

TEHNIČKA ŠKOLA RUĐERA BOŠKOVIĆA
GETALDIĆEVA 4, ZAGREB

ZAVRŠNI STRUČNI RAD:

Čitanje RFID i
zapisivanje u bazu
podataka te
enkripcija



MENTOR:
Braslav Erpačić dipl. ing.

UČENIK: Luka Šarić
RAZRED: 4.F

Sadržaj

1. UVOD	1
1.1 KRATKI OPIS STRUČNOG ZAVRŠNOG RADA	1
1.2 SHEMA FIZIČKOG SPOJA	1
1.3 SHEMA PROGRAMSKOG DIJELA	2
1.4. NACRT KUĆIŠTA ARDUINA I RFID ČITAČA	2
2. ARDUINO I HARDVER	3
2.1. UVOD U ARDUINO I ARDUINO IDE	3
2.2. ARDUINO U ZAVRŠNOM RADU	3
2.3. RGB DIODA	4
2.4. RFID-RC522	4
2.5. KOD NA ARDUINU	5
3. LOKALNA BAZA PODATAKA	6
3.1 XAMPP	6
3.2. BAZA PODATAKA XAMPP	7
3.3. APACHE XAMPP	8
4. GLAVNI PROGRAM	10
4.1. UKRATKO O PROGRAMU „MIND“	10
4.2. KOD PROGRAMA „MIND“	10
4.3. PRIMJER RADA PROGRAMA	13
5. PROGRAM ZA UPIS	15
5.1. UKRATKO O PROGRAMU „UPIS“	15
5.2. UKRATKO O WIN FORMS	15
5.3. KOD PROGRAMA „UPIS“	15
5.4. PRIMJER RADA PROGRAMA „UPIS“	17

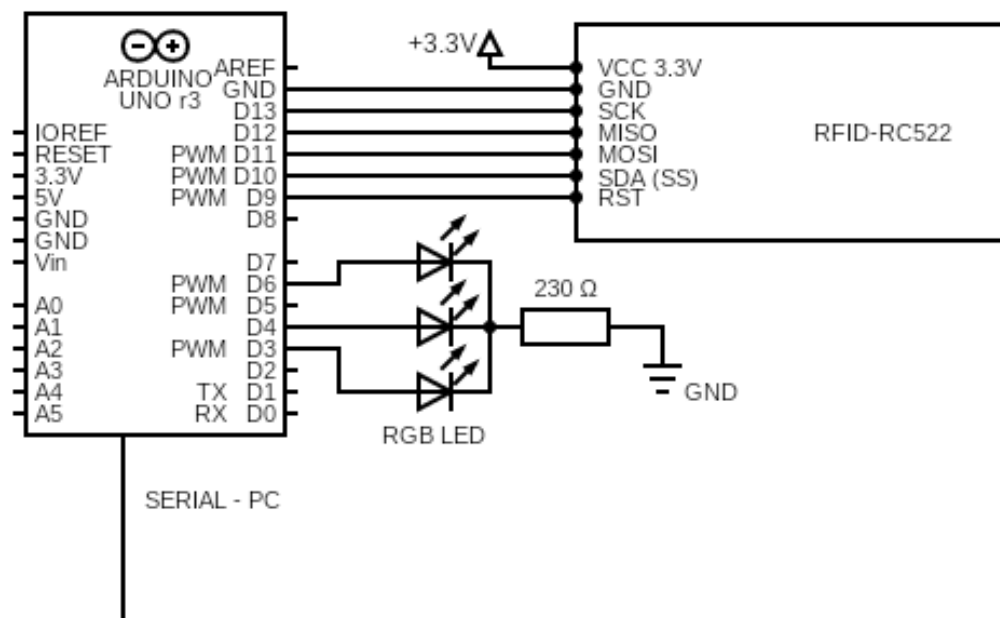
6. WEB STRANICA ZA ADMINISTRACIJU BAZE PODATAKA	19
6.1. UKRATKO O ADMINISTRACIJSKOJ WEB STRANICI	19
6.2. NEKI OD BITNIJIH KODOVA STRANICE	20
6.3. PRIMJER RADA ADMINISTRACIJSKE STRANICE	23
7.SIGURNOST SISTEMA	27
8. ZAKLJUČAK	28
9. LITERATURA	29

1. Uvod

1.1 Kratki opis stručnog završnog rada

Rad se temelji na čitanju RFID kartice pomoću čitača koji radi pomoću Arduino platforme. Pročitana kartica na siguran se način uspoređuje sa lokalnom bazom podataka koja radi pomoću XAMPPa. Dodatne funkcionalnosti su: administracija lokalne baze (upis u bazu, ispis iz baze, brisanje podataka baze, izmjena podataka baze...), RGB led indikator pri korištenju čitača koji simulira otvorena vrata.

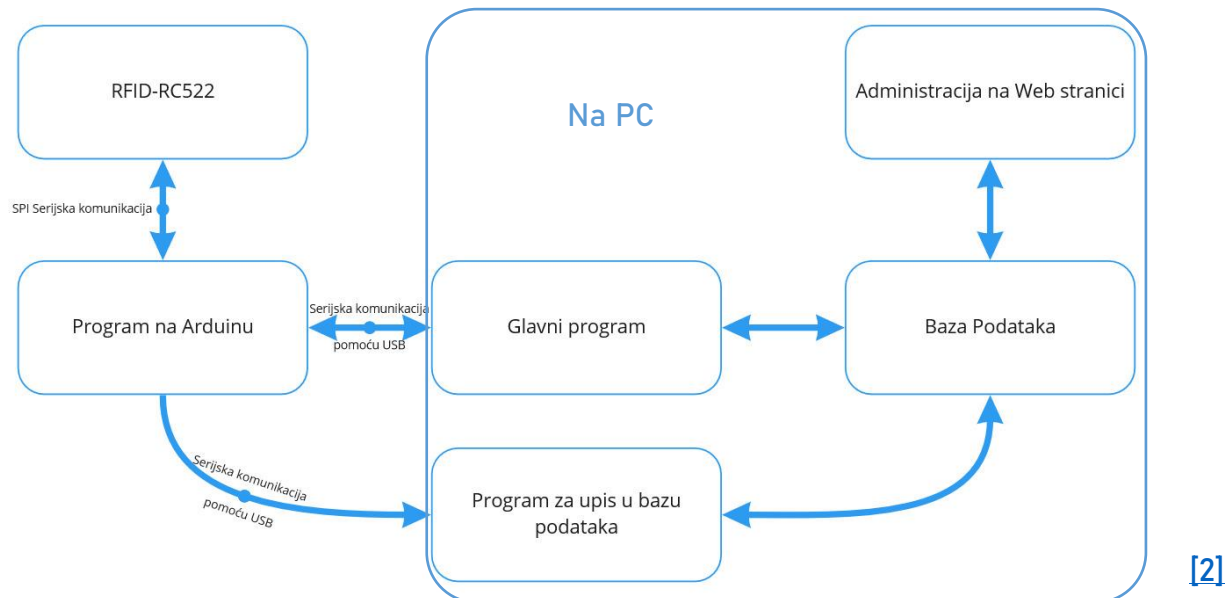
1.2 Shema fizičkog spoja



[1]

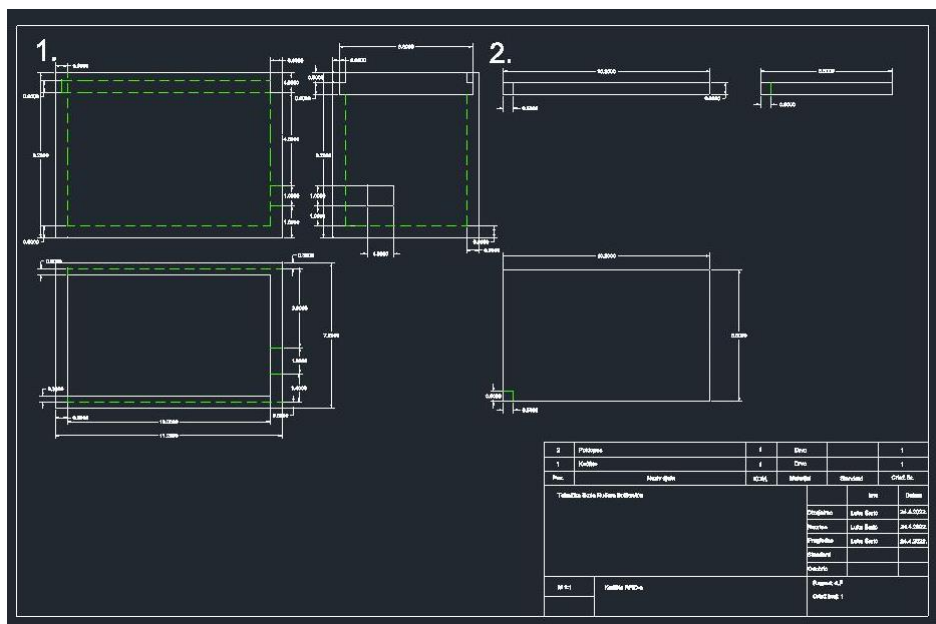
Prikaz načina spajanja komponenta na Arduino koji se serijski spaja na željeno računalo s bazom podataka i ostalim programima. Arduino koristi SPI komunikaciju kako bi komunicirao s RC522 te USB komunikaciju kako bi komunicirao sa programima na PC.

