

# Web aplikacija za prijavu kontakata za potrebe samoizolacije

---

Licht, Bruno

Undergraduate thesis / Završni rad

2022

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

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:398880>

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

*Download date / Datum preuzimanja:* **2024-12-24**

*Repository / Repozitorij:*

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



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

**Sveučilišni studij**

**Web aplikacija za prijavu kontakata za potrebe  
samoizolacije**

**Završni rad**

**Bruno Licht**

**Osijek, 2022**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac ZIP - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 26.08.2022.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na  
preddiplomskom sveučilišnom studiju**

<b>Ime i prezime Pristupnika:</b>	Bruno Licht
<b>Studij, smjer:</b>	Preddiplomski sveučilišni studij Računarstvo
<b>Mat. br. Pristupnika, godina upisa:</b>	R 4383, 22.07.2019.
<b>OIB Pristupnika:</b>	01046359336
<b>Mentor:</b>	Prof. dr. sc. Krešimir Nenadić
<b>Sumentor:</b>	,
<b>Sumentor iz tvrtke:</b>	
<b>Naslov završnog rada:</b>	Web aplikacija za prijavu kontakata za potrebe samoizolacije
<b>Znanstvena grana rada:</b>	<b>Programsko inženjerstvo (zn. polje računarstvo)</b>
<b>Zadatak završnog rad:</b>	Kratko opisati problematiku određivanja kontakata koji se navode za potrebe samoizolacije. Potrebno je modelirati i izraditi bazu podataka koja će se koristiti za pohranu svih podataka. Izraditi korisničko sučelje i povezati ga s poslužiteljskim funkcionalnostima. Tema rezervirana: Bruno Licht
<b>Prijedlog ocjene završnog rada:</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene od strane mentora:</b>	26.08.2022.
<b>Datum potvrde ocjene od strane Odbora:</b>	
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

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

Osijek, 07.09.2022.

Ime i prezime studenta:	Bruno Licht
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R 4383, 22.07.2019.
Turnitin podudaranje [%]:	8

Ovom izjavom izjavljujem da je rad pod nazivom: **Web aplikacija za prijavu kontakata za potrebe samoizolacije**

izrađen pod vodstvom mentora Prof. dr. sc. Krešimir Nenadić

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

<b>1. UVOD.....</b>	<b>5</b>
<b>1.1. Zadatka rada .....</b>	<b>5</b>
<b>2. PREGLED PODRUČJA .....</b>	<b>6</b>
<b>2.1. CovidGo .....</b>	<b>6</b>
<b>2.2. Stop Covid-19 .....</b>	<b>7</b>
<b>2.3. Corona-Warn app.....</b>	<b>8</b>
<b>2.4. Aarogya Setu .....</b>	<b>9</b>
<b>2.5. Immuni.....</b>	<b>10</b>
<b>3. POSTUPAK IZRADE APLIKACIJE .....</b>	<b>11</b>
<b>3.1. Rješenje problematike zadatka.....</b>	<b>11</b>
<b>3.2. Web stranice i funkcionalnosti.....</b>	<b>11</b>
<b>3.3. Baza podataka .....</b>	<b>12</b>
<b>3.4. Apache poslužitelj .....</b>	<b>13</b>
<b>3.5. Implementacija funkcionalnosti .....</b>	<b>14</b>
3.5.1. Dodavanje kontakata u bazu podataka.....	14
3.5.2. Proširivanje modalnog obrasca.....	15
3.5.3. Dohvaćanje podataka kontakta .....	15
3.5.4. Upozorenje neispravno unesenih podataka.....	16
3.5.5. Brisanje kontakata .....	16
<b>4. RAD S APLIKACIJOM .....</b>	<b>18</b>
<b>4.1. Analiza dijelova aplikacije .....</b>	<b>22</b>
4.1.1. Glavna stranica .....	22
4.1.2. Stranica za prijavu i registraciju .....	23
<b>5. ZAKLJUČAK.....</b>	<b>24</b>

# 1. UVOD

Suočeni s epidemiološkom situacijom u digitalnoj eri, nastala je potreba za stvaranjem jedinstvenih digitalnih rješenja u borbi protiv širenja zaraze kako bi se što brže vratili u normalnu svakodnevicu.

Zadatak ovog završnog rada je izrada web aplikacije koja omogućuje vođenje evidencije kontakata sa zaraženim osobama. To je postignuto stvaranjem baze podataka koja sadrži ključne podatke o prijavljenim osobama čiji će detalji biti pojašnjeni u kasnijem poglavlju. Ti podaci su vidljivi, na javnoj adresi računala (u trenutku rada na završnom) koje postavlja web aplikaciju, bilo kojoj osobi s valjanim pristupnim podacima. To predstavlja ozbiljnu moguću povredu GDPR-a (eng. *General Data Protection Regulation*) zato što iako su ti podaci nužni za evidenciju neće sve osobe koje su unesene u sustav biti sretne što su njihovi podaci dostupni na Internetu. Zato bi sve unesene osobe i korisnici aplikacije trebali biti upozoreni o vidljivosti podataka svima unutar aplikacije.

Korisnici aplikacije su u mogućnosti pregledati cijeli popis zaraženih osoba, pretraživati po određenim kriterijama, dodavati nove osobe na popis te čak brisati i mijenjati podatke osoba na popisu u slučaju greške ukoliko imaju odgovarajuće ovlasti prilikom prijave u aplikaciju. Pristupni podaci mogu jedino biti dodijeljeni od strane osobe koja ima fizički pristup računalu na kojoj se postavlja aplikacija zbog postavljene konfiguracije pristupa bazi podataka.

U drugom poglavlju su ukratko opisana postojeća slična rješenja te je dan općeniti izgled. Nakon toga je opisan čitav proces stvaranja aplikacije od odabira potrebnih alata, potrebnih izmjena postavki te do primjera kako su alati korišteni. U zadnjem poglavlju prije zaključka se nalazi cijeli izgled gotove aplikacije zajedno sa objašnjenjima kako se koriste pojedine funkcionalnosti te kratka objašnjenja kako su neka od njih implementirana.

## 1.1. Zadatka rada

Kratko opisati problematiku određivanja kontakata koji se navode za potrebe samoizolacije. Potrebno je modelirati i izraditi bazu podataka koja će se koristiti za pohranu svih podataka. Izraditi korisničko sučelje i povezati ga s poslužiteljskim funkcionalnostima.

## 2. PREGLED PODRUČJA

Kako je svijet suočen s novom i opasnom epidemijom virusa tako su poduzimane razne mjere kako bi se spriječilo širenje zaraze. Kako smo izloženi digitalnoj tehnologiji i aktivno dijelimo podatke svaki dan bilo je samo pitanje vremena kada će nastati nove aplikacije koje će na jedinstven način pomoći u obrani od virusa.

