Univerzitet u Novom Sadu

Tehnički fakultet “Mihajlo Pupin”

Zrenjanin

Seminarski rad

Veb programiranje

Veb aplikacija za recepte – „Mirkova Kujna“

Profesor: Dr. Ljubica Kazi Student: Mirko Popović

Asistent: MSc Vuk Amižić Smer: SI 21/21

Zrenjanin 2024.

Sadržaj

[1. OPIS POSLOVNOG KONTEKSTA PRIMENE APLIKACIJE 3](#_Toc177397161)

[1.1 *Cilj izrade aplikacije* 3](#_Toc177397162)

[1.2 *Alati korišćeni pri radu* 3](#_Toc177397163)

[2. KORISNIČKO UPUTSTVO 5](#_Toc177397164)

[3. SQL SKRIPT KREIRANJA RELACIONE BAZE PODATAKA 13](#_Toc177397165)

[4. KLJUČNI DELOVI PROGRAMSKOG KODA 21](#_Toc177397166)

[4.1 *Obavezni deo* 21](#_Toc177397167)

[4.1.1 Login 21](#_Toc177397168)

[4.1.2 Unos 26](#_Toc177397169)

[4.1.3 Brisanje 29](#_Toc177397170)

[4.1.4 Izmena 31](#_Toc177397171)

[4.1.5 Štampa 32](#_Toc177397172)

[4.2 *Opcioni deo* 37](#_Toc177397173)

[4.2.1 Parametarska štampa 37](#_Toc177397174)

[5. ZAKLJUČAK 41](#_Toc177397175)

[6. LITERATURA 42](#_Toc177397176)

# OPIS POSLOVNOG KONTEKSTA PRIMENE APLIKACIJE

## 1.1 *Cilj izrade aplikacije*

Aplikacija “Mirkova Kujna” predstavlja reiteraciju originalnog veb sajta, sa dodatim novim funkcionalnostima. Cilj sajta je da se kreira platforma na kojoj korisnici mogu kreirati naloge kako bi dalje mogli da dele svoje recepte koji su dostupni svima, korisnici takođe mogu da lajkuju tuđe recepte koji se prikazuju na dinamičnoj rang listi.

## 1.2 *Alati korišćeni pri radu*

Aplikacija je rađena u programskim jezicima PHP, JavaScript, CSS i HTML, sav naveden kod se kucao u okruženju „Visual Studio Code“. Za potrebe baze podataka i lokalnog hostinga sajta korišćen je „MAMP“ (Aplikacija za SQL i ApacheServer na MacOS platformi).

A white elephant in a circle

Description automatically generated

Slika 1 - MAMP

A screenshot of a computer

Description automatically generated

Slika 2 – Pokrenut MAMP

# KORISNIČKO UPUTSTVO

Početna stranica nam upućuje dobrodošlicu na sajt, I sadržaj je generalno dekorativnog karaktera

A black and white text

Description automatically generated

Slika 3 – Početni banner



Slika 4 – Slajder sa popularnim receptima

A screenshot of a website

Description automatically generated

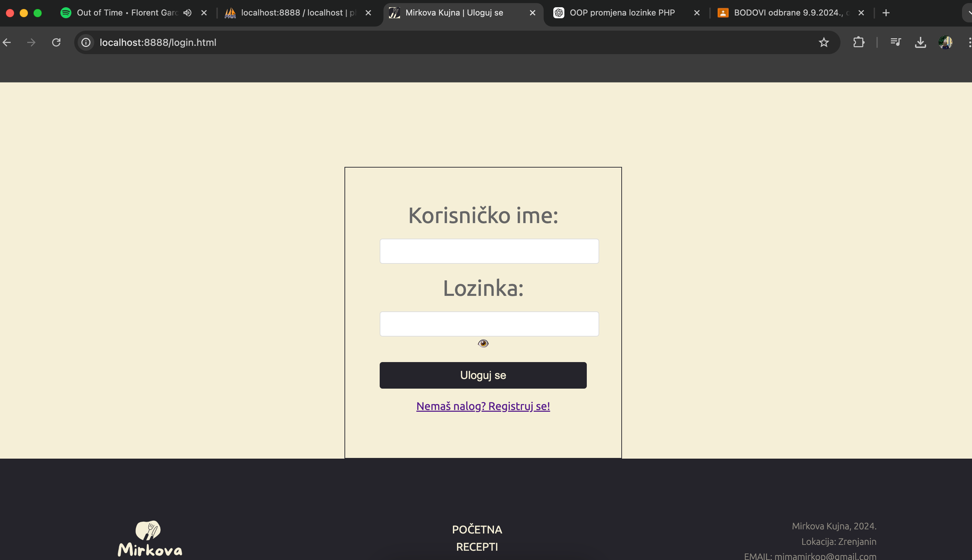
Slika 5 – Dno početne stranice, galerija i futer

