on user role.
      if ($user->can('administer_site')) {
        header('Location: admin_groups.php');
      } elseif ($user->isClient()) {
        header('Location: reports.php');
      } else {
        header('Location: time.php');
      }
      exit();
    } else
      $err->add($i18n->get('error.auth'));
  }
```

Slika 4.2 *Autentifikacija korisnika.*

4.2 Administrator

Administrator će imati mogućnost prikaza izvješća svakog zaposlenika te će tako vidjeti koliko je radnih sati zaposlenik odradio kroz mjesec i biti će mu omogućen uvid u radno vrijeme na

pojedinom projektu. Na slici 4.3 prikazan je prikaz izvješća pomoću grafa. Programski kôd na slici 4.3 predstavlja sliku 3.4 u prethodnom poglavlju te opisuje radno vrijeme jednog zaposlenika.

```
$chart_form->addInput(array('type' => 'combobox',
'onchange' => 'this.form.submit();',
'name' => 'type',
'value' => $cl_type,
'data' => $types
// Calendar.
$chart_form->addInput(array('type'=>'calendar','name'=>'date','value'=>$cl_date));

// Get data for our chart.
$totals = ttChartHelper::getTotals($user_id, $cl_type, $cl_date, $cl_interval);
$smarty->assign('totals', $totals);

$chart = new PieChartEx(300, 300);
$data_set = new XYDataSet();
foreach($totals as $total) {
  $data_set->addPoint(new Point( $total['name'], $total['time']));
}
$chart->setDataSet($data_set);
```

Slika 4.3 *Prikaz izvješća pomoću grafa.*

Također jedna od funkcija koju će administrator imati u odnosu na zaposlenika je dodavanje i brisanje zaposlenika. Administrator će biti u mogućnosti dodati novog zaposlenika te obrisati zaposlenika. Za dodavanje novog zaposlenika u bazu podataka se dodaje njegovo ime, korisničko ime s kojim se prijavljuje na portal, lozinka i email. Na slici 4.4 je prikaz dodavanja zaposlenika, a na slici 4.5 brisanje zaposlenika.

```php
static function insert($fields, $hash = true) {
  global $user;
  $conn = getConnection();

  $password = $conn->quote($fields['password']);
  if($hash)
    $password = 'md5('.$password.')';
  $email = isset($fields['email']) ? $fields['email'] : '';

  $sql = "insert into tt_users (name, login, password,email) values (".
    $conn->quote($fields['name']).", ".$conn->quote($fields['login']).
    ", $password, ".$conn->quote($email).")";
  $affected = $conn->exec($sql);
```

Slika 4.4 *Dodavanje novog zaposlenika.*

```php
// Delete user.
$sql = "delete from tt_users where id = $user_id";
$affected = $conn->exec($sql);
```

Slika 4.5 *Brisanje zaposlenika.*

Kako bi dobio bolji uvid u radno vrijeme, administrator ima mogućnost određivanja kvota za radni mjesec te tako može vidjeti koliko još sati zaposleniku preostaje za tekući mjesec te koliko sati je odradio prekovremeno. Prikaz programskog kôda prikazan je na slici 4.6.

```php
$deleteSql = "delete from tt_monthly_quotas".
  " where year = $year and month = $month and group_id = $this->group_id and org_id = $this->org_id";
$this->db->exec($deleteSql);
if ($hours){
  $insertSql = "insert into tt_monthly_quotas (group_id, org_id, year, month, hours)".
    " values ($this->group_id, $this->org_id, $year, $month, $hours)";
  $affected = $this->db->exec($insertSql);