### 2.1. CovidGo

Mobilna aplikacija CovidGO omogućava validaciju QR kôdova na EU digitalnim COVID potvrđama izdanim u Republici Hrvatskoj i državama članicama EU. Mobilnom aplikacijom moguće je provjeriti potvrde o testiranju, potvrde o cijepljenju i potvrde o preboljenju.

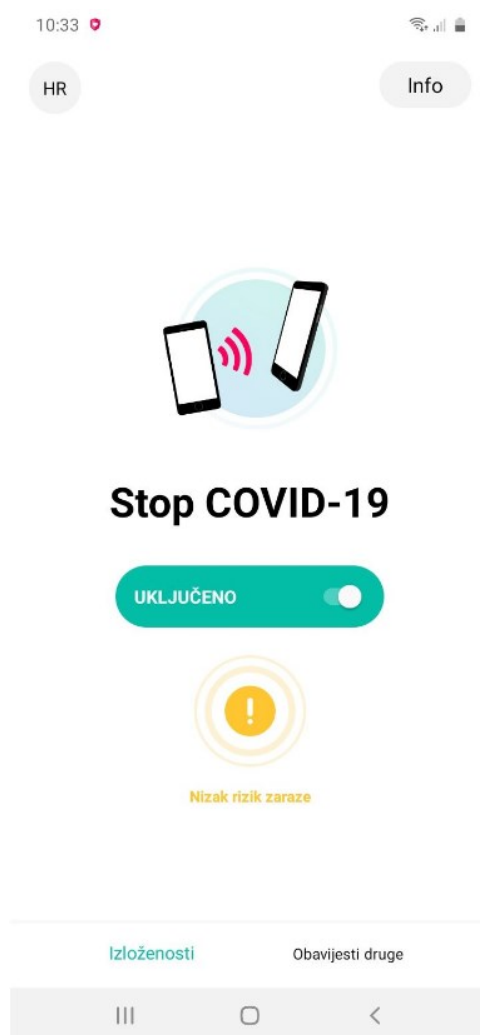
CovidGO ne obrađuje osobne podatke, već provjerava nacionalni digitalni potpis zapisan u QR kôdu[1]. U slučaju valjane potvrde ekran će se prikazati u zelenoj boji, a u slučaju da potvrda iz bilo kojeg razloga nije valjana prikazat će se crveni ekran s dodatnim obrazloženjem, oba slučaja vidljiva na slici 2.1.



Sl. 2.1. CovidGO

## 2.2. Stop Covid-19

Stop COVID-19 je aplikacija koja služi jednostavnom upozoravanju građana da su se možda našli u epidemiološki rizičnom kontaktu. Može pomoći korisnicima donijeti ispravnu odluku u slučaju pojave simptoma. Također nudi upozorenja u slučaju epidemiološki bliskog kontakta te time korisnik može pojačati oprez i higijenu u slučaju da još nije razvio simptome[2] .

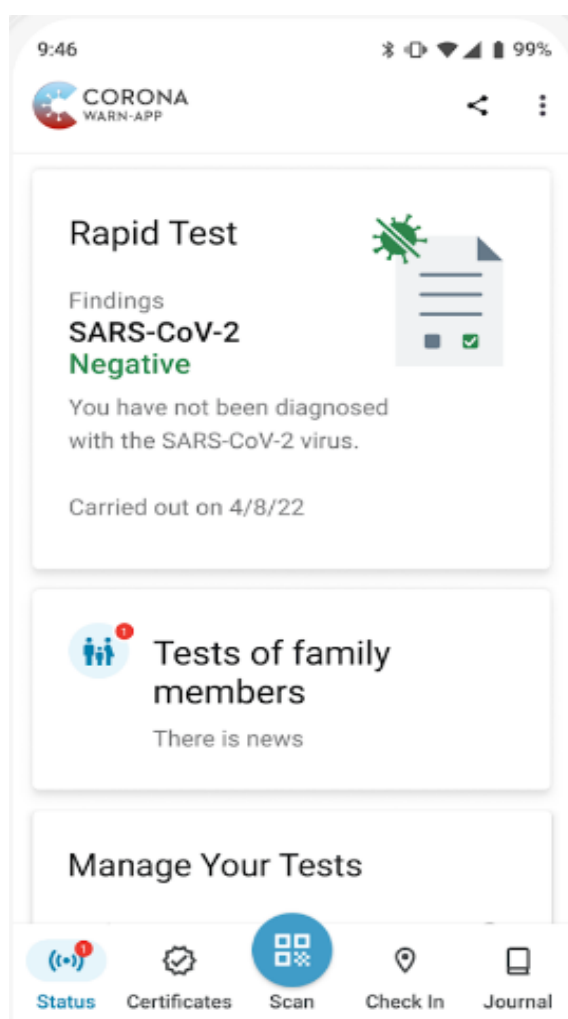


Sl. 2.1. Stop COVID-19



## 2.3. Corona-Warn app

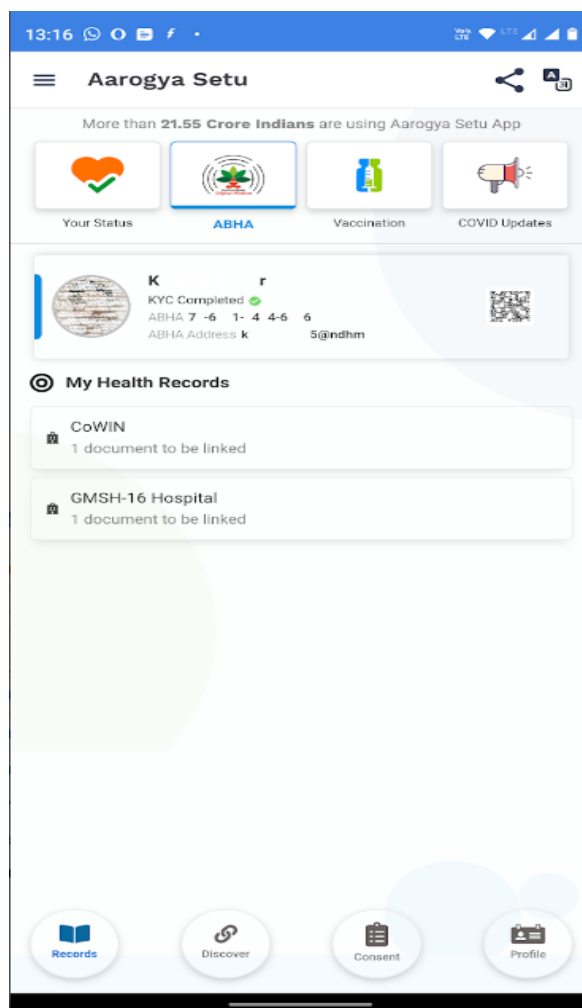
Corona-Warn-App je aplikacija koja pomaže u praćenju lanaca infekcije SARS-CoV-2 (koji može uzrokovati COVID-19) u Njemačkoj. Aplikacija se temelji na tehnologijama s decentraliziranim pristupom i obavještava korisnike ako su bili izloženi SARS-CoV-2. Transparentnost je ključna za zaštitu krajnjih korisnika aplikacije i za poticanje usvajanja. Izvorni kod aplikacije i dokumentacija su dostupni na internetu tako da bilo tko može pridonijeti razvoju projekta[3] .