1.3 Shema programskog dijela



Prikaz sadržaja programa cjelokupnog rada te prikaz komunikacije između njih. Program Arduina napisan je u Arduino IDE, glavni program te program za upis napisani su u C#. Baza podataka napravljena je u MySQLu pomoću XAMPP te se za web stranicu koristi HTML, PHP, CSS.

1.4. Nacrt kućišta Arduina i RFID čitača



Primjer nacrtu za kućište Arduina i RFID čitača u programu AutoCAD.

2. Arduino i hardver

2.1. Uvod u Arduino i Arduino IDE

Arduino je kompanija koja se bavi proizvodnjom hardvera i softvera. Svojim uspjehom su kreirali veliku open-source platformu. Oni proizvode jedno-pločne mikro kontrolere koji su open-source i programiraju se pomoću Arduino IDE. Arduino IDE je IDE napisan u Javi, te služi za lakše pisanje koda za Arduino mikro kontroler. Sama struktura koda i kompiliranje slični su C i C++. Danas je mnogo električnih komponenta kompatibilno s Arduinom te ga to čini odličnom platformom za učenje i testiranje. [\[3\]](#)

2.2. Arduino u završnom radu

U završnom radu Arduino sam koristio kao procesor informacija i komponentu koja može koristiti RFID-RC522 čitač.



Arduino Uno R3

Već navedenim načinom spajamo RFID-RC522 modul te RGB LED diodu.

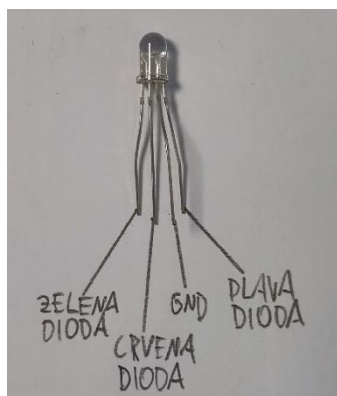
Arduino Pin	Pin na komponenti
D3~	RGB Led plava dioda
D5~	RGB Led crvena dioda
D6~	RGB Led zelena dioda
D9~	RC522 RST

D10~	RC522 SDA
D11~	RC522 MOSI
D12	RC522 MISO
D13	RC522 SCK
3.3V	RC522 3.3V
GND	RC522 GND, RGB Led zajednička katoda

Razlog spajanja dioda RGB Led na D3,D5,D6 je zbog PWM. PWM omogućava lažan analogni pin. To radi na način da šalje pulsove el. Energije. Ako želimo veću vrijednost (0-255) slat će ih češće, a ako želimo manju rjeđe. Kako bi RGB dioda radila, njezina zajednička katoda treba biti spojena na GND.

Razlozi spajanja D10,D11,D12,D13 su zbog SPI komunikacije, u realnosti mogli bismo ih i spojiti i na neke druge pinove, ali bi tako trebali definirati u programu. D9 je RST pin RC522, te također mora biti spojen i definiran u programu. 3.3V i GND spajamo kako bi modul dobivao el. struju potrebnu za rad.

2.3. RGB dioda



RGB dioda je na prvi pogled komplicirana komponenta, ali zapravo je jednostavna. Ona je ništa više od tri diode spojene zajedno. Može biti izvedena kao zajednička anoda ili zajednička katoda (kao na slici).

Zajednička katoda radi tako da joj šaljemo određeni napon. Što je napon veći to ona jače svijetli. Različite boje dobivamo kombinacijom jače svjetlećih dioda i manje svjetlećih dioda.

2.4. RFID-RC522



RFID-RC522 je modul koji može čitati RFID kartice, tagove i slično, a RFID kartice u sebi imaju antenu koja omogućuje slanje podataka čitaču kada se nađe u blizini. RFID je vrsta bežične komunikacije koja se služi elektromagnetskim ili elektrostatičkim spajanjem u jedinstvenom radio frekvencijskom spektru. [4]

U današnje vrijeme koristi se u različitim aspektima života. Identifikacija osoba, životinja, brzo plaćanje karticom...

2.5. Kod na Arduinou

Na Arduinou u jedno vrijeme može biti samo jedan kod te za ovaj završni rad Arduino koristi ovaj kod:

```
#include "SPI.h"
#include "MFRC522.h"
```

1.

1. Library potrebni za rad SPI i RC522.

```
const int pinRST = 9;
const int pinSDA = 10;
const int pinB = 3;
const int pinR = 5;
const int pinG = 6;
```

2.

2. Deklaracija varijabli za lakše korištenje („sta“ je varijabla za status jesu li vrata otvorena ili zatvorena).

```
char sta = 'z';
```

3.

3. Deklaracija RC522

```
MFRC522 mfrc522(pinSDA, pinRST);
```

void setup(){} - funkcija koja se pokreće pri pokretanju Arduinoa.

```
void setup() {
  pinMode(pinB, OUTPUT);
  pinMode(pinR, OUTPUT);
  pinMode(pinG, OUTPUT);
```

4.

4. Definiranje pinova kao ulaznih ili izlaznih (Za RGB diodu)

```
  analogWrite(pinR, 255);
  analogWrite(pinG, 0);
  analogWrite(pinB, 0);
```

5. Početak SPI komunikacije, čitanja s RC522 te Serijske komunikacije s PC.

```
  SPI.begin();
  mfrc522.PCD_Init();
  Serial.begin(9600);
}
```

5.

loop(){} - funkcija koja se pokreće svakih nekoliko milisekundi (stalno).

6. Petlja koja čeka očitavanje kartice, nakon očitavanja čita njezin ID i šalje ga na Serijski port.

7. Nakon čekanja odgovora, ova petlja otvara „vrata“ ili ih zatvara ovisi jeli dobila odgovor „o“ -otvori ili „z“ - zatvori.

```
void loop() {
  if (mfrc522.PICC_IsNewCardPresent()) {
    if (mfrc522.PICC_ReadCardSerial()) {
      for (byte i = 0; i < mfrc522.uid.size; ++i) {
        Serial.print(mfrc522.uid.uidByte[i], HEX);
        Serial.print(" ");
      }
      Serial.println();
      delay(300);
```

6.

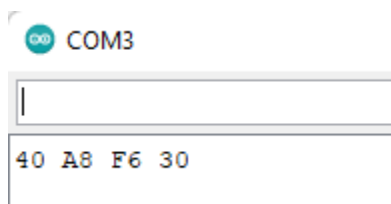
```
      if (Serial.available() > 0) {
        if (Serial.read() == 'o') {
          analogWrite(pinR, 0);
          analogWrite(pinG, 255);
          analogWrite(pinB, 255);
          sta = 'o';
        }
      }
      delay(5000);
      analogWrite(pinR, 255);
      analogWrite(pinG, 0);
      analogWrite(pinB, 0);
      sta = 'z';
    }
  }
}
```

7.

„Vrata“ su predstavljena RGB diodom.

Kada dioda svijetli crveno vrata su zatvorena, a kada svijetli kombinacijom plave i zelene su otvorena.

Primjer ispisa ID-a kartice:

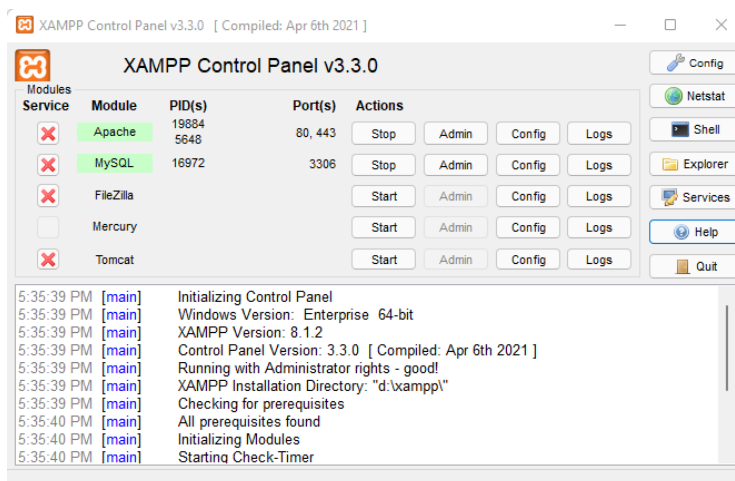


3. Lokalna baza podataka

3.1 XAMPP

Kao lokanu bazu podataka odabrao sam XAMPP koji je također i Apache s MariaDB i PHP. XAMPP je open-source paket koji je napravljen da bude jednostavan i lagan za instalirati. [\[5\]](#)