}
```

Slika 4.6 *Dodavanje mjesečnih kvota u bazu podataka.*

Na slici 4.6 prikazan je unos kvota radnih sati u bazu podataka. Nakon prve prijave administratora sve kvote imaju početnu vrijednost nula te se pri unosi novih kvota u bazu podataka stare brišu. Nakon određivanja kvote radnih sati za svaki mjesec u tekućoj godini, ti podaci se spremaju u MySQL bazu podataka kako je prikazano na slici 4.7.

	group_id	org_id	year	month	minutes
Edit Copy Delete	4	4	2020	1	10800
Edit Copy Delete	4	4	2020	2	10200
Edit Copy Delete	4	4	2020	3	10800
Edit Copy Delete	4	4	2020	4	10560
Edit Copy Delete	4	4	2020	5	10800
Edit Copy Delete	4	4	2020	6	10500
Edit Copy Delete	4	4	2020	7	9600
Edit Copy Delete	4	4	2020	8	9600
Edit Copy Delete	4	4	2020	9	10800
Edit Copy Delete	4	4	2020	10	10560
Edit Copy Delete	4	4	2020	11	10200
Edit Copy Delete	4	4	2020	12	11160

Slika 4.7 *MySQL baza podataka za mjesečne kvote.*

4.3 Zaposlenik

Nakon prijave na portal, zaposlenik je preusmjeren na stranicu gdje može prijaviti svoje vrijeme pri dolasku na posao kako je prikazano na slici 3.3. Programski kôd zapisa odrađenog radnog vremena prikazan je na slici 4.8.

```
$duration = toDuration($start, $finish);
if ($duration === false) $duration = 0;
if (!$duration && getUncompleted($user_id)) return false;
if ($duration) {
  $sql = "insert into tt_log (user_id, group_id, org_id, date, duration, project_id,created_by) ".
    "values ($user_id, $group_id, $org_id, ".$conn->quote($date).", '$duration', ".$conn->quote($project).",
     ".$conn->quote($note).",$created_by)";
  $affected = $conn->exec($sql);
```

Slika 4.8 *Prikaz zapisa radnog vremena u bazu podataka.*

Također, zaposleniku nisu omogućene opcije poput dodavanja novog zaposlenika, novog projekta i slično. Sve funkcije koje obavlja zaposlenik, administrator također može obavljati, no sve funkcije koje obavlja administrator nisu omogućene zaposleniku. Zaposlenik je u mogućnosti pogledati samo svoje izvješće i samo svoj broj obavljenih sati. Nije mu omogućeno rukovanje sa samim radom tvrtke i njenim zaposlenicima. Zaposlenicima neće biti omogućeno biranje lozinke već će administrator generirati lozinku. Kôd je prikazan na slici 4.9.

```
function randomPassword(length) {
    var chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOP1234567890";
    var pass = "";
    for (var x = 0; x < length; x++) {
        var i = Math.floor(Math.random() * chars.length);
        pass += chars.charAt(i);
    }
    return pass;
}

