Aplikacija za upravljanje radom baza podataka

Lacković, Rea

Undergraduate thesis / Završni rad

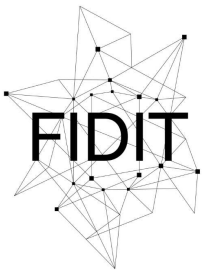
2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:195:017822

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-08-08**



Sveučilište u Rijeci
Fakultet informatike i digitalnih tehnologija

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository

Sveučilište u Rijeci, Fakultet informatike i digitalnih tehnologija

Sveučilišni prijediplomski studij Informatika

Rea Lacković

Aplikacija za upravljanje radom baza podataka

Završni rad

Mentor: doc. dr. sc. Martina Ašenbrener Katić

Komentor: Sanjin Maržić, mag. inf.

Rijeka, srpanj 2023.

Sveučilište u Rijeci
FIDIT Fakultet informatike i digitalnih tehnologija
www.inf.uniri.hr

Rijeka, 12. lipnja 2023.

Zadatak za završni rad

Pristupnica: Lacković Rea

Naziv završnog rada: Aplikacija za upravljanje radom baza podataka

Naziv završnog rada na engleskom jeziku: Application for database management

Sadržaj zadatka:

Zadatak završnog rada je obraditi temu izrade web aplikacije za upravljanje radom baza podataka „OCC", odnosno „Oracle Cloud control". U radu je potrebno opisati alate, tehnologije i jezike korištene pri izradi te opisati proces izrade aplikacije.

Mentor

Doc. dr. sc. Martina Ašenbrener Katić

Martina Ašenbrener Katić

Voditelj za završne radove

Doc. dr. sc. Miran Pobar

Komentor

Sanjin Maržić, mag. inf.

Sanjin Maržić

Zadatak preuzet: 19. lipnja 2023.

Rea Lacković

(potpis pristupnika)

Adresa: Radmile Matejčić 2
51000 Rijeka, Hrvatska
Tel: +385(0)51 584 700
E-mail: ured@inf.uniri.hr
OIB: 64218323816
IBAN: HR1524020061400006966

UNIRI

Sadržaj

Sažetak

Ovaj završni rad će obraditi temu izdrade web aplikacije za upravljanje radom baza podataka „OCC“, odnosno „Oracle Cloud control“. U radu se prvo obrađuje teorijski dio u kojem se ukratko opusuju alati, tehnologije i jezici korišteni pri izradi, a zatim se drugi dio bazira na praktični dio, odnosno samu izradu aplikacije. Pri izradi ove aplikacije su korišteni PHP, JavaScript, HTML, Bootstrap, MySQL, XAMPP, phpMyAdmin.

Ključne riječi: web aplikacija, PHP, JavaScript, HTML, Bootstrap, MySQL, XAMPP, phpMyAdmin

Summary

This final thesis will deal with the topic of creating a web application for database management “OCC”, i.e. “Oracle Cloud control”. The thesis first deals with the theoretical part in which the tools, technologies and languages used in the creation are briefly discussed, and then the second part is based on the practical part, i.e. the creation of the application itself. PHP, JavaScript, HTML, Bootstrap, MySQL, XAMPP, phpMyAdmin were used to create this application.

Keywords: web application, PHP, JavaScript, HTML, Bootstrap, MySQL, XAMPP, phpMyAdmin

1 Uvod

Aplikacije su računalni programi koji se koriste na lokalnom računalu za obavljanje raznih zadataka. Najčešće korištene aplikacije su web preglednici (kao što su Google Chrome i Firefox), uređivači dokumenata (kao što je Word) i klijenti e-pošte (kao što su Outlook ili Thunderbird).

Web aplikacije su web stranice koje pokušavaju izgledati i ponašati se kao desktop aplikacije. Napisani su za pokretanje unutar web preglednika, a ne izravno na računalu. Web aplikacije se oslanjaju na web preglednik za funkcije koje bi inače morale biti kodirane (kao što je gumb za povratak, renderiranje stranice i tako dalje).

Web aplikacije, za razliku od desktop aplikacija, nisu ograničene na jedan operativni sustav. Web aplikacija radi u pregledniku, pa gdje god se preglednik pokrene, pokrenut će se i web aplikacija.

Web aplikacije imaju arhitekturu klijent-poslužitelj te je njihov kod podijeljen u dvije komponente — skripte na strani klijenta i skripte na strani poslužitelja (AWS, 2023). Skripte na strani poslužitelja služe za upravljanje pohranjivanjem i dohvaćanjem informacija, odnosno obrađivanje zahtjeva klijenta i slanje odgovora klijentu, a skripte na strani klijenta za prikaz informacija korisnicima (StackPath, 2023).

U ovom radu slijedi objašnjenje tehnologija i alata korištenih za razvoj aplikacije OCC te samog tijeka razvoja aplikacije. Aplikacija je izrađena u sklopu stručne prakse u tvrtki RIS d.o.o te je dopunjena i dodana dokumentacija za potrebe završnog rada. Cilj aplikacije je upravljanje radom baza u „Oracle Autonomous Database“ usluzi. Također, aplikacija je prvo rađena na lokalnom računalu, a zatim je prebačena na server.

2 Motivacija za izradu aplikacije

Tvrtka RIS d.o.o ima baze podataka u Oracle Cloudu te veći dio cijene se plaća po jezgri u jedinici vremena. Kada je baza ugašena se plaća samo pohrana, a kada je upaljena cijena ide po jezgri. Ideja je da si zaposlenici mogu sami upaliti bazu bez da moraju nekoga pitati, a s druge strane baza ne mora biti upaljena neradnim satima i danima.

Do sada je tvrtka funkcionirala na način da je određen fiksni raspored koji se definira u Oracle Cloudu. Definirati raspored mogu samo administratori te je mijenjanje bilo čega išlo preko njih. Kako su administratori jedini u mogućnosti mijenjati postavke dogodilo bi se da se zaborave vratiti iste. Također, stvara se dodatna komunikacija programera s administratorima i vjerojatnost čekanja na administratorovu reakciju. Aplikacijom „OCC“ se izbjegava komunikacija programera i administratora, čekanje administratora te mogućnost dodatnog troška.

3 Opis tehnologija korištenih u izradi aplikacije

3.1 HTML

HTML, odnosno Hyper Text Markup Language, je označiteljski jezik (engl. *markup language*) korišten pri izradi web stranica i web aplikacija. Označiteljski jezici su jezici koji se koriste za označavanje i strukturiranje sadržaja u elektroničkom formatu. Oni ne služe za programiranje, već za organiziranje i formatiranje podataka, tako da se mogu lakše interpretirati i prikazivati. Za oblikovanje teksta se koriste elementi. Znakovi elemenata su < za početak koda za oblikovanje i > za kraj. HTML elementi su poput ključnih riječi koje definiraju kako će web preglednik formatirati i prikazati sadržaj. Uz pomoć oznaka, web preglednik može razlikovati HTML sadržaj od jednostavnog sadržaja. HTML elementi sadrže tri glavna dijela: početnu oznaku, sadržaj i završnu oznaku, no neki HTML elementi su nezatvoreni elementi što znači da nije potrebna završna oznaka (Harris, 2017.).

Fizičar Tim Berners-Lee je 1989. godine napisao dopis u kojem predlaže internetski sustav hiperteksta te je krajem 1990. godine specificirao HTML i napisao preglednik i poslužiteljski softver. HTML se temeljio na jeziku koji se već koristio za označavanje dokumenata – SGML (Standardni generalizirani označiteljski jezik). Dokument „HTML oznake“ je bio prvi javno dostupan opis HTML-a, a prvi ga je koristio Tim Berners-Lee krajem 1991. godine. Navedeni dokument spominje 18 elemenata jednostavan početni dizajn HTML-a. Prva standardna verzija HTML-a je izdana 1995. godine pod nazivom „HTML 2.0“. Godine 2000. HTML je postao međunarodni standard (ISO/IEC 15445:2000). Krajem 1999. godine je objavljen HTML 4.01, s daljnjim pogreškama objavljenim do 2001. Godine 2004. započeo je razvoj HTML5 u Radnoj skupini za tehnologiju web hipertekstualnih aplikacija (WHATWG), koji je postao zajednički proizvod s W3C-om 2008., te je dovršen i standardiziran dana 28. listopada 2014. (Kyrnin, 2011.), (W3C, 2014). U tablici 1 se nalazi koje godine je pojedina verzija HTML izdana.

Verzija HTML-a	Godina izdanja
HTML	1991.
HTML+	1993.
HTML 2.0	1995.
HTML 3.2	1997.
HTML 4.01	1999.
HTML5	2014.

Tablica 1: Povijesni pregled HTML verzija

Osnovni elementi HTML stranice:

1. deklaraciju vrste dokumenta
2. HTML oznaka
3. oznaka glave
4. oznaka naslova
5. oznaka tijela

HTML se u razvoju aplikacije „OCC“ koristi za kreiranje izgleda i kostura stranica.

3.2 Bootstrap

Bootstrap je najpopularniji CSS okvir za razvoj responzivnih i mobilnih web stranica. Otvorenog koda je te besplatan za korištenje. Bootstrap pruža sintaksu za razvoj predložaka (engl. *template*) dizajna stranica. Bootstrap sadrži osnove razvoja responzivnih web stranica pa programeri imaju zadatak samo umetnuti kod u unaprijed definirani grid sistem. Bootstrap je izgrađen na HTML-u, CSS-u i JavaScriptu. Web programeri koji koriste Bootstrap mogu puno brže izraditi web stranice bez trošenja vremena na brigu o osnovnim vizualnim elementima iste. Primarni cilj Bootstrapa je stvaranje responzivnih web stranica namijenjenih prvenstveno mobilnim uređajima. Omogućuje da se svi dijelovi sučelja web stranice prikazuju na adekvatan način s obzirom na veličinu zaslona (Zola, 2022).