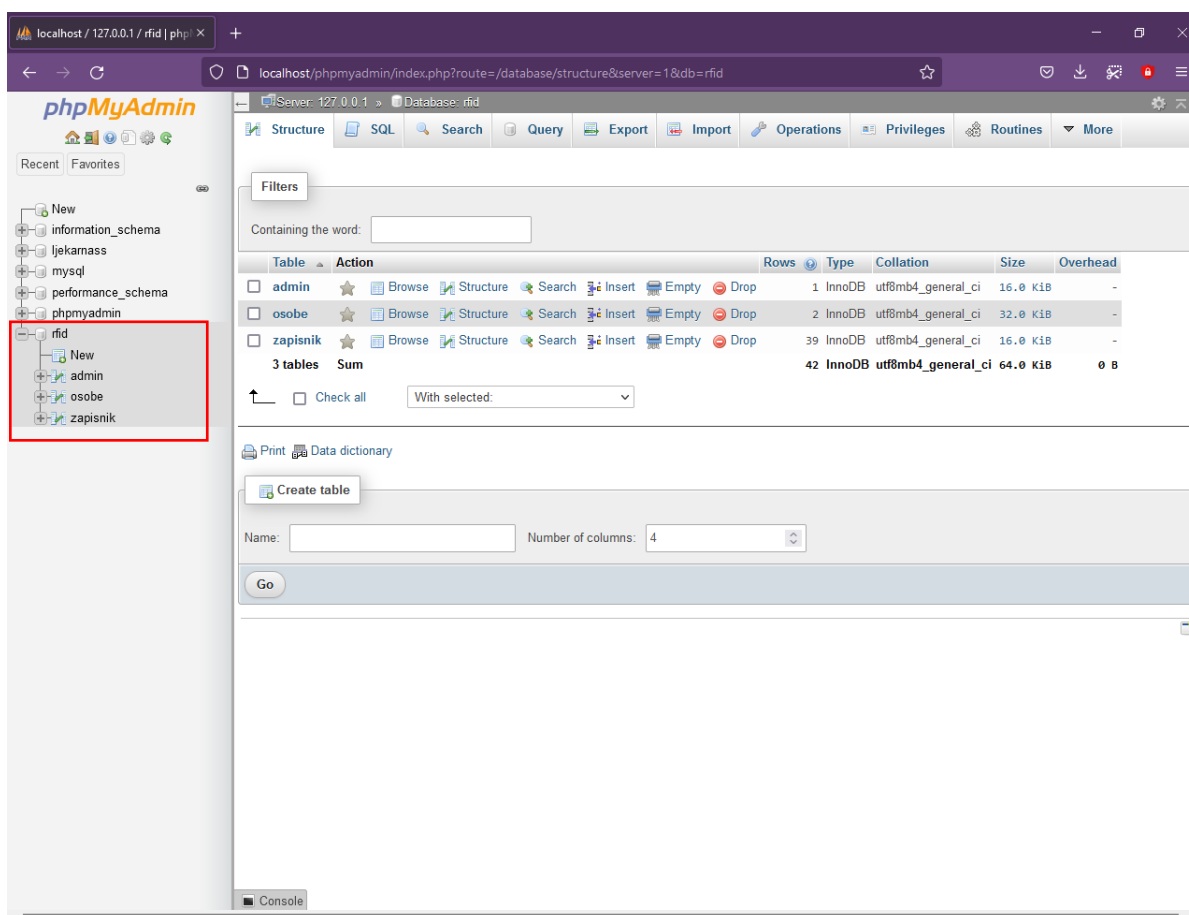
U radu je korišten kako bi se kreirala baza te kasnije pokrenuo PHP web stranice s kojim se uzimaju podatci iz te iste baze podataka.



Izgled XAMPP control panela. Preko kojeg lagano možemo pristupiti Apacheu ili MySQL bazi podataka.

Nakon što upalimo Apache i MySql pritiskom na gumb „Start“, možemo pristupiti određenom aspektu tako da stisnemo gumb „admin“.

Završni stručni rad – Lokalna baza podataka



MySQL baza podataka, s već kreiranom bazom „rfid“.

3.2. Baza podataka XAMPP

Prva tablica podataka je „osobe“:

+ Options		IB	ID	Ime	Prezime	Salt	Email
		Identifikacijski broj osobe		HASHirani i SALTani identifikacijski broj kartice			
<input type="checkbox"/>	Edit Copy Delete	1	C6D2D124099111E5B8DB179F8167CB949A7096E123015F7A12...	Luka	Šarić	+HeVaTnLzjoTqIXFJKUBfui88MEi73DV	saricluka76@gmail.com
<input type="checkbox"/>	Edit Copy Delete	2	A6D6DF82D482553F6CC7054A216A74213BC93DE08DE6610FC0...	Ivo	Ivić	0it8yZXVKhQ69x3D3d3lUKTJt7QC1q6U	ivicivo@gmail.com

U njoj se nalaze općenite informacije o svakom registriranom korisniku te ID kartice koji je bila HASHiran i SALTan zajedno s SALTom koji je bio korišten. Ova tablica koristi se pri pregledu postoji li korisnik u bazi. Ako je korisnik u ovoj tablici vrata će se otvoriti, kao nije neće.

Završni stručni rad – Lokalna baza podataka

Druga tablica podataka je „zapisnik“:

	ZapisID Identifikacijski broj zapisa	Vrijeme Vrijeme korištenja RFID	IB Identifikacijski broj osobe
<input type="checkbox"/> Edit Copy Delete	1	4/15/2022 10:31:24 PM	1
<input type="checkbox"/> Edit Copy Delete	2	4/15/2022 10:31:42 PM	1
<input type="checkbox"/> Edit Copy Delete	3	4/15/2022 10:31:56 PM	2
<input type="checkbox"/> Edit Copy Delete	4	4/15/2022 10:37:12 PM	3
<input type="checkbox"/> Edit Copy Delete	5	4/15/2022 10:37:27 PM	2
<input type="checkbox"/> Edit Copy Delete	6	4/15/2022 10:37:34 PM	1
<input type="checkbox"/> Edit Copy Delete	7	4/15/2022 10:37:42 PM	2
<input type="checkbox"/> Edit Copy Delete	8	4/16/2022 1:49:52 PM	1
<input type="checkbox"/> Edit Copy Delete	9	4/16/2022 1:50:17 PM	2
<input type="checkbox"/> Edit Copy Delete	10	4/16/2022 1:50:23 PM	1
<input type="checkbox"/> Edit Copy Delete	11	4/16/2022 1:50:35 PM	2
<input type="checkbox"/> Edit Copy Delete	12	4/16/2022 1:50:45 PM	2
<input type="checkbox"/> Edit Copy Delete	13	4/16/2022 1:52:15 PM	2

U ovoj tablici pohranjuju se dnevnički zapisi pristupanja registriranih i prijašnje registriranih korisnika RFID čitaču (vratima).

Treća tablica podataka je „admin“:

	AdminID Identifikacijski broj admina	username korisničko ime admina	password HASHirana i SALTana lozinka admina
<input type="checkbox"/> Edit Copy Delete	1	luka	\$2y\$12\$4rJrvepyp8bKMX5fZHnBi.KdP6GPsWEI2JrUiIKIETW...

U ovoj tablici pohranjena su korisnička imena i HASHirane te SALTane lozinke administratora, s kojim im je dopuštena prijava na stranicu administracije baze.

Ova baza podataka parti sve norme kao što su atomizacija podataka, ne potrebno ponavljanje podataka i slično... IB se ponavlja u dvije tablice kako bi ih kasnije mogli spajati. Ova baza podataka ne bi trebala biti vidljiva bilo kome pa čak ne ni administratoru koji vrši administraciju.

3.3. Apache XAMPP

Apache je web server koji prima zahtjeve korisnika i nakon toga im šalje željene podatke. XAMPP koristim kao Apache za PHP koji uzima podatke iz baze podataka i prikazuje ih na web stranici.

Index of /

Name	Last modified	Size	Description
 Stranica - Ljekarna/	2022-04-09 18:00	-	
 Završni/	2022-04-16 14:40	-	
 applications.html	2019-08-27 16:02	3.5K	
 bitnami.css	2019-08-27 16:02	177	
 dashboard/	2022-01-21 17:46	-	
 favicon.ico	2015-07-16 17:32	30K	
 img/	2022-02-15 08:16	-	
 webalizer/	2022-02-15 08:16	-	
 xampp/	2022-02-15 08:16	-	

Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2 Server at localhost Port 80

Izgled Apache XMAPP-a, od ovuda možemo pokrenuti bilo koju datoteku koja se nalazi u folderu (npr. D:\xampp\htdocs\nasadatoteka.php).

Također možemo i pristupiti našoj stranici tako da jednostavno upišemo:

localhost/imefolderastranice

u bilo koji browser na našem računalu.

