**VISOKA TEHNIČKA ŠKOLA**

**STRUKOVNIH STUDIJA**

**S** **U B O T I C A**

**PROJEKAT**

**iz predmeta Web programiranje**

|  |  |
| --- | --- |
| **KANDIDAT(I)**  **Nikola Marjanušić** | **MENTOR**  **prof. dr Zlatko Čović** |
| **SUBOTICA, 2024.** | |

Sadržaj

[1. Opis zadatka 3](#_Toc156903284)

[2. Realizacija zadatka 4](#_Toc156903285)

[3. Struktura baze podataka (MySQL Workbench) 5](#_Toc156903286)

[4. Opis funkcionalnosti 6](#_Toc156903287)

[5. Korišćena literatura 7](#_Toc156903288)

# Opis zadatka

Cilj projekta je da se kreira web aplikacija-sajt za zakazivanje termina u zubarskoj ordinaciji. Aplikacija razlikuje četiri nivoa pristupa: gost, registrovan/prijavljen korisnik, zubar-radnik i administrator.

**web adresa projekta:<https://rr.stud.vts.su.ac.rs/>**

# Realizacija zadatka

U ovom delu treba prikazati i delove programskog koda. Programski kod napisati uz pomoć fonta **Courier, veličina slova 10**.

**Prvo povezujemo bazu podataka:**

<?php

$host = 'localhost';

$dbname = 'rr';

$username = 'rr';

$password = '6B0JAU1RlwqQAFM';

try {

    // Create a new PDO instance

    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);

    // Set PDO error mode to exception

    $pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

    //echo "Connected successfully";

} catch(PDOException $e) {

    echo "Connection failed: " . $e->getMessage();

}

**Moramo da procesujemo Login I da proverimo dali je informacija uneta dobro:**

    if (validateForm()) {

      submitForm();

  }

**validateForm uzima elemente forme I proverava dail su dobro popinjeni.**

function validateForm() {

    var flag = true

    const email = document.getElementById('email');

    const password = document.getElementById('password');

    const confirmPassword = document.getElementById('confirmPassword');

    emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

    if (!emailPattern.test(email.value)) {

        email.style.outline = "2px solid red";

        email.addEventListener('click', function(){removeWorning(email)});

        flag = false;

    }

    const passwordPattern = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&.])[A-Za-z\d@$!%\*?&.]{8,}$/;

    if (!passwordPattern.test(password.value)){

        password.style.outline = "2px solid red";

        password.addEventListener('click', function(){removeWorning(password)});

        flag = false;

    }

    else if(!(password.value === confirmPassword.value)){

        confirmPassword.style.outline = "2px solid red";

        confirmPassword.addEventListener('click', function(){removeWorning(confirmPassword)});

        flag = false;

    }

    return flag;

}

**submitForm salje podatke.**

function submitForm() {

    const formData = new FormData(document.getElementById('logIn'));

    console.log(formData)

    fetch('../../Backend/PHP/db\_Insert\_Login.php', {

        method: 'POST',

        body: formData

    })

    .then(response => response.json())

    .then(data => {

        console.log(data)

        if (data.success) {

            //alert('Form submitted successfully!');

            form.reset();

            window.location.href = 'index.php';s

        } else {

            alert('Form submission failed: ' + data.message);

        }

    })

    .catch(error => {

        // console.error('Error:', error);

        // alert('An error occurred during form submission.');

    });

}

function removeWorning(x){

x.style.outline = "none";

}

**Kada posajemo informaciju sa AJAX-om:**

**Poroverimo ponovo dal je sva informacija poslata.**

$email = $\_POST['email'] ?? '';

$password = $\_POST['password'] ?? '';

if (empty($email) || empty($password)){

    echo json\_encode(['success' => false, 'message' => 'Not all requierd elements are filled in']);

    exit;

}

**I ako jeste napravimo promene u bazi podataka:**

$stmt = $pdo->prepare('INSERT INTO user\_logins (user\_id, login\_date, login\_time) VALUES (:user\_id, :login\_date, :login\_time)');

            $stmt->bindParam(':user\_id', $user\_id);

            $stmt->bindParam(':login\_date', $login\_date);

            $stmt->bindParam(':login\_time', $login\_time);

            $stmt->execute();

            $\_SESSION['user\_loggedin'] = true;

            $\_SESSION['user\_id'] = $user\_id;

**Ovaj kod i slicni drugi kodovi su koristeni za Log in, Sign in i Create Doctor.**

**Rezervisanje Za tretman:**

    async function fetchData() {

        try {

            const response = await fetch('../getData.php');

            data = await response.json();

            return data

        } catch (error) {

            console.error('Error fetching doctor:', error);

        }

    }

    async function fetchUser() {

        try {

            const response1 = await fetch('../getUser.php');

            const user = await response1.text();

            return user;