Kada korisnik malo počne da lista ka dole na vrhu stranice će se pojaviti navigacioni meni, takođe pojedinim stranicama može se pristupiti I u futeru.



Slika 6 – Navigacioni meni

Prva stranica koju ćemo posetiti u ovom korisničkom uputstvu je stranica za prijavu korisnika:



Slika 7 – Prijava korisnika

Budući da zasad još nemamo korisnički nalog, klikom na link za registraciju odlazimo na drugu stranicu:

A screenshot of a login form

Description automatically generated

Slika 8 – Registracija korisnika

Pri registraciji biramo korisničko ime I lozinku, sada recimo da želimo da uzmemo korisničko ime Mirko, a ono je pritom zauzeto dobićemo sledeći ekran:

A computer screen with a white background

Description automatically generated

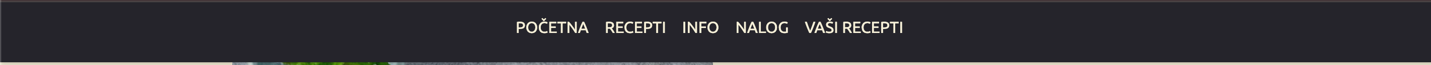
Slika 9 – Zauzeto korisničko ime

A black and white text

Description automatically generated

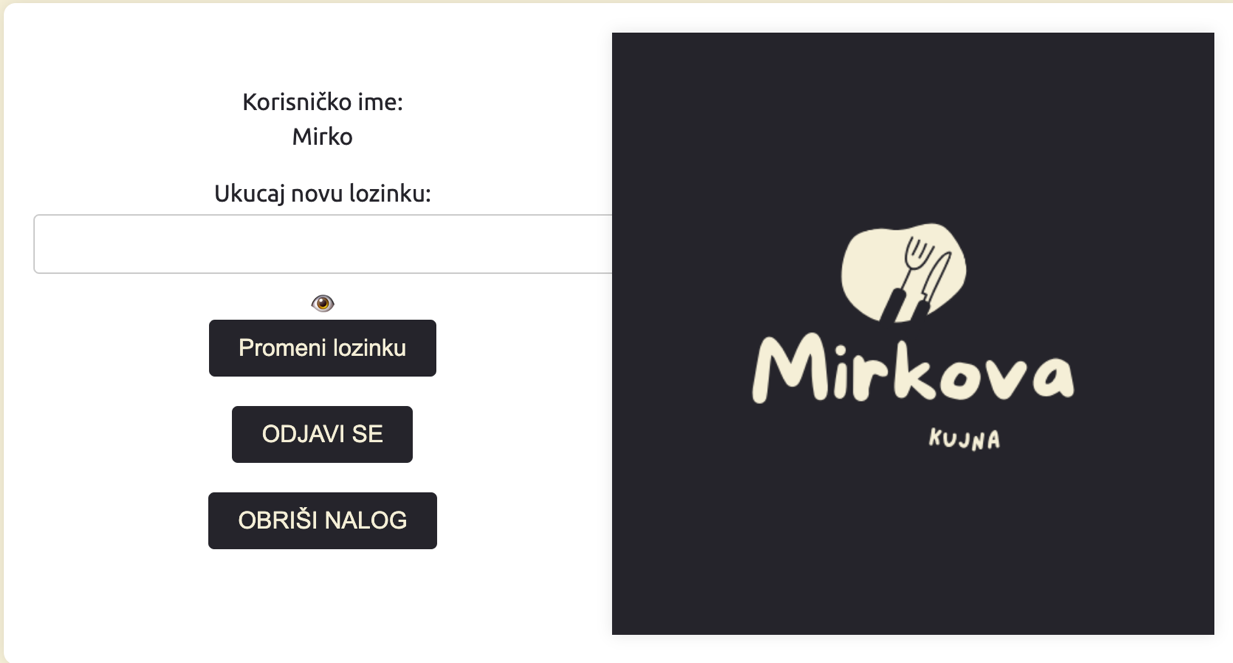
Slika 10 – Druga verzija sajta

Budući da smo mi već registrovali nalog, registracija nas redirektuje na početnu stranicu koja prati sesiju (Ukoliko korisnik nije ulogovan sesija se gasi I vrši se redirekcija na stranice za neregistrovanog korisnika)



Slika 11 – Druga verzija navigacionog menija

Sada navigacioni meni pruža korisniku dve nove stranice, prva stranica je stranica za naš nalog:



