Razvoj web aplikacije u okolini oblaka računala

Došlić, Antonio

Undergraduate thesis / Završni rad

2017

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:200:324362

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Preddiplomski sveučilišni studij računarstva

RAZVOJ WEB APLIKACIJE U OKOLINI OBLAKA RAČUNALA

Završni rad

Antonio Došlić

Osijek, 2017.

SADRŽAJ

1. UVOD 1
   1.1. Zadatak završnog rada 1
2. PRIKAZ SLIČNIH RJEŠENJA 2
3. PLAN I PRIKAZ VLASTITOG RJEŠENJA 3
   3.1. Plan rješenja 3
   3.2. Oblak računala 4
      3.2.1 Arhitektura oblaka računala 4
      3.2.2 Podjela oblaka prema modelu isporuke 4
      3.2.3 Modeli oblaka računala 5
   3.3. Parametri korišteni u web aplikaciji 5
      3.3.1 Težine parametara korištenih u web aplikaciji 5
4. PROGRAMSKO RJEŠENJE 7
   4.1. Pregled korištenih alata 7
      4.1.1. Microsoft Azure 7
      4.1.2. GitHub 8
      4.1.3. Razvojna okolina Bootstrap 9
      4.1.4. jQuery 10
      4.1.5. Programski jezik PHP 11
      4.1.6. AJAX 12
      4.1.7. MySQL 12
   4.2. Razvoj web aplikacije 13
      4.2.1. Kreiranje web aplikacije 13
      4.2.2. Povezivanje web aplikacije i GitHub projekta 15
      4.2.3. Kreiranje baze podataka i povezivanje s web aplikacijom 16
      4.2.4. Uređivanje baze podataka 19
   4.3. Pisanje koda web aplikacije 21
5. TESTIRANJE I ANALIZA WEB APLIKCIJE 29
   5.1 Testiranja sučelja za registraciju 29
   5.2 Testiranja sučelja za prijavu 30
   5.3 Testiranja web aplikacije 30
6. ZAKLJUČAK 34
LITERATURA 35

SAŽETAK 36
ŽIVOTOPIS 37
PRILOZI (na DVD-u) 38

1. UVOD

Zdravlje predstavlja glavni dio svakog ljudskog života i kao takvo ima neizmjernu važnost. Usprkost toj činjenici ljudi ga često zanemaruju. Jedan od najvećih problema zdravlja predstavlja dijabetes popularno nazvan tihi ubojica zbog svojih prekrivenih simptoma. Oko 35% svijetske populacije boluje od dijabetesa i nema nikakvih naznaka da se broj oboljelih neće i dalje povećavati. Znanstvenici čak predviđaju da će se broj oboljelih udvostručiti u vrlo kratkom vremeskom periodu. Kao rješenje takvoj situaciji znanstvenici potiču ljude da se "probude" i krenu voditi računa o svome zdravlju. Između ostalog to se ostvaruje educiranjem ljudi o bolesti, o njenim simtomima te o načinu liječenja ukoliko je prekasno za njeno spriječavanje. Tu je stvorena ideja da se napravi rad s kojim bi bila poboljšana svijest ljudi. Nakon dublje analize došlo se do konačne ideje o web aplikaciji s kojom bi bila mjerena rizičnost oboljevanja od dijabetesa na osnovu nekih parametara koje bi bile upisane od strane korisnika. Parametri koje korisnik upisuje bili bi spremljeni u bazu podataka kako bi promjene bile praćene u njegovom zdravstvenom ponašanju.

U drugom poglavlju obradit će se već postojeća rješenja koja obrađuju izračun rizičnosti oboljevanja od dijabetesa te spomenuti neke primjere. U trećem poglavlju govorit će se o planu i prikazu vlastitog rješenja. Nakon toga, u četvrtom poglavlju navest će se korišteni alati i tehnologije u izradi web aplikacije te opisati izradu iste. U zadnjem poglavlju bit će opisano testiranje web aplikacije te komentirati dobivena rješenja.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada je opisati postupak razvoja web aplikacije u okolini Microsoft Azure te izbor prikladnog predloška i programske tehnologije. Također, treba razraditi model skalabilne web aplikacije za podršku oboljelima od dijabetesa. Koristeći prikladne programske jezike i okvire, web aplikaciju treba programski ostvariti, omogućiti njen rad u oblaku računala, te je ispitati i analizirati.

2. PRIKAZ SLIČNIH RJEŠENJA

Na internetu postoje brojne web aplikacije koje su namjenjene računanju rizičnosti obolijevanja korisnika od dijabatesa, većinom na engleskom jeziku. Sve te web aplikacije, osim što su statične, imaju i vrlo skroman izbor parametara na osnovu kojih vrše izračun. Osim toga, te web aplikcije nisu namijenjene za višestruku uporabu zato što ne omogućuju korisniku pohranjivanje parametara koje su upisane te tako korisnik ne može pratiti promjene u svom zdravstvenom ponašanju. Uzmimo za primjer kalkulator za izračunavanje dijabetesa s web stranice [1]. Taj kalkulator nakon samo sedam parametara daje konačan izračun. Izračun je prikazan kao broj iz intervala 1-10. Drugi primjer je kalkulator sa stranice navedena na [2] koji se temelji na većem broju parametara, ali uz ograničenja. Neka od ograničenja bi bila da je web aplikacija namijenjena samo osobama starijim od 45 godina. Rješenje toga bi bilo maknuti nepotrebna ograničenja.

3. PLAN I PRIKAZ VLASTITOG RJEŠENJA

Web aplikacija koja je izrađena, bazirana je na već postojećim rješenjima koja su spomenuta u prethodnom poglavlju, ali za razliku od tih rješenja svi nedostatci su ispravljeni. Glavni nedostaci koji su nadoknađeni su statičnost, ograničenost i pohrana podataka.

3.1. Plan rješenja

Prva stvar na koju je obraćena pažnja su nedostaci prijašnjih rješenja i kako bi što bolje bili usklađeni u izradu rada. Kako bi to bilo ostvareno, povezivanje je planirano između web aplikacije i baze podataka. Osim toga bilo je potrebno prikupiti veći broj parametara kako bi bio dobiven stvarniji izračun rizičnosti obolijevanja od dijabetesa.

Nakon toga odabrani su alati, programi i razvojna okruženja najpogodnija za razvoj web aplikacije. Više o tome nalazi se na slici 3.1 na kojoj je prikazan dijagram toka plana i razvoja izrade web aplikacije.

Slika 3.1. Dijagram toka razvoja web aplikacije

3.2. Oblak računala

Za razvoj web aplikacije izabrano je razvojno okruženje u oblaku računala zbog toga što je svepopularnije i sveprisutnije uz brojne prednosti koje su ponuđene. Prednosti razvojnog okruženja u oblaku računala su da programi ne moraju biti kupljeni niti instalirani za izradu web aplikacije nego su dostupni na oblaku računala. Razvojno okruženje oblaka naplaćivano je metodom "pay as you go", odnosno naplaćiva se samo ono što je korišteno. Dodatna prednost razvojnog oblak okruženja je što je omogućeno skaliranje tako da su ostvarene bolje performanse i dostupnost uz manje operacijske troškove.

Oblak računala, prema [3], nije nova tehnologija, nego novi model koji već postojane tehnologije dovodi u poslovno okruženje na novi način. On je predstvaljen kao novi model kojim je osiguran pogodan pristup mreži na zahtjev raspodijeljenim resursima, koji mogu biti poslužitelji, mreže, aplikacije, usluge i memorije. Usluge koje se nalaze na oblaku računala su web orijentirane te su stoga lako dostupne brojnim uređajima putem interneta. Resursi su vrlo lako alocirani te zatim locirani putem zahtjeva.

3.2.1 Arhitektura oblaka računala

Prema [4], arhitektura oblaka računala sastoji se od četiri sloja: sklopovkog sloja, sloja infrastrukture, sloj platforme i sloj aplikacije. Zadaća sklopovskog sloja je upravljanje fizičkim resursima oblaka. Sloj infrastrukture sadržan je od računalni resursa i sustava za spremanje podataka. Sloj platforme sastojan je od operacijskih sustava i razvojnih okruženja za izradu aplikacija. Sloj aplikacije sastojan je od aplikacija.

3.2.2 Podjela oblaka prema modelu isporuke

Oblak računala pruža tri modela isporuke usluga: poslužitelj kao usluga (SaaS), Platforma kao usluga (PaaS) i Infrastruktura kao usluga (IaaS).

U SaaS-u korisnik iznajmljuje programe od strane proizviđača. Neke od prednosti SaaS-a su da je omogućen brži pristup novim tehnologijama, pristup podacima s bilo koje lokacije i lakše su predviđeni troškovi.

Model PaaS omogućuje aplikaciju ili razvojnu platformu u kojoj korisnici sami kreiraju aplikacije koje će pokrenuti na oblaku. Prednosti PaaS-a su da korisnik ne mora kupovati i instalirati programe potrebne za razvoj aplikacija.

IaaS je predstavena kao korištenje infrastrukture na bazi virtualni ili fizičkih resursa. Prednosti IaaS-a su da je omogućeno dinamičko skaliranje, više korisnika može istovremeno biti pristupljeno odgovarajućoj infrastrukturi te mnoge druge.

3.2.3 Modeli oblaka računala

Prema [4], osnovni modeli oblaka su: javni (eksterni), privatni (interni), hibridni i virtualni oblaci. Javni oblak predstavlja model kod kojeg je poslužitelji usluga pružaju svoje resurse kao javno dostupnim. Privatni oblak je formiran za potrebe jednog korisnika. Korisnik ima potpunu kontrolu nad podacima i kvalitetom usluge.Hibridni oblak sastojan je od kombinacije javnog i privatnog oblaka. Virtualni oblak (VPC) predstavljen je kroz platformu koja je postavljena na vrh javnog oblaka te je predstvaljena kao alternativno rješenje nedostatcima javnih i privatnih oblaka.

3.3. Parametri korišteni u web aplikaciji

Osim razvojnog okruženja i parametri imaju veliku ulogu pri izradi web aplikacije te ih je potrebno detaljnije opisati. Parametre su podijeljeni u tri skupite zbog jednostavnosti razumjevanja.

Prvu skupinu parametara čine osobni podaci, a to su godine, spol, indeks tjelesne mase i opseg struka.

Druga skupina parametara skrivena je iza naziva "Ponašanje i simptomi". Tu skupinu parametara čine sljedeći podaci: broj tjednih tjelesnih aktivnosti, konzumacija voća i povrća, povišena razina šećera u krvi, pušenje, polidpsija, poliurija, polifagija, stalan umor, nejasan vid, neobjašnjiv gubitak težine, problemi sa stopalima, rane koje ne zacjeljuju i nasljeđe. Najčešće su odgovori na ta pitanja DA/NE.

Zadnja skupina parametara nazvana je laboratorijski podaci koju čine sistolički tlak, dijastolički tlak, LDL kolesterol, HDL kolesterol i glukoza u krvi.