        } catch (error) {

            console.error('Error fetching user:', error);

        }

    }

    fetchData().then(data => {

fetchUser().then(user => {

…

…

**Prvo uzmeno informaciju o zubarima I ulogovanom koriskiku.**

**Zubar je u obliku objekta dok korisnik je osnovna informacija koju smo sacuvali u $\_SESSION.**

**Doctor object:**  
class Doctor {

    public $id;

    public $fname;

    public $lname;

    public $email;

    public $img;

    public $msg;

    public $treatments;

    public $workHours;

    public function setId($id) {

        $this->id = $id;

    }

    public function getId() {

        return $this->id;

    }

    public function setFName($fname) {

        $this->fname = $fname;

    }

    public function getFName() {

        return $this->fname;

    }

    public function setLName($lname) {

        $this->lname = $lname;

    }

    public function getLName() {

        return $this->lname;

}

…

…

…

…

    public function setTreatments($treatments) {

        $this->treatments = $treatments;

    }

    public function getTreatments() {

        return $this->treatments;

    }

    public function setWorkHours($workHours) {

        $this->workHours = $workHours;

    }

    public function getWorkHours() {

        return $this->workHours;

    }

}

class Treatment {

    public $treatment;

    public $length;

    public function \_\_construct($treatment, $length) {

        $this->treatment = $treatment;

        $this->length = $length;

    }

}

class WorkDay {

    public $day;

    public $start\_time;

    public $end\_time;

    public function \_\_construct($day, $start\_time, $end\_time) {

        $this->day = $day;

        $this->start\_time = $start\_time;

        $this->end\_time = $end\_time;

**Kada pravimo Kalendar za Rezervacije proveravamo dali je selektovan Zubar I tretman I dali Zubar radi tog dana.**

for (let i = 1; i <= daysInMonth; i++) {

      const day = document.createElement("div");

      day.textContent = i;

 if (doctor.length == 1 && FilteredTreatmentsList.length == 1){

  var start = doctor[0].workHours[(firstDay + i -1 -1) % 7].start\_time;

    var end = doctor[0].workHours[(firstDay + i -1 -1) % 7].end\_time;

//first -1 is because i=1 and the second -1 is because the calendar starts from Sunday

      if (start != end) {

    var exists = doctor[0].treatments.some(function(t) {

          return t.treatment === FilteredTreatmentsList[0];

          });

// Create pop up --------------------- ------------------------//

        if(exists){

           day.classList.add('clickable');

           day.addEventListener("click", function () {

           modalFunction(i,firstDay);

        });

      }

    }

  }

  daysContainer.appendChild(day);

}

**Kada korisnik Izabere datum saljemo informaciju I sacuvamo je u bazu.**

**Profil za zubare:**

document.addEventListener('DOMContentLoaded', function() {

    async function fetchDoctorData() {

        try {

            const response = await fetch('../getDoctor.php');

            doctor = await response.json();

            return doctor

        } catch (error) {

            console.error('Error fetching doctor:', error);

        }

    }

var doctor = fetchDoctorData().then(() => {

…

…

…

**Vecina koda nije poseban samo uzima informaciju koju smo dobili koristici fech.**

**Takodje stranica ima edit dugme koje prikaze nova polja za popunavanje I dozvoljava zubaru da menja informaciju o njemu.**

    editBtn.addEventListener('click', function() {

    changeName.style.display = "block";

    changeLastName.style.display = "block";

    changePasswordConteiner.style.display = "block";

    editTreatments.style.display = "block";

    saveBtn.style.display = "block";

    cancleBtn.style.display ="block";

    editBtn.style.display ="none";

    treatmentsConteiner.querySelectorAll('tr').forEach((row)=>{

    var none = row.querySelector('.none');

    none.style.display ="block"

});

  document.querySelectorAll('.dayCard').forEach((card) => {

  const startSlider = card.querySelector('#StartSliderConteiner');

  const endSlider = card.querySelector('#EndSliderConteiner');

    startSlider.style.display="block";

    endSlider.style.display="block";

});

**Takodje save dugme se pojavi I kada se klikne sacuva bilo koje promene.**

saveBtn.addEventListener('click',function(){

   var formData = new FormData();

    formData.append('id',doctor.id);

    formData.append('changeName', changeName.value);

    formData.append('changeLastName', changeLastName.value);

…

…

…

   if (validateForm(formData)) {

        submitForm(formData);

      }

**Kada posaljemo informaciju na PHP stranu uzmemo ID zubara i sa nim ostatak informacije u bazi. Tu informaciju uporedjujemo sa promenjenom informacijom i cuvamo novu informaciju.**

session\_start();

$id = $\_POST['id'] ?? '';

try {

    $stmt = $pdo->prepare("SELECT fname, lname, password FROM doctors WHERE id = :id");

    $stmt->bindParam(":id",$id);

    $stmt->execute();

    $result = $stmt->fetch(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

    echo json\_encode(['success' => false, 'message' => 'Database error: ' . $e->getMessage()]);

}

$fname = $\_POST['changeName'] ?? '';

if (empty($fname)) {

    $fname = $result['fname'];

}

$lname = $\_POST['changeLastName'] ?? '';

if (empty($lname)) {

    $lname = $result['lname'];

}...