Sl. 2.3. Corona-warn

## 2.4. Aarogya Setu

Aarogya Setu koristi praćenje kontakata za snimanje pojedinosti o svim ljudima s kojima su korisnici možda došli u kontakt prilikom obavljanja uobičajenih aktivnosti. Ako netko od njih u nekom kasnijem trenutku bude pozitivan na COVID-19, korisnik je odmah obaviješten i za njega je organizirana proaktivna medicinska intervencija. Prije korištenja aplikacije potrebno se registrirati s indijskim brojem telefona[4] .



Sl. 2.4. Aarogya Setu

## 2.5. Immuni

Aplikacija je razvijena u potpunosti u skladu s talijanskim i europskim zakonodavstvom za zaštitu privatnosti kako bi se pokušao spriječiti nastanak novih žarišta zaraze. Davanjem svoje trenutne lokacije, korisnici će biti obaviješteni u slučaju da su bili blizu osobe koje su testirane pozitivne na COVID-19 te uz pristanak korisnika aplikacija obavještava zdravstvene radnike o kontaktu[5] .



Sl. 2.5. Immuni

### **3. POSTUPAK IZRADE APLIKACIJE**

Prije početka izrade same aplikacije bilo je potrebno odrediti koji će se jezici i razvojna okruženja koristiti. Riječ je o web aplikacije dakle HTML (eng. *HyperText Markup Language*) i CSS (eng. *Cascading Style Sheets*) za stvaranje osnovnog izgleda aplikacije su neizbježni a uz pomoć Javascript-a su ostvarene mnoge funkcionalnosti. Za potrebe postavljanja aplikacije odabran je XAMPP program koji sadrži Apache poslužitelj i MySQL koji je korišten kao baza podataka. Kao veza između baze podataka i aplikacije je odabran PHP (eng. *PHP: Hypertext Preprocessor*) koji dolazi kao dio XAMPP-a. Kao razvojno okruženje je odabran Notepad++ zbog jednostavnosti uporabe i zato što ne postoji potreba za stvaranje izvršne datoteke.

#### **3.1. Rješenje problematike zadatka**

Nakon odabranih alata potrebno je odrediti ključne podatke za praćenje samoizolacije zaraženih osoba i njihovih kontakata. Svi ti podaci će se kasnije organizirati u strukturu podataka i kao takvi unijeti u bazu podataka kako bi se mogli prikazati na unutar web aplikacije. Prvi i najbitniji podatak bi bio osobni identifikacijski broj (OIB) jer predstavlja jedinstven podatak kojim je moguće identificirati osobu. Zatim je potrebno zapisati gdje ta osoba živi i kada se zarazila kako bi se mogao pratiti smjer zaraze te omogućiti praćenje da li osoba poštiva mjere samoizolacije. Uz te podatke su također u strukturu dodane ime, prezime i broj telefona osobe kako bi se izbjeglo odnošenje prema osobi kao prema broju u sustavu. Uz to će se u strukturu dodati i OIB potvrđene zaražene osobe s kojom je kontakt nedavno bio u bliskom susretu, dalje u radu se spominje kao primarni kontakt, kako bi se osobe koje su zajedno ošle u karantenu mogle grupirati te zajedno prikazati unutar aplikacije. Potrebne funkcionalnosti za stvaranje popisa poput dodavanje kontakata, brisanje, pretraživanje i uređivanje podataka u slučaju greške su objašnjene dalje u poglavlju.

#### **3.2. Web stranice i funkcionalnosti**

Idući korak je bio stvaranje skice općenitog izgleda aplikacije te pravljenja popisa svih funkcionalnosti koji su potrebni za ostvarenje same ideje. Prema toj skici je zatim, kao što je navedeno ranije u poglavlju, bilo potrebno odrediti na koji će se način izrađivati aplikacija te obraditi teorijsku podlogu i upoznati se sa svakim od odabranih alata.

Nakon toga su izrađene potrebne stranice pomoću HTML i CSS koda. Također prilikom izrađivanja stranica korišten je Bootstrap 3 koji sadrži vlastiti CSS i Javascript kod kako bi se postigao bolji izgled. Za potrebe aplikacije napravljene su 3 stranice: stranica za prijavu, registraciju i glavna stranica. Kako se je za ulazak u aplikaciju potrebno prijaviti bilo je potrebno omogućiti komunikaciju između uređaja i servera koji sadrži bazu podataka čija će logika biti objašnjena kasnije. U slučaju uspješne prijave korisniku se otvara glavna

stranica na kojoj dobiva pogled na popis svih zaraženih osoba unesenih u bazu podataka. Uz to mu se nudi mogućnost dodavanja novih osoba na popis i pretraživanja popisa po kriterijima. Uz odgovarajuću razinu pristupa može također i brisati osobe iz baze podatke i mijenjati im podatke (zamišljeno je da takvu razinu pristupa ima mali broj administratora). Dinamičnost web stranice je postignuta Javascript-om pomoću kojeg su ostvarene funkcionalnosti poput upozorenja korisnika na neispunjeni obrazac za dodavanje osoba na popis. Način uključivanja Javascript datoteke je prikazan na slici 3.1. PHP je korišten za komunikaciju sa poslužiteljem kako bi se dohvatili stvarni podaci iz baze podataka te se nakon toga prikazuju u formatiranom

```
<script src="mainFunctions.js"></script>
```

Sl. 3.1. Uključivanje Javascript datoteke u HTML kod

obliku pomoću „echo“ naredbe koja omogućuje ispis *string-a* na stranicu, ali ako se unutar *string-a* umetnu ispravne HTML oznake web preglednik će prepoznati će taj dio *string-a* prepoznati kao HTML kod te će se on izvršiti. PHP datoteka može sadržavati HTML oznake i skriptni kod koji će se izvršiti, a na slici 3.2 je prikazan način uključivanja postojeće PHP skripte u drugu PHP skriptu.

```
<?php include "maindisplaytable.php"?>
```

Sl. 3.2. Uključivanje PHP skripte iz jedne datoteke u drugu

### 3.3. Baza podataka

Baza podataka je strukturirana tako da se sastoji od 3 tablice prikazane na slikama 3.3, 3. i 3.5. Prilikom stvaranja tablice koja sadrži podatke o zaraženim osobama trebalo je prepoznati koji su sve podaci nužni kako bi se primjerice mogao odrediti broj zaraženih na nekom području ili pratiti osobe koje trebaju biti u samoizolaciji. Za tu svrhu odabrani su kao što je spomenuto ranije: OIB, ime i prezime, adresa i grad trenutnog boravka i broj telefona. Na slici su prikazana još dva podatka: datum zaraze i OIB primarnog kontakta koji služe kako bi se moglo pratiti moguće širenje zaraze.