3.3.1 Težine parametara korištenih u web aplikaciji

Nisu svi parametri jednako važni, odnosno nemaju svi jednaku težinu. Parametri koji imaju najveću važnost pri izračunu rizika obolijenja od dijabetesa su primjereice kada je korisnik stariji od 64 godine, te ima poliuriju, polidipsiju, povišenu razinu šečera u krvi, glukozu u krvi veću od 5.72 i nekoga u bližoj obitelji koji već boluje od dijabetesa. Kako bi se lakše

dočarala, težina parametara je prikazana u tri slučaja skupova parametara koja prestavljaju nizak, srednji i visoki rizik oboljevanja od dijabetesa na osnovu najvažnijih parametara. Težina parametara izražena je u bodovima od nula do maksimalni pet bodova.

Tablica 3.1. Težina parametara web aplikacije

	Nizak rizik	Srednji rizik	Visoki rizik
Godine	<45	45-54	>64
Indeks tjelesne mase	<25	25-30	>30
Opseg struka (cm)	Muškarci<94 Žene<80	Muškarci 94-102 Žene 80-88	Muškarci>102 Žene>88
Povišena razina šećera u krvi	Ne	Ne	Da
Polidipsija	Ne	Ne	Da
Poliurija	Ne	Da	Da
Problemi sa stopalima	Ne	Da	Da
Nasljeđe	Ne	Dalja obitelj	Bliža obitelj
Sistolički tlak	<120	120-135	>135
Dijastolički tlak	<80	80-90	>90
Glukoza u krvi	<5.27	5.27-5.72	>5.72

4. PROGRAMSKO RJEŠENJE

4.1. Pregled korištenih alata

Alati korišteni za izradu web aplikacije su Microsoft Azure, GitHub, Bootstrap, jQuery, PHP, Ajax i MySQL baza podataka. U nastavku slijedi detaljniji opis tih alata.

4.1.1. Microsoft Azure

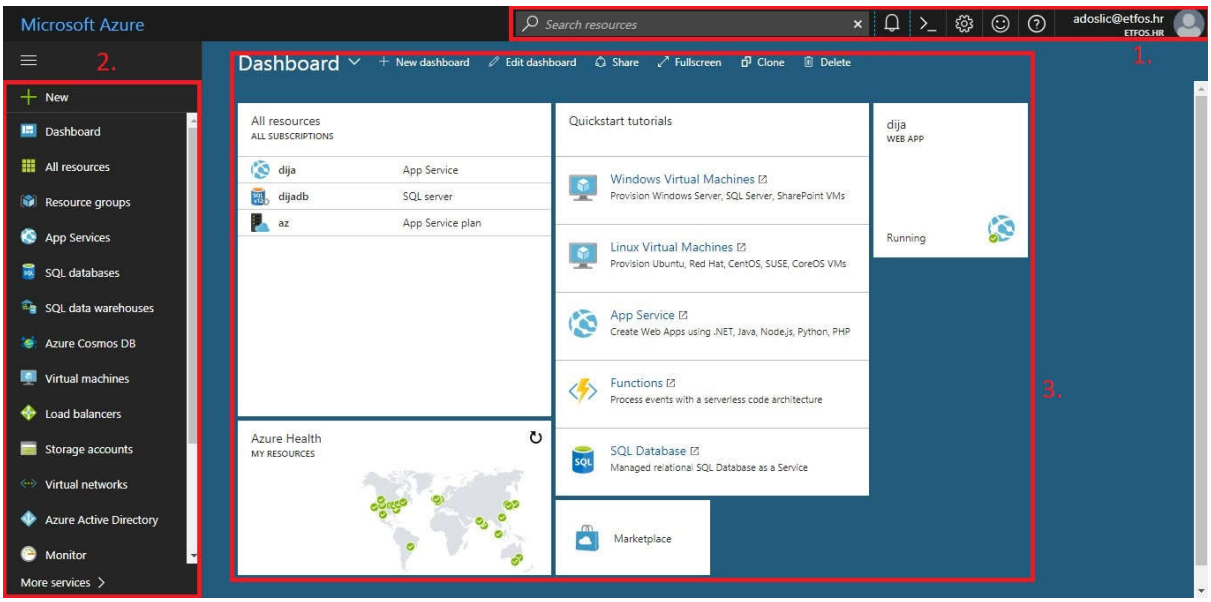
Microsoft Azure [4], prvobitno nazvan Windows Azure, je okolina oblaka računala koja je kreirana od strane Microsoft za izgradnju, testiranje, razvoj te upravljanje aplikacijama i uslugama preko mreže podatkovnih središta. Pružane usluge su poslužitelj kao usluga (SaaS), platforma kao usluga (PaaS), infrastruktura ako usluga (IaaS) te su podržavani brojeni programski jezici, alati i razvojna okruženja.

Microsoft Azure dostupan je u 38 zemalja i pruža preko 600 različitih usluga od kojih će biti spomenute samo najvažnije

- Windows ili Linux virtualno računalo je omogućeno pokretanjem IaaS modela
- Aplikacijske usluge, PaaS okruženje omogućuje razvojnim inženjerima lako objavljivanje i upravljanje web stranicama
- Omogućeno je kreiranje web aplikacija uz pomoć ASP.NET, PHP, Node.js ili Python koje mogu biti razvijene preko FTP-a, Git-a, Mercular-a, Team Foundation Poslužitelj ili drugih načina
- Podaci o ponašanju korisnika u stvarnom vremenu prikupljani su od strane mobilnog angažmana te je omogućeno obaviještavanje korisnika preko mobitela
- HockeyApp je korišten za razvoj, planiranje i testiranje beta inačica mobilnih aplikacija
- Usluge skladištenja pružane su od strane REST i SDK API za spremanje i pristupanje podatakaa na oblaku
- Koristi NoSQL nerelacijsku bazu podataka
- Redis Cache je implementacija Redisa na Microsoftovom oblaku
- Asinkrona komunikaciju između programa je omogućena od strane Queue-a
- Zadaću pohrane podataka između lokalnog računala i oblaka je obavljana od strane StorSimple-a

- SQL baza podataka rađena je na kreiranju, skaliranju i proširivanju aplikacija na oblak koristeći Microsoft SQL tehnologiju
- PaaS ponude mogu biti korištene za enkripciju i zaštitu sadržaja

Na slici 4.1 je prikazano sučelje Microsoft Azure. U prvom crvenom pravokutniku smještena je tražilica, ikona za obavijesti, postavke, podrška te osobni podaci korisnika. Drugi crveni pravokutnik predstavlja brzi izbornik. Zbog njega nam je omogućeno lako kreiranje web aplikacija, mobitnih aplikacija, baza podataka, stvaranje virtualni računala, pregled skladištenja podataka, virtualnih mreža te mnoge druge funkcije. Treći crveni pravokutnik predstavlja kontrolnu ploču. Na kontrolnu ploču mogu biti smješteni projekti i ostale važne stvari.



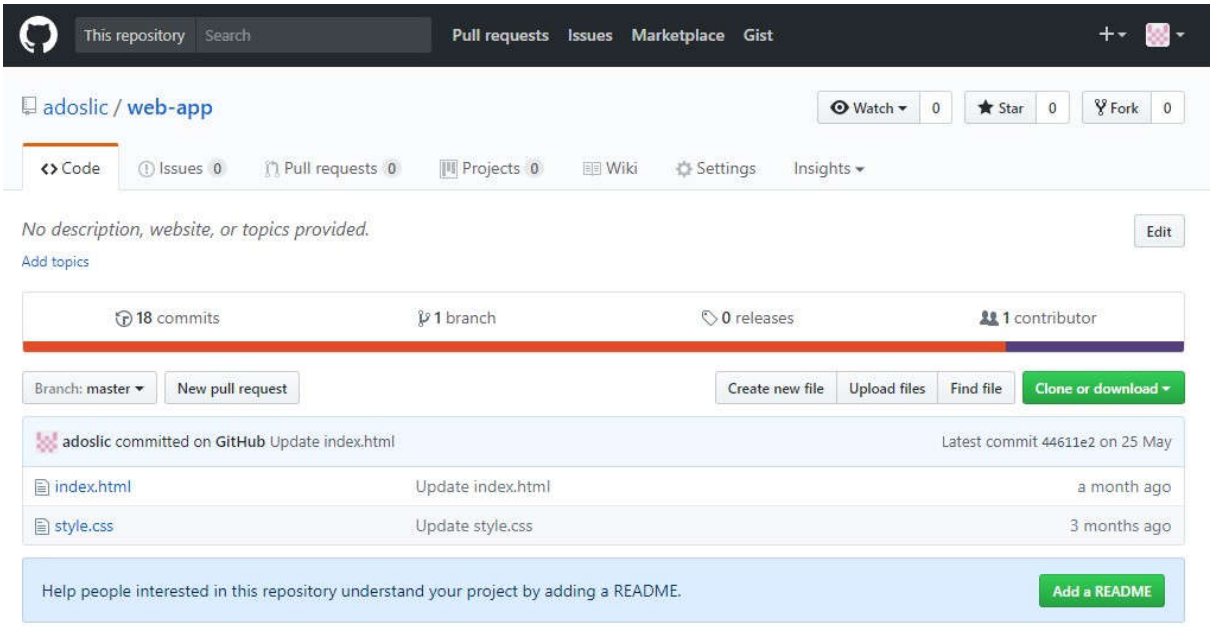
Slika 4.1. Microsoft Azure sučelje

4.1.2. GitHub

GitHub [5] je web platforma za kontroliranje skladištenja podataka i pružanje internetskih usluga. Najčešće je korištena za kodiranje. Osim osnovnih funkcija koje su podnuđene, kao što su kontroliranje raspodjele i upravljanje izvora koda, ponuđene su i neke GitHub-ove vlastite značajke. Omogućena je kontrola pristupa, praćenje greške u kodu, planiranje zadataka i druge funkcije. Ponuđeni su privatni i javni način skladištenja. Javni se koristi za projekte otvorenog koda.

Osim navedenih funkcija, omogućena je suradnja više ljudi na istom projektu te tako projekt može biti brže i učinkovitije ostvaren. Kako bi to bilo ostvareno, svaka promjena glavnog dokumenta je zabilježena. Ako je promjena valjana, novonastali dokument se sjedinjuje s glavnim.

NA slici 4.2 je prikazano GitHub sučelje, točnije pregled projekta pod nazivom "web-app". Vidljivo je da je projekt sastojan od dva dokumenta tipa html i css te kada je zadnji puta izvršena promjena nad tim dokumentima.



Slika 4.2. Izgled GitHub sučelja

4.1.3. Razvojna okolina Bootstrap

Bootstrap [6], originalno nazvan Twitter Blueprint, je besplatano front-end web okruženje za dizajniranje web stranica i web aplikacija. Sastojan je od HTML i CSS predložaka za formu, gumbove, navigaciju i ostalih komponenti za sučelje kao i od opcionalnih JavaScript predložaka. Razvijen je os strane Marka Ottoa i Jacobaa Thorntonaa koji su ujedno i osnivači Twittera. Bootstrap je razvijen na Twitteru kao predložak unutrašnjih alata, no nakon što je razvijen, objavljen je kao projekt otvorenog tipa 2011. godine.