…

…

    $stmt = $pdo->prepare("UPDATE doctors

           SET fname = :fname, lname =:lname, password = :password

           WHERE id = :id;");

    $stmt->bindParam(':fname', $fname);

    $stmt->bindParam(':lname', $lname);

    $stmt->bindParam(':password', $password);

    $stmt->bindParam(':id', $id);

    $stmt->execute();

**Za prikaz rezervacija fecujemo informaciju I lupujemo kroz nju za svaku rezervaciju I napravimo poseban element.**

fetchData().then(data => {

        console.log(data);

        data.forEach(apointment => {

                // Create the outer modal div

                const modalDiv = document.createElement('div');

                modalDiv.id = 'modal';

                modalDiv.className = '';

                // Create the modal-content div

                const modalContentDiv = document.createElement('div');

                modalContentDiv.className = 'modal-content';

                // Create the inner content div

                const innerDiv = document.createElement('div');

                innerDiv.className = '';

                // Create and append the paragraphs

                const dateP = document.createElement('p');

                dateP.id = 'modal-date';

                dateP.textContent = `Apoitment at: ${apointment.Apointment\_date}`;

                innerDiv.appendChild(dateP);

                var [hours, minutes, seconds] = apointment.Apointment\_time.split(":");

                // Remove leading zero from hours

                if (hours.startsWith("0")) {

                    hours = hours.slice(1);

                }

                time =`${hours}:${minutes}`;

                const timeP = document.createElement('p');

                timeP.id = 'modal-time';

                timeP.textContent = `Time of Apoitment: ${time}`;

                innerDiv.appendChild(timeP);

                const doctorP = document.createElement('p');

                doctorP.id = 'modal-doctor';

                doctorP.textContent = `Dentis e-mail: ${apointment.doctor\_email}`;

                innerDiv.appendChild(doctorP);

                const patientP = document.createElement('p');

                patientP.id = 'modal-patiant';

                patientP.textContent = `Patiant e-mail: ${apointment.user\_email}`;

                innerDiv.appendChild(patientP);

                const treatmentP = document.createElement('p');

                treatmentP.id = 'modal-treatmant';

                treatmentP.textContent = `for:  ${apointment.treatments}`;

                innerDiv.appendChild(treatmentP);

                // Append the inner div to modal-content div

                modalContentDiv.appendChild(innerDiv);

                // Append the modal-content div to modal div

                modalDiv.appendChild(modalContentDiv);

                // Append the modal div to the body or any specific element

                document.body.appendChild(modalDiv);

        });

});

**Na Admin strani uzimamo odredjenu informaciju I prikazujemo je u formi tabele.**

fetch('../../Backend/PHP/db\_Select\_Users.php')

        .then(response => response.json())

        .then(data => {

            const usersTableBody = document.querySelector('#Users\_table tbody');

            data.forEach(user => {

                const row = document.createElement('tr');

                // Create a cell for each data field

                row.innerHTML = `

                    <td>${user.id}</td>

                    <td>${user.fname}</td>

                    <td>${user.lname}</td>

                    <td>${user.email}</td>

                    <td>${user.phone}</td>

                    <td>${user.gender}</td>

                    <td>${user.birthDate}</td>

                    <td>

                        <button class="btn btn-danger btn-sm" onclick="deleteUser(${user.id}, this)">Delete</button>

                    </td>

                `;