OIB	Ime	Prezime	Broj_telefona	Datum_zaraze	Adresa_prebivališta	PB_prebivališta	OIB_Primarnog_kontakta
12345678945	Nenad	Jukić	0996642215	2022-06-13	Vinkovačka 36	10000	12345678945
32165478945	Mirna	Bilić	0789456123	2022-06-23	Vinkovačka 10	10000	32165478945
45678936921	Novak	Nović	0976642211	2022-06-23	Panonska 25	31000	45678936921

Sl. 3.3. Tablica podataka zaraženih osoba – „popis\_zarazenih“

Prije prijave u aplikaciju potrebno je ili se registrirati s tokenom čiji način dobivanja još nije specificiran ili dobiti korisničke podatke od postojećeg korisnika (moguće je više istovremenih prijava s istim podacima). Tako da se tablica o korisnicima sastoji od korisničkog imena, zaporke i razine pristupa aplikaciji.

ID	mail	username	pass	razina_pristupa
1	█████@gmail.com	admin	██████████	1
2	██████████@gmail.com	basic	██████████	0

Sl. 3.4. Tablica korisnika – „users“

Zadnja tablica, slika 3.5, sadrži token koji je potrebno unijeti kako bi se dobila određena razina pristupa. Zasad još nije osmišljen način kako bi se vrijednost tokena mijenjala te je bitno naglasiti da je sigurnije davati samo korisničke podatke budućim korisnicima.

Razina	Šifra
0	██████████
1	██████████

Sl. 3.5. Tablica korisnika – „access\_level\_codes“

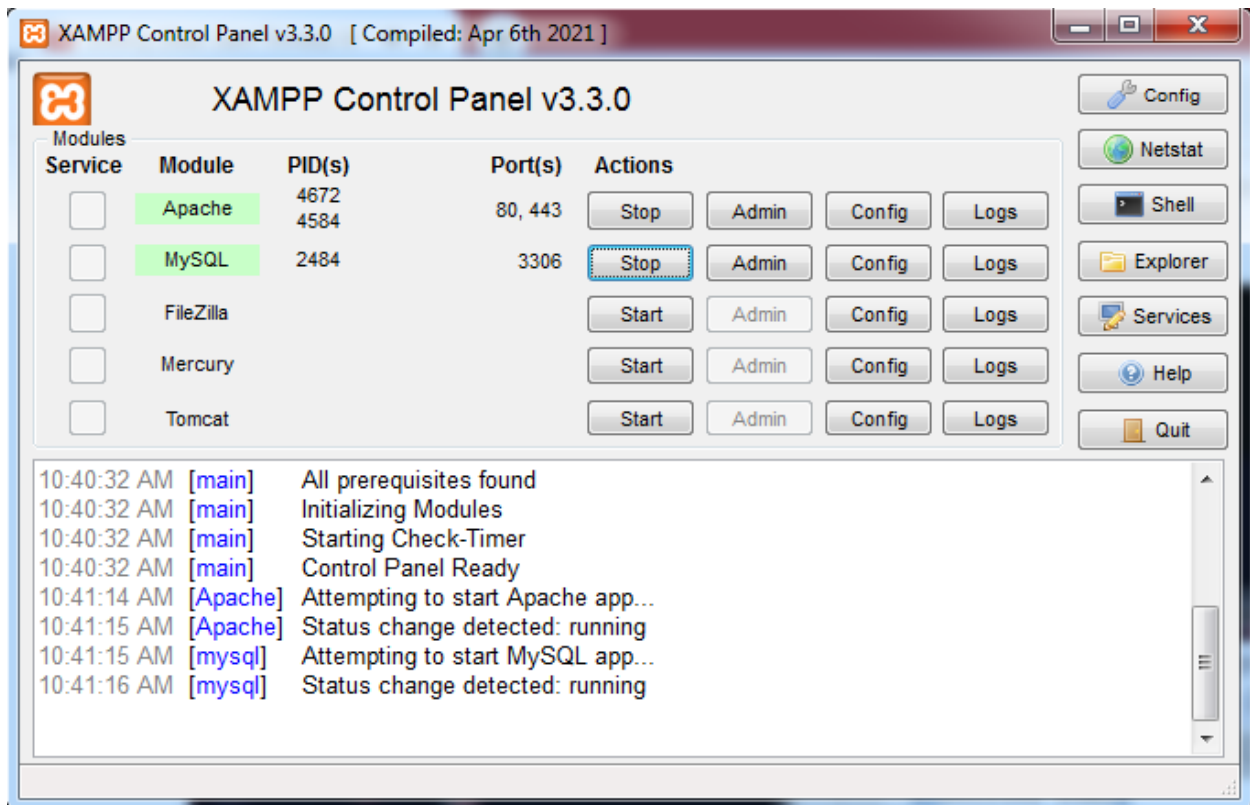
Za bilo kakav pristup bazi podataka potrebno je izvršiti kod sa slike 3.63.63.63.6. `mysqli_connect` naredba prima 4 parametra koja su opisana u kodu te njihove vrijednosti ovise o konfiguraciji MySQL baze podataka. Tek nakon uspješnog spajanja je moguće slati zahtjeve za izvođenjem SQL naredbi.

```
$hName='localhost'; // host name
$uName='zavrnsni'; // database user name
$password=''; // database password
$dbName = "zavrnsni"; // database name
$dbCon = mysqli_connect($hName, $uName, $password, $dbName);
```

Sl. 3.6. Način povezivanja s bazom podataka

### 3.4. Apache poslužitelj

Kako bi se testiralo funkcionira li aplikacija ispravno na različitim uređajima bilo je potrebno postaviti poslužitelj na lokalnom računalu. Instalacijom XAMPP programa i pokretanjem Apache-a korisnicima je omogućen pristup na lokalnoj mreži ukoliko vatrozid na poslužitelju to dopušta. Aplikacija je namijenjena da se koristi na globalnoj razini te su zato na lokalnom usmjerivaču otvoreni portovi koje su dodijeljene Apache-u, vidljivi na slici 3.7. Kako to predstavlja mogući veliki sigurnosni rizik poput dohvaćanja datoteka, unutar poslužitelja u konfiguraciji XAMPP-a je bilo potrebno ograničiti pristup preko interneta samo na mapu koja sadrži datoteke aplikacije i to na način da nije moguće vidjeti sadržaj te mape nego samo ono što aplikacija prikazuje[6].



Sl. 3.7. XAMPP kontrolna ploča

### 3.5. Implementacija funkcionalnosti

U ovom dijelu rada su pojašnjeni načini ostvarivanja različitih funkcionalnosti web aplikacije koje su nužne za njen rad. Dio koda je pisan uz pomoć stranice sa primjerima sličnih rješenja ali je naravno oblikovan za potrebe razvoja aplikacije[7].