Programski kod 4.1 prikazuje primjer bootstrap koda za izradu jednog gumba na kojem je napisano "Izračunaj". Osim toga vidljivo je da se taj gumb nalazi unutar **div** elemenata od kojih mu je omogućeno responzivnost, odnosno odgovarajući prikaz neovisno o veličini zaslona.

```
<div class="form-group col-xs-12 col-sm-10 col-md-8">
    <button class="btn btn-primary" id="finishBtn">Izračunaj</button>
</div>
```

Programski kod 4.1. Primjer bootstrap kod za izradu gumba

Do sada su objavljene četiri inačice te je najnovija 4.0. inačica. Od inačice 2.0. podržavan je responzivni web dizajn što bi značilo da je raspored web aplikacije dinamično prilagođavan uređaju bilo da se radi o tabletu, mobitelu ili računalu. Od inačice 3.0. podržavana je nova filozofija kojom su mobiteli stavljani na prvo mjesto svojim nenametljivim i progresvinim poboljšanjem. Inačica 4.0. ima nadogradnju na SaaS umjesto Less-a koji je do tada bio korišten.

4.1.4. jQuery

JQuery [7] je jedna od JavaScript biblioteka dizajnirana da pojednostavni pisanje HTML koda. Besplatan je program, no pod licencom MIT-a. Najrasprostranjenija je biblioteka JavaScript-a. Sintaksa jQuery-ja je dizajnirana da olakša navigiranje dokumenata, odabir DOM (Dokument Objekt Model) elemenata, kreiranje animacija, upravljanje događajima i razvoj Ajax aplikacija. Također, jQueryjem je omogućeno pisanje koda preko JavaScript biblioteke čime je programerima omogućeno stvaranje naprednih efekata, animacija i ostalih stvari koje čine aplikacije posebnima. Modularnim pristupom jQuery biblioteka omogućeno nam je stvaranje naprednih, ali ujedno i dinamičnih web stranica i web aplikacija. Skup njegovih osnovnih svojstava kao što su selekcija, obuhvaćanje i manipulacija DOM elemenata omogućujen je novi programerski stil kombiniranjem alogoritama i strukture DOM podataka.

Programski kod 4.2 predstava primjer jQuery koda koji traži element s **ID**-om "progress-bar" te mu biva promijenjen atribut "**data-percent**" u broj koji je dobiven izračunom **Math.round** funkcije.

```
$('#progress-bar').attr('data-percent',Math.round((bodovi/62)*100));
```

Programski kod 4.2. jQuery kod web aplikcije za promjenu atributa

JQuery je podržavan od strane Microsoft i Nokia koji je ugrađen u njihove platforme. JQurey je ugrađen u Visual Studio kako bi mogao biti korišten pri izradi ASP.NET AJAX i ASP.NET MVC aplikacija.

4.1.5. Programski jezik PHP

PHP [8] je skriptni programski jezik namjenjen razvoju dinamičnih web aplikacija. Prva inačica stvorena je 1995. godine te je do danas objavljeno 7 inačica. Trenutno najnovija inačica je 7.2. PHP je po sintaksi vrlo sličan C-u i Perl-u. Karakteriziran je sa velikim brojem ugrađenih funkcija koje su stavljene na raspolaganje korisnicima, ali osim njih funkcije mogu biti pisane i od strane korisnika. Osim velikog broja funkcija, PHP ima i petlje koje su dosta slične onima u C-u, a to su if petlja, while petlja, do...while petlja, for petlja i druge. Od PHP je omogućen rad s internet protokolima i bazama podataka. Sve popularne baze podataka su podražane kao što su ODBC, Oracle, MySQL i druge.

PHP kod je smješten unutar "<?php" i "?>" oznaka. Može biti umetnut u HTML kod ili stajati zasebno. Neovisno o tome je li umetnut u HTML kod ili stoji zasebno, PHP kod ne može biti viđen od strane korisnika.