                // Append the row to the table body

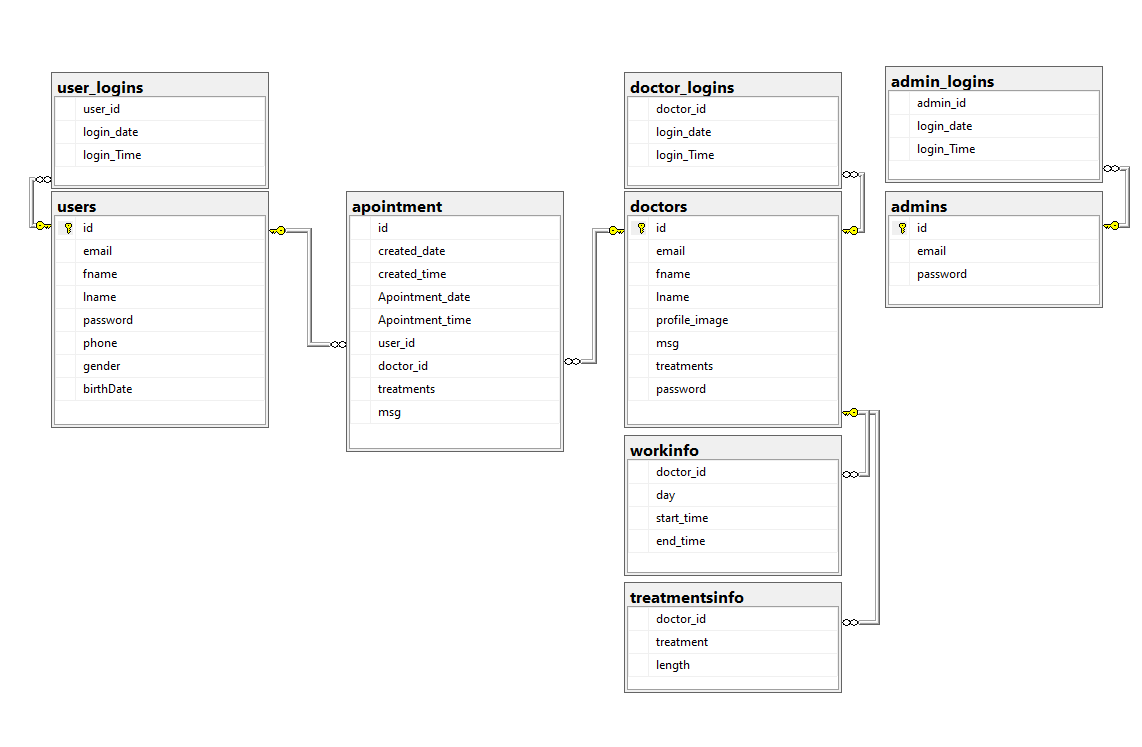
                usersTableBody.appendChild(row);

            });

        })

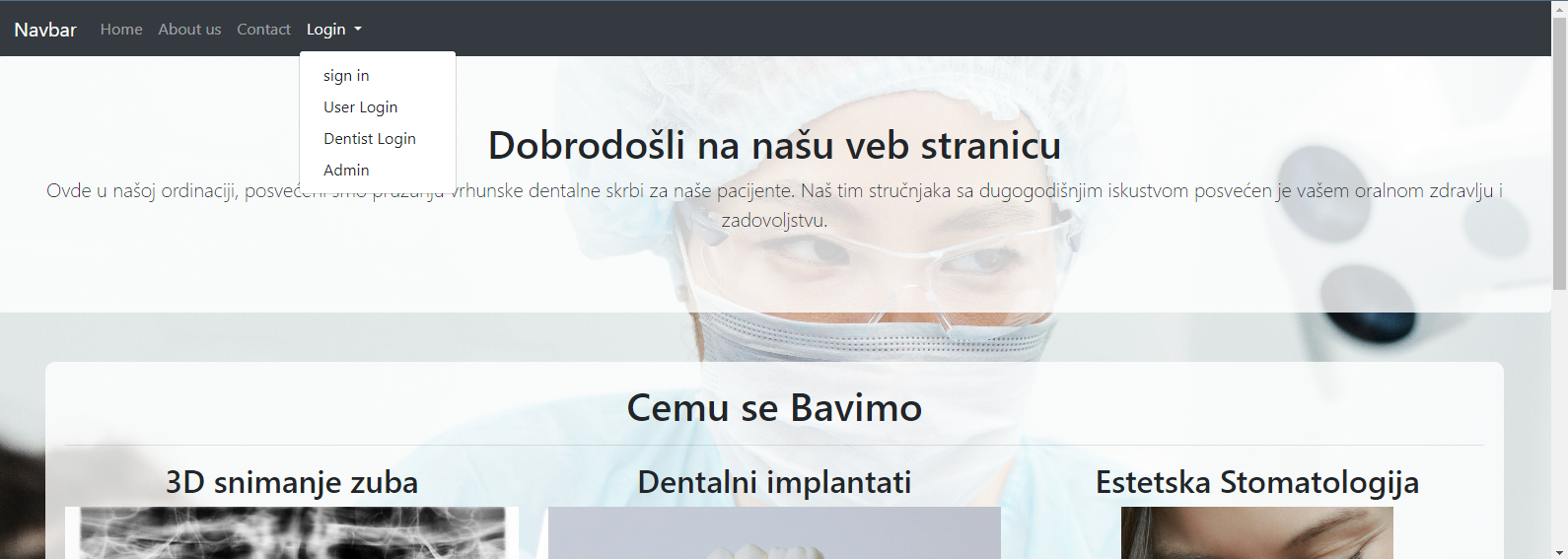
    .catch(error => console.error('Error fetching users:', error));