4. Glavni program

4.1. Ukratko o programu „Mind“



Program je pisan u C# u programu Visual Studio 2022 kao Console application. Visual Studio 2022 je program koji služi kao prevoditelj, služi za lako pisanje koda i sve ostale moderne funkcije današnjeg prevoditelja. Program bi trebao biti uključen na PC na koji je spojen Arduino. Trebao bi biti pokrenut svo vrijeme koje mislimo provjeravati pristup RFID čitaču. Iako je program stalno pokrenut, nije potrebno ga

gledati ili slično.

4.2. Kod programa „Mind“

Svi korišteni library za izradu programa:

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO.Ports; //library za port funkcije
using System.Security.Cryptography; //library za SHA512
using MySql.Data.MySqlClient; //library za spajanje i slanje zahtjeva na bazu
```

Početak glavnog dijela programa:

```
class Program
{
    static SerialPort port; //inicijalizacija varijable
    static void Main(string[] args)
    {
        int status = 0;
        Hash podaci = new Hash(); //kreiranje objekta
        Baza sad = new Baza();
        Check checkid = new Check();

        port = new SerialPort
        {
            PortName = "COM3", //definiranje imena porta
            BaudRate = 9600 //definiranje BaudRatea
        };
        port.Open(); //otvaranje porta
    }
}
```

„port“ varijabla služi za kreiranje virtualnog Serijskog porta COM3. Omogućuje nam čitanje Serijskog porta Arduina.

Kreiranje objekata omogućuje zvanje metoda tih klasa.

Definicija „port“ te inicijalizacija.

Završni stručni rad – Glavni program

```
while (true)
{
    1. string temp = port.ReadLine();
    if ((temp != "z\r") && (temp != "o\r") && (temp != "") && (temp != "\r"))
    {
        2. sad.Connect();
        3. temp = temp.Replace(" ", "");
        temp = temp.Replace("\r", ""); //formatiranje
        Console.WriteLine(DateTime.Now.ToString());
        //podaci.data = temp; //slanje plaintexta
        //podaci.Hashthis(); //zvanje metode za hash
        4. status = checkid.checkthis(podaci, temp, sad.ids, sad.ibs, sad.imena, sad.prezimana, sad.salts, sad.emails);

        if(status == 1)
        {
            port.Write("o");
        }
        else
        {
            port.Write("z");
        }

        //provjera
        //Console.WriteLine(podaci.hashData);
        //Console.WriteLine(DateTime.Now.ToString() + " RFID ID: " + temp);

        5. sad.ids.Clear();
        sad.imena.Clear();
        sad.prezimana.Clear();
        sad.salts.Clear();
    }
}
```

while(true){} petlja čini da se sve operacije unutar nje ponavljaju dok je program uključen.

1. Čitanje porta te eliminacija provjere bespotrebnih podataka. U slučaju slanja ID-a izvršava se ostatak naredbi.
2. Zvanje metode za spajanje na bazu.
3. Formatiranje upisa. I ispis trenutnog vremena na Console.
4. Zvanje metode za provjeru, koja vraća 1 ili 0 ovisеći postoji li korisni u bazi ili ne. Kada postoji vraća 1 te tako sljedeća petlja šalje „o“ Arduino, koji tada otvara vrata.
5. Čisti varijable korištene za spremanje trenutnog stanja baze.

Metoda Connect() kalse Baza:

```
public class Baza
{
    public List<string> ids = new List<string>();
    public List<string> ibs = new List<string>();
    public List<string> imena = new List<string>();
    public List<string> prezimana = new List<string>();
    public List<string> salts = new List<string>();
    public List<string> emails = new List<string>();
}
```

1 reference

Izgled klase Baze i metode Connect().

U klasi baza upisuju se svi podaci tablice osobe iz baze podataka. Upis obavlja upravo metoda Connect().

Završni stručni rad – Glavni program

```
public void Connect()
{
    MySqlConnection dbCon;
    MySqlConnectionStringBuilder builder = new MySqlConnectionStringBuilder();

    builder.Server = "127.0.0.1";
    builder.Database = "rfid";
    builder.UserID = "root";
    builder.Password = "";
    String connString = builder.ToString();

    dbCon = new MySqlConnection(connString);

    String query = "SELECT * FROM osobe";
    MySqlCommand getID = new MySqlCommand(query, dbCon);
    dbCon.Open();
    Console.WriteLine("*****");
    Console.WriteLine("Uspješno spojeno na Bazu podataka."/* + connString*/);
    MySqlDataReader reader = getID.ExecuteReader();

    while (reader.Read())
    {
        ids.Add(reader["ID"].ToString());
        ibs.Add(reader["IB"].ToString());
        imena.Add(reader["Ime"].ToString());
        prezimena.Add(reader["Prezime"].ToString());
        salts.Add(reader["Salt"].ToString());
        emails.Add(reader["Email"].ToString());
    }

    dbCon.Close();
    Console.WriteLine("Odspojeno.");
}
```

1. Kreiranje zahtjeva i spajanje na bazu.

2. Ispis na Console, te početak zapisa svih poda - taka na liste. Zatvaranje konekcije te slanje povratnih podataka na Console.

Metoda checkthis() klase Check:

```
2 references
public class Check
{
    Baza upis = new Baza();
    1 reference
    public int checkthis(Hash podaci, string plaintext, List<string> ids, List<string> ibs, List<string> imena, List<string> prezimena, List<string> salts, List<string> emails )
    {
        int brojilo = 0;
        foreach (string id in ids)
        {
            brojilo++;
            podaci.data = plaintext + salts[brojilo-1]; //slanje plaintexta
            podaci.Hashthis(); //zvanje metode za hash
            //Console.WriteLine(podaci.hashData);
            //Console.WriteLine(id);
            if (id == podaci.hashData)
            {
                Console.WriteLine("Otvoreno za: " + imena[brojilo-1] + " " + prezimena[brojilo-1] + " IB: " + ibs[brojilo-1]);
                upis.Log(ibs[brojilo - 1]);
                brojilo = 0;
                return 1;
            }
        }
        return 0;
    }
}
```

Ova metoda za svaku osobu iz tablice osobe uzima slat, upotrebljuje ga s metodom HashThis() kako bi dobio HASH za novo pročitani ID kartice. Novi HASH tada uspoređuje s HASHom iz baze podataka, te ako je jednak vraća 1, ispisuje povratne informacije na Console te poziva metodu Log(). U protiv vraća 0.

Završni stručni rad – Glavni program

Metoda HashThis() klase Hash:

```
3 references
public class Hash
{
    public string data; //plaintext hasha
    byte[] hashed; //hash text
    public string hashedData; //poljepsani hash text

    1 reference
    public void Hashthis()
    {
        SHA512 hashSvc = SHA512.Create(); //kreiranje SHA512

        hashed = hashSvc.ComputeHash(Encoding.UTF8.GetBytes(data)); //hashanje
        hashedData = BitConverter.ToString(hashed).Replace("-", ""); //uljepšavanje
    }
}
```

Ova metoda na jednostavan način korištenjem librarya HASHa ID kartice zajedno sa SALTom pomoću algoritma SHA512.

Metoda Log() klase Baza:

```
public void Log(string ib)
{
    MySqlConnection dbCon;
    MySqlConnectionStringBuilder builder = new MySqlConnectionStringBuilder();

    builder.Server = "127.0.0.1";
    builder.Database = "rfid";
    builder.UserID = "root";
    builder.Password = "";
    String connString = builder.ToString();

    dbCon = new MySqlConnection(connString);

    DateTime sada = DateTime.Now;
    String query = "INSERT INTO zapisnik (Vrijeme, IB) VALUES ('" + sada + "', '" + ib + "')";
    MySqlCommand upis = new MySqlCommand(query, dbCon);
    dbCon.Open();

    Console.WriteLine("-----");
    Console.WriteLine("Uspješno spojeno na Bazu podataka."/* + connString*/);
    MySqlDataReader reader = upis.ExecuteReader();

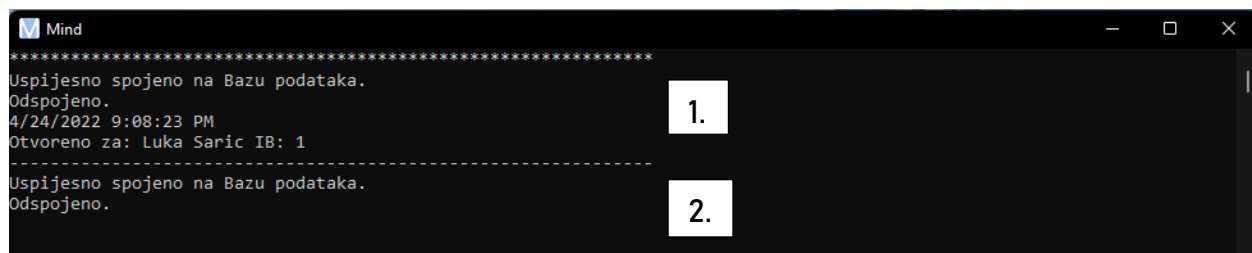
    while (reader.Read())
    {
    }
    dbCon.Close();
    Console.WriteLine("Odspojeno.");
}
```

Slično metodi Connect(), razlikuje se po zahtjevu.

Zahtjev u ovom slučaju nije čitanje podataka iz baze već zapis u nju.

Ova metoda u tablicu zapisnik zapisuje IB osobe i vrijeme pristupanja RFID čitaču, ostatak je identičan.

4.3. Primjer rada Programa



















