#### 3.5.1. Dodavanje kontakata u bazu podataka

Osim pregledavanja popisa jednako bitan dio je dodavanje novih kontakata u bazu podataka. Nakon unošenja podataka u obrazac koji je objašnjen na idućem poglavlju pokreće se skripta koja dohvaća te podatke koji se unose u SQL Insert naredbu koja dodaje podatke u bazu podataka. U slučaju da neki podaci nisu uneseni na njihovo mjesto se stavlja *null* vrijednost ukoliko je dozvoljena. U kasnije objašnjenom modalnom prozoru za dodavanje kontakata je moguće proširiti obrazac unosa kako bi te osobe bile povezane u bazi podataka preko podatka OIB\_primarnog\_kontakta. Na slici 3.8 je prikazano dohvaćanje imena kontakata i provjera da li je OIB unesen ali se obje operacije provode za sve podatke iz baze podataka.3.8

```

include 'connect.php';
if(!isset($_POST['save'])) {
    $name = "".$_POST['name']."";

if($oib=="") {
    $oib='NULL';
}

$query = "INSERT INTO popis_zarazenih (OIB, ime, prezime, broj_telefona, datum_zaraze, adresa_prebivališta, pb_prebivališta, OIB_primarnog_kontakta) VALUES ($oib, $nam
try {
    mysqli_query($dbCon, $query);

```

Sl. 3.8 Dodavanje kontakata u bazu

### 3.5.2. Proširivanje modalnog obrasca

Kako bi se omogućilo korisnicima da dodaju nedavne kontakte zaražene osobe, odlučeno je da će se postojeći modalni prozor za dodavanje proširivati koliko god je potrebno. Prvo se stvara zahtjev za poslužitelja te mu se pridodaje funkcija da ukoliko je zahtjev uspješno izvršen da će se na sadržaj elementa nadodati tekst. Zatim je odabrana datoteka koja sadrži podatke, odnosno tekst koji se treba nadodati, te se zahtjev šalje poslužitelju.

```

function addSubContactForm() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("extendedform").innerHTML = document.getElementById("extendedform").innerHTML+this.responseText;
        }
    };
    xhttp.open("GET", "subContactForm.txt", true);
    xhttp.send();
}

```

Sl. 3.9 Proširivanje modalnog obrasca

### 3.5.3. Dohvaćanje podataka kontakta

S ciljem da se stranica nikad ne mora osvježavati kako bi dohvatila podatke koji se moraju dohvatiti izvršavanjem PHP skripti, korišten je AJAX (eng. *Asynchronous JavaScript and XML*) za čije je korištenje potrebno instalirati jQuery. AJAX se koristi na način da se definiraju zaglavlja, podaci koji se šalju, te funkcije koje se izvršavaju u slučaju uspješnog odnosno neuspješnog izvršavanja zahtjeva. U primjeru sa slike 3.10 definirani su PHP skripta kojoj se šalje OIB osobe kojoj se žele promijeniti podaci. U slučaju uspješnog izvršavanja skripte odnosno dohvaćanja podataka, podaci se razdvajaju i kao takvi koriste kao parametri za funkciju koja otvara modalni prozor s ispunjenim podacima kontakta.



```

function getContactData(coib){
$.ajax({
  type: "POST",
  url: "changecontact.php",
  dataType:"text",
  async: false,
  data: {
    oib:coib
  },
  success: function(data) {
    var myArr=data.split("&");
    displayContactInfo(myArr[0],myArr[1],myArr[2],myArr[3],myArr[4],myArr[5],myArr[6]);
  },
  error:function(data, exception){
    console.log(exception);
    console.log(data.status);
  }
});
}

```

Sl. 3.10. Dohvaćanje podataka

### 3.5.4. Upozorenje neispravno unesenih podataka

Novi korisnici rijetko kad znaju koja su ograničenja prilikom unosa podataka i zato ih je potrebno upozoriti da slučajno ne pošalju nepotpune podatke ili još gore pogrešnu vrstu podatka, primjerice korisnik slučajno unese riječ umjesto broj u polje „broj telefona“. Kako bi se to spriječilo dohvaća se polje čije se promjene žele promatrati, u primjeru slike 3.11 OIB, te se njemu pridodaje „EventListener“ komponenta koja je namještena da kad god korisnik napusti polje da se automatski provjerava niz uvjeta. U ovom slučaju je dan primjer samo za upozorenje u slučaju da je polje ostalo prazno za svako polje postoje bar 2 uvjeta kako bi se osigurao uspješan unos. Važno je upozoriti da u slučaju da se nikad ne klikne na polje da se nikad neće pokazati upozorenje da je polje prazno.

```

const oib = document.getElementById('oib')
oib.addEventListener("focusout", function(){
  if(oib.value==""){
    document.getElementById("warning").innerHTML="Niste unjeli OIB.";
    submitter.style.pointerEvents = "none";
  }
}

```

Sl. 3.11. Upozorenje neispravnosti podataka

### 3.5.5. Brisanje kontakata

Za bilo kakvo manipuliranje bazom podataka prvo se preko PHP skripte potrebno spojiti na bazu pomoću `mysqli_connect` naredbe s parametrima koji odgovaraju postavkama tražene baze. Zatim se priprema valjana SQL naredba na koju se mogu nadodati dodatne opcije. Nakon toga provjeravamo da li postoje kriteriji po kojima se filtriraju kontakti za brisanje. Na slici 3.11 je naveden samo kriterij imena kontakta ali je implementirana provjera za sve podatke u bazi podataka. U slučaju da postoji kriterij na postojeću SQL naredbu se nadodaje dio naredbe koji ovisi o tome je li riječ o prvom nađenom kriteriju te se za kraj šalje zahtjev bazi da se izvrši naredba.

```

$dbCon = mysqli_connect($hName, $uName, $password, $dbName);
$sql="DELETE FROM popis_zarazenih ";
$searched=false; //maindisplayt.
@$nameCrit= $_SESSION['name'];

if($nameCrit!="") {
    if($searched) {
        $sql.=" AND ime='".$nameCrit."'";
    }
    else {$sql.="WHERE ime='".$nameCrit."'";}
    $searched=true;
}

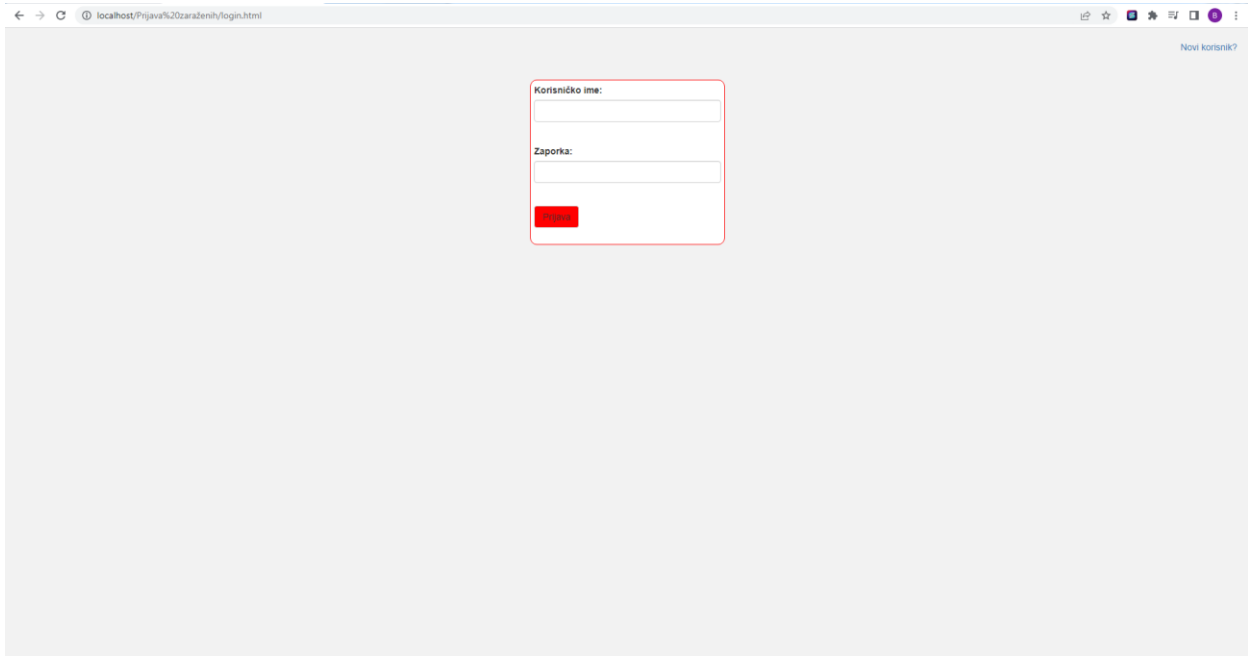
if (mysqli_query($dbCon, $sql)) {
    $_SESSION["change"]=true;
    include 'main.php';
}