Programski kod 4.3 prikazuje primjer PHP koda koji služi za spajanje na bazu podataka. Za spajanje na bazu podataka potrebno nam je ime poslužitelja, korisničko ime, lozinka te ime baze podataka na koju se spaja. Sve te nabrojene parametre predaju se funkciji mysqli_connect kako bi se spojili na bazu podataka.

```php
<?php
$servername = "eu-cdbr-azure-west-b.cloudapp.net";
$username = "bb037235b20d44";
$password = "c30237a7";
$dbname = "dijabetes";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

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

Programski kod 4.3. Primjer PHP kod korišten za spajanje na bazu podataka

4.1.6. AJAX

AJAX[10], predsstavljena skraćenica za asinkroni Javascript i XML. To je skup trikova namjenjenih razvoju weba koje koriste web tehnologije na klijentovoj strani za kreiranje asinkronih web aplikacija. S AJAX-om, podaci mogu biti poslani i primnjeni sa poslužitelja asinkrono (u pozadini) bez uplitanja u prikaz i ponašanje postojeće stranice. Razdvajanjem podataka razmjene i prikaza, sa AJAX-om je omogućena promjenu sadržaja web aplikacijama bez potrebe za učitavanjem cijele stranice. AJAX nije jedinstvena tehnologija, nego grupa tehnologija koja može biti kombinirana s HTML-om i CSS-om.

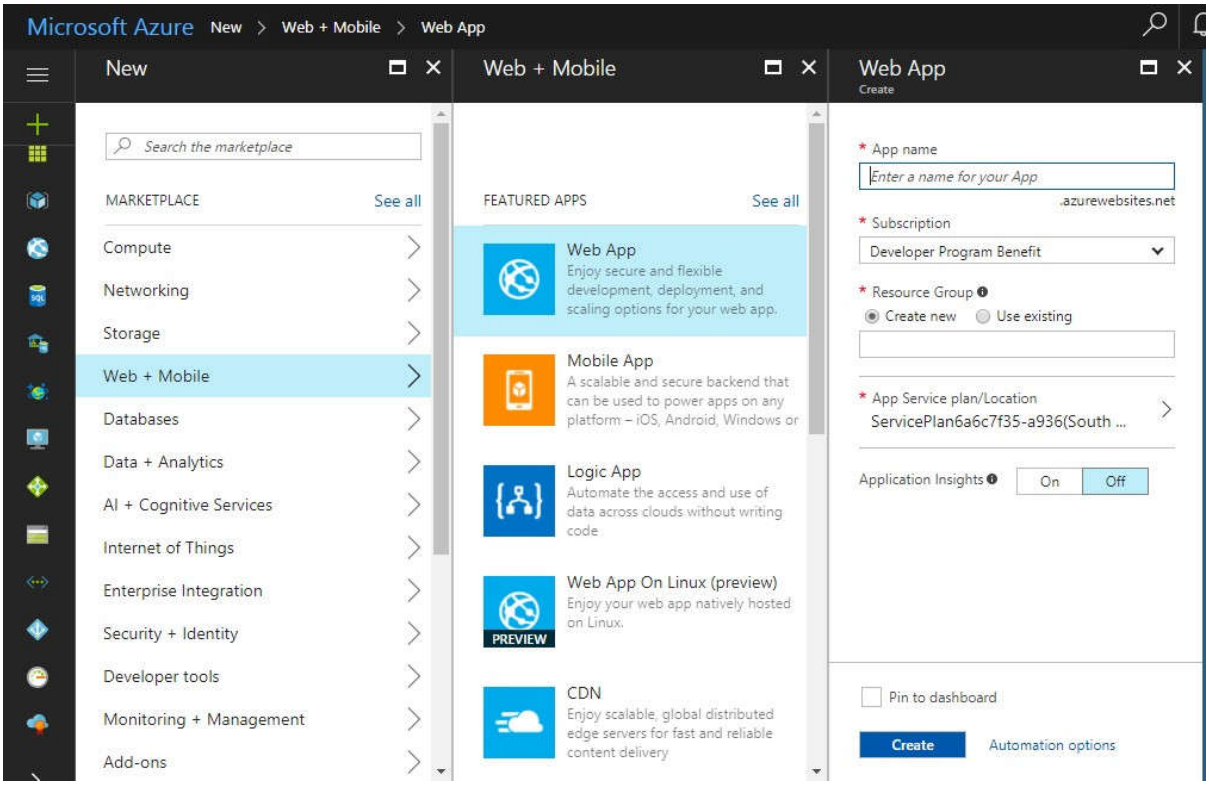
4.1.7. MySQL

MySQL [9] je sustav za upravljanje bazom podataka. MySQL baze su relacijskog tipa, koji je pokazan kao naljbolji način spremanja i pretraživanja velikih količina podataka. Kao takav predstavljen čini osnovu svakog informacijskog sustava. Jedan je od najzastupljenijih sustava u svijetu, a to je postignuto time što su besplatni za korištenje u većini slučajeva. Tijekom svog razvoja MySQL je bio spriječen poteškoćama jer su mu bile nedostajane neke od osnovnih funkcija definirane SQL standardom. Nedugo zatim je optimiran kako bi bio brz u funkcionalnosti. Ima podršku brojnih programskih jezika kao što su PHP, Java, Perl, Python i drugi. Trenutno najnovija inačica je 5.7.

4.2. Razvoj web aplikacije

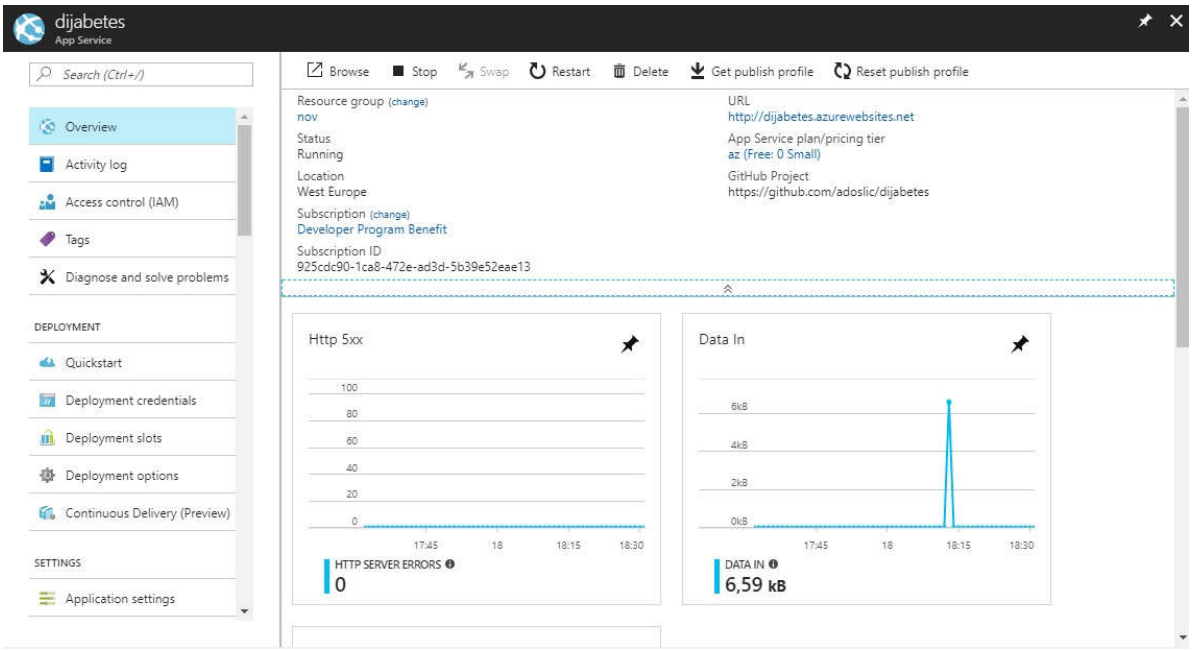
4.2.1. Kreiranje web aplikacije

Nakon istraživanja i odabira najprikladnijih alata za razvoj web aplikacije, možw se krenuti u izradu web aplikacije. Za izradu web aplikacije odabrana je razvojna okolina Microsoft Azure. Prvi korak je izradea korisničkog računa. Iz brzog izornika prikazanog i objašnjenog u prethodnom poglavlju na slici 4.1, odabrana je opcija "New", nakon toga "Web+Mobile", te "Web App" kako bi web aplikacija bila kreirana kao što je na slici 4.3 prikazano.



Slika 4.3. Prikaz izrade web aplikacije

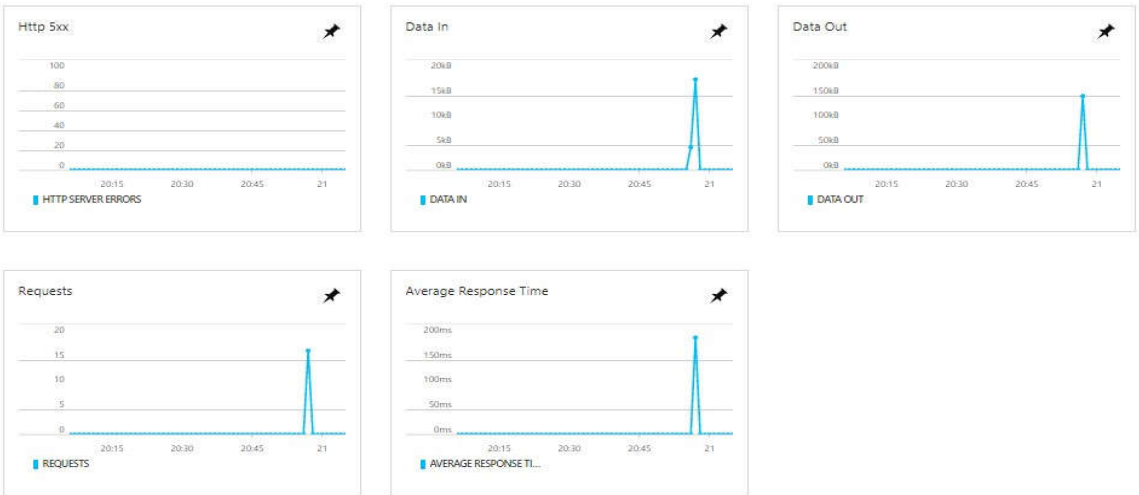
U zadnjem prozoru prikazanom na slici 4.3 bira se ime aplikacije, vrstu pretplate, grupu resursa i aplikacijski plan te zatim označi "Pin to dashboard" i klikne na "Create".Nakon toga, web aplikacija je prikazana na kontrolnoj ploči. Klikom na aplikaciju dobivene su detaljne informacije o njoj prikaze na slici 4.4.



Slika 4.4. Detaljan prikaz ostvarene web aplikacije

S lijeve strane je novi brzi izbornik prilagođen web aplikaciji. U gornjem prozoru vidljivi su podaci o web aplikaciji kao što su grupa resursa, status aplikacije, lokaciju poslužitelja na kojem su aplikacije pokrenute, vrsta pretplate, aplikacijski plan te url ostvarene aplikacije.

Na slici 4.5 prestavljeni su podaci kojima smo pratili rad web aplikacije. Pračeni su količina podataka koja dolazi/odlazi, broj zahtjeva te vrijeme odziva.



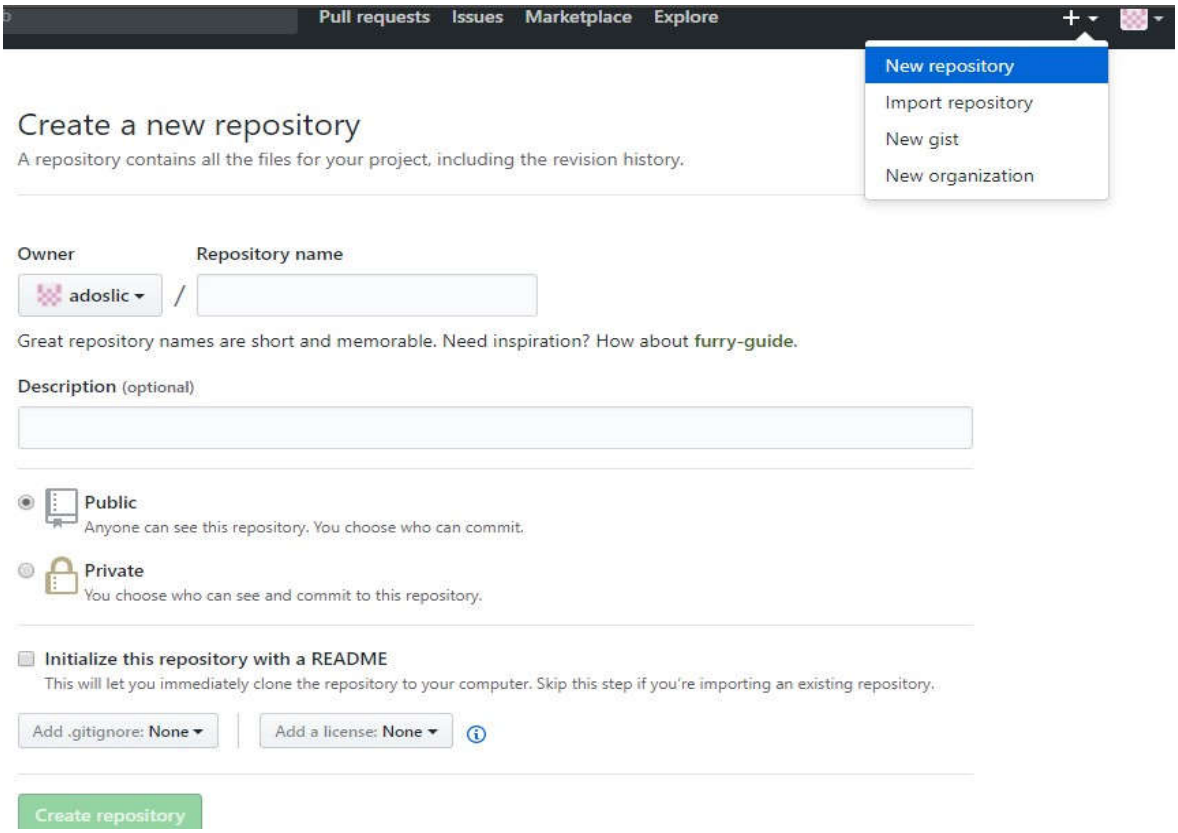
Slika 4.5. Praćenje rada web aplikacije

4.2.2. Povezivanje web aplikacije i GitHub projekta

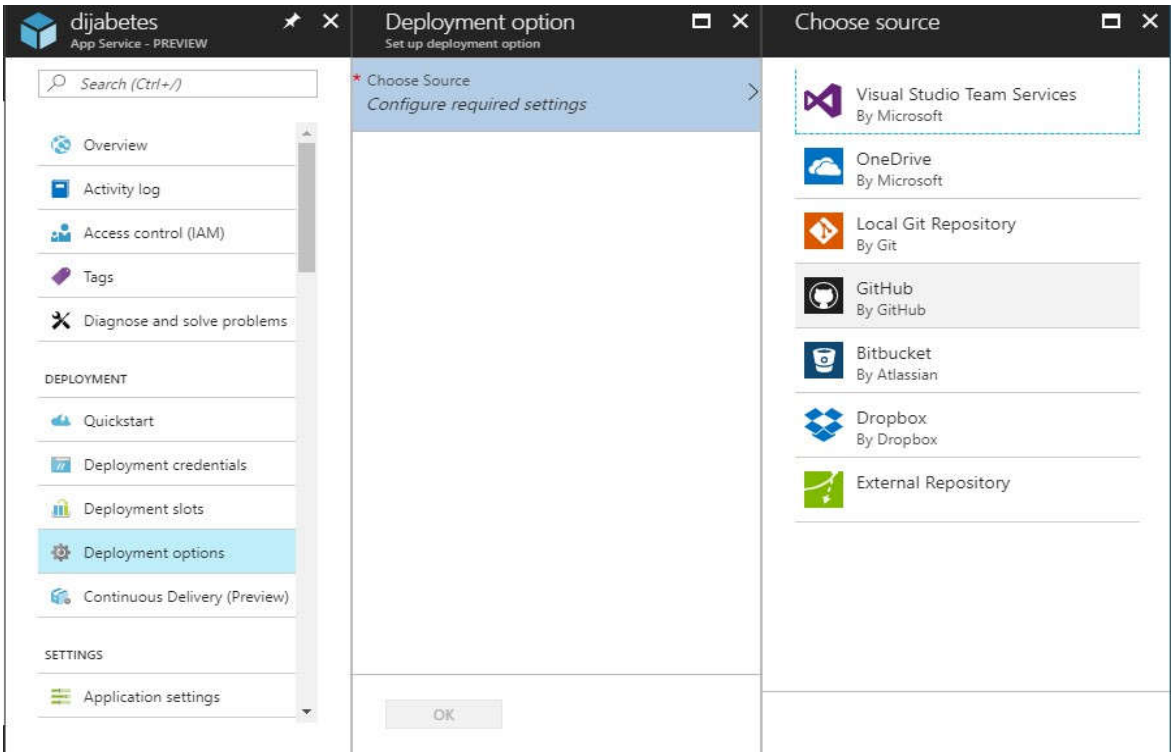
Budući da smo se odlučili potpuna izrada web aplikacije na oblaku računala web aplikacija biva povezana s izvorom koda za kojeg je izabran GitHub jer Microsoft Azure ne nudi mogućnost pisanja kodova u jezicima koji su potrebni. Prije povezivanja GitHub-a i web aplikacije na Microsoft Azure-u, potreban je korisnički račun na GitHubu i izrada projekta gdje će biti spremljeni svi dokumenti važni za izradu web aplikacije. Izrada korisničkog računa je vrlo jednostavna i standardna te je nema potrebe detaljnije objašnjavati, pa slijedi prikaz izrade projekta.

Nakon što je odabran "New repository" u gornjem desnom kutu otvoren je sadržaj prikazan na slici 4.6. Na slici 4.6 vidljivo je ime vlasnik projekta, mjesto za pisanje naziva projekta, opis projekta, odabir kome će projekt biti vidljiv i mogućnost stvaranja gotovih sigurnosnih dokumenata.

Nakon izrade projekta povezani smo s našom web aplikacijom. Povezivanje je vidljivo na slici 4.7.



Slika 4.6. Prikaz izrade GitHub projekta

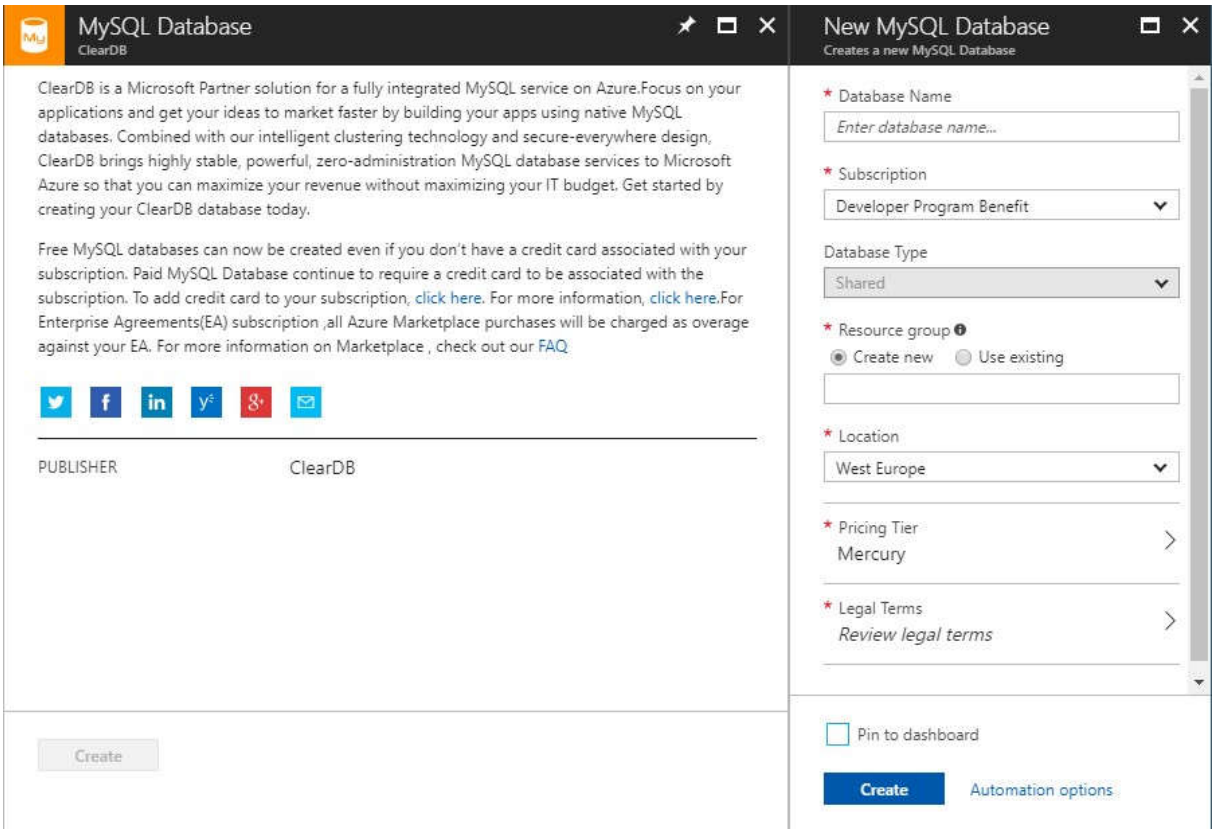


Slika 4.7. Spajanje web aplikacije na GitHub

Iz brzog izbornika s lijeve strane prikazanog na slici 4.7. odabere se "Deployment options". Zatim odabere se GitHub kao izbor koda.

4.2.3. Kreiranje baze podataka i povezivanje s web aplikacijom

Nakon spajanja web aplikacije i projekta na GitHub-u kreirana je baza podataka i povezana s web aplikacijom. Budući da se radi s MySQL bazom podataka, kreirana je tako da se u tražilici brzog izbornika upiše "MySQL Database" i odabere MySQL Database iz ponuđenog izbora. Nakon klika na MySQL Database otvoren je detaljniji opis baze podataka i obrazac za kreiranje baze podataka prikazan na slici 4.8. Obrazac prikazan u desnom prozoru na slici 4.8 sadrži polja za unos imena baze podataka, pretplatu, tip baze podataka, izvor grupe, lokaciju servera te uvijete korištenja.

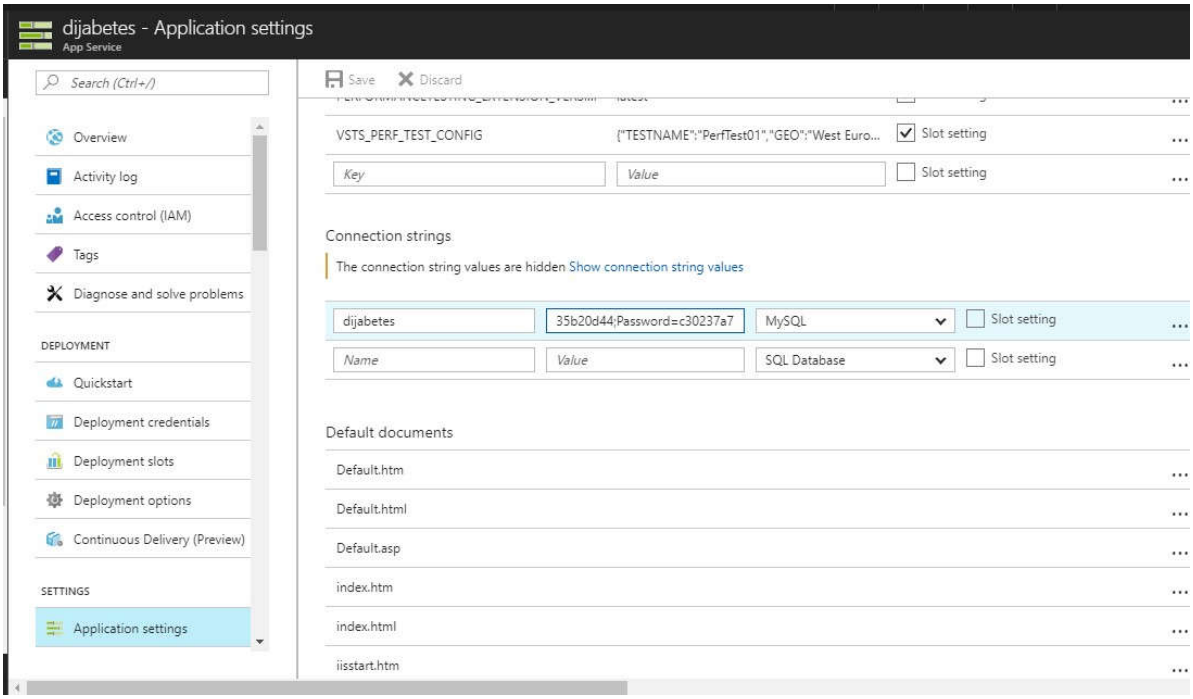


Slika 4.8. Prikaz izrade MySQL baze podataka

Nakon ispunjavanja obrasca i klikom na "Create" završeno je kreiranje baze podataka. Sljedeća stvar koja je napravljena je povezivanje baze podataka s web aplikacijom, a to se čini tako da se otvori novonastala baza podataka i odabere "Properties" iz brzog izbornika baze podataka. Zatim se otvora prozor prikazan na slici 4.9.