Slika 12 – Stranica za nalog

Stranica za nalog nam pruža opcije menjanja lozinke, odjave (koja nas vraća na originalnu verziju sajta) I opciju brisanja naloga. Lozinka se menja kucanjem nove lozinke I klikom na dugme za promenu:

A screenshot of a phone

Description automatically generated

Slika 13 – Podaci korisnika iz baze

A screenshot of a menu

Description automatically generated

Slika 14 – Menjanje lozinke

A black and white screen

Description automatically generated

Slika 15 – Uspešno promenjena lozinka

A screenshot of a phone

Description automatically generated

Slika 16 – Promenjena lozinka u bazi

Sada da pređemo na brisanje naloga:

A screenshot of a computer

Description automatically generated

Slika 17 – Brisanje naloga

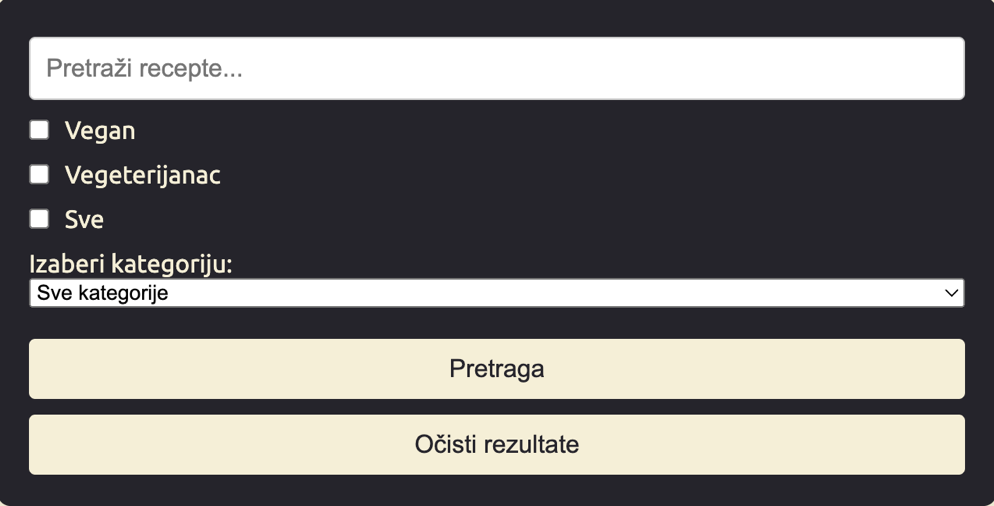
Klikom na dugme “Obriši nalog” dobijamo ekran gore. Kada potvrdimo naš nalog se briše konačno iz baze podataka. Dugme za odjavu korisnika funkcioniše tako što prekida sesiju I vrši redirekciju na početnu stranicu za neregistrovane korisnike. Registrovanim korisnicima je takođe dostupna opcija kreiranja sopstvenih recepata, te njihove objave na sajtu.

A screenshot of a computer

Description automatically generated

Slika 18 – Kreator recepata

Gore je slika koja predstavlja formu za lak način postavljanja recepata na sajt. Ovi recepti