```

Sl. 3.11. Branjanje kontakata prema kriteriju

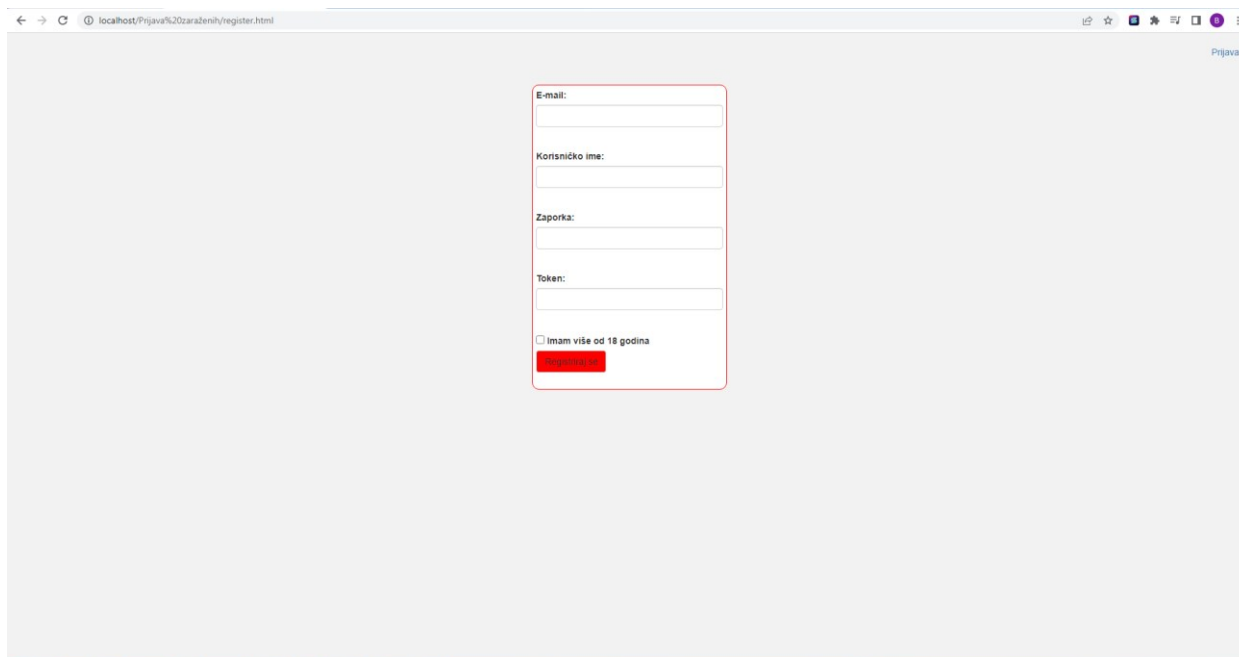
## 4. RAD S APLIKACIJOM

Na slikama 4.1, **Pogreška! Izvor reference nije pronađen.** i 4 je dan završni izgled web aplikacije. U slučaju da nije već prijavljen, korisnik je automatski preusmjeren na stranicu za prijavu, slika 4.1, gdje



Sl. 4.1. Prijava u aplikaciju

unosni pristupne podatke koje je dobio ranije. U slučaju da je potrebno stvoriti pristupne podatke za novog korisnika, potrebno je kontaktirati odgovarajućeg administratora koji će zatim otići na stranicu za registraciju, slika 4.2, te će unijeti nove pristupne podatke u sustav pomoću tokena koji je potreban za uspješnu registraciju i kojemu samo administratori baze podataka imaju pristup. Prilikom unosa tokena administrator može stvoriti običnog korisnika ili korisnika s većim pravima pristupa.



Sl. 4.2. Stranica za registraciju

Na glavnoj stranici, slika 4.3, korisnik ima mogućnost pregleda cijelog popisa zaraženih osoba i dodavanje osoba na isti popis pritiskom na gumb „Dodaj osobu“ preko modalnog prozora, prikazan slikom 4.4, a preko istog prozora se mogu dodati i nedavni kontakti te osobe preko „+“ gumba . Gumbom „Dodaj“ se svi ispravni podaci o novim osobama unose u bazu podataka. Važno je napomenuti da iako postoji funkcija koja upozorava jesu li pojedinačni podaci ispravno uneseni, moguće je pokušati unijeti djelomične u bazu podataka ali postoji programski dio nevidljiv korisniku koji sprječava unošenje takvih podataka te će izbaciti upozorenje.