```
Mind
*****
Uspijesno spojeno na Bazu podataka.
Odspojeno.
4/24/2022 9:08:23 PM
Otvoreno za: Luka Saric IB: 1
-----
Uspijesno spojeno na Bazu podataka.
Odspojeno.
```

1. Primjer rada za postojećeg korisnika
2. Primjer rada za nepostojećeg korisnika.

Završni stručni rad – Glavni program

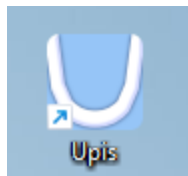
+ Options

		ZapisID	Vrijeme	IB
		Identifikacijski broj zapisa	Vrijeme korištenja RFID	Identifikacijski broj osobe
<input type="checkbox"/>	 Edit	 Copy	 Delete	26
<input type="checkbox"/>	 Edit	 Copy	 Delete	27
<input type="checkbox"/>	 Edit	 Copy	 Delete	28
<input type="checkbox"/>	 Edit	 Copy	 Delete	29
<input type="checkbox"/>	 Edit	 Copy	 Delete	30
<input type="checkbox"/>	 Edit	 Copy	 Delete	31
<input type="checkbox"/>	 Edit	 Copy	 Delete	32
<input type="checkbox"/>	 Edit	 Copy	 Delete	33
<input type="checkbox"/>	 Edit	 Copy	 Delete	34
<input type="checkbox"/>	 Edit	 Copy	 Delete	35
<input type="checkbox"/>	 Edit	 Copy	 Delete	36
<input type="checkbox"/>	 Edit	 Copy	 Delete	37
<input type="checkbox"/>	 Edit	 Copy	 Delete	38
<input type="checkbox"/>	 Edit	 Copy	 Delete	39
<input type="checkbox"/>	 Edit	 Copy	 Delete	40
<input type="checkbox"/>	 Edit	 Copy	 Delete	41

Prikaz novog zapisa u tablici zapisnik nakon skeniranja postojećeg korisnika.

5. Program za upis

5.1. Ukrato o programu „Upis“



Program je pisan u C# u programu Visual Studio 2022 kao Windows Forms application. Služi za registraciju novih korisnika. Prije pokretanja potrebno je ugasiti „Mind“. Nakon pokretanja, potrebno je samo skenirati RFID karticu, upisati podatke o korisniku te kliknuti gumb za upis. Nakon toga u bazu će uspješno biti dodan novi korisnik.

5.2. Ukratko o Win Forms

Windows Forms je dio programa Visual Studio 2022. On omogućuje lagano kreiranje aplikacija za Windows. Lagano se može promijeniti te nadodati bilo koji aspekt aplikacije što čini Windows Forms odličnim za brzo kreiranje manjih aplikacija.

5.3. Kod programa „Upis“

```
1 using System;
2 using System.Windows.Forms;
3 using System.IO.Ports; //library za port funkcije
4 using System.Security.Cryptography; //library za SHA512
5 using System.Text;
6 using System.ComponentModel;
7 using MySql.Data.MySqlClient;
8 using System.Drawing;
```

Svi libraryji potrebni za rad programa.

```
public partial class Form1 : Form
{
    Hash podaci = new Hash(); //kreiranje objekta
    string salt;
    1 reference
    public Form1()
    {
        InitializeComponent();
        backgroundWorker1.RunWorkerAsync();
        ispis.ForeColor = Color.Blue;
        ispis.Text = "Skenirajte karticu.";
    }
    2 reference
```

Kreiranje globalnog objekta i varijable.

Inicijalizacija Forme, zadavanje početnih vrijednosti i prikaza.

Završni stručni rad – Program za upis

```
private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e)
{
    SerialPort port; //inicijalizacija varijable

    port = new SerialPort
    {
        PortName = "COM3", //definiranje imena porta
        BaudRate = 9600 //definiranje BaudRatea
    };
    port.Open(); //otvaranje porta

    while (true)
    {
        string temp = port.ReadLine();
        if ((temp != "z\r") && (temp != "o\r") && (temp != "") && (temp != "\r"))
        {
            temp = temp.Replace(" ", "");
            temp = temp.Replace("\r", ""); //formatiranje
            salt = GetSalt();
            podaci.data = temp + salt; //slanje plaintexta
            podaci.Hashthis(); //zvanje metode za hash
            ispis.ForeColor = Color.Green;
            ispis.Text = "Generiran Hash i Salt. Nakon upisa ostalih informacija, možete kliknuti upis.";
        }
    }
}
```

Posebni Event, služi da možemo koristiti petlju bez kraja (while(true){}), u Windows Forms programu. Radi tako da jedan thread procesora radi jedan dio programa dok drugi, drugi dio. Ostatak koda je gotovo identičan glavnom dijelu koda „Mind“ programa.

Razlika je u tome što

nema ispisa na port, jer nemamo potrebu. Još jedna razlika je što imamo dodatnu metodu GetSalt().

Metoda HashThis() identična je istoj metodi programa „Mind“.

Metoda UpisBaza() gotovo je jednaka metodi Log() programa „Mind“:

```
void UpisBaza()
{
    MySqlConnection dbCon;
    MySqlConnectionStringBuilder builder = new MySqlConnectionStringBuilder();

    builder.Server = "127.0.0.1";
    builder.Database = "rfid";
    builder.UserID = "root";
    builder.Password = "";
    String connString = builder.ToString();

    dbCon = new MySqlConnection(connString);

    String query = "INSERT INTO osobe (Ime, Prezime, Email, ID, Salt) VALUES ('" + ime.Text + "','" + prezime.Text + "','" + email.Text + "','" + podaci.hashData + "','" + salt + "')";
    MySqlCommand upis = new MySqlCommand(query, dbCon);
    dbCon.Open();

    MySqlDataReader reader = upis.ExecuteReader();

    while (reader.Read())
    {
    }

    dbCon.Close();
    ispis.ForeColor = Color.Green;
    ispis.Text = "Upis obavljen!";
}
```

Razlika je u zahtjevu odnosno sadržaju koji se upisuje ,upisu u tablicu osobe a ne zapisnik te se ispis vrši na određeno polje u formi, a ne Console line.

Metoda GetSalt():

```
private static string GetSalt()
{
    var random = new RNGCryptoServiceProvider();
    byte[] salt = new byte[24];
    random.GetBytes(salt);
    return Convert.ToBase64String(salt);
}
```

Služi za generiranje nasumičnog 32 dugog SALTa koristeći library.

Nakon pozivanja vraća dobiveni SALT.

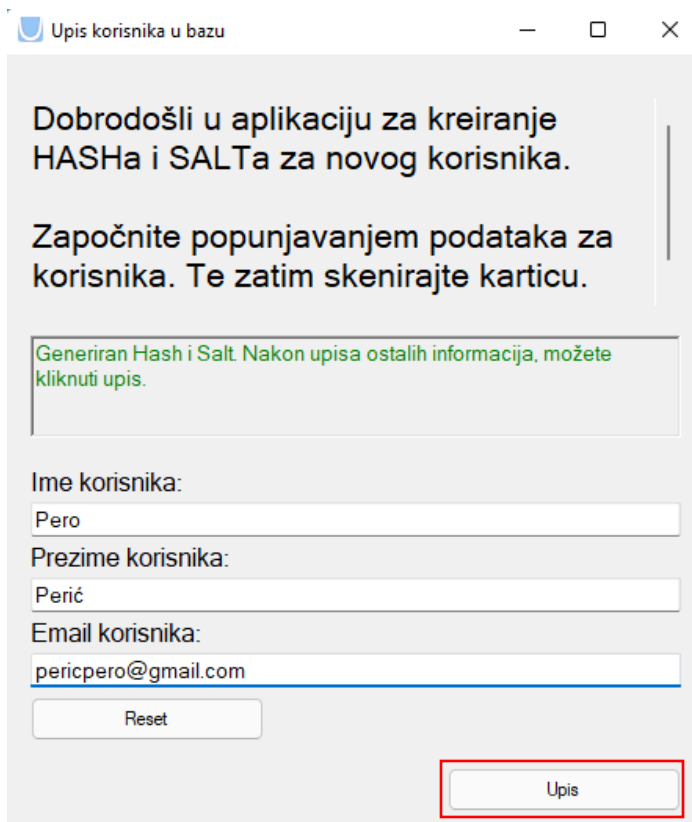
```
1 reference
private void reset_Click(object sender, EventArgs e)
{
    podaci.hashData = "";
    salt = "";
    ime.Text = "";
    prezime.Text = "";
    email.Text = "";
    ispis.ForeColor = Color.Blue;
    ispis.Text = "Skenirajte karticu.";
}
```

Event koji se aktivira pritiskom gumba reset. Resetira zasad upisane podatke, pripremajući za novi upis.