Na slici 4.9 prikazan je detaljan prikaz baze podataka. Vidljiv je status baze, lokacija poslužitelja, pretplata, ime baze podataka, korisničko ime i lozinka, te niz za povezivanje. Upravo taj niz je kopiran i zalijepljen na zato predviđeno mjesto u web aplikaciji. Mjesto gdje smo taj niz zalijepili je prikazan na slici 4.10.

Slika 4.9. Detalji kreirane baze podataka

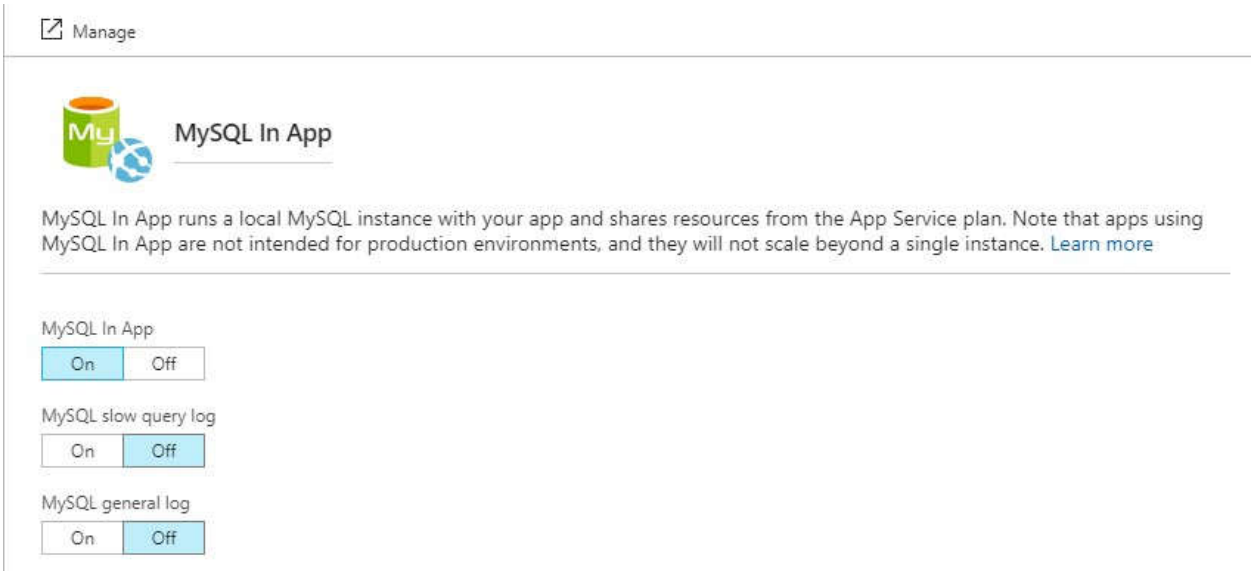


Slika 4.10. Prikaz povezivanja web aplikacije s bazom podataka

Nakon spremanja načinjenih izmjena u postavkama, povezivanje web aplikacije s bazom podataka je završeno.

4.2.4. Uređivanje baze podataka

Sljedeći korak je uređivanje baze podataka, a to je učinjeno klikom na "MySQL in App" iz brzog izbornika web aplikacije. U novootvorenom prozoru važno je da je postavka "MySQL in App" podešena na "On" kako bi našu baza podataka bila uređena. Taj novootvoreni prozor prikazan je na slici 4.11.



Slika 4.11. Postavke naše baze podataka

Klikom na "Manage" otvaren nam je phpMyAdmin okruženje u kojem je uređena naša bazu podataka. Baza podataka sastojana je od dvije tablice koje su međusobno povezane. Prva tablica naziva **korisnik** sprema podatke koje korisnik unosi pri izradi korisničkog računa, a to su ime, prezime, e-mail adresa i lozinka. Osim tih podataka u tablici se nalazi i korisnikID radi lakšeg razlikovanja korisnika koji je ujedno i primarni ključ. Druga tablica naziva **parametri** sprema parametre koje korisnik unosi. Tablica parametri osim parametara ima i podatak ID koji je primarni ključ, te podatak korisnikID_FK koji je strani ključ. Kao što je već rečeno, te dvije tablice su međusobno povezane, a to je ostvareno preko primarnog ključa korisnikID u tablici korisnik i stranog ključa korisnikID_FK u tablici parametri.

Na slici 4.12 je prikazan izgled tablice **korisnik**.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	korisnikID	int(11)			No	None		AUTO_INCREMENT	Change Drop Primary Unique Index More
2	ime	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
3	prezime	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
4	email	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
5	lozinka	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More

Slika 4.12. Izgled tablice korisnik

Na slici 4.13 je prikazan dio tablice **parametri**.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ID	int(11)			No	None		AUTO_INCREMENT	Change Drop Primary Unique Index More
2	korisnikID_FK	int(11)			No	None			Change Drop Primary Unique Index More
3	godine	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
4	spol	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
5	ITM	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
6	opseg	int(5)			No	None			Change Drop Primary Unique Index More
7	aktivnost	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
8	voce	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
9	secer	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
10	pusenje	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
11	polidipsija	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
12	poliurija	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
13	polifagija	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
14	umor	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
15	vid	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
16	tezina	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
17	stopala	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More
18	rane	varchar(255)	utf8_general_ci		No	None			Change Drop Primary Unique Index More

Slika 4.13. Izgled tablice parametri

Kao što je vidljivo na slikama 4.12 i 4.13 predviđen je tip podatka koji će biti spremljen u tablice kao i duljinu zapisa. Uzmimo za primjer ime korisnika kojemu najbolje odgovara tip varchar. Broj koji stoji u zagradi pokraj tipa podatka označava duljinu zapisa. Kod tipa varchar to ne znači da će biti zauzeto toliko mjesta u memoriji nego predstavlja maksimalno moguće mjesta u memoriji rezervirano za taj podatak.

Sljedeći korak u izradi web aplikacije bio je planiranje izgleda stranice i pisanje koda.

4.3. Pisanje koda web aplikacije

Kao što je spomenuto u jednom od prethodnih poglavlja, ako se želi ostvariti veća povezanost između korisnika i web aplikacije, a to će biti ostvareno tako što će korisniku biti omogućen ulazak u web aplikaciju samo ako ima korisnički račun. Budući da su postojana dva tipa korisnika tako su ponuđene dvije mogućnosti na početnoj stranici aplikacije. Prvi tip korisnika je "novi" tip koji prvi puta pristupa aplikaciji te budući da nema postojeći korisnički račun, potrebno mu je dati mogućnost za izradu istog. Drugi tip korisnika je "stari" tip koji je već pristupio aplikaciji tj. koji imaveć izrađen korisnički račun. Kao i prvom tipu, i drugom tipu je potrebno dati ono što mu je potrebno, odnosno ako već ima korisnički račun nema potrebe za izradom novoga nego mu omogućiti preusmjeravanje na stranicu za prijavu. Dakle, početna stranica sadrži dva dijela namjenjena za oba tipa korisnika. Novi korisnici klikom na gumb "registracija" budu preusmjereni na stranicu za kreiranje korisničkog računa, a stari korisnici klikom na gumb "prijava" budu preusmjereni na stranicu za prijavu.

Stranica za registriranje zahtjeva od korisnika popunjavanje pet polja, a to su ime, prezime, e-mail, lozinka te potvrda lozinke. Ispod polja za unos podataka nalazi se gumb "pošalji" kojemu je zadaća proslijediti unesene podatke bazi podataka. Ukoliko korisnik ne ispuni samo jedno od tih polja korisnički račun neće moći biti kreiran. Ukoliko korisnik unese netočnu potvrdu lozinke, također korisnički račun neće moći biti kreiran te će dobiti poruku o pogrešci. Ako korisnik unese već postojeće podatke, korisnički račun neće moći biti kreiran te će dobiti poruku o postojanju korisničkog računa.

Programski kod 4.4 prikazuje primjer koda jednog od polja za unos podataka. Kod je vrlo jednostavan, sastojan je od jednog elementa za unos koji se nalazi unutar nekoliko div elemenata. Zbog ljepšeg izgleda dodana je ikona pokraj polja za unos, a to je vidljivo u programskom kodu između elemenata **span**.