# Struktura baze podataka (MySQL Workbench)

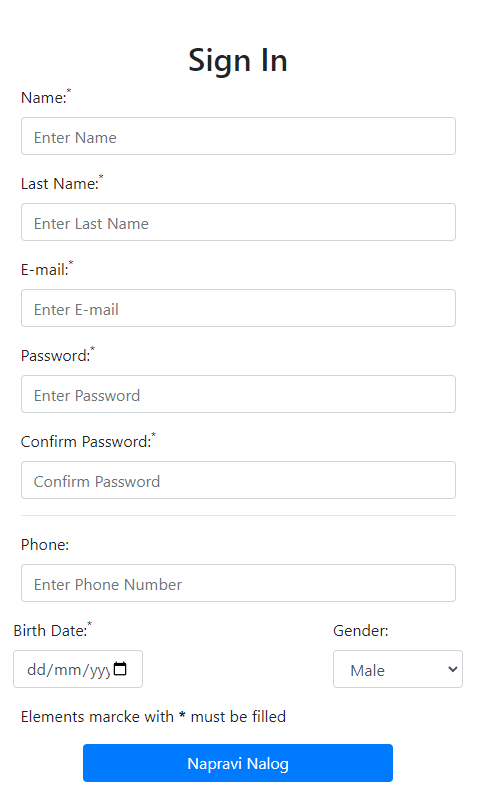


# Opis funkcionalnosti

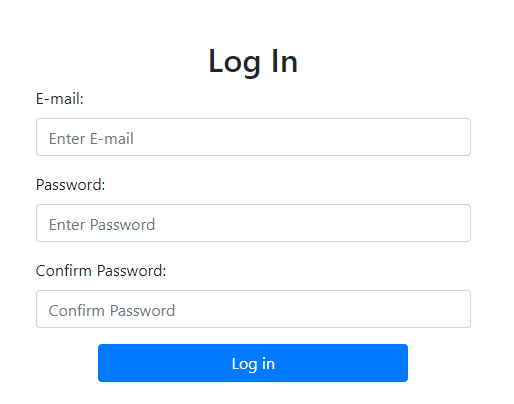
U ovom delu obavezno opisati način funkcionalnosti projekta. Koristiti slike (*screenshots*).



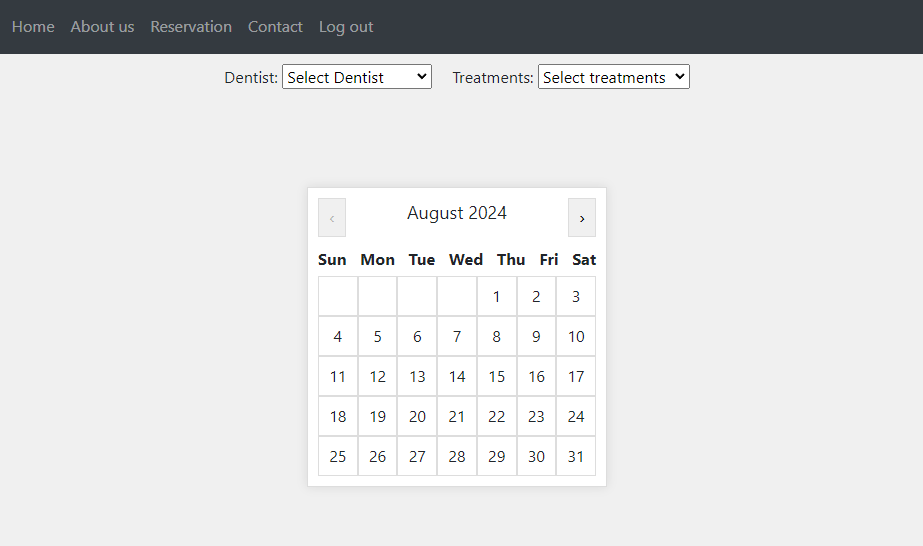
Pocetna strana za goste ima opcije za login



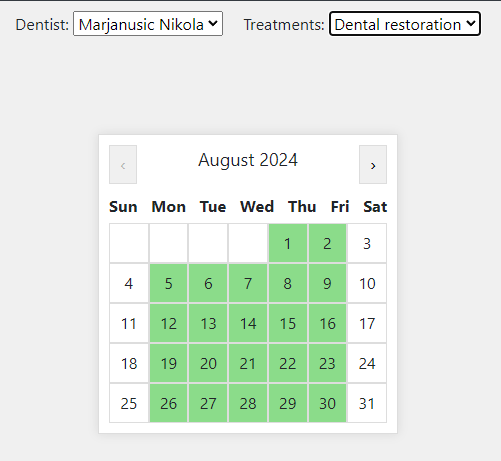
Ako korisnik nema akaunt moze da se prijavi.Ako je prijava tacna automacki vodi usera na log in stranicu