```
1 reference
private void upis_Click(object sender, EventArgs e)
{
    if ((ime.Text == "") || (prezime.Text == "") || (email.Text == "") || (podaci.hashData == "") || (salt == "")) {
        reset.PerformClick();
        ispis.ForeColor = Color.Red;
        ispis.Text = "Neka polja su prazna. Proces resetiran. Započnite ispočetka.";
    }
    else
    {
        UpisBaza();
    }
}
```

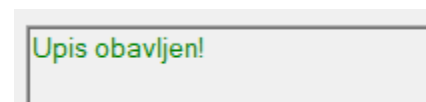
Event koji se aktivira pritiskom gumba upis. Provjerava jesu li sva polja popunjena. Ako nisu, pritišće gumb Reset i informira korisnika. Ako jesu, poziva funkciju UpisBaza().

5.4. Primjer rada programa „Upis“



Primjer programa nakon učitavanja kartice i prije klika gumba upis.

Promjena teksta nakon klika gumba upis:



Završni stručni rad – Program za upis

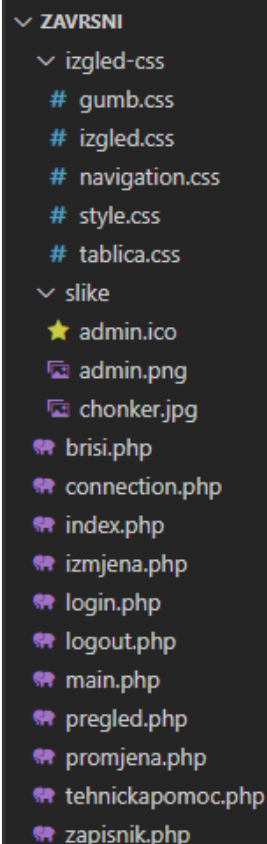
		IB	ID	Ime	Prezime	Salt	Email
		Identifikacijski broj osobe	HASHirani i SALTani identifikacijski broj kartice	Ime osobe	Prezime osobe	Salt	Email osobe
<input type="checkbox"/>	 Edit  Copy  Delete	1	C6D2D124099111E5B8DB179F8167CB949A7096E123015F7A12...	Luka	Šarić	+HeVaTnLzjoTqIXFKUBfkl88MEi73DV	saricluka76@gmail.com
<input type="checkbox"/>	 Edit  Copy  Delete	2	A6D6DF82D482553F6CC7054A216A74213BC93DE08DE6610FC0...	Ivo	Ivić	0it8yZXVKhQ69x3D3d3IUkTjt7QC1q6U	ivicivo@gmail.com
<input type="checkbox"/>	 Edit  Copy  Delete	6	A255C27728D94826A91A9371B70BF05211FE21665CE3560378...	Pero	Perić	CiZMsUT4GWFOTu4zKmdJdpYNIldMyciGy	pericpero@gmail.com

Prikaz upisanog korisnika u bazi podataka.

6. Web stranica za administraciju baze podataka

6.1. Ukratko o administracijskoj web stranici

Web stranica izrađena je u mnogo datoteka. Svi kodovi su pisani u Visual Studio Code programu, koji je jedan od najboljih za laku i kvalitetnu izradu web stranica. Sve stranice pisane su u html5 programskom jeziku zajedno sa PHP, koji stranici daje funkcionalnost i mogućnost komuniciranja s bazom podataka. Izgled stranice nalazi se u datotekama pisanih pomoću CSS-a.



```

ZAVRSNI
├── izgled-css
│   ├── gumb.css
│   ├── izgled.css
│   ├── navigation.css
│   ├── style.css
│   └── tablica.css
├── slike
│   ├── admin.ico
│   ├── admin.png
│   └── chonker.jpg
├── brisi.php
├── connection.php
├── index.php
├── izmjena.php
├── login.php
├── logout.php
├── main.php
├── pregled.php
├── promjena.php
├── tehnickapomoc.php
└── zapisnik.php

```

Ovo su sve datoteke Web stranice. Sve se nalaze u direktoriju Zavrzni. Kodovi web stranica nalaze se u tom istom direktoriju, dok se slike koje stranice koriste nalaze u direktoriju slike, a CSS kodovi u direktoriju izgled-css.

Završni stručni rad – Web stranica za administraciju baze podataka

3. main.php ima primjer koda koji koristi sesiju kako bi provjerio jeli korisnik valjano prijavljen na stranicu:

```
<?php
    session_start();
    if( ! isset($_SESSION["username"]) ){
        header("refresh:5;url=index.php");
        die('Samo prijavljeni korisnici mogu pristupiti ovoj stranici. Preusmjeravam...');
    }
    else{
        $username=$_SESSION["username"];
    }
?>
```

U slučaju da nije preusmjerava ga na index.php.

Primjer spajanja CSS datoteka s stranicom:

```
<link rel = "stylesheet" href = "izgled-css/izgled.css">
<link rel = "stylesheet" href = "izgled-css/navigation.css">
```

U CSS datotekama definirani su izgledi pojedinih elemenata stranice. CSS koji definira izgled horizontalne navigacije stranice (navigation.css):

```
main.php
1 <?php
2     session_start();
3     if( ! isset($_SESSION["username"]) ){
4         header("refresh:5;url=index.php");
5         die('Samo prijavljeni korisnici mogu pristupiti ovoj stranici. Preusmjeravam...');
6     }
7     else{
8         $username=$_SESSION["username"];
9     }
10 }>
11 <!DOCTYPE html>
12 <html lang="en">
13 <head>
14     <link rel="icon" href="slike/admin.png" type="image/x-icon">
15     <meta charset="UTF-8">
16     <meta http-equiv="X-UA-Compatible" content="IE=edge">
17     <link rel = "stylesheet" href = "izgled-css/izgled.css">
18     <link rel = "stylesheet" href = "izgled-css/navigation.css">
19     <meta name="viewport" content="width=device-width, initial-scale=1.0">
20     <title>Administracija - Main</title>
21 </head>
22 <body>
23     <ul>
24         <li></li>
25         <li><a class="active">Početna</a></li>
26         <li><a href="zapisnik.php">Zapisi posjeta</a></li>
27         <li><a href="pregled.php">Izmjena podataka baze</a></li>
28         <li><a href="tehnickapomoc.php">Tehnička pomoć</a></li>
29         <li><a href="logout.php">Odjava</a></li>
30     </ul>
31     <div class="div1">
32         <?php
33         echo "<h1 class='dobro'>Dobrodošli u administraciju baze podataka " . $username . "</h1>";
34     </div>
35     <h3>Administracija RFID čitača i baze podataka:</h3><br>
36     <p><b>Zapisi posjeta</b> - <i>Zapisi korištenja kartica na RFID čitaču.</i></p>
37     <p><b>Izmjena baze podataka</b> - <i>Brisanje, dodavanje i pregled korisnika</i></p>
38     <p><b>Tehnička pomoć</b> - <i>Kontakt kreatora za tehničku pomoć i ostala pitanja</i></p>
39     <p><b>Tehnička pomoć</b> - <i>Odjava iz administracije</i></p>
40     </div>
41 </body>
42 </html>
```

```
izgled-css > # navigation.css > li
1 ul {
2     list-style-type: none;
3     margin: 0;
4     padding: 0;
5     overflow: hidden;
6     background-color: #93ACB5;
7     position: static;
8     top: 0;
9     width: 100%;
10 }
11
12 li {
13     float: left;
14     border-right: 1px solid black;
15 }
16
17 li a {
18     display: block;
19     color: black;
20     text-align: center;
21     padding: 16px;
22     text-decoration: none;
23 }
24
25 li a:hover:not(.active) {
26     background-color: #67767C;
27 }
28
29 .active{
30     background-color: #A0D3FF;
31 }
32
33 .slika{
34     padding: 7px 7px 0px;
35     width: 38px;
36     height: 38px;
37 }
38
39 .slika2{
40     margin-top: 10%;
41     margin-left: 15%;
42     margin-right: 15%;
43     width: 70%;
44     height: 70%;
45 }
46
47 .slika2:active{
48     opacity: 100%;
49 }
```


4. connection.php PHP datoteka koja se koristi na svakom aspektu stranice koji treba komunicirati sa bazom, sličan je kodu iz programa „Mind“:

```
<?php
    $host = "localhost";
    $user = "root";
    $password = '';
    $db_name = "rfid";