```
<div class="row">
    <div class="col-sm-12">
        <div class="form-group">
            <div class="input-group">
                <span class="input-group-addon"><span class="glyphicon glyphicon-envelope"></span></span>
<input type="email" name="email" id="email" class="form-control input-sm" placeholder="Upišite email adresu" required="required">
            </div>
        </div>
    </div>
</div>
```

Programski kod 4.4. Kod za unos adrese e-pošte

Kada korisnik ispuni sva polja za unos i klikne na gumb "pošalji" ti će podaci biti proslijeđeni bazi podataka, a da bi to bilo tako prvo mora biti ostvarena konekcija između baze podatka i našeg koda. Kod s kojim je to ostvareno prikazan je u jednom od prethodnih poglavlja u programskom kodu 4.3. gdje je i detaljno opisan.

Kada je konekcija s bazom podataka ostvarena, napravljen je dokument koji će biti aktiviran nakon što korisnik klikne na gumb "pošalji". Izgled toga dokumenta je vidljiv u programskom kodu 4.5.

```php
<?php
$error=''; // Variable To Store Error Message
if(isset($_POST['submit'])){

    $first_name = $_POST['first_name'];
    $last_name = $_POST['last_name'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $password_confirmation = $_POST['password_confirmation'];

        if($password==$password_confirmation){
            $encrypted = md5($password);
            $sql = "SELECT * FROM korisnik WHERE ime = '$first_name' AND prezime = '$last_name' AND email = '$email' AND loz
            $query = mysqli_query($conn, $sql);
            $rows = mysqli_num_rows($query);

            if($rows>0){
                $error = "Korisnik već postoji";
            }
            else{
                $sql = "INSERT INTO korisnik (ime, prezime, email, lozinka)
                VALUES ('$first_name','$last_name','$email','$encrypted')";

                if(!mysqli_query($conn,$sql)){
                    $error = "Greška s bazom";
                }else{
                    header("location: login.php");
                }
            }
        }else{
            $error = "Netočna potvrda lozinke";
        }
    mysqli_close($conn); // Closing Connection
}
?>
```

Programski kod 4.5. Izgled koda koji šalje podatke u bazu podataka

U programskom kodu je prikazano spremanje unesenih vrijednosti u varijable, te prosljeđivanje tih varijabli u bazu podataka ako su zadovoljeni uvjeti. Prvi uvjet je da je lozinka ista kao potvrda lozinke, a drugi da se uneseni paramtri ne nalaze već u bazi podataka. U programskom kodu 4.5 je vidljivo korištenje funkcije md5 koja ima ulogu kriptiranja, u našem slučaju lozinke, iz sigurnosnih razloga. Ukoliko nije bilo grešaka tijekom slanja podataka u bazu podataka korisnik će biti preusmjeren na stranicu za prijavu.

Stranica za prijavu zahtjeva od korisnika unos dva polja, a to su adresa e-pošte i lozinka. Ispod tih polja za unos nalazi se gumb "prijava" kojemu je zadaća provjeravanje podatke iz

unesenih polja s podacima u bazi podataka. Ukoliko korisnik ne ispuni neko polje, neće biti ptijavljen te će dobiti poruku o pogrški. Ukoliko korisnik unese netočne podatke dobit će poruku o netočnim podacima.Ukoliko korisnik sva polja za unos ispuni točno i klikne na gumb "prijava" bit će aktiviran dokument kojemu je zadaća podatke iz unesenih polja provjeriti s podacima u bazi podataka. Izgled toga koda je vidljiv u programskom kodu 4.6.

```php
<?php
session_start(); // Starting Session
$error=''; // Variable To Store Error Message
if(isset($_POST['submit'])){
    if(empty($_POST['email']) || empty($_POST['password'])){
        $error = "Netočan email ili lozinka";
    }else{
    // Define $email and $password
    $email=$_POST['email'];
    $password=$_POST['password'];

    // To protect MySQL injection for Security purpose
    $email = stripslashes($email);
    $password = stripslashes($password);
    $email = mysqli_real_escape_string($conn, $email);
    $password = mysqli_real_escape_string($conn, $password);
    $password = md5($password);

    // SQL query to fetch information of registerd users and finds user match.
    $sql = "SELECT * FROM korisnik WHERE lozinka='$password' AND email='$email'";
    $query = mysqli_query($conn, $sql);
    $rows = mysqli_num_rows($query);
    if ($rows == 1){
        $row = mysqli_fetch_row($query);
        $_SESSION['user_id']= $row[0];
        $_SESSION['ime']= $row[1];
        $_SESSION['prezime']= $row[2];

        header("location: firstPage.php"); // Redirecting To Other Page
    }else{
        $error = "Netočan email ili lozinka";
    }
    mysqli_close($conn); // Closing Connection
    }
}
?>
```

Programski kod 4.6. Izgled koda koji provjerava unesene podatke s podacima u bazi podataka

U kodu je prikazano spremanje vrijednosti unesenih polja u varijable, te korištenje sigurnosnih funkcija **stripslashes**, **mysqli_real_escape_string** i **md5**. Funkcija **stripslashes** uklanja znak "/" iz unosa, a funkcija **mysqli_real_escape_string** izbjegava specijalne

znakove iz unosa i time je spriječen neovlašteni upad u web aplikaciju. Funkcija **md5** je objašnjena u prethodnom poglavlju. Nakon tih funkcija provjeravaju se uneseni podaci s podacima u bazi podataka te ukoliko se podaci podudaraju korisnik će bit prosljeđen u web aplikaciju.

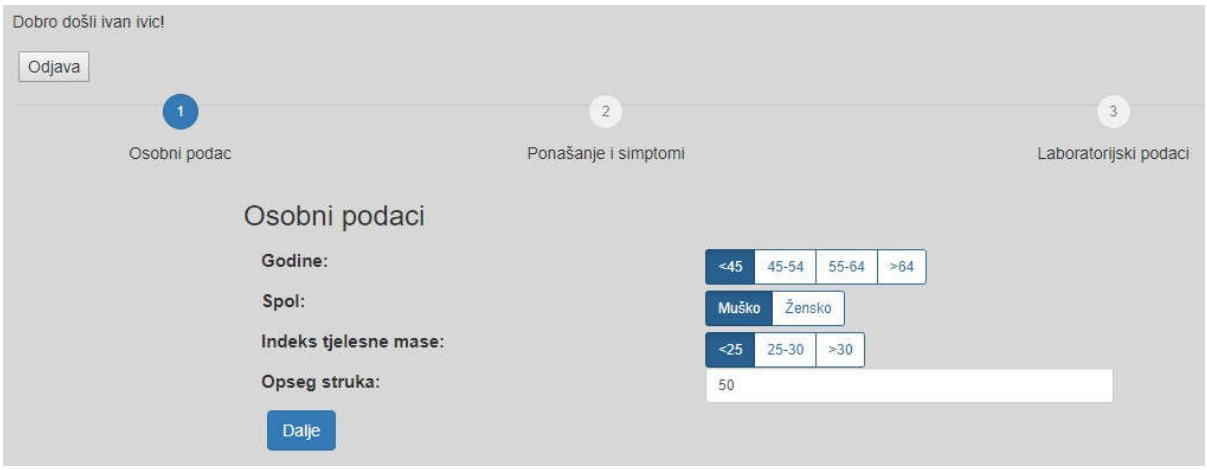
Kada korisnik uđe u web aplikaciju u gornjem lijevom kutu piše njegovo ime i prezime te ispod toga gumb za odjavu. U programskom kodu 4.7 prikazan je primjer koda za unos parametara.