function generatePassword() {
    userForm.pas1.value = randomPassword(userForm.length.value);
}
```

Slika 4.9 *Generiranje lozinke za novog zaposlenika*

5. ZAKLJUČAK

Web aplikacija opisana u ovom radu poslužila bi tvrtkama za evidenciju radnog vremena svojih zaposlenika. Iako već postoje web aplikacije u tu svrhu, u ovom radu omogućen je prikaz jedne takve aplikacije. Prednost ove aplikacije u odnosu na već postojeće aplikacije je ta što aplikacija opisana u ovom radu koristi samo minimalne funkcionalnosti koje su potrebne za rad jedne takve aplikacije. Zbog toga je ova aplikacija jednostavna za korištenje. Aplikacija opisana u ovom radu omogućava administratoru uvid u radno vrijeme svih zaposlenika te mu je omogućeno dodjeljivanje projekta na kojem pojedini zaposlenik radi. Zaposlenik će imati mogućnost evidentirati radne sate prilikom prijave na portal. Kada završi s poslom, zaposlenik evidentira vrijeme te se odjavljuje sa portala. Pri evidentiranju sati zaposlenik odabire projekt na kojem trenutno radi kako bi administrator imao lakši uvid u radno vrijeme na pojedinom projektu. Pomoću ove web aplikacije zaposlenik ne treba sam voditi evidenciju radnih sati jer je ovako ima prilikom dolaska na posao. Također, svi radni sati koje je zaposlenik odradio biti će spremljeni u SQL bazu podataka te se neće moći izgubiti. Nadalje, kroz rad ove aplikacije korištene su brojne tehnologije. Za izradu backend dijela zadatka korišten je PHP programski jezik i MySQL baza podataka dok su za izradu frontenda korišteni HTML, CSS i JavaScript. Aplikacija se pokreće pomoću APACHE web servera koji nam je omogućen pomoću programa XAMPP.

LITERATURA

[1] Harvest - https://www.getharvest.com/ [22.5.2020]

[2] Toggl - https://toggl.com/ [22.5.2020]

[3] Everhour - https://everhour.com/ [24.5.2020]

[4] Timely - https://www.gettimely.com/ [24.5.2020]

[5] HTML općenito - https://developer.mozilla.org/en-US/docs/Web/HTML [25.5.2020]

[6] HTML oznake - https://html.com [25.5.2020]

[7] CSS (Cascading Style Sheets) - https://developer.mozilla.org/enUS/docs/Web/CSS [25.5.2020]

[8] Bootstrap - https://getbootstrap.com/ [1.6.2020]

[9] JavaScript osnove - https://javascript.info/ [4.6.2020]

[10] jQuery - https://www.w3schools.com/jquery/ [4.6.2020]

[11] JSON - https://developer.mozilla.org/enUS/docs/WEB/JavaScript [4.6.2020]

[12] PHP Objetkno-orijentirano programiranje greycampus.com/codelabs/php/OOPS [6.6.2020]

[13] PHP - https://www.javatpoint.com/php-tutorial [6.6.2020]

[14] Smarty- https://www.smarty.net/ [6.6.2020]

[15] SQL - https://www.w3schools.com/sql/ [21.6.2020]

[16] MySQL - https://www.mysql.com/ [21.6.2020]

[17] Visual Studio Code anketa - https://insights.stackoverflow.com/survey/2019 [28.6.2020]

SAŽETAK

Tema: Web aplikacija za evidenciju radnog vremena

U ovom radu opisana je web aplikacija za evidenciju radnog vremena zaposlenika u nekoj tvrtki. Web aplikacija opisana u ovom radu napravljena je pomoću modernih tehnologija koje se danas koriste. Cilj rada je prikazati i opisati tehnologije koje su korištene pri izradi aplikacije te prikazati njenu praktičnu funkcionalnost. Izrađena web aplikacija omogućuje zaposlenicima i administratorima lakše praćenje radnih sati. Zaposleniku je omogućen unos radnog vremena koje je obavio za pojedini dan dok je administratoru omogućen pregled sati koje je svaki zaposlenik obavio na dnevnoj ili mjesečnoj bazi. Treba naglasiti da je za backend dio aplikacije korišten pretežito PHP dok su za izradu frontenda korišteni HTML, CSS, Bootstrap i JavaScript. Svi podaci vezani za korisnike i unošene sate spremaju se u lokalnu MySQL bazu podataka koja je omogućena putem lokalnog servera.

Ključne riječi: baza, evidencija, radno vrijeme, web aplikacija, zaposlenik

ABSTRACT

Title: Web application for tracking working hours

This final paper describes the development of a web application for measurement of working hours in a company. Web application described in this paper is created using modern web technologies used today. The goal is to demonstrate and describe technologies that are used in the development of application and to show it's practical functionality. The developed web application can be used by both users and administrators to easily track working hours. The user is allowed to log in his working hours each day while the administrator is allowed to view the hours that each user has done on daily or monthly basis. It is important to note that most of the backend part of the application is made with PHP while frontend part of the application is created with mix of HTML, CSS and JavaScript. All user-related data and working hours that are logged are stored in local MySQL database which is enabled through local server.

Key words: database, employee, tracking, web application, working hours

ŽIVOTOPIS

Dominik Bjelčević, rođen 22. listopada 1996. godine u Osijeku. Odrastao je u Čemincu, selu pokraj Osijeka gdje i danas živi. Školovanje započinje u Osnovnoj školi Čeminac u Čemincu. Tamo je pohađao svih 8 razreda, a nakon toga upisuje Prvu srednju školu Beli Manastir. Nakon 4 godine stekao je zvanje Računalni tehničar. Svoje obrazovanje nastavlja na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku 2016. godine. Upisao je stručni studij Informatike u trajanju od 3 godine. Praksu je odradio 2020. godine kao full-stack developer u MONU gdje je upotpunio svoja znanja objekto orijentiranog programiranja te C# programskog jezika.

PRILOZI

Projektna mapa s izvornim kôdom i priloženim word i pdf dokumentom nalazi se na optičkom disku koji je priložen uz ispisanu verziju završnog rada.