    $con = mysqli_connect($host, $user, $password, $db_name);
    if(mysqli_connect_errno()) {
        die("Failed to connect with MySQL: ". mysqli_connect_error());
    }
?>
```

5. zapisnik.php je dio stranice koji služi za ispis tablice zapisnik. On pruža dodatnu funkcionalnost koristeći se INNER JOIN u zahtjevu koji šalje bazi podataka.

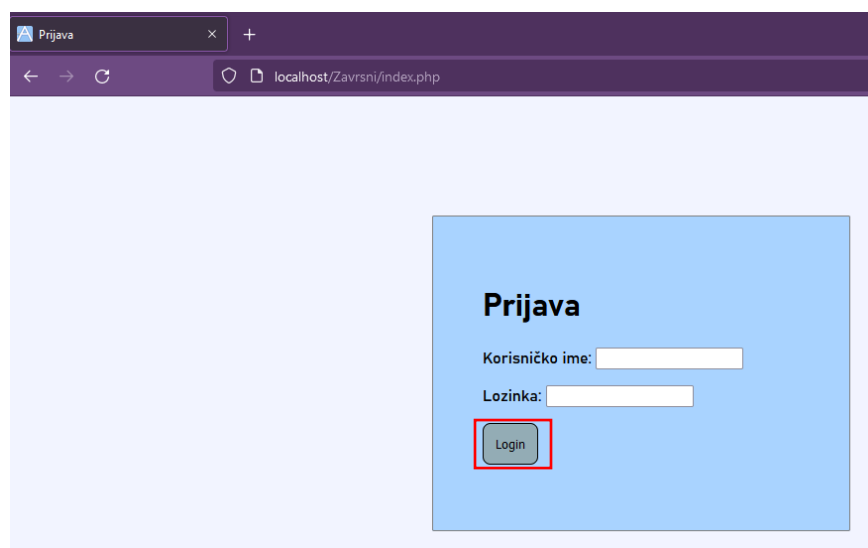
```
<?php
    include('connection.php');
    $sql = "SELECT * FROM zapisnik INNER JOIN osobe ON zapisnik.IB = osobe.IB;";
    $result = mysqli_query($con, $sql);
    if (mysqli_num_rows($result) == 0) {
        die("Sending the query failed! ");
    }
    while ($row = mysqli_fetch_array($result)) {
        echo "<div>";
        echo "<tr>";
        echo "<td class=\"ime\">" . $row['Vrijeme'] . "</td>";
        echo "<td class=\"ime\">" . $row['Email'] . "</td>";
        echo "<td class=\"ime\">" . $row['Ime'] . "</td>";
        echo "<td class=\"ime\">" . $row['Prezime'] . "</td>";
        echo "</tr>";
        echo "</div>";
    }
    mysqli_close($con);
?>
```

INNER JOIN služi za ispis podataka iz tablice osobe koji nisu napisani u tablici zapisnik a bitni su. Korištenjem INNER JOIN pratimo norme baze podataka, ali još uvijek možemo vidjeti sve korisne podatke.

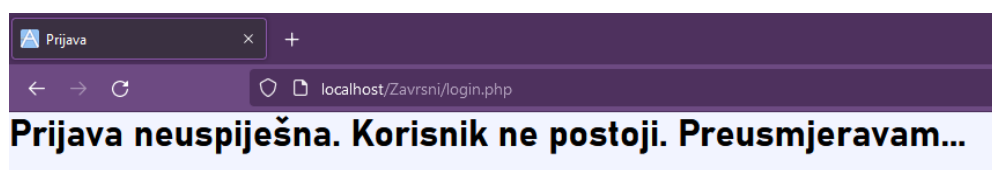
6. Ostali aspekti web stranice su slični normalnim web stranicama odnosno slični već obrađenim kodovima. Objašnjavanje svih kodova web stranice detaljno oduzelo bi mnogo prostora, energije i vremena, a mnogo kodova su repetitivni i monotoni.

6.3. Primjer rada administracijske stranice

1. Prijava korisnika (indeks.php + login.php):



Primjer neuspješne prijave:

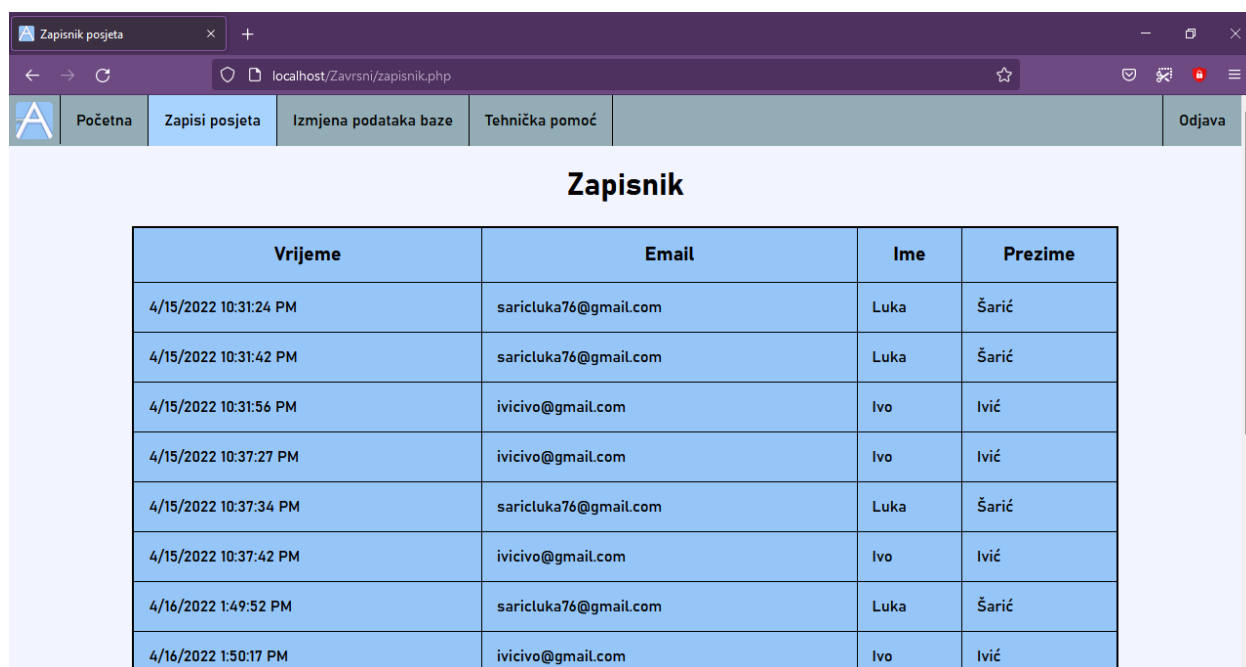


Otvaranje početne strance nakon uspješne prijave:



-stranica uzima korisničko ime administratora i koristi ga za pozdrav

2. Ispis zapisa posjeta na zapisnik.php:

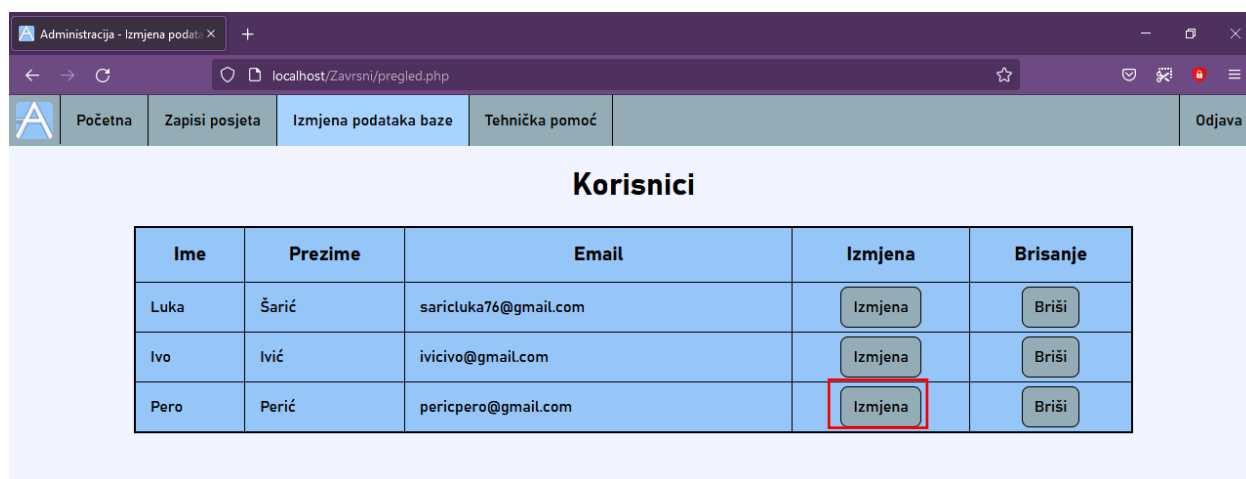


Vrijeme	Email	Ime	Prezime
4/15/2022 10:31:24 PM	saricluka76@gmail.com	Luka	Šarić
4/15/2022 10:31:42 PM	saricluka76@gmail.com	Luka	Šarić
4/15/2022 10:31:56 PM	ivicivo@gmail.com	Ivo	Ivić
4/15/2022 10:37:27 PM	ivicivo@gmail.com	Ivo	Ivić
4/15/2022 10:37:34 PM	saricluka76@gmail.com	Luka	Šarić
4/15/2022 10:37:42 PM	ivicivo@gmail.com	Ivo	Ivić
4/16/2022 1:49:52 PM	saricluka76@gmail.com	Luka	Šarić
4/16/2022 1:50:17 PM	ivicivo@gmail.com	Ivo	Ivić

-možemo vidjeti kako tablica ispisuje email, ime, i prezime iz tablice osobe, a vrijeme iz tablice zapisnik.

3. Izmjena baze podataka:

Izmjenu podataka korisnika i brisanje korisnika možemo obaviti na pregled.php



Ime	Prezime	Email	Izmjena	Brisanje
Luka	Šarić	saricluka76@gmail.com	Izmjena	Briši
Ivo	Ivić	ivicivo@gmail.com	Izmjena	Briši
Pero	Perić	pericpero@gmail.com	Izmjena	Briši

Završni stručni rad – Web stranica za administraciju baze podataka

-klikom na gumb Izmjena možemo izmijeniti podatke:

Ime:

Prezime:

Email:

Uspiješno promjenjeni podaci.

Korisnici

Ime	Prezime	Email	Izmjena	Brisanje
Luka	Šarić	saricluka76@gmail.com	<input type="button" value="Izmjena"/>	<input type="button" value="Briši"/>
Ivo	Ivić	ivicivo@gmail.com	<input type="button" value="Izmjena"/>	<input type="button" value="Briši"/>
Ivan	Perić	peric@gmail.com	<input type="button" value="Izmjena"/>	<input type="button" value="Briši"/>

-nakon izmjene podataka uistinu možemo vidjeti izmjenu.

Klikom na gumb briši brišemo korisnika iz baze:

Korisnici

Ime	Prezime	Email	Izmjena	Brisanje
Luka	Šarić	saricluka76@gmail.com	<input type="button" value="Izmjena"/>	<input type="button" value="Briši"/>