```
<div class="form-group">
    <label for="spol" class="col-xs-12 col-sm-6 col-md-6 control-label">Spol:</label>
    <div class="col-sm-6 col-md-6 pull-right">
        <div class="input-group">
            <div id="radioBtn" class="btn-group spol">
                <a class="btn btn-primary btn-sm active" data-toggle="spol" data-title="M">Muško</a>
                <a class="btn btn-primary btn-sm notActive" data-toggle="spol" data-title="Z">Žensko</a>
            </div>
            <input type="hidden" name="spol" id="spol" required="required">
        </div>
    </div>
</div>
```

Programski kod 4.7. Primjer koda za unos parametara

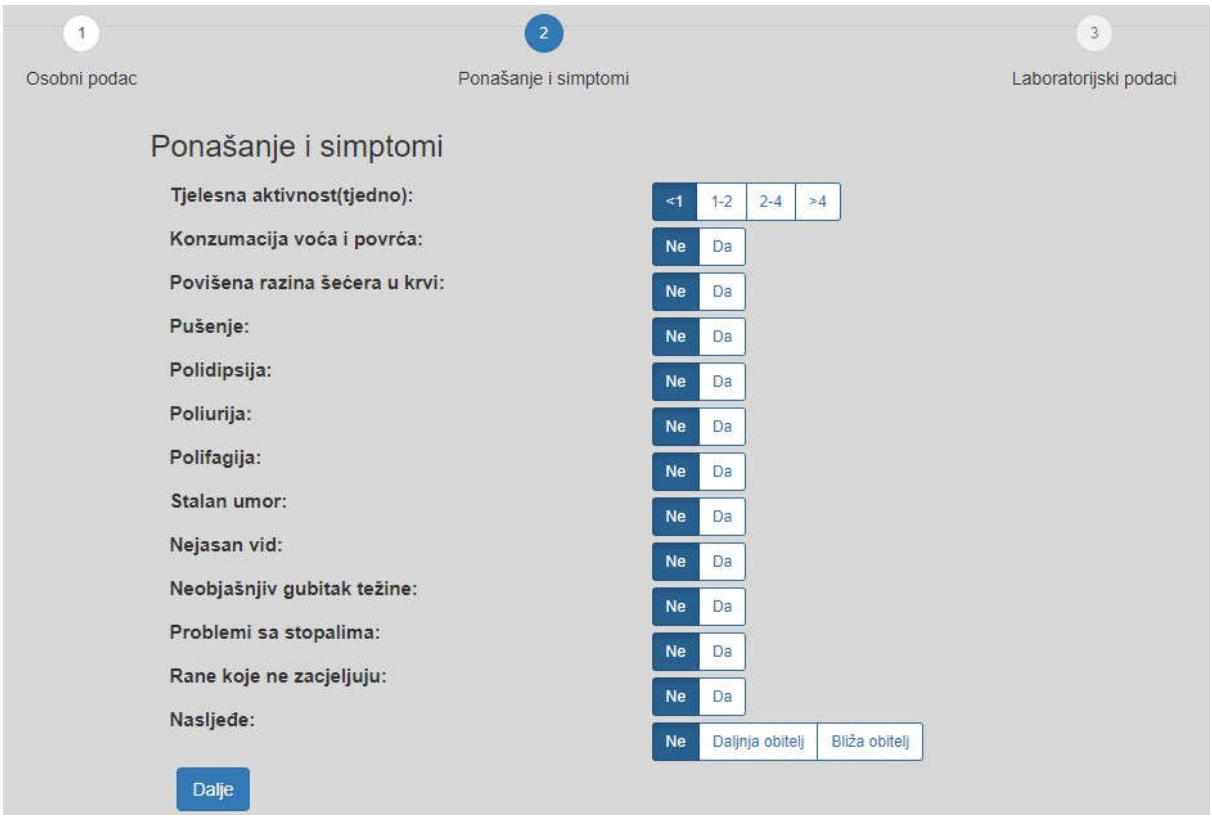
Kod je sastojan od dva **a** elementa koja imaju ulogu gumbova i koji se nalaze unutar nekoliko div elemenata. Ukoliko korisnik klikne na prvi **a** element, on je označen kao aktivan (oboji se u plavo) i njegovi se podaci šalju u bazu podataka. Isto vrijedi i za drugi element.

Unos parametara je ostvaren u tri koraka. Prvi korak predstavlja osobne podatke vidljivo na slici 4.14.



Slika 4.14. Prvi korak korisničkog sučelja

Klikom na gumb "Dalje" korisniku je otvoren drugi korak korisničkog sučelja prikazanog na slici 4.15.



Slika 4.15. Drugi korak unosa podataka kroz korisničko sučelje

Klikom na gumb "Dalje" korisniku je otvoren treći korak korisničkog sučelja prikazanog na slici 4.16.

Slika 4.16. Treći korak unosa podataka putem korisničkog sučelja

Klikom na gumb "Finish" Ajax dokument za slanje unesenih parametara u bazu podataka biva aktiviran. Izgled aktiviranog Ajax dokumenta je vidljiv u programskom kodu 4.8.

```
var userID ="<?php echo $_SESSION['user_id'];?>";
$('#finishBtn').on('click', function () {
    $.ajax({
        type: 'POST',
        url: 'php/core/userTest.php',
        data: {
            'userID': userID,
            'godine': godine,
            'spol': spol,
            'ITM': ITM,
            'struk': struk,
            'aktivnost': aktivnost,
            'voce': voce,
            'secer': secer,
            'pusenje': pusenje,
            'polidipsija': polidipsija,
            'poliurija': poliurija,
            'polifagija': polifagija,
            'umor': umor,
            'vid': vid,
            'tezina': tezina,
            'stopala': stopala,
            'rane': rane,
            'nasljede': nasljede,
            'sistolicki': sistolicki,
            'dijastolicki': dijastolicki,
            'LDL': LDL,
            'HDL': HDL,
            'glukoza': glukoza
        },
        error:function(data){
            alert("error occured"); //===Show Error Message====
        }
    });

});
```

Programski kod 4.8. AJAX kod kojim se šalju uneseni parametri u bazu podataka

Nakon što se klikne na gumb s ID-om "finishBtn" programski kod 4.8 je aktiviran. Osim što je klikom na gumb "Finish" AJAX dokument aktiviran, aktiviran je i kod kojim je prikazan rezultat naše web aplikacije, odnosno prikazan rizik obolijevanja od dijabetesa. Izračun jednog dijela tog rizika je vidljiv u programskom kodu 4.9.

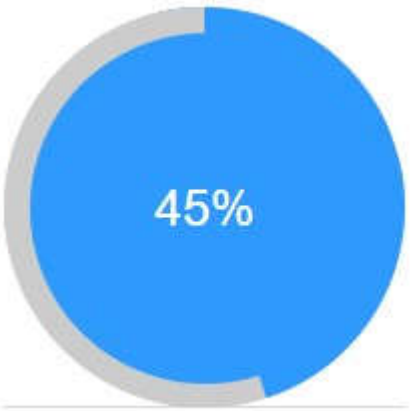
```
$(".glukoza a").each(function(){
if($(this).hasClass("active")){
    glukoza = $(this).attr("data-title");
    console.log(glukoza);

    if(glukoza == "<5.27"){
        //0
    }else if(glukoza == "5.27-5.72"){
        bodovi+=2;
    }else{
        bodovi+=5;
    }
}
});

$('#progress-bar').attr('data-percent',Math.round((bodovi/62)*100));
```

Programski kod 4.9. Primjer dijela izračuna rizika oboljenja od dijabetesa

Programski kod 4.9 prikazuje provjeru "aktivnih" članova te ovisno o težini parametara varijabli "bodovi" je pribrojen određeni broj bodova. Programski kod 4.10. prikazuje ispis rezultata u obliku grafa. Primjer toga grafa je vidljiv na slici 4.17.

```
    <!-- Modal -->
<div id="myModal" class="modal fade" role="dialog">
  <div class="modal-dialog">

    <!-- Modal content-->
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Rezultat</h4>
      </div>
      <div class="modal-body" id="Modalcontainer">
           <p>Vjerojatnost da bolujete od dijabetesa iznosi:</p>
        <div class="progress-bar position" id="progress-bar" data-duration="5000" data-color="#ccc,#2E9AFE"></div>

      </div>
      <div class="modal-footer">

           <button type="button"  class="btn btn-default" data-dismiss="modal">Zatvori</button>