Za razvoj Bootstrapa su zaduženi Mark Otto i Jacob Thornton koji su ga razvili u Twitteru s ciljem poboljšanja dosljednosti alata korištenih na stranici i smanjenja održavanja. Originalno je izdan 19. kolovoza 2011. godine, a prije je bio poznat kao Twitter Blueprint te Twitter Bootstrap (Bootstrap, 2023).

Verzija Bootstrapa	Godina izdanja
Bootstrap	2011.
Bootstrap 2	2012.
Bootstrap 3	2013.
Bootstrap 4	2014.
Bootstrap 5	2021.

Tablica 2: Povijesni pregled Bootstrap verzija

U razvoju aplikacije „OCC“ Bootstrap se koristi kako bi se na lagan način postigli vizualno ugodni elementi i postigao lijep raspored sadržaja na stranici.

3.3 JavaScript

JavaScript je programski jezik kojeg programeri koriste za izradu interaktivnih web stranica. Od osvježavanja feedova društvenih medija do prikaza animacija i interaktivnih karata, JavaScript funkcije mogu poboljšati korisničko iskustvo web stranice. Kao skriptni jezik na strani klijenta, to je jedna od temeljnih tehnologija World Wide Weba (AWS, 2023).

JavaScript je izumio Brendan Eich 1995. godine. Razvijen je za Netscape 2 i postao je ECMA-262 standard 1997. godine. Nakon što je Netscape predao JavaScript ECMA-i, zaklada Mozilla nastavila je razvijati JavaScript za preglednik Firefox. U svibnju 1995., Marc Andreessen je napisao prvi kod JavaScripta pod nazivom 'Mocha'. Kasnije je marketinški tim zamijenio naziv s 'LiveScript'. Ali, zbog zaštitnih znakova i nekih drugih razloga, u prosincu 1995. jezik je konačno preimenovan u 'JavaScript' (javatpoint, 2023).

JavaScript se koristi za izradu dinamičkih web stranica, web i mobilnih aplikacija, igara, web poslužitelja, pozadinske infrastrukture itd. Najčešće se pokreće u web pregledniku korisnika, ne na poslužitelju. To znači da JavaScript može komunicirati s korisnikom, odgovarati na korisničke unose i dinamički ažurirati sadržaj stranice bez potrebe za komunikacijom s poslužiteljem.

JavaScript se može koristiti na dva načina:

1. u HTML dokument unutar <script> elementa
2. u zasebnoj .js datoteci koja se povezuje s HTML elementom <link> unutar <head> elementa

JavaScript se u ovom projektu koristi za prikaz grafova i podataka u njima, prikaz podataka u pojedinim tablicama, sortiranje podataka u tablici i pretraživanje tablice.

3.4 PHP

PHP (Hypertext Preprocessor) jedan je od najpopularnijih jezika koji se koriste za web. Jezik se razvio kako bi programeru omogućio brzo razvijanje dobro oblikovanih programa bez grešaka koristeći i proceduralne i objektno orijentirane tehnike programiranja. Pruža mogućnost korištenja mnogih već postojećih biblioteka koda koje dolaze s osnovnom instalacijom ili se mogu instalirati unutar PHP okoline. Pruža više fleksibilnosti od mnogih drugih jezika. Lakoća s kojom se dodatne biblioteke koda mogu dodati okruženju jedna je od mnogih pokretačkih snaga njegove popularnosti (Prettyman, 2020).

PHP je jezik otvorenog koda. Kao takav, svaka verzija jezika stvorena je korištenjem unosa pojedinaca koji ga koriste - samih programera. To omogućuje jeziku da se s vremenom razvija u smjeru koji pokreću korisnici (Prettyman, 2020).

Od svog prvog izdanja 1995. godine kao Personal Home Page Tool (PHP) od strane Rasmusa Lerdorfa, verzije su objavljene na Internetu s forumima kako bi korisnicima pružili mogućnost davanja prijedloga, pa čak i davanja izmjena koda i dodataka. Danas je www.php.net službena PHP web stranica (Prettyman, 2020).

PHP je u ovom projektu korišten za komunikaciju s bazom podataka „occ“, upravljanje sesijama i upravljanje bazama u Oracle Cloudu.

3.5 XAMPP

XAMPP je paket rješenja web poslužitelja otvorenog koda. Uglavnom se koristi za testiranje web aplikacija na lokalnom web poslužitelju. Razvili su ga Apache Friends, a njegov izvorni kod može revidirati ili izmijeniti publika. Sastoji se od Apache HTTP poslužitelja, MariaDB i tumača za različite programski jezike poput PHP i PERL. Dostupan je na 11 jezika i podržan je na različitim platformama kao što je IA-32 paket Windows & X64 paketa MacOS-a i Linuxa (Javatpoint, 2023) .

XAMPP pomaže programeru razviti i testirati svoju web aplikaciju u lokalnom razvojnom okruženju putem računala i prijenosnih računala prije isporuke na produkciju. To je platforma koja

osigurava prikladno okruženje za testiranje i provjeru rada projekata na temelju Apache, Perl, MySQL baze podataka i PHP-a kroz sustav samog domaćina. Perl se u XMAPP-u koristi kao programski jezik za web razvoj, PHP je rezervni scenarijski jezik, a MariaDB je baza podataka razvijena od strane MySQL-a (Javatpoint, 2023) .

XAMPP znači:

X = cross-platform

A = Apache poslužitelj

M = Mariadb (ranije MySQL)

P = PHP

P = Perl

XAMPP je korišten za lokalni razvoj aplikacije, prije nego se prebaci na server. Korišten je Apache i MySQL server.

3.6 phpMyAdmin

PhpMyadmin je besplatni softverski alat namijenjen za upravljanje administracijom MySQL-a preko Interneta te je napisan u PHP-u. PhpMyAdmin pruža korisničko sučelje kojim se mogu napraviti često korištene operacije nad bazama, no i dalje postoji mogućnost izravnog izvršavanja bilo kojeg SQL upita. Također, podržava velik raspon operacija na MySQL-u i MariaDB-u (phpMyAdmin, 2023).

phpMyAdmin se u projektu koristi za upravljanje bazom podataka aplikacije nakon što se aplikacija prebaci na server.

3.7 DBeaver

DBeaver je besplatan i univerzalni alat otvorenog koda za baze podataka namijenjen programerima i administratorima baza podataka. Pokrenut je 2010. godine kao hobi projekt Serge-a Ridera, a prvo službeno izdanje bilo je 2011. godine na FreeCodeu. Brzo je postao popularno sredstvo u zajednici otvorenog koda. Iste godine je osnovana službena web stranica i stvoren je forum za podršku zajednici (premješten 2016. godine u GitHub) (DBeaver, 2023).

Neke od glavnih značajki DBeavera:

1. Grafičko korisničko sučelje (GUI)
2. Uređivanje podataka
3. Analiza baze podataka
4. SQL Editor
5. Specifične značajke baze podataka

U razvoju aplikacije je koristeći DBeaver stvorena baza podataka te se upravljalo bazom putem njega dok aplikacija nije prebačena na server.

3.8 LDAP

LDAP je 1993. godine razvio Tim Howes i njegovi kolege na Sveučilištu u Michiganu. Stvoren je kao lagana verzija protokola pristupa direktoriju (DAP). Njegov izvorni cilj bio je pružiti pristup direktoriju X.500, ali alat sada ima širu raznolikost uporabe (Bhargava, 2021) .

LDAP (Lightweight Directory Access Protocol) je protokol koji se koristi za pristup, upravljanje i pretraživanje direktorija podataka. LDAP je posebno popularan za upravljanje korisničkim identitetima, kao što su korisnička imena, lozinke, adrese e-pošte itd.

LDAP se temelji na klijent-poslužitelj arhitekturi. Klijenti, kao što su aplikacije ili korisnička sučelja, šalju zahtjeve poslužitelju LDAP-a za pretraživanje ili ažuriranje podataka. Poslužitelji LDAP-a pohranjuju direktorij podataka, provode upite i vraćaju rezultate klijentima. LDAP je dizajniran da bude jednostavan i efikasan protokol koji omogućava brz pristup i manipulaciju velikim količinama podataka (onelogin, 2023) .

LDAP podržava različite operacije koje omogućavaju čitanje, pisanje, ažuriranje i brisanje podataka. Najčešće korištene operacije uključuju pretraživanje (search) za dohvaćanje podataka koji zadovoljavaju određene kriterije, dodavanje (add) i ažuriranje (modify) za unos ili izmjenu podataka te brisanje (delete) za uklanjanje podataka (Mmeje, 2021).

LDAP ima široku primjenu, posebno u kontekstu upravljanja korisničkim identitetima i autentikacije. Mnoge organizacije koriste LDAP za centraliziranu pohranu korisničkih računa i informacija o pristupu. Također, koristi se za integraciju različitih aplikacija i sustava, omogućavajući im dijeljenje korisničkih podataka.

Uz LDAP protokol, postoje i različiti LDAP poslužitelji koji implementiraju taj protokol, kao što su OpenLDAP, Microsoft Active Directory, Novell eDirectory i Oracle Internet Directory.

LDAP je bio potreban kako bi se korisnici mogli prijaviti sa svojim Windows korisničkim računima u aplikaciju.

3.9 MySQL

MySQL je najpopularnija baza podataka otvorenog koda. Prema DB-Enginesu, MySQL se svrstava kao druga najpopularnija baza podataka, iza Oracle baze podataka. MySQL pokreće mnoge od najpoznatijih aplikacija, uključujući Facebook, Netflix, Twitter, Airbnb, Uber, Booking.com i Shopify. Budući da je MySQL otvorenog koda, uključuje brojne značajke razvijene u uskoj suradnji s korisnicima više od 25 godina (Oracle Cloud Infrastructure, 2023) .