Ivo	Ivić	ivicivo@gmail.com	<input type="button" value="Izmjena"/>	<input type="button" value="Briši"/>
Ivan	Perić	peric@gmail.com	<input type="button" value="Izmjena"/>	<input type="button" value="Briši"/>

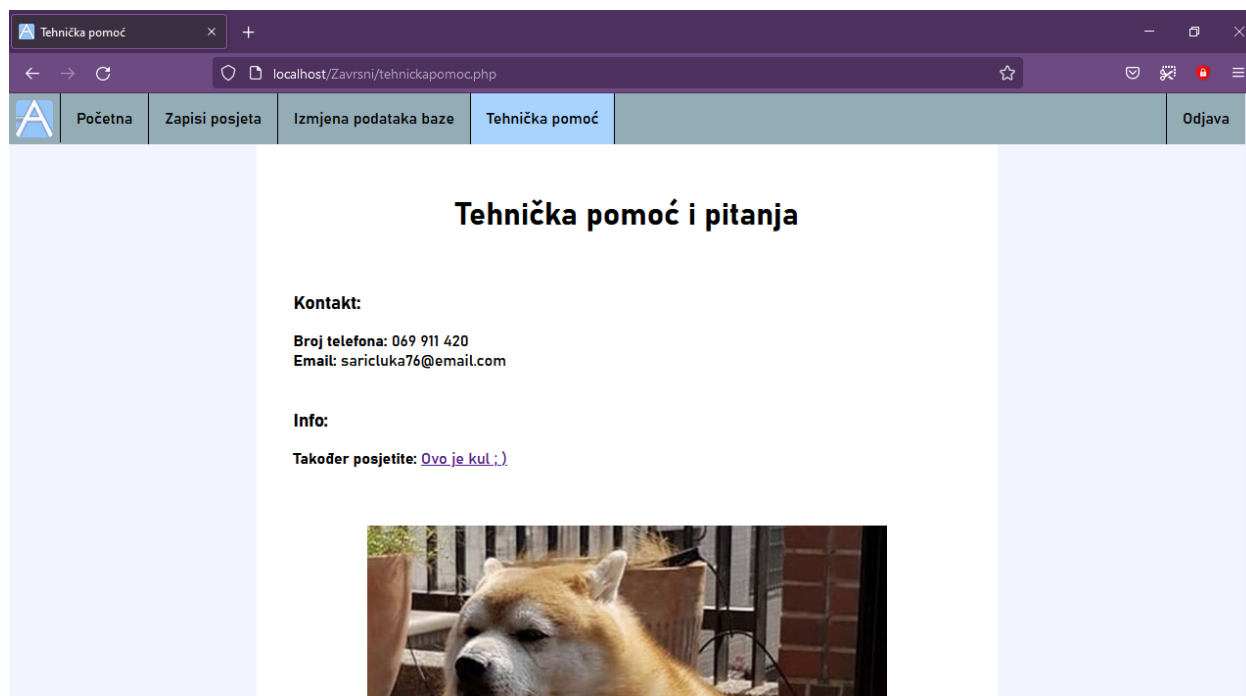
Korisnik uspješno izbrisan.

-povratkom možemo vidjeti da je korisnik izbrisan:

Korisnici

Ime	Prezime	Email	Izmjena	Brisanje
Luka	Šarić	saricluka76@gmail.com	<button>Izmjena</button>	<button>Briši</button>
Ivo	Ivić	ivicivo@gmail.com	<button>Izmjena</button>	<button>Briši</button>

4. Kontakt stranica



-stranica koja samo služi za tehničku pomoć i kontakt kreatora.

5. Odjava sa stranice:

Odjavljujemo se tako da jednostavno pritisnemo gumb odjava:



Nakon toga biti ćemo uspješno odjavljeni i preusmjereni na indeks.php.

Uspiješno odjavljeno, preusmjeravam...

7. Sigurnost sistema

Sigurnost svakog sistema jedna je od najbitnijih aspekta. Ako naš sistem na bilo koji način sprema vrijedne podatke ili služi za povećanje sigurnosti, užasno je važno da i sam sistem bude siguran. Naravno nikad nije moguće postići potpunu sigurnost, ali dodavanjem nekoliko elemenata koji služe za povećanje sigurnosti znatno možemo usporiti i demoralizirati napadača.

U mojem sistemu koji sam izradio kao završni rad, postoji nekoliko dodatnih elemenata sigurnosti:

1. Prijava administratora na administracijsku web stranicu
2. SQL injection prevencija
3. HASH i SALT lozinka odnosno ID-a RFID-a kartice

Prijava administratora na administracijsku web stranicu pomaže sigurnosti tako da samo određene povjerljive osobe mogu mijenjati podatke i čitati ih. Postojanje stranice samo po sebi ograničuje koje sve podatke se može pročitati (npr. HASH stalno ostaje skriven, niti administrator baze ga ne može vidjeti).

SQL injection prevencija sprječava moguće napade na bazu kojima bi se došlo do tajnih podataka.

HASH i SALT lozinke administratora povećavaju sigurnost u slučaju razotkrivanja baze podataka. Također skrivaju ID RFID kartice jer je u današnje vrijeme moguće nabaviti kartice kojima se ID može mijenjati. Ako bi napadom na bazu podataka dobili korištene ID-je mogli bi ih kopirati na lažnu karticu i pristupiti vratima.

8. Zaključak

Ovaj završnim radom dotaknuo je mnogo stručnih predmeta s kojim smo se susreli u ove protekle četiri godine. Izradom rada moglo se naučiti mnoge stvari koje nismo stekli u školovanju.

U zaključku, izrada završnog rada bila je odličan izvor znanja. Završetkom rada dobro se može uočiti mogućnost povezivanja raznolikih alata danas, način rada kojim se može proizvesti proizvod koji je gotovo spreman za tržište.

Samim završetkom rada svako malo dobivao sam ideju koja bi mogla još više unaprijediti sistem, te tako dok ne bude savršen. Savršen rad je teško postići kao amater, ali je važno znati svoje greške i dijelove koje bi htio promijeniti.

U svojem radu bi u realnoj implementaciji htio promijeniti:

1. Način komunikacije Arduina s PC
2. Spajanje programa na PC u jedan
3. Spajanje web stranice na Internet
4. Dodatna sigurnost sistema

9. Literatura

- [1] El. shema kreirana pomoću- <https://www.circuit-diagram.org>
- [2] Umna mapa kreirana pomoću- <https://miro.com>
- [3] Arduino i Arduino IDE - <https://www.arduino.cc/en/about>
- [4] RFID - <https://www.techtarget.com/iotagenda/definition/RFID-radio-frequency-identification>
- [5] XAMPP - <https://www.apachefriends.org/index.html>
- [6] Bcrypt slika - <https://en.wikipedia.org/wiki/Bcrypt>