      </div>
    </div>
  </div>
</div>
```

Programski kod 4.10. Izgled koda rezultata prikazan grafom



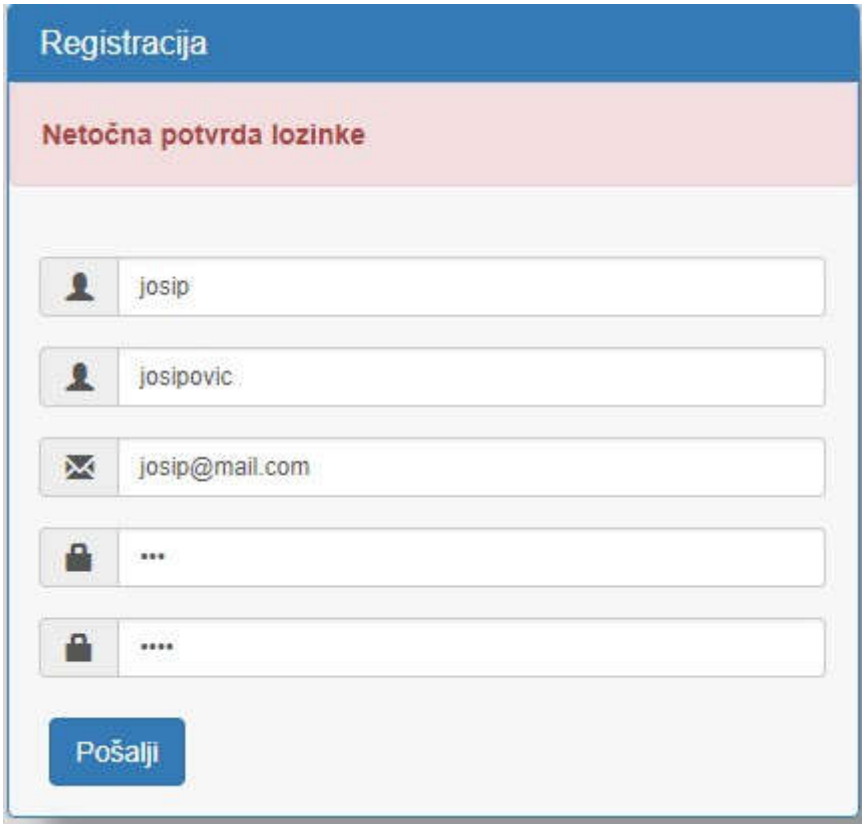
Slika 4.17. Ispis rezultata pomoću grafa

5. TESTIRANJE I ANALIZA WEB APLIKCIJE

Nakon izrade web aplikcije potrebno je testirati njen rad. Tim testiranjem pokušalo se predvidjeti sve moguće situacije i testirati kako će se aplikacija ponašati. Zbog jednostavnosti i preglednosti testiranje je podjeljeno na tri dijela koja ćemo detaljnije opisati.

5.1 Testiranja sučelja za registraciju

U prvom slučaju registriranja nije ispravno potvrđena lozinka te je ispisana poruka o netočnoj potvrdi lozinke kao što je prikazano na slici 5.1. Reakcija aplikacije je takva zato što nije bio zadovoljen jedan od uvjeta prikazan u programskom kodu 4.5.

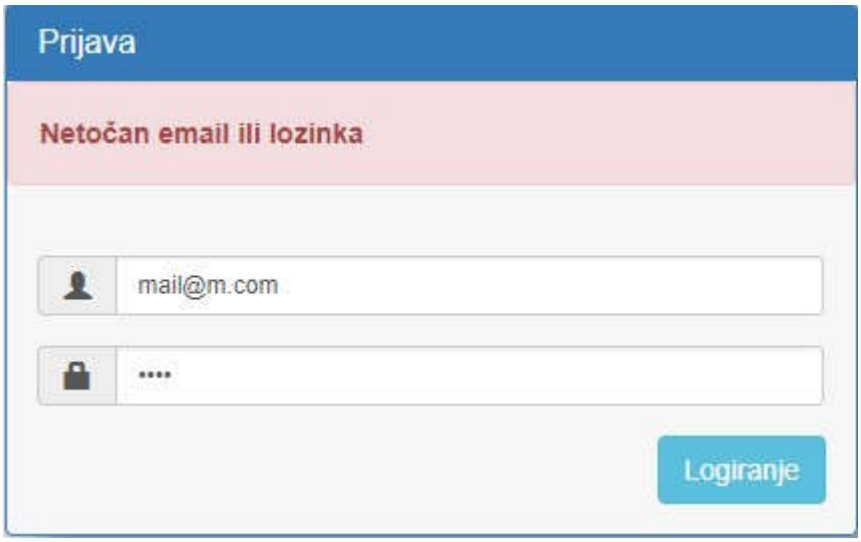


Slika 5.1. Rezultat netočne potvrde lozinke

U drugom slučaju nije upisano ime, te je ispisana poruka da unesemo sva polja. U trećem slučaju popunjeni su svi traženi podaci te smo preusmjereni na stranicu za prijava. Reakcija aplikacije je takva jer su svi uvjeti ispunjeni te je korisnik uspješno registriran.

5.2 Testiranja sučelja za prijavu

Nakon što smo registrirani, otvorena je stranica za prijavu. U prvom slučaju upisana je netočna adresa e-pošte te je dobivena poruka o neispravnosti unesene email adrese kao što je prikazano na slici 5.2. Reakcija aplikacije je takva jer nije pronađeno podudaranje unesene adrese e-pošte i adresa u bazi podataka.



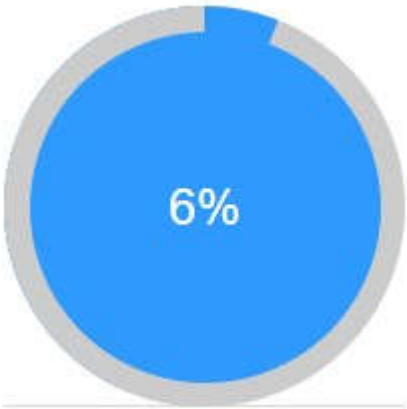
Slika 5.2. Rezultat netočnog unosa adrese e-pošte

U drugom slučaju upisana je netočna lozinka te je ispisana poruka o neispravnosti lozinke. U trećem slučaju upisana je ispravna adresa e-pošte i lozinka te smo preusmjereni unutar web aplikacije. Reakcija aplikacije je takva zato jer je pronađena adresa e-pošte i lozinka koja se podudara s onima u bazi podataka.

5.3 Testiranja web aplikacije

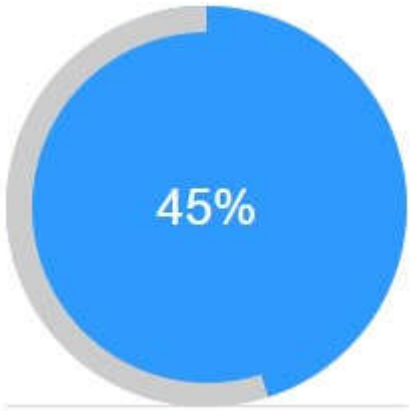
Testiranje web aplikacije obavit će se za tri slučaja opisana u prethodnom poglavlju u tablici 3.1 za nizak, srednji i visok rizik obolijenja od dijabetesa.

Nakon unosa skupa parametara, za nizak rizik obolijevanja od dijabetesa danih u tablici 3.1 dobije se rezultat prikazan na slici 5.3.



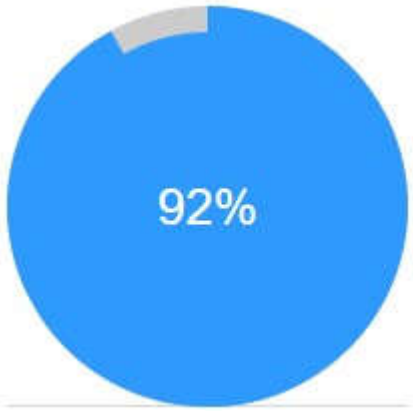
Slika 5.3. Rezultat unosa skupa parametara niskog rizika oboljenja od dijabetesa

Nakon unosa skupa parametara za srednji rizik obolijevanja od dijabetesa danih u tablici 3.1 dobije se rezultat prikazan na slici 5.4.



Slika 5.4. Rezultat unosa skupa parametara srednjeg rizika oboljenja od dijabetesa

Nakon unosa parametara za visoki rizik obolijevanja od dijabetesa danih u tablici 3.1 dobije se rezultat prikazan na slici 5.5.



Slika 5.5. Rezultat unosa skupa parametara visokog rizika oboljenja od dijabetesa

Tablica 5.1 prikazuje dobivene rezultate na osnovu unesenih parametara.

Tablica 5.1. Prikaz unesenih parametara i dobivenih rezultata

	Uneseni parametri za prvi slučaj (nizak rizik)	Uneseni parametri za drugi slučaj (srednji rizik)	Uneseni parametri za treći slučaj (visok rizik)
Godine	<45	45-55	>64
Spol	Muško	Muško	muško
Indeks tjelesne mase	<25	25-30	>30
Opseg struka (cm)	80	95	110
Tjedna tjelesna aktivnost	>4	1-2	<1
Konzumacija voća i povrća	Da	Da	ne
Povišena razina šećera u krvi	Ne	Ne	Da
Pušenje	Ne	Da	Da
Polidipsija	Ne	Ne	Da
Poliurija	Ne	Da	Da
Polifagija	Ne	Ne	Da
Stalan umor	Ne	Da	Da
Nejasan vid	Ne	Da	Da
Neobjašnjiv gubitak težine	Da	Da	Ne
Problemi sa stopalima	Ne	Ne	Da
Rane koje ne zacjeljuju	Ne	Ne	Da
Nasljeđe	Ne	Dalja obitelj	Bliža obitelj
Sistolički tlak	<120	120-135	>135
Dijastolički tlak	<80	80-90	>90
LDL kolesterol	<3.38	<3.38	>3.38
HDL kolesterol	>1.32	1.06-1.32	<1.06

Glukoza u krvi	<5.27	5.27-5.72	>5.72
Rezultat	**6%**	**45%**	**92%**

Rezultati iz sva tri slučaja su očekivani iz čega se može zaključiti da web aplikacija funkcionira kao što je očekivano.

6. ZAKLJUČAK

Cilj ovog završnog rada je upoznavanje s razvojnom okolinom oblaka računale te izrada web aplikacije u toj razvojnoj okolini. Razvojna okolina koja je odabrana je Microsoft Azure, a tema web aplikacije dijabetes, točnije računanje rizika obolijevanja od dijabetesa na osnovu unesenih parametara.

Glavni problem rada je proučavanje načina rada Microsoft Azure-a te mogućnosti povezivanja s bazom podataka i izvorom koda. Kao izvor koda je odabran GitHub. Osim razumjevanja načina rada oblaka računala, važno je bilo načiniti logički plan razvoja web aplikacije kako bi bilo uspješno razvijanje web aplikacije.

Aplikacija je namjenjena svima te je vrlo jednostavna za korištenje. Glavna zadaća aplikacije je podići svijest korisnika o svome zdravlju jer je zdravlje jako zapostavljeno. Za svaku bolest je važno rano prepoznavanje, pa tako i za dijabetes te se nadamo da će ova aplikacija pomoći korisnicima u otkrivanju istog.

Aplikacija se može poboljšati povećanjem baze podataka i brojem parametara na osnovu kojih je dobiven rezultat.

LITERATURA

[1] American Diabetes Association, Type 2 Diabetes Risk Test, http://www.diabetes.org [posjećeno kolovoz 2017.]

[2] Framingham Heart Study, Diabetes Risk Calculator, http://www.framinghamheartstudy.org [posjećeno rujan 2017.]

[3] A. Samčović, CLOUD COMPUTING OKRUŽENJE U MREŽAMA NAREDNE GENERACIJE, XXX Simpozijum o novim tehnologijama u poštanskom i telekomunikacionom saobraćaju – PosTel 2012, Beograd, 04. i 05. prosinac 2012. [posjećeno rujan 2017.]

[4] Microsoft, Microsoft Azure, https://azure.microsoft.com [posjećeno rujan 2017.]

[5] GitHub, https://github.com [posjećeno kolovoz 2017.]

[6] Bootstrap, http://getbootstrap.com [posjećeno kolovoz 2017.]

[7] Jquery, https://jquery.com [posjećeno kolovoz 2017.]

[8] PHP, http://php.net [posjećeno kolovoz 2017.]

[9] MySQL, https://www.mysql.com [posjećeno kolovoz 2017.]

[10] W3Schools Online Web Tutorials, https://www.w3schools.com [posjećeno kolovoz 2017.]

SAŽETAK

U radu je prikazana izrada web aplikacije u razvojnom okruženju oblaka računala. Prikazani su kodovi korišteni pri izradi popraćeni detaljnim objašnjenjima. Aplikacija je povezana s bazom podataka te je omogućeno registracija i prijava korisnika. Osim korisničkih podataka u bazu su spremljeni i parametri koje korisnik unosi u aplikaciju putem korisničkog sučelja. Kao rezultat aplikacija daje postotak rizičnosti obolijevanja od dijabetesa prikazan u obliku grafa. Testiranje aplikacije ostvareno na tri ekstremna slučaja pokazuje da web aplikacije funkionira kako smo i očekivali.

Ključne riječi: dijabetes, oblak računala, Microsoft Azure, web aplikacija

ABSTRACT

The paper presents a web application development in the cloud computing environment. The codes shown are used in the production followed by detailed explanations. The application is linked to the database and the registration and user login is enabled. In addition to the user data, the database also stores parameters that the user enters into the application through the user interface. As a result of the application, the percentage of risk of diabetes is shown in the form of a graph. Testing applications found in three extreme cases show that web applications function as we expected.

Keywords: cloud computing, diabetes, Microsoft Azure ,web application

ŽIVOTOPIS

Antonio Došlić rođen je 31.7.1995. godine u Osijeku. Živi u Kuševcu na adresi Ljudevita Gaja 11. Osnovnoškolsko obrazovanje započinje 2002. godine u OŠ Josipa Antuna Ćolnića u Đakovu. Nakon osnovnoškolskog obrazovanja, 2010. godine se upisuje u Gimnaziju Antuna Gustava Matoša u Đakovu, smjer Prirodoslovna matematička gimnazija. Nakon srednjoškolskog obrazovanja upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija, smjer Računarstvo. Posjeduje znanje u govoru, čitanju i pisanju engleskog jezika te vozačku dozvolu B kategorije. Osim navedenog posjeduje i osnovno znanje HTML, CSS, jQuery, PHP, MySQL, bootstrap, Ajax, C, C++ i C#, te upravljanje Microsoft Office alatima.

PRILOZI (na DVD-u)

Prilog 1. Završni rad u .docx i .pdf formatu

Prilog 2. Programski kodovi web aplikacije