Švedska tvrtka pod nazivom MySQL AB prvotno je razvila MySQL 1994. godine. Američka tehnološka kompanija Sun Microsystems je preuzela potpuno vlasništvo kada su kupili MySQL AB

2008. godine. Američki tehnološki gigant Oracle 2010. godine kupio je Sun Microsystes, a MySQL je od tada u vlasništvu Oracle-a (B., 2023) .

Ključne prednosti:

1. Jednostavnost korištenja
2. Pouzdanost i skalabilnost
3. Fleksibilnost
4. Podrška za više platformi
5. Široka podrška
6. Sigurnost

MySQL dolazi u dvije glavne verzije - MySQL Community Edition i MySQL Enterprise Edition. Community Edition je besplatna i otvorena za korištenje, dok Enterprise Edition pruža dodatne značajke i napredne alate uz plaćenu podršku.

3.10 Metodologija MIRIS i metoda entiteti – veze

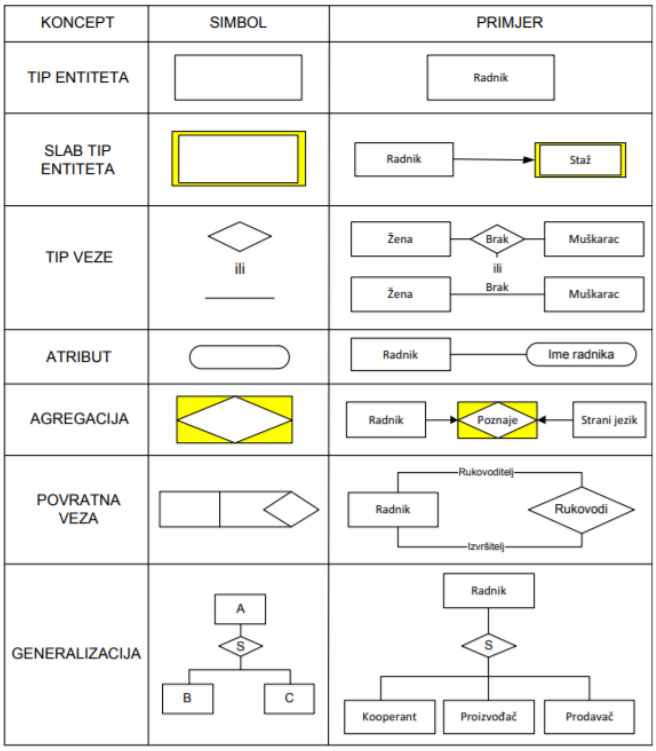
Metodologija MIRIS (Metodologija za Razvoj Informacijskih Sustava) je skup metoda čiji je cilj projektirati i izgraditi Informacijski sustav (IS). Metodologiju MIRIS razvio je prof. dr. sc. Mile Pavlić, a objavljena je 1995. godine. MIRIS se koristi s tri osnovne metode: metodom za modeliranje podataka, metodom za modeliranje procesa i metodom za modeliranje aplikacija (Pavlić, Informacijski sustavi, 2011).

Metoda entiteti - veze (skraćeno EV) je grafički prikaz međusobno povezanih grupa podataka promatranoga sustava. EV je semantički bogata metoda za modeliranje podataka jer raspolaže ljudski bliskim konceptima. EV se odlikuje prirodnošću opisa, a njezini koncepti su bliski korisniku, pa je shema modela podataka laka za razumijevanje i komunikaciju korisnika i projektanta (Pavlić, Oblikovanje baza podataka, 2011.).

Osnovni koncepti metode entiteti - veze od kojih se gradi struktura modela entiteti - veze su (Pavlić, Oblikovanje baza podataka, 2011.):

- Entitet i tip entiteta
- Veza i tip veze
- Atribut tipa entiteta
- Slab tip entiteta i specijalni tipovi veza
- Agregirani tip entiteta
- Povratni tip veze
- Generalizacijski tip veze

Model podataka metodom EV gradi se upotrebom grafičkih simbola prikazanih na slici 1 (Pavlić, Oblikovanje baza podataka, 2011.).



Slika 1: Grafički simboli metode EV

Dijagram strukture modela naziva se dijagram entiteta i veza, skraćeno: DEV, EV dijagram, shema EV (engl. *Entity - Relationship Diagram*, skraćeno ERD) (Pavlić, Oblikovanje baza podataka, 2011.).

Metodologija MIRIS, odnosno model EV, je korišten za kreiranje modela podataka aplikacije.

4 Izrada web aplikacije

U ovom poglavlju je opisan sam razvoj aplikacije, od modeliranja podataka, odnosno dizajniranja baze podataka do konačnog proizvoda.

4.1 Izrada baze podataka

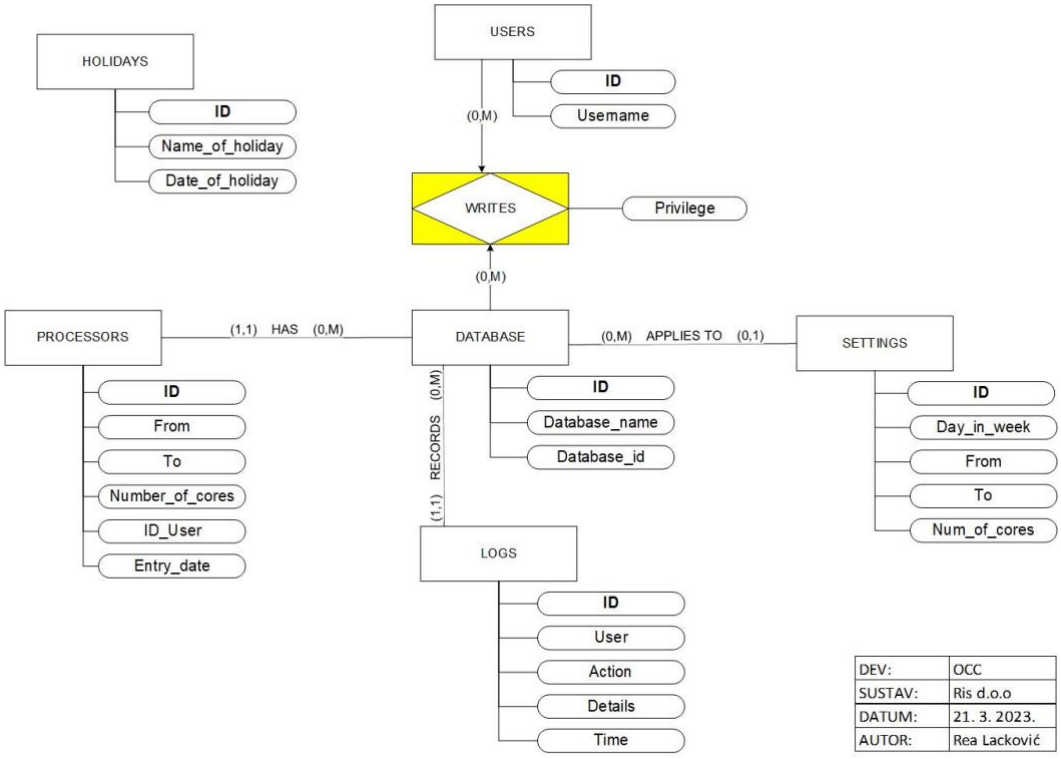
Ovo poglavlje opisuje razvoj baze podataka. Prvo je opisano koje su potrebe baze, zatim je izmodeliran model podataka i relacijski model te na kraju je implementirana baza u alatu DBeaver.

4.1.1 Opis potrebe baze podataka

Tvrtka ima više baza podataka u Oracle Cloud-u te je potrebno da one nekada rade, odnosno ne rade. Također, svaka baza ima određen broj jezgri koje mogu raditi u određeno vrijeme. Ponekada je potrebno da određena baza radi drugačije od uobičajenog vremena i postavki zbog veće količine posla i sl. Postoje razni korisnici, odnosno zaposlenici tvrtke i nemaju svi iste mogućnosti pristupa određenim bazama.

4.1.2 Dijagram entiteta i veza

Na slici 2 je prikazan model podataka aplikacije OCC modeliran metodom EV.



Slika 2: Model podataka aplikacije OCC

Određena baza podataka (Database tablica) pod svojim ID-em ima ime te Database_id što je zapravo ID koji dobije unutar Oracle Cloud-a. Svaka baza ima zadano vrijeme u danu kada radi te se ti podaci nalaze u tablici Settings. Kako je rad baze određen prema danima u tjednu tablica Settings, uz samo vrijeme i broj jezgri, ima atribut naziva dana. Tablica Proccesors služi za spremanje neuobičajenih termina rada baza. U nju se spremaju vremenski podaci, od kada do

kada treba baza raditi, koliko jezgri treba raditi, datum unosa kako bi se znala zadnja informacija te ime korisnika koji je upisao podatak. U tablici Users se nalaze imena zaposlenika pod username-om. Agregacija Writes spaja tablicu Users i Database te ima atribut Privilege. Dopuštenja o tome koji korisnik može što raditi nad određenom bazom se čitaju kroz atribut Privilage. Tablica Holiday u ovome projektu nije implementirana, ali sadržavala bi datum i naziv praznika te bi se iz nje provjeravalo je li praznik i ako je baza bi radila na određeni način.