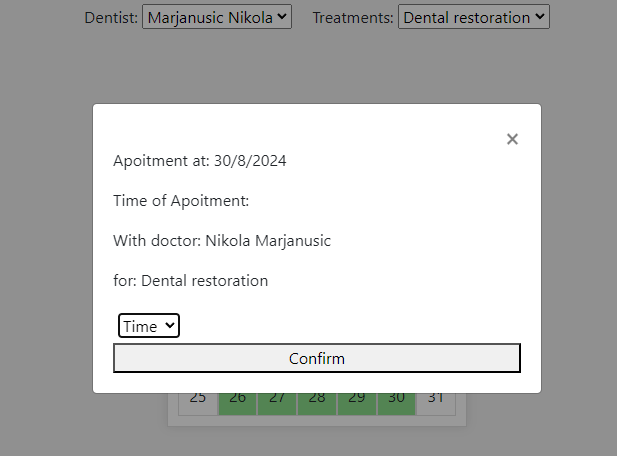
Korisnik se ovde moze ulogovati na akaunt I dobiti pristup rezervacijama



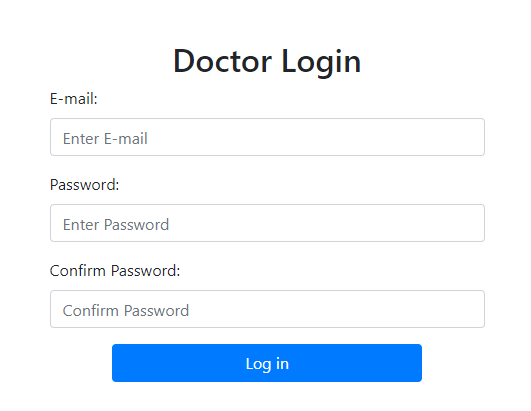
Korisnikov navbar se menja I sada ima novu opciju “Reservation” za rezervacije takodje Log in dugme je zamenjeno sa Log out.



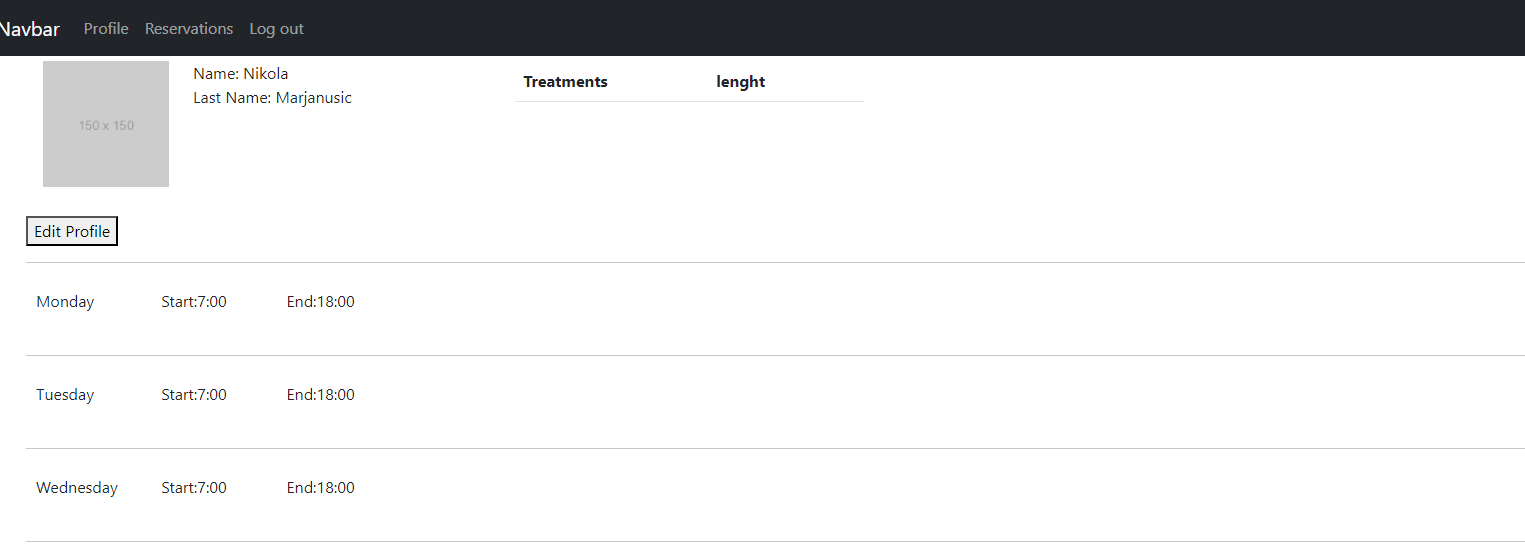
Kada korisnik selektuje Zubara i vrstu Tretmana dani na kojima zubar radi se promene u zelenu boju i dobijaju mogucnost da se kliknu