OIB	Ime	Prezime	Broj telefona	Adresa	Poštanski broj	Datum zaraze	X	
12345678945	Nenad	Jukić	0996642215	Vinkovačka 36	10000	2022-06-13	X	
32165478945	Mima	Bilić	0789456123	Vinkovačka 10	10000	2022-06-23	X	
45678936921	Novak	Nović	0976642211	Panonska 25	31000	2022-06-23	X	
46512379650	Mima	Narik	0953547892	Radićeva 12	10000	2022-06-23	X	
46512379651	Mima	Satiš	0976642211	Panonska 25	10000	2022-06-23	X	
46512379652	Gojo	Jukić	0976642211	Radićeva 12	32100	2022-06-14	X	
46512379655	Mima	Rdić	0976642211	Vinkovačka 36	10000	2022-06-23	X	
46512398752	Rick	Bilić	0996642211	Radićeva 12	10000	2022-06-23	X	
78912345697	Mima	Morić	0976642211	Radićeva 12	10000	2022-06-23	X	
78965412357	Gojo	Narik	0976642211	Vinkovačka 36	10000	2022-06-23	X	

Sl. 4.3. Glavna stranica sa popisom zaraženih

**Dodavanje osoba**

OIB:

Ime:

Prezime:

Broj\_telefona:

Adresa:

Poštanski broj adrese:

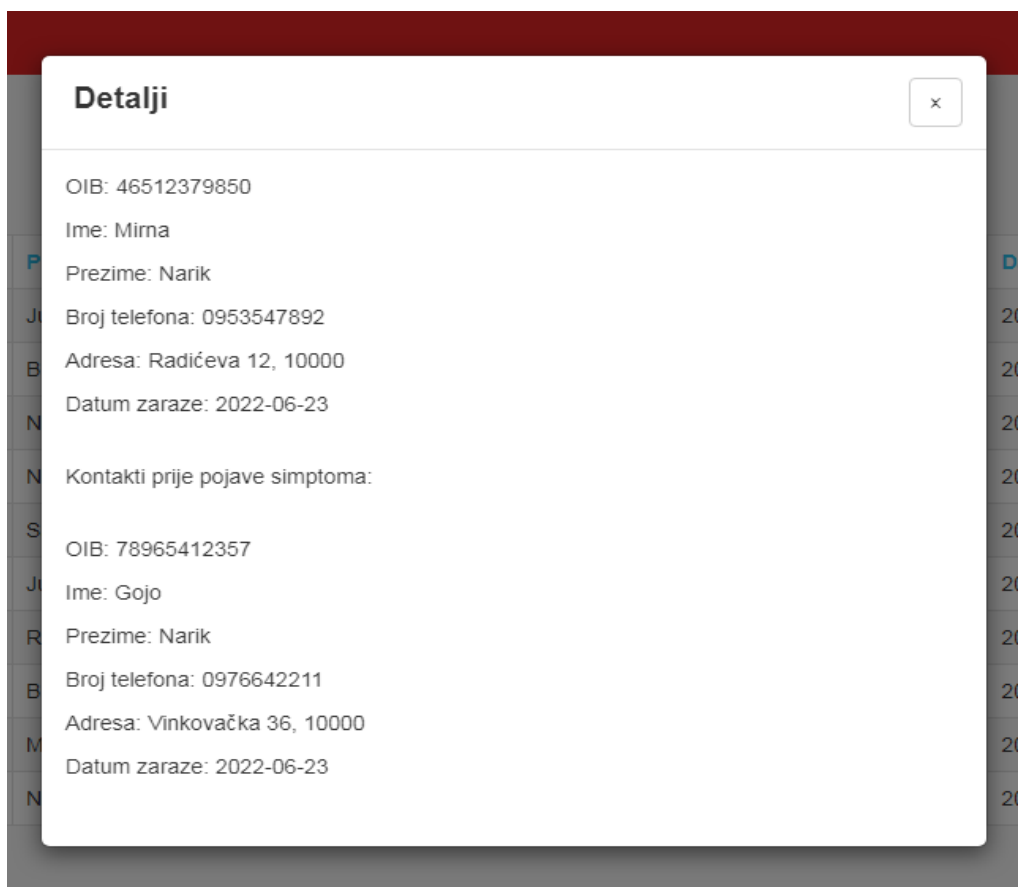
Datum zaraze:

**Niste unjeli ime.**

Sl. 4.4. Modalni obrazac za dodavanje osoba

Klikom na OIB osobe na popisu otvara se modalni prozor, vidljiv na slici 4.5, sa ispisom informacija o toj osobi te podaci svih osoba koji su uneseni zajedno s tom osobom.

S desne strane su smješteni gumbovi za printanje trenutno vidljivog popisa i za pretraživanje popisa po kriterijima. Unutar prozora za pretraživanje, slika 4.6, unose se podaci te se pritiskom na gumb „Traži“ stranica osvježava i popis sadržava osobe koje odgovaraju kriterijima. Gumbom „Reset“ se poništavaju svi kriteriji i popis se vraća u prvotno stanje. Također je moguće sortirati trenutni popis prema primjerice prezimenu pritiskom na riječ „Prezime“ pri vrhu tablice, a ponovnim klikom se sortira obrnutim redoslijedom.



Sl. 4.5. Podaci o bliskim kontaktima



Sl. 4.6. Opcije pretraživanja

Zadnja dva stupca u tablici predstavljaju gumbове za brisanje i uređivanje podataka. Oni su vidljivi samo povlaštenim korisnicima web aplikacije iz sigurnosnih razloga i mogućnosti davanja pristupa popisu većem broju ljudi. Pritiskom na „X“ pri vrhu stranice iskače prozor koji zahtjeva potvrdu brisanja osobe u pritisnutom redu. „X\*“ na vrhu stranice služi za brisanje cijelog trenutno vidljivog popisa što je korisno primjerice ako je potrebno obrisati osobe koje su se zarazile u određenom vremenskom periodu. Pritiskom na olovku iskače modalni prozor, sličan slici 4.4, koji sadrži podatke osobe te pritiskom na gumb „Promjeni“ se šalje zahtjev bazi podataka da promjeni podatke osobe.

## 4.1. Analiza dijelova aplikacije

Od prethodno spomenute 3 stranice najzahtjevnija za izradu je bila glavna stranica sa popisom kontakata jer je na njoj bilo potrebno dodati modalne obrasce za dodavanje kontakata, pregled popisa te omogućiti pretraživanje i sortiranje popisa po kriterijima unutar iste stranice bez da korisnik bude opterećen mogućnostima i informacijama na stranici.

### 4.1.1. Glavna stranica

Prilikom učitavanja glavne stranice potrebno je popuniti tablicu kako bi korisnik mogao vidjeti kontakte. Sa slike 4.7, nakon što je zahtjev za dohvaćanje popisa podataka izvršen sprema se u varijablu „result“ te se na ekran ispisuje broj dohvaćenih kontakata. U slučaju da su uspješno dohvaćeni kontakti iz baze pokreće se petlja koja uzima jedan po jedan kontakt te ispisuje red po red tablice sa podacima koji pripadaju kontaktu. Običnom korisniku su samo vidljivi podaci, ali ako je riječ o korisniku s povišenim povlasticama uz podatke se također dodaju opcije za brisanje i mijenjanje podataka unutar tablice. Kako je korišten Bootstrap za dizajniranje stranice bilo je potrebno dodati atribute „data-toggle“ i „data-target“ kako bi bilo moguće otvoriti modalni prozor klikom na OIB kontakta koji prikazuje sve njegove kontakte koji su završili

```
$result=$dbCon->query($sql);
echo '<script type="text/javascript">displayResultCount(',$result->num_rows,')</script>';
if($result->num_rows>0){
    while($row=$result->fetch_assoc()){
        echo "<tr><td><a data-toggle='modal' data-target='#myModal' class='oib'>",$row["oib"],"</a></td>
<td>",$row["ime"],"</td><td>",$row["prezime"],"</td>
<td>",$row["broj_telefona"],"</td><td>",$row["adresa_prebivališta"],"</td>
<td>",$row["pb_prebivališta"],"</td><td>",$row["datum_zaraze"],"</td>";
        if(intval($_SESSION["level"])>0){
            echo "<td><a href='javascript:confirmDeletion(',$row["oib"],")'>&#10006;</a></td>
<td><button data-toggle='modal' data-target='#myModal' role='dialog' class='convertedButton' onclick='getContactData(',$row["oib"],")>&#9998;</button></td>";
        }
    }
echo "</tr>";
```