4.1.3 Relacijski model

HOLIDAYS (**ID**, Name_of_holiday, Date_of_holiday)

USERS (**ID**, Username)

DATABASE (**ID**, Database_name, Database_id)

WRITES (**ID(Users)**, **ID(Database)**, Privilege)

PROCESSORS(**ID**, From, To, Number_of_cores, Entry_date, *ID(Database)*)

SETTINGS (**ID**, Day_in_week, From, To, Nun_of_cores, *ID(Database)*)

LOGS (**ID**, Username, Action, Details, *ID(Database)*)

4.1.4 Fizičko oblikovanje

Za fizičko oblikovanje baze podataka upotrijebljen je DBeaver. Pomoću njega se lako stvaraju tablice u bazi podataka i pune s podacima. Na slici 3 su prikazane sve tablice u bazi podataka „occ“. Kroz sučelje se i postojeći podaci u tablicama mogu mijenjati, brisati i dodavati.

Table Name	Engine	Auto Increment	Data Length	Partitioned	Description
processors	InnoDB	1	16K	[ ]	
users	InnoDB	1	16K	[ ]	
database	InnoDB	1	16K	[ ]	
settings	InnoDB	1	16K	[ ]	
logs	InnoDB	1	16K	[ ]	
writes	InnoDB	0	16K	[ ]	

Slika 3: Prikaz kreiranih tablica u DBeaveru

4.2 Logo

Na slici 4 je prikazan logo aplikacije koji se koristi. Postoji varijanta loga i u bijeloj boji.

Slika 4: Logo aplikacije

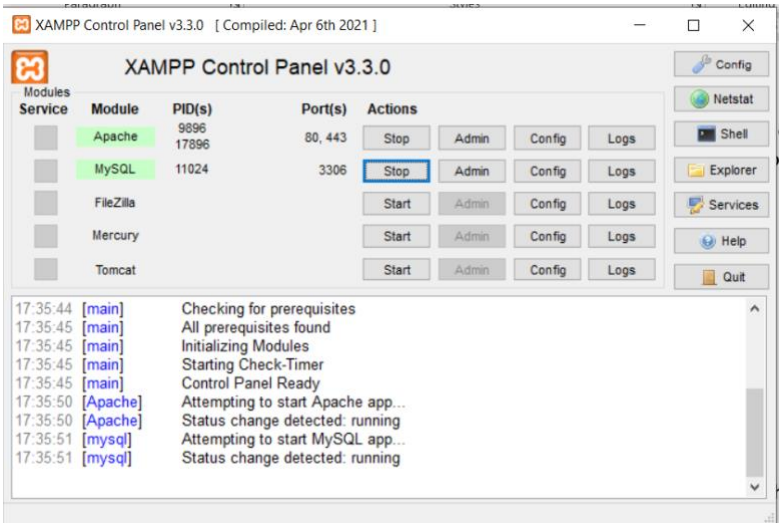
4.3 Razvoj stranica

Aplikacija se sastoji od stranica:

1. Home – početna stranica sa statusom, grafovima i tablicama
2. Dashboard – stranica pregleda i brisanja podataka u tablici Processors
3. Add – stranica za dodavanje podataka u tablicu Processors
4. Settings – stranica za dodavanje podataka u tablicu Settings
5. Sign – stranica za prijavu u aplikaciju

Razvoj navedenih stranica je opisan u narednim poglavljima, a zatim su objašnjene datoteke koje se nalaze u pozadini kako bi aplikacija funkcionirala na pravi način.

Kako bi krenuli s izradom aplikacije potrebno je pokrenuti lokalni server u XAMPP sučelju tako da se pokrene Apache server i server za MySQL bazu podataka. Pokretanjem servera dobivamo mogućnost pokretanja onoga što programiramo i spajanje aplikacije s bazom podataka. Postupak je prikazan na slici 5.



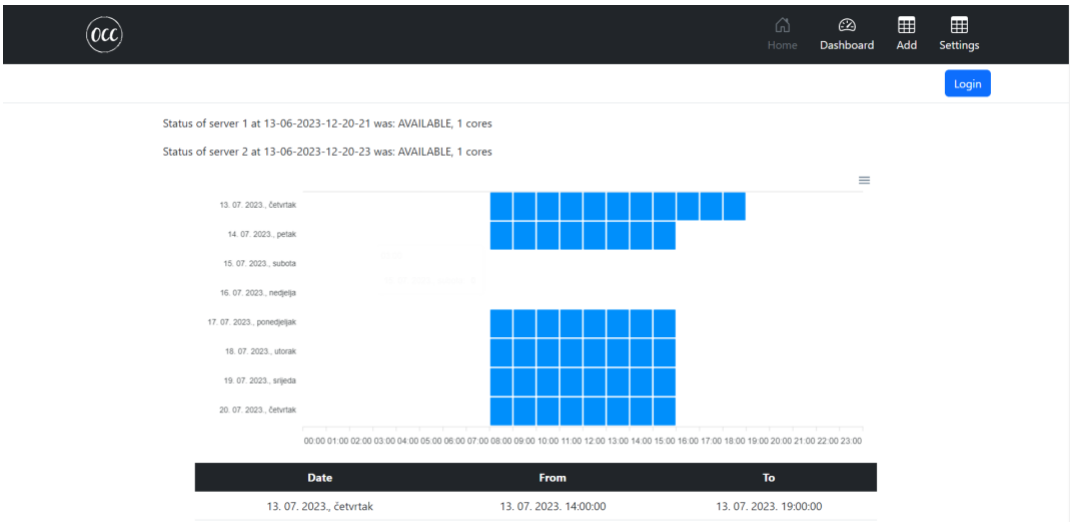
Slika 5: XAMPP sučenje

Na računalu na lokaciji xmapp/htdocs je kreirana mapa pod nazivom „occ“ u kojoj će se nalaziti sve datoteke potrebne za aplikaciju. Aplikaciji se pristupa putem preglednika pod adresom localhost/occ. Sve datoteke će se kreirati u Notepad++-u.

4.3.1 Stranica „Home“

Izgled stranica je preuzet s Bootstrap stranice te je po potrebi dorađen.

Na stranici „Home“ (slika 6) se trebaju prikazati statusi servera za vrijeme kada su zadnji put provjereni, heatmap grafovi te pripadajuće tablice kako koja baza radi u narednih 7 dana što se vidi na slici.



Slika 6: Stranica "Home"

Za stranicu „Home“ se koristi datoteka index.php. Datoteka počinje s php kodom kako bi se pokrenula sesija što se radi naredbom session_start().

U HTML dijelu datoteke se nalaze elementi na čijim će se mjestima prikazati status, grafovi i tablice (slika 7).

```
<div style="padding-left: 15%; padding-right: 10%;"
id="status"></div>

<div style="padding-left: 10%; padding-right: 10%;"
id="table1"></div>
```

Slika 7: Elementi HTML-a za prikaz statusa, grafova i tablica

Izračun što se treba prikazati na stranici se nalazi u HTML elementu <script> te su korišteni PHP i JavaScript. U PHP dijelu se vrši spajanje na bazu te čitanje podataka iz tablica Processors, Settings i Logs po tome kojoj bazi koji podatak pripada. Čitajući te podatke se stvaraju varijable koje sadrže prikladne podatke za pojedine baze te se ti podaci spremaju u listu koja će se koristiti u daljnjem kodu. Logika čitanja podataka se nalazi na slici 8 gdje je prikazan kod za čitanje podataka iz tablice Processors.

```
echo "var data = [];";


for ($i = 0; $i < count($databases); $i++) {
$db = $conn;
$query = "SELECT `From`, `To` FROM processors WHERE Database_id = " .
$databases[$i]["ID"];
$r = $db->query($query);

if($r == true){
if ($r->num_rows > 0) {
$row = mysqli_fetch_all($r, MYSQLI_ASSOC);
$msg = $row;
} else {
$msg = "No Data Found";
}
}
else{
$msg= mysqli_error($db);
}

echo "data" . $i . " = " . json_encode($row) . ";";
echo "if(data" . $i . " != null) data.push(data" . $i . "); else
data.push([]);";
}
```

Slika 8: Kod za čitanje podataka iz tablice Processors

Iz tablice Logs se dobije status pojedine baze (slika 9) koji se onda šalje HTML-u putem funkcije getElementById().innerHTML.

```
echo "var statuus = [];";


for ($i = 0; $i < count($databases); $i++) {
    $query = "SELECT `Details` FROM `logs` WHERE `Action` = 'Status'
AND Database_id = " . $databases[$i]["ID"] . " ORDER BY Time DESC
limit 1";
    $r = $db->query($query);
    $row3 = mysqli_fetch_all($r, MYSQLI_ASSOC);

    echo "statuus" . $i . " = " . json_encode($row3) . ";";
    echo "if(statuus" . $i . " != null) statuus.push(statuus" . $i .
");";
}
```

Slika 9: Kod za čitanje zadnjeg statusa u tablici Logs

Za provjeru kada baze trebaju raditi je potrebno stvori listu objekata u koju će se spremati podaci kada koja baza treba raditi (slika 10). Ti objekti se sastoje od 7 podataka čiji nazivi predstavljaju datume narednih 7 dana, a vrijednosti pod tim nazivima su liste od 24 nule. Liste od 24 podatka, odnosno nula, predstavljaju 24 sata u danu. Kasnije u provjeri će se te nule mijenjati u zavisnosti o satu kada baza treba raditi.

```
dat2 = [new Date(), new Date(Date.now() + 24 * 60 * 60 * 1000),
    new Date(Date.now() + 2 * 24 * 60 * 60 * 1000),
    new Date(Date.now() + 3 * 24 * 60 * 60 * 1000),
    new Date(Date.now() + 4 * 24 * 60 * 60 * 1000),
    new Date(Date.now() + 5 * 24 * 60 * 60 * 1000),
    new Date(Date.now() + 6 * 24 * 60 * 60 * 1000),
    new Date(Date.now() + 7 * 24 * 60 * 60 * 1000)]
dataShow = [];
for(var o = 0; o < data.length; o++){
    window['dataShow' + o] = {};
    dataShow.push(window['dataShow' + o]);
}

for(var o=0; o < dataShow.length; o++){
    for(var i = 0; i < 8; i++)
    {
        dataShow[o][dat2[0].toISOString()] =  [0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
        dataShow[o][dat2[1].toISOString()] = [0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
        dataShow[o][dat2[2].toISOString()] = [0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
        dataShow[o][dat2[3].toISOString()] = [0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
        dataShow[o][dat2[4].toISOString()] = [0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
        dataShow[o][dat2[5].toISOString()] = [0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
        dataShow[o][dat2[6].toISOString()] = [0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
        dataShow[o][dat2[7].toISOString()] = [0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
    }
}
```

Slika 10: Stvaranje liste objekata

Kada imamo pripremljene podatke, onda možemo ići u provjeru. Provjera se vrši tako što se učitani podaci iz baze uspoređuju s tim koji datumi su u narednih 7 dana. Prvo se vrši provjera podataka iz tablice Processors, a zatim iz tablice Settings. Tako dobijemo konačne liste za sate u danima kada baza treba biti upaljena.