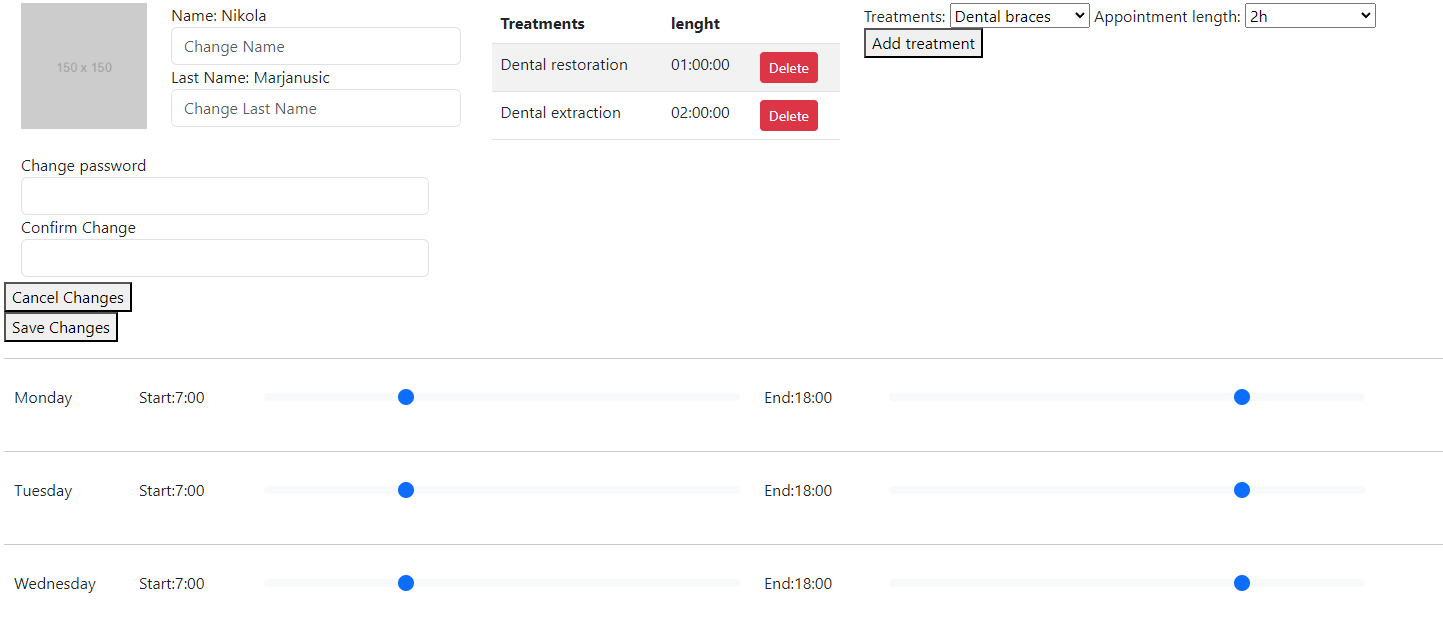
Kada se dan klikne izpaci se informacija o rezervaciji I korisnik moze da bira vreme I potvrdi rezervaciju.



Logovanje za zubara je slicko kao logovanje za korisnika I kada se potvrsi automacki vodi korisnika na profilnu stranu



Na profilnoj strani ispisana je informacija o zubaru i kada se Edit Profile dugme kilkne Zubar moze da menja informaciju



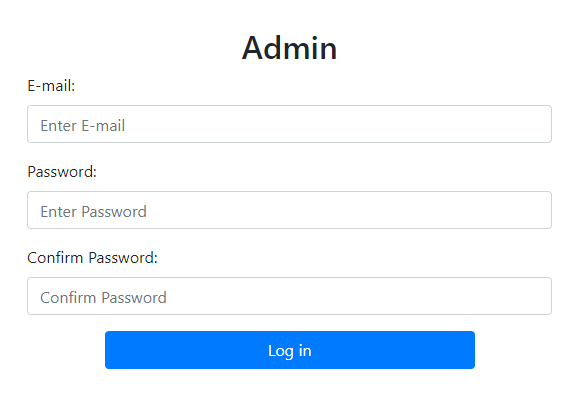
Ovde zubar moze da menja kojim danima radi kada pocinje da radi koje tretmane radi I koliko dugo mu traju. Takodnje moze da menja ime I šifru.

Kada se kilikne Cancel dugme stranica se vraca na prvobitni izgled bez ikakvih promena.