Sl. 4.7. Popunjavanje tablice podacima

u samoizolaciji. Sortiranje i pretraživanje po kriteriju funkcionira na način da se prilikom svake promjene kriterija pamte vrijednosti kriterija u sesiji ponovo učita stranica te dohvaćaju podaci iz baze prema zadanim uvjetima.

### 4.1.2. Stranica za prijavu i registraciju

Stranica za prijavu i registraciju se sastoji od obrasca koji se popunjava te pritiskom na gumb za prijavu odnosno registraciju se poziva `attemptlogin.php` odnosno `checkuser.php` skripta koja prvo provjerava da li su uneseni zabranjeni znakovi u polja obrasca. U slučaju da jesu proces registracije/prijave se obustavlja i stranica ispisuje upozorenje, a ako ne aplikacija šalje zahtjev bazi podataka da provjeri da li uneseni korisnički podaci već postoje. Kod sa slike 4.8, koji predstavlja prijavu, nakon izvršavanja zahtjeva provjerava da li je pronađen korisnik u bazi podataka i ako je u aplikaciju se učitavaju podaci bitni za rad te ga se navodi na glavnu stranicu. Kod za registriranje korisnika je sličan samo se razlikuje u provjerama koje se rade. Šalje se zahtjev bazi podataka za provjeru postojećeg korisnika s istim podacima te ako ne postoji takav korisnik, novi se dodaje u bazu te se korisnika preusmjeruje na glavnu stranicu.

```
$excludedsymbols=array('+','-','/','*','$',';');
$username = $_POST['username'];
$password=$_POST['password'];
$warning=false;
foreach($excludedsymbols as $banned){
    if (str_contains($username,$banned)){
        $warning=true;
        break;
    }
}
if(!$warning){
    $query = "SELECT * FROM users WHERE '$password'=password AND '$username'=username";
    $result = $dbCon->query($query);
    if($result->num_rows==1){
        $_SESSION["username"]=$username;
        $_SESSION["level"]=$result->fetch_assoc()["razina_pristupa"];
        include 'main.php';
    }
}
```

Sl. 4.8. Provjera ispravnosti unosa i postojećih podataka



## 5. ZAKLJUČAK

Kao što se moglo primjetiti prilikom čitanja ovog rada, izrada web aplikacija može se ubrzo zakomplicirati s obzirom na funkcionalnosti koje ta aplikacija mora imati i na koju se tehnologiju oslanja. Čak i uključivanje Bootstrap-a, koji se zapravo sastoji od gotove CSS i Javascript datoteke, na stranicu zahtjeva određenu količinu vremena i proučavanja dokumentacije dostupne na internetu kako bi se postigao željeni izgled stranice. Najbitniji zadatak tijekom izrade aplikacije je zapravo bio ograničiti pristup poslužitelju na samo određene datoteke koje pripadaju web aplikaciji kako bi se spriječio neovlašten pristup stranici a time podacima na bazi podataka. To je uspješno napravljeno za sve „ne-korisnike“ koji su automatski preusmjereni na stranicu za prijavu ili su u potpunosti odbijeni. No čak i uza sve rečeno, autor ne može s potpunom sigurnošću potvrditi da je aplikacija zaštićena od neovlaštenih pristupa te je poželjno da se aplikacija analizira i ponovo testira kako bi se utvrdili sigurnosni propusci. Svaki od programa korištenih u izradi aplikacije je besplatan za korištenje, osim u slučaju korištenja u komercijalne svrhe.

## LITERATURA

- [1] COVID potvrde, <https://eudigitalnacovidpotvrda.hr/hr/covid-potvrde/mobilna-aplikacija-covidgo>  
28.6.2022
- [2] Stop COVID-19, <https://www.koronavirus.hr/stop-covid-19-723/723> 28.6.2022
- [3] Corona-Warn app, <https://www.coronawarn.app/en/> 28.6.2022
- [4] Aarogya Setu, <https://www.aarogyasetu.gov.in/> 28.6.2022
- [5] Immuni app, <https://www.governo.it/it/approfondimento/immuni/15336> 28.6.2022 numeriraj i u  
tekstu poveži
- [6] XAMPP, <https://www.apachefriends.org/> 24.6.2022
- [7] W3Schools, <https://www.w3schools.com/> 26.6.2022

## SAŽETAK

Glavni cilj ovog rada bio je napraviti web aplikaciju koja omogućuje jednostavan pregled popisa zaraženih kontakata prijavljenog korisnika i prikazom nedavnih kontakata pojedinih osoba. Aplikacija omogućuje bilo kojoj osobi s odgovarajućim pristupnim podacima da se prijavi i dobije uvid i cijeli popis koji se nalazi u bazi podataka koja se nalazi na udaljenom poslužitelju. Svi korisnici imaju mogućnost dodavanja osoba na popis te pretraživanje popisa po određenim kriterijima. Uz to, korisnici s većim privilegijama imaju mogućnost mijenjanja podataka osoba na popisu, u slučaju da postoje krivo unešeni podaci, te brisanje pojedinaca s popisa. Web aplikacija je dostupna u svakom trenutku svakoj osobi dokle god je poslužitelj spojen s Internetom. Teorijski dio sadrži opis načina izrade stranice pomoću korištenja Bootstrap-a i Javascript-a i način uspostavljanja Apache poslužitelja i spajanje na bazu podataka pomoću PHP-a.

**Ključne riječi:** Apache, Bootstrap, Javascript, korisnik, PHP, popis

## **ABSTRACT**

### **Web application for logging contacts for self-isolation purposes**

The main goal of this undergraduate paper was to create a web application that enables a simple view of the list of infected contacts of logged in user with a display of recent contacts of individuals. The application allows anyone with the valid access credentials to log in and see the entire list contained in a database located on a remote server. All users have the option of adding people to the list and searching the list according to certain criteria. In addition, users with higher privileges have the possibility to change the data of people on the list, in case there is false data, and to delete individuals from the list. The web application is available at any time to anyone as long as the server is connected to the Internet. The theoretical part contains a description of how Bootstrap and Javascript are used and how to establish an Apache server and connect to a database using PHP.

Keywords: Apache, Bootstrap, Javascript, user, PHP, list