Za prikaz heatmap-a se koristi ApexCharts koji definira dvije varijable:

- options – atributi grafa
- series – podaci za prikaz

Options varijabla je ista za sve grafove, a series se mijenja pa je to zapravo dvodimenzionalna lista za pojedinu bazu. Liste za prikaz se sastoje od objekata, koji za naziv imaju datum, a za podatke, odnosno „data“ listu objekata za sate. Objekti koji predstavljaju sate pod atributom „x“ imaju sat u obliku hh:mm, a pod atributom„y“ nulu ili jednicu, zavisno o tome treba li baza raditi toga sata ili ne. Prikaz dodavanja podataka u varijablu series je prikazan na slici 11.

```
for(var p = 0; p < dataShow.length; p++)
{
    var l = dat2.length - 1;
    var l1 = 0;
    l0++;
    for (const d in dataShow[p])
    {
        series[l0][l1] = {name: new
Date(Object.keys(dataShow[p])[l]).toLocaleDateString() + ', ' +
        new Date(Object.keys(dataShow[p])[l]).toLocaleString("hr", {
weekday: "long" }), data: []};
        for(var i = 0; i < 24; i++)
        {
            series[l0][l1]['data'].push({x: x1[i], y:
dataShow[p][dat2[l].toISOString()][i]});
        }
        l1++;
        l--;
        if(l < 0 || l1 >= 8) break;
    }
}
```

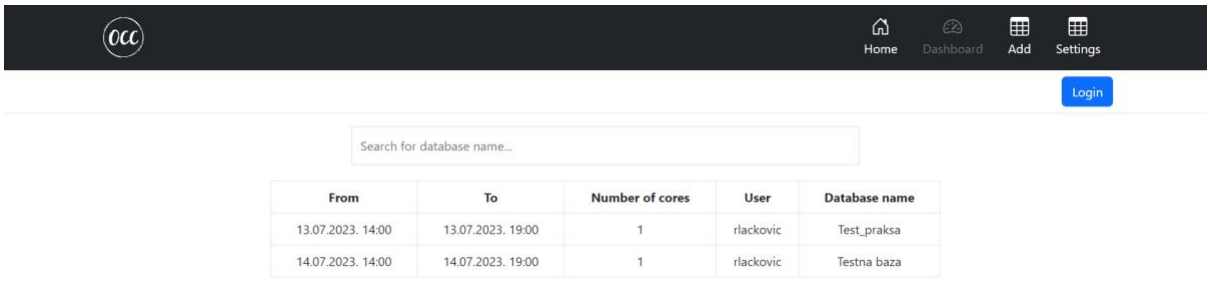
Slika 11: Dodavanje podataka u listu series

Nakon što se popuni varijabla series s potrebnim podacima slijedi prikaz grafa i tablice.

4.3.2 Stranica „Dashboard“

Na stranici „Dashboard“ (slika 12) su tablično prikazani svi podaci uneseni u tablicu Processors. Iznad same tablice se nalazi traka za pretraživanje po nazivu baze. Također,

implementirana je mogućnost sortiranja tablice pomoću gotovog koda dostupnog na internetu. Na slikama 13 i 14 se nalazi kod za prikaz podataka iz tablice Processors u obliku tablice.



Slika 12: Stranica "Dashboard"

```
<?php
    if(is_array($fetchData)){
    $sn = 1;
    foreach($fetchData as $data){
?>
    <tr>
    <td sorttable_customkey="<?php echo $data['From']; ?>" ><?php echo
date('d.m.Y. H:i', strtotime($data['From'])); ?></td>
    <td sorttable_customkey="<?php echo $data['To']; ?>"><?php echo
date('d.m.Y. H:i', strtotime($data['To'])); ?></td>
    <td><?php echo $data['Number_of_cores'];?></td>
    <td><?php echo $data['ID_User']; ?></td>
    <td>
    <?php
    $list = mysqli_query($db, "SELECT Database_name FROM `database`
WHERE ID = '" . $data['Database_id'] ."'");
    echo mysqli_fetch_assoc($list)['Database_name'];
    ?></td>
```

Slika 13: Kod za prikaz podataka na stranici "Dashboard" - 1

```
    <?php
    if (isset($_SESSION['user_id'])) {
    echo '<td> <a href="delete.php?ID=' . $data["ID"] . '&From=' .
$data['From'] . '&To=' . $data['To'] . '&Database=' .
$data["Database_id"] . '"
    onclick="return confirm(\'Are you sure to delete?\')"> <i
class="material-icons" style="font-size: 23px; color:
red;">delete</i></a>';
}
?>
    </tr>
    <?php
    $sn++;}}else{ ?>
    <tr>
    <td>
<?php echo $fetchData; ?>
</td>
<tr>
<?php
}?>
```

Slika 14: Kod za prikaz podataka na stranici "Dashboard" - 2

Varijabla $fetchData u kodu na slikama 13 i 14 je uvezena iz datoteke developer.php čiji je sadržaj na slikama 14 i 15. U tu varijablu se spremaju podaci iz tablice Processors u obliku liste listi te se oni čitaju u datoteci table.php.

```
<?php

include("database.php");

$db = $conn;
$columns= ['ID', ``From`', ``To`', 'Number_of_cores', 'ID_User',
'Database_id'];
$fetchData = fetch_data($db, tableName, $columns);

function fetch_data($db, $tableName, $columns){
     if(empty($db)){
      $msg= "Database connection error";
     }elseif (empty($columns) || !is_array($columns)) {
      $msg="columns Name must be defined in an indexed array";
     }elseif(empty(tableName)){
       $msg= "Table Name is empty";
     }
```

Slika 15: Sadržaj datoteke developer.php - 1

```
    else{
        $columnName = implode(", ", $columns);
        $query = "SELECT ".$columnName." FROM " . tableName;
        $result = $db->query($query);
        if($result == true){
            if ($result->num_rows > 0) {
                $row = mysqli_fetch_all($result, MYSQLI_ASSOC);
                $msg = $row;
            } else {
                $msg = "No Data Found";
            }
        }
        else{
            $msg= mysqli_error($db);
        }
    }
    return $msg;
}
?>
```

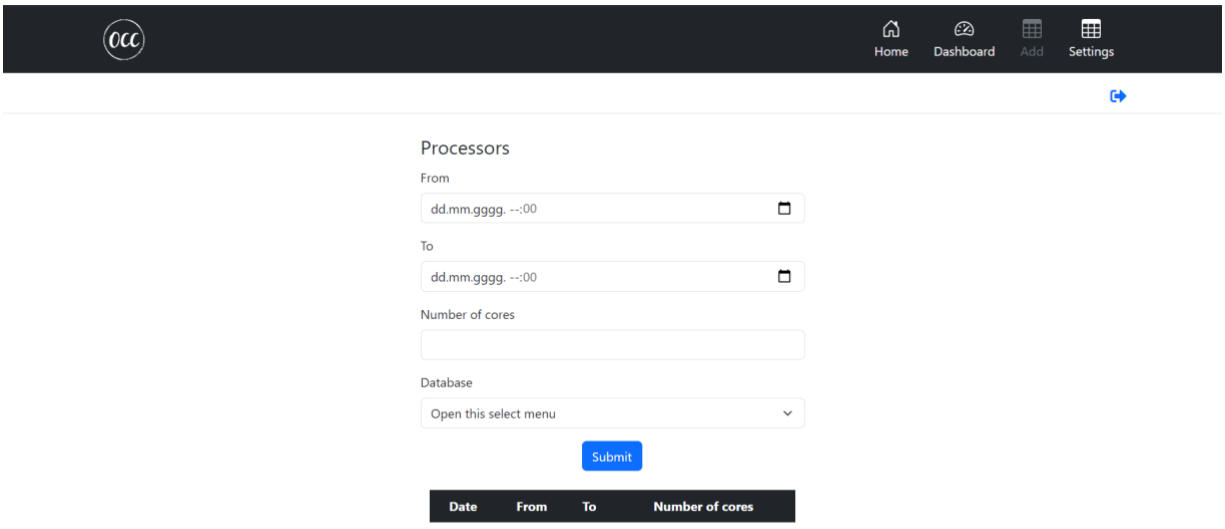
Slika 16: Sadržaj datoteke developer.php – 2

4.3.3 Stranica „Add“

Stranica „Add“ je dostupna samo prijavljenim korisnicima, ukoliko korisnik nije prijavljen klikom na njezinu poveznicu se vraća na „Home“ stranicu.

Funkcija ove stranice je unos podataka za tablicu Processors, odnosno unos termina kada želimo da pojedina baza radi. Obrazac za unos izgleda kao na slici 17 te je napravljen pomoću Bootstrap-a. Kod za obrazac je prikazan na slikama 18 i 19. Rad baze se može namjestiti samo na puni sat, tako da se provjerava unos sata. Također unos datuma i baze je obvezan dok ako se ne unese broj jezgri on će biti 1. Kod unosa datuma i vremena je onemogućeno označavanje prošlih datuma.