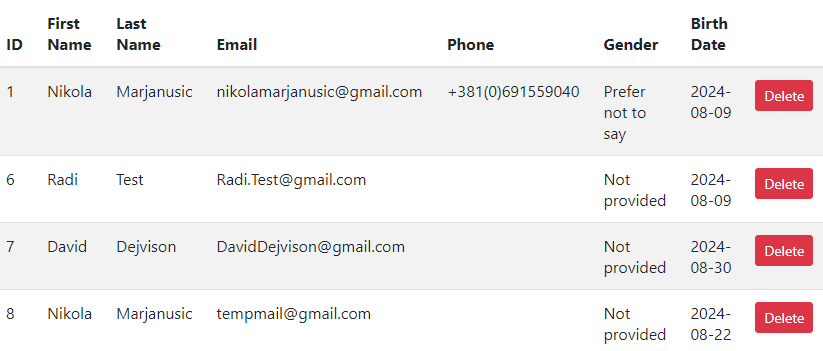
Dok Save dugme sacuva promena.



Takodje u Navbaru zubar ima opciju reyervations koja prikazuje sve rezervacije rezervisane za njega.



Admin login je isti kao i prosli sem što vodi na databse view





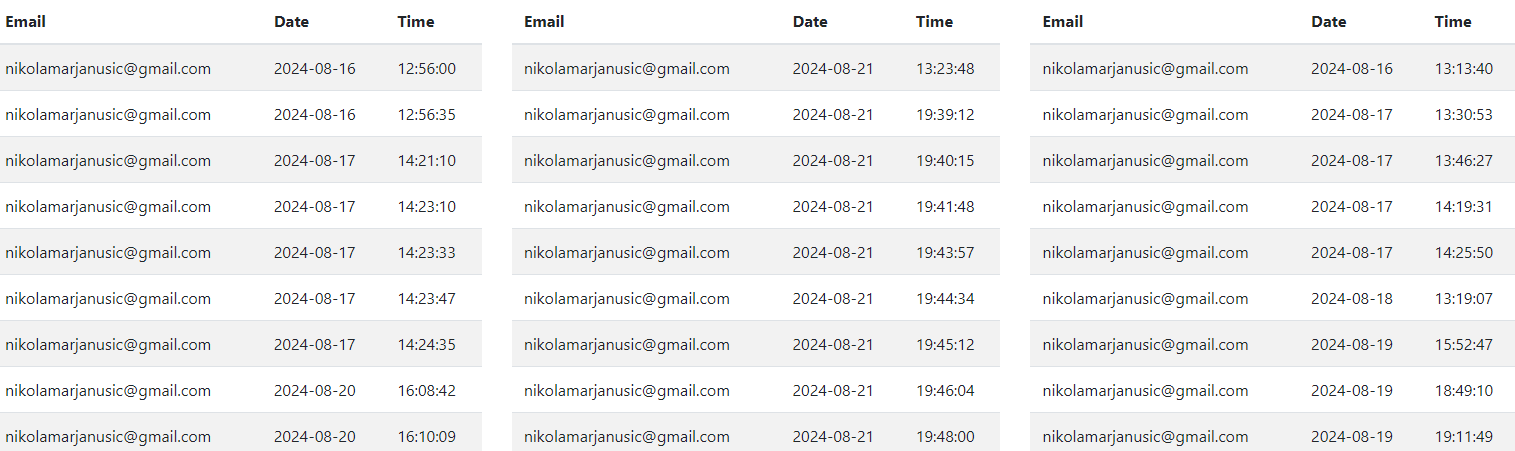
databse view prikazuje dve tabele prva je za korisnike a druga je za Zubare.



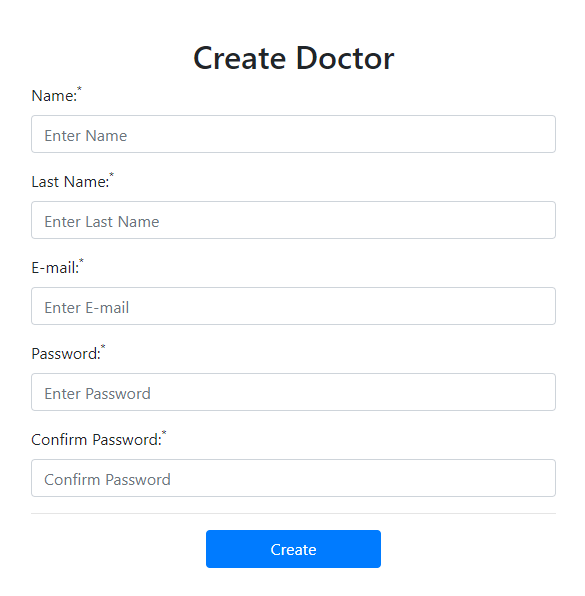
Takodje u navbaru ima jos opcija.



View Apointments prikazuje sve trenutne apointmentove za svakog Zubara/Pacijenta



View Login prikazuje ko se kada prijavio za Korisnike,Zubare I Admine



Stranica Create doctor dozvoljava Administratoru da napravi nove doktore

# Korišćena literatura

W1. <http://www.w3schools.com/>

W2. <http://www.jquery.com>