S odabirom baze se ispod prikazuje tablica s podacima kako inače ta baza radi, čitajući podatke iz tablice Settings za odabranu bazu. Također, provjerava se koji je korisnik prijavljen te će mu biti omogućen odabir samo onih baza za koje ima privilegije da im može pristupiti.



Slika 17: Stranica „Add“

```
<fieldset>
    <legend>Processors</legend>
        <form name="frmProcessors" method="post" action="processors.php">
            <p>
                <label for="From" class="form-label">From</label>
                  <input type="datetime-local" required class="form-control"
step="3600" name="datFrom" id="datFrom" min="<?=date('Y-m-d\Th:00')?>">
            </p>
            <p>
                <label for="To" class="form-label">To</label>
                  <input type="datetime-local" required class="form-control"
step="3600" name="datTo" id="datTo" max="2023-01-2023T20:00">
            </p>
            <p>
                  <label for="Number of cores" class="form-label">Number of
cores</label>
                   <input type="number" class="form-control" name="numCores"
id="numCores" min="1" max="8">
            </p>
            <p>
                <label for="Database" class="form-label">Database</label>
                  <select required name="database" id="database" class="form-
select" aria-label="Default select example" onchange="showInfo()">
              <option value="" disabled selected>Open this select menu</option>
```

Slika 18: Kod za obrazac stranice "Add" - 1

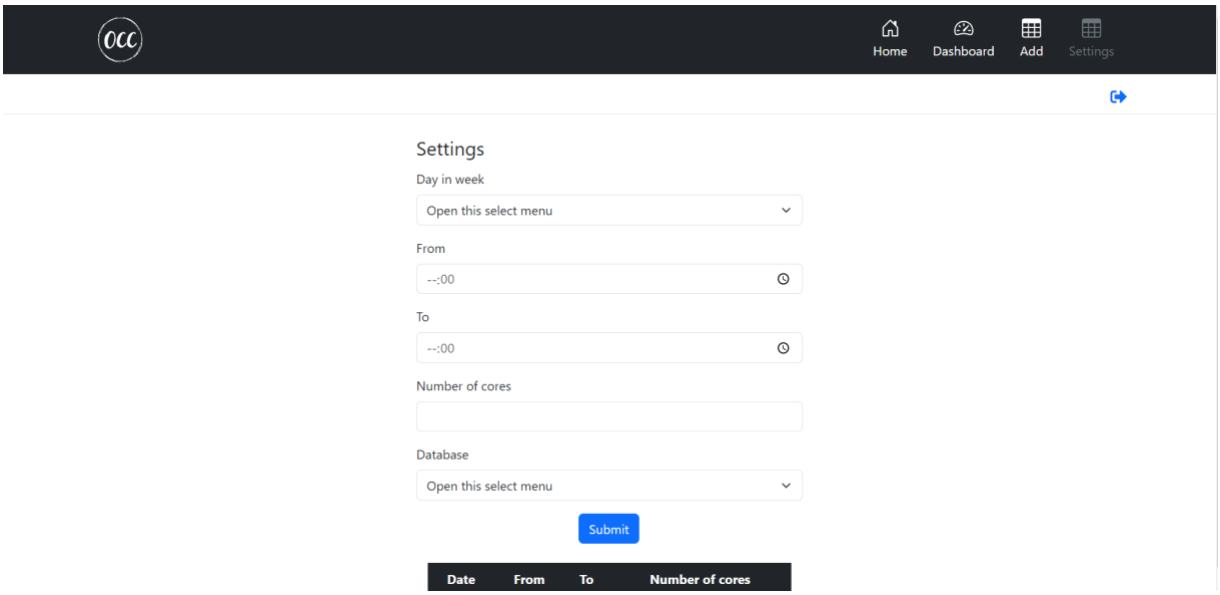
```
                <?php
                include("database.php");
                $list = mysqli_query($conn, "SELECT ID, Database_name FROM
`database` WHERE ID IN (SELECT Database_id from writes WHERE User_id = (SELECT ID
from users WHERE Username = '" . $_SESSION['user_id'] . "'))");
                while($row_list = mysqli_fetch_assoc($list)){
                ?>
            <option value="<?php echo $row_list['ID']; ?>"><?php echo
            $row_list['Database_name']; ?></option>
                <?php
                }
                ?>
                </select>
            </p>
            <p style = "padding-left: 42%">
           <input class="btn btn-primary" type="submit" value="Submit" name="Submit"
id="Submit" >
            </p>
        </form>
</fieldset>
```

Slika 19: Kod za obrazac stranice "Add" - 2

4.3.4 Stranica „Settings“

Stranica „Settings“ je također dostupna samo prijavljenim korisnicima, ukoliko korisnik nije prijavljen klikom na njezinu poveznicu se vraća na „Home“ stranicu.

Funkcija ove stranice je unos podataka za tablicu Settings, odnosno unos kako želimo da pojedina baza svakodnevno radi. Obrazac za unos izgleda kao na slici 20 te je napravljen pomoću Bootstrap-a. Kod za obrazac je prikazan na slikama 21 i 22. Informacije od kada do kada su u obliku vremena, koje se može namjestiti samo na puni sat. Također, kao i na stranici „Add“ odabirom baze se prikazuje tablica kako inače radi odabrana baza. Nadalje, ako se ne upiše podatak za broj jezgri on će biti automatski 1. U slučaju da želimo „obrisati“ vrijeme za pojedini dan, tada ostavimo prazna polja za informacije „From“ i „To“. Naravno svaki prijavljeni korisnik može odabrati samo onu bazu za koju ima dopuštenje.



Slika 20: Stranica „Settings“

```
<fieldset>
    <legend>Settings</legend>
        <form name="frmSettings" method="post" action="settings.php">
            <p>
                <label for="Dayinweek" class="form-label">Day in week</label>
                 <select required class="form-select" aria-label="Default select
example" name='dayinweek' id="dayinweek">
                        <option value="" disabled selected>Open this select
menu</option>
                    <option value="Monday">Monday</option>
                    <option value="Tuesday">Tuesday</option>
                    <option value="Wednesday">Wednesday</option>
                    <option value="Thursday">Thursday</option>
                    <option value="Friday">Friday</option>
                    <option value="Saturday">Saturday</option>
                    <option value="Sunday">Sunday</option>
                </select>
            </p>
            <p>
                <label for="From" class="form-label">From</label>
              <input type="time" class="form-control" step="3600" name="timeFrom"
id="timeFrom">
            </p>
```

Slika 21: Kod za obrazac stranice "Settings" - 1

```
            <p>
                <label for="To" class="form-label">To</label>
               <input type="time" class="form-control" step="3600" name="timeTo"
id="timeTo">
            </p>
            <p>
                    <label for="Number of cores" class="form-label">Number of
cores</label>
                     <input type="number" class="form-control" name="numCores"
id="numCores" min="1" max="8">
            </p>
            <p>
                <label for="Database" class="form-label">Database</label>
                  <select required name="database" id="database" class="form-
select" aria-label="Default select example" onchange="showInfo()">>
               <option value="" disabled selected>Open this select menu</option>
                <?php
                   $list = mysqli_query($conn, "SELECT ID, Database_name FROM
`database` WHERE ID IN (SELECT Database_id from writes WHERE User_id = (SELECT
ID from users WHERE Username = '" . $_SESSION['user_id'] . "') AND Privilege =
'administrator')");

                while($row_list = mysqli_fetch_assoc($list)){
                ?>
                    <option value="<?php echo $row_list['ID']; ?>"><?php echo
$row_list['Database_name']; ?></option>
                <?php
                }
                ?>
                </select>
            </p>
            <p style = "padding-left: 42%">
                 <input class="btn btn-primary" type="submit" value="Submit"
name="Submit" id="Submit" >
            </p>
        </form>
</fieldset>
```

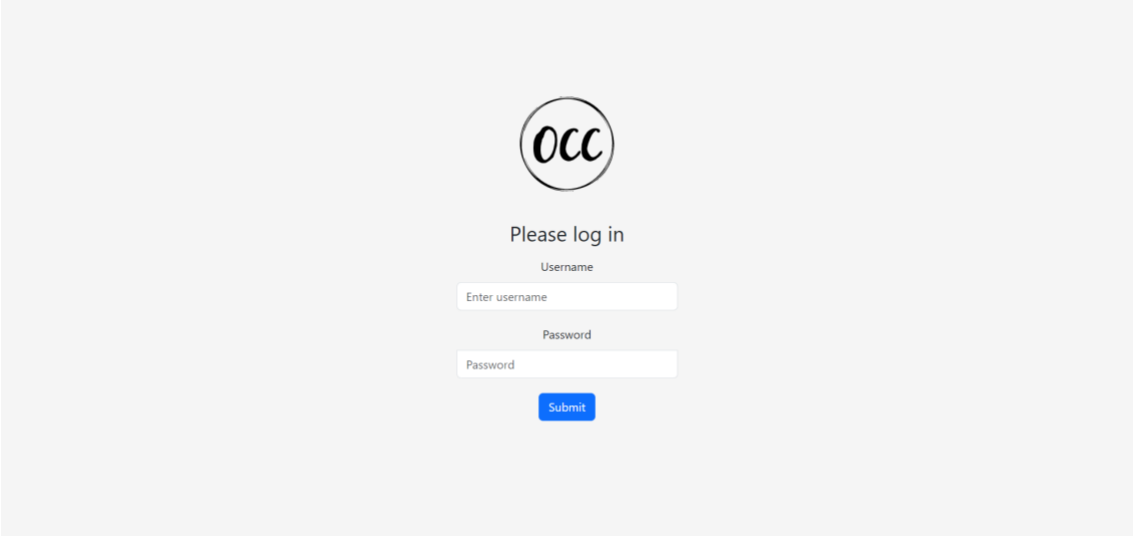
Slika 22: Kod za obrazac stranice "Settings" - 2

4.3.5 Stranica za prijavu

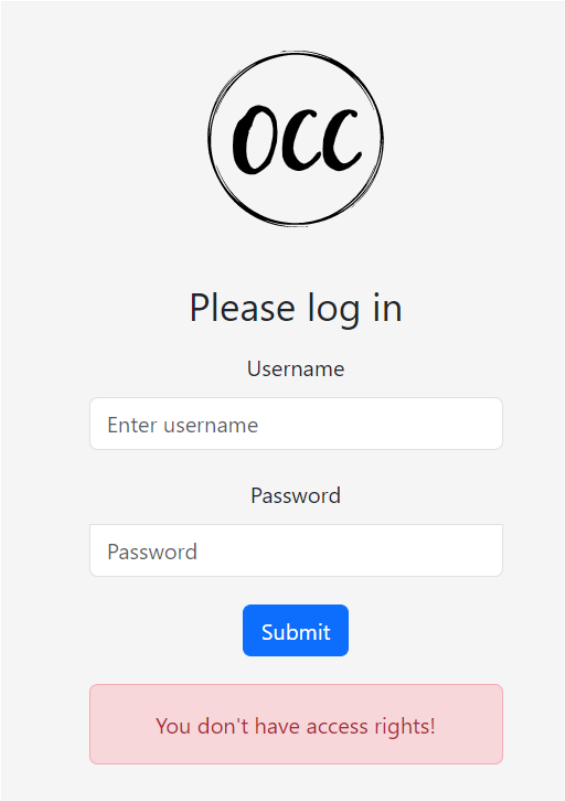
Stranica za prijavu (slika 23) je također implementirana pomoću gotovog Bootstrap tamplate-a s potrebnim prilagodbama.

Za prijavu se potrebno prvo spojiti na LDAP server. Nakon spajanja se provjerava postoji li bind user i ako postoji provjerava se je li korisnik u ActiveDirectoryu. Kada se provjeri postoji li korisnik slijedi pokušaj spajanja s dobivenim podacima. Spajanjem korisnik pretražuje sam sebe te ukoliko dođe do pogreške ispisuje se odgovarajuća poruka (slike 24 i 25). Uspješnim pronalaskom sebe provjerava se postoji li korisnik i u tablici Users te ako ne postoji odbija se prijava.

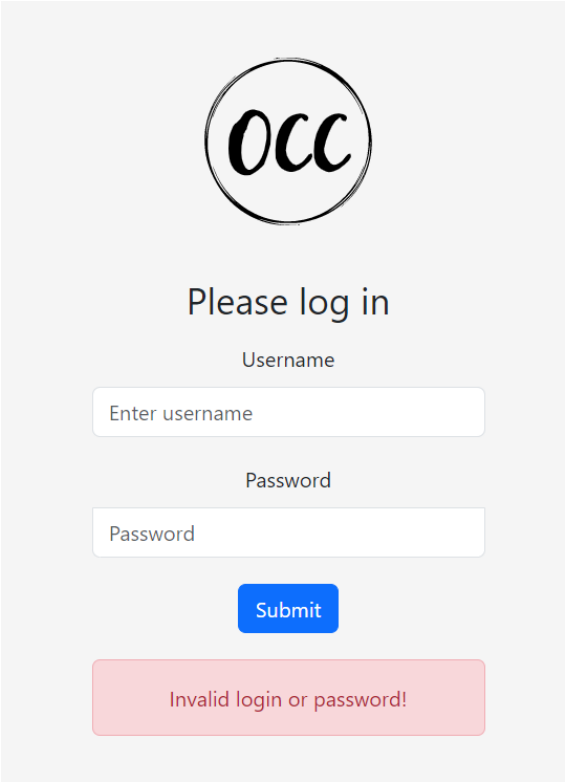
U slučaju uspješnog spajanja i pronalaska korisnika u varijablu 'user_id' unutar sesije se stavlja korisnikovo ime kako bi se znalo da je korisnik prijavljen te se preusmjerava na „Home“ stranicu.



Slika 23: Stranica za prijavu



Slika 24: Poruka pogreške pri upisu username-a bez privilegija



Slika 25: Poruka pogreške pri upisu krive lozinke

4.3.6 Pomoćne datoteke

4.3.6.1 Config.php

Datotoka config.php sadrži definiciju varijabli potrebnih za spajanje na bazu i njezin sadržaj se nalazi na slici 26.

```php
<?php
define("hostName", 'localhost');
define("userName", 'occ');
define("password", '***************');
define("databaseName", 'occ');
define("tableName", 'processors');
?>
```

Slika 26: Sadržaj dototeke config.php

4.3.6.2 Database.php

Datoteka database.php sadrži kod (slika 27) za spajanje na bazu kako ne bi bilo potrebno ponavljati isti kod kroz veći broj datoteka. Potrebno je uključiti datoteku config.php te s naredbom mysqli() otvoriti novu vezu na MySQL server. Zatim se provjerava veza i u slučaju greške se ona ispisuje.

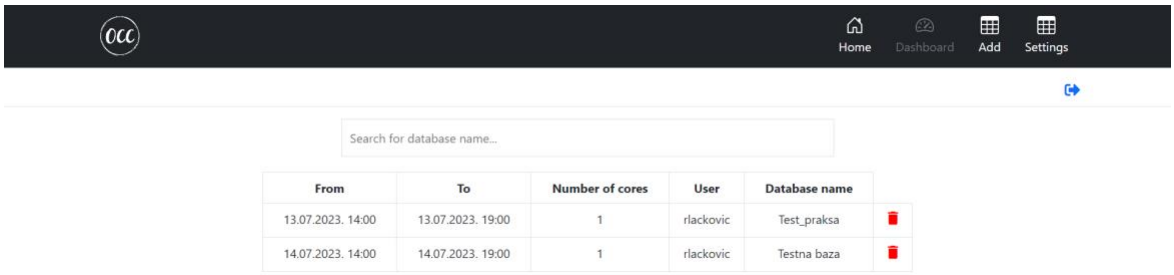
```php
<?php
include("config.php");
$conn = new mysqli(hostName, userName, password, databaseName);
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
?>
```

Slika 27: Sadržaj datoteke database.php

4.3.6.3 Delete.php

Delete.php datoteka služi za brisanje podatka iz tablice Processors. Prijavom u aplikaciju se na stranici „Dashboard“ u tablici doda stupac s ikonom kante (slika 28) te klikom na nju se briše željeni podatak. Na slici 29 je kod za prikaz simbola kante u datoteci table.php stranice „Dashboard“. Provjera se postoji li vrijednost varijable 'user_id' u sesiji te ako postoji prikazuje se simbol kante. Klikom na kantu se pojavljuje dijalog za potvrdu brisanja te kada korisnik potvrdi brisanje poziva se datoteka delete.php kojoj se prosljeđuju vrijednosti:

- ID reda, odnosno podatka
- From
- To
- Database ID



Slika 28: Stranica "Dashboard" nakon prijave

```
<?php
if (isset($_SESSION['user_id'])) {
    echo '<td> <a href="delete.php?ID=' . $data["ID"] . '&From=' .
$data['From'] . '&To=' . $data['To'] . '&Database=' .
$data["Database_id"] . '"
    onclick="return confirm(\'Are you sure to delete?\')"> <i
class="material-icons" style="font-size: 23px; color:
red;">delete</i></a>';
}
?>
```

Slika 29: Kod za prikaz ikone smeća

U datoteci delete.php (slika 30) se izvršava DELETE naredba nad bazom. Također, bilježi se u tablicu Logs akcija brisanja, gdje se spremaju podaci tko je obrisao podatak i koji. Nakon uspješnog brisanja se korisnik vraća na stranicu „Dashboard“.

```php
<?php
include("database.php");
session_start();
$log_user = $_SESSION['user_id'];
$id = $_GET['ID'];
$from = $_GET['From'];
$to = $_GET['To'];
$database_id = $_GET['Database'];
$query = "DELETE FROM processors WHERE ID = '$id';";
$result = mysqli_query($conn, $query);

$log_msg = 'Delete';
$details = $log_user . ' deleted time from ' . $from . ' to ' . $to . '
for ' . $database_id;

$sql = "INSERT INTO `logs` (`User`, `Action`, `Details`) VALUES
('$log_user', '$log_msg', '$details')";
$rs = mysqli_query($conn, $sql);

header("location: table.php");
?>
```

Slika 30: Sadržaj datoteke delete.php

4.3.6.4 Logout.php

Kada korisnik stisne ikonu za odjavu na bilo kojoj od stranica, datoteka logout.php (slika 31) se poziva kako bi se korisnik odjavio i sesija uništila.

Naredba koja se koristi u datoteci logout.php je session_destroy(). Ova naredba briše sve podatke o sesiji, uključujući sve varijable koje su vezane za tu sesiju. To znači da će se svi podaci o korisniku i njegovoj prijavi ukloniti iz aktivne sesije.

Nakon što se sesija uništi korisnika se preusmjerava na početnu stranicu „Home“ naredbom header('Location: index.php').

```php
<?php
session_start();
session_destroy();
header('Location: index.php');
exit();
?>
```

Slika 31: Sadržaj datoteke logout.php

4.3.6.5 Processors.php

Kada korisnik unese podatke u obrazac na stranici "Add", datoteka processors.php (slika 32) se poziva kako bi se ti podaci dodavali u bazu podataka.

U datoteci processors.php prvo se dohvaćaju podaci uneseni u obrazac. To se radi koristeći $_POST.

Nakon što su podaci dohvaćeni, izvršava se naredba INSERT nad tablicom Processors. Ova naredba se izvršava koristeći funkcija mysqli_query() koja izvršava SQL upit na bazi podataka.

Nakon uspješnog dodavanja podataka u tablicu Processors, u tablicu Logs se bilježi koji je korisnik unio podatak. Podatak o tome koji je korisnik unio podatak se dobije iz varijable 'user_id' spremljene u sesiji.

```php
<?php
include("database.php");
session_start();
$datFrom = $_POST['datFrom'];
$datTo = $_POST['datTo'];
if($_POST['numCores'] == null)
    $numCores = 1;
else $numCores = $_POST['numCores'];
$ID_User = $_SESSION['user_id'];
$database_id=$_POST['database'];
$log_details = $log_user . ' added time from ' .$datFrom . ' to ' .
$datTo . ' for database ' . $database_id;
$sql = "INSERT INTO `processors` (`From`, `To`, `Number_of_cores`,
`ID_User`, Database_id) VALUES ('$datFrom', '$datTo', '$numCores',
'$ID_User', '$database_id')";
$rs = mysqli_query($conn, $sql);
if($rs)
{
    $log_msg = 'Add';
    $sql = "INSERT INTO `logs` (`User`, `Action`, `Details`) VALUES
('$ID_User', '$log_msg', '$log_details')";
    $rs = mysqli_query($conn, $sql);
    header('Location: add.php');
}
?>
```

Slika 32: Sadržaj datoteke processors.php

4.3.6.6 Settings.php

Kada se na stranici „Settings“ žele unijeti podaci za zadani način rada pojedine baze, datoteka settings.php (slike 33 i 34) se poziva kako bi se ti podaci ažurirali u bazi podataka.

U datoteci settings.php, prvo se dohvaćaju podaci uneseni u obrazac na stranici "Settings". Kao i u prethodnom primjeru, to se radi koristeći $_POST.

Nakon što su podaci dohvaćeni, izvršava se naredba UPDATE SET nad tablicom Settings. Ova naredba se izvršava koristeći funkcija mysqli_query().

Podaci se mijenjaju na temelju uvjeta, kao što su dan u tjednu i identifikator baze.

```php
<?php
include("database.php");
session_start();

$dayinweek = $_POST['dayinweek'];
$timeFrom = $_POST['timeFrom'];
$timeTo = $_POST['timeTo'];
$database_id = $_POST['database'];
if($_POST['numCores'] == null)
    $numCores = 1;
else $numCores = $_POST['numCores'];
$ID_User = $_SESSION['user_id'];
$log_details = $ID_User . ' changed time from ' . $timeFrom . ' to ' .
$timeTo . ' for ' . $dayinweek . ' for Database ' . $database_id;

if($timeFrom == NULL)
{
    $sql = "UPDATE `settings` SET `From`=NULL, `To`=NULL,
Num_of_cores=NULL WHERE Day_in_week='$dayinweek'";
}
else{
    $sql = "UPDATE `settings` SET `From`='$timeFrom', `To`='$timeTo',
Num_of_cores='$numCores' WHERE Day_in_week='$dayinweek' AND
Database_id='$database_id'";
}
```

Slika 33: Sadržaj datoteke settings.php - 1

```php
$rs = mysqli_query($conn, $sql);
if($rs)
{
    $log_msg = 'Update settings';
    $sql = "INSERT INTO `logs` (`User`, `Action`, `Details`) VALUES
('$ID_User', '$log_msg', '$log_details')";
    $rs = mysqli_query($conn, $sql);
    header('Location: add-settings.php');
}
?>
```

Slika 34: Sadržaj datoteke settings.php - 2

4.4 Definiranje job-ova

Nakon što se lokalno kreirala aplikacija bilo je potrebno prebaciti istu na server. Kada je aplikacija prebačena na server definirane su skripte job.php i job-2.php koje služe za provjeru rada baza i mijenjanje stanja istih ukoliko je to potrebno.

Datoteka job.php se koristi za rad s testnom bazom, dok se job-2.php koristi za rad s produkcijskom. S obzirom da se produkcijska baza ne gasi, u datoteci job-2.php nema mogućnost gašenja baze nego se samo provjerava koliko jezgri bi u određenom trenutku trebalo raditi te se broj jezgri mijenja po potrebi. Testna baza ima jednu jezgru pa u datoteci job.php ne postoji provjera broja jezgri nego samo treba li baza biti upaljena ili ugašena.

U Windows Task Scheduler su definirani zadaci koji pokreću datoteke job.php i job-2.php te se ti zadaci obavljaju svakih 5 minuta.

Iz tablica Processors i Settings se provjerava kako bi baza trebala raditi u određenom trenutku te kada se dobije stanje baze se ono po potrebi mijenja. Provjera stanja i mijenjanje istog se vrši Oracle Cloud Infrastructure command line interface-om. CLI pruža istu temeljnu funkcionalnost kao i Oracle Cloud konzola, plus dodatne naredbe. Neki od njih, kao što je mogućnost pokretanja skripti, proširuju funkcionalnost konzole.

Naredbe:

- Dobivanje stanja:
    - oci db autonomous-database get --autonomous-database-id
- Paljenje baze
    - oci db autonomous-database start --autonomous-database-id
- Gašenje baze
    - oci db autonomous-database stop --autonomous-database-id
- Mijenjanje broja jezgri
    - oci db autonomous-database update --autonomous-database-id … --cpu-core-count

5 Zaključak

Cilj ovog završnog rada je bio dokumentirati izradu web aplikacije za upravljanje bazama podataka. Na samom početku istog je objašnjenje tehnologija i alata koji su korišteni za sami razvoj, a zatim i opisan postupak izrade. Aplikaciju se može jednostavno i intuitivno koristiti za upravljanje radom baza unutar tvrtke.

Korisničko sučelje je stvoreno pomoću HTML-a i Bootstrapa. Serverske skripte su napisane u PHP-u, a baza podataka izgrađena pomoću DBeavera lokalno i phpMyAdmina kada je aplikacija prebačena na server.

U daljnjem razvoju aplikacije se može dodati definiran rad baza tijekom praznika te prikaz broja jezgri koji treba raditi u određenim trenucima na grafovima na stranici „Home“.

6 Literatura

AWS. (2023). Preuzeto 6. srpnja 2023 iz What Is A Web Application?: https://aws.amazon.com/what-is/web-application/

AWS. (2023). Preuzeto 7. srpnja 2023. iz What Is Javascript (JS)?: https://aws.amazon.com/what-is/javascript/

B., R. (31. siječnja 2023). *Hostinger Tutorials*. Preuzeto 10. sprnja 2023. iz What is MySQL: MySQL Explained For Beginners: https://www.hostinger.com/tutorials/what-is-mysql#So_What_is_MySQL

Bhargava, R. (11. kolovoza 2021). *jumpcloud*. Preuzeto 10. sprnja 2023. iz What Is LDAP? The Ultimate Guide: https://jumpcloud.com/blog/what-is-ldap

Bootstrap. (2023). Preuzeto 7. srpnja 2023. iz History: https://getbootstrap.com/docs/4.0/about/history/

DBeaver. (2023). Preuzeto 10. srpnja 2023. iz Company: https://dbeaver.com/company/

Harris, P. (2017.). *What Is HTML Code?* The Rosen Publishing Group. Preuzeto 6. srpnja 2023.

Haverbeke, M. (2018.). *Eloquent JavaScript: A Modern Introduction to Programming.* No Starch Press.

javatpoint. (2023). Preuzeto 7. srpnja 2023. iz Learn JavaScript Tutorial: https://www.javatpoint.com/javascript-tutorial

Javatpoint. (2023). Preuzeto 10.. srpnja 2023. iz XAMPP TUTORIAL: https://www.javatpoint.com/xampp

Khan, S. (15. ožujka 2021). *GENERAL ASSEMBLY*. Preuzeto 7. sprnja 2023. iz WHAT IS THE POWER OF JAVASCRIPT USED FOR?: https://generalassemb.ly/blog/what-is-the-power-of-javascript-used-for/

Kyrnin, J. (2011.). *Sams Teach Yourself HTML5 Mobile Application Development in 24 Hours.* Sams Publishing. Preuzeto 6. srpnja 2023.

Mmeje, C. (14. ožujka 2021). *Sensu*. Preuzeto 10. srpnja 2023. iz What is LDAP and how does it work?: https://sensu.io/blog/what-is-ldap

onelogin. (2023). Preuzeto 10. srpnja 2023. iz What is LDAP?: https://www.onelogin.com/learn/what-is-ldap

Oracle Cloud Infrastructure. (2023). Preuzeto 10. srpnja 2023. iz What is MySQL?: https://www.oracle.com/mysql/what-is-mysql/

Pavlić, M. (2011). *Informacijski sustavi.* Školska knjiga. Preuzeto 12. srpnja 2023.

Pavlić, M. (2011.). *Oblikovanje baza podataka.* Rijeka: Odjel za informatiku Sveučilišta u Rijeci. Preuzeto 12. srpnja 2023.

phpMyAdmin. (2023). Preuzeto 10. srpnja 2023. iz About: https://www.phpmyadmin.net/

Prettyman, S. (2020). *Learn PHP 8: Using MySQL, JavaScript, CSS3, and HTML5.* Preuzeto 10. srpnja 2023.

StackPath. (2023). Preuzeto 6. srpnja 2023. iz WHAT IS A WEB APPLICATION?: https://www.stackpath.com/edge-academy/what-is-a-web-application/

W3C. (28. listopada 2014). Preuzeto 6. srpnja 2023. iz Open Web Platform Milestone Achieved with HTML5 Recommendation: https://www.w3.org/press-releases/2014/html5-rec/

Zola, A. (kolovoz 2022). *TechTarget*. Preuzeto 7. srpnja 2023. iz Bootstrap: https://www.techtarget.com/whatis/definition/bootstrap

7 Popis slika

8 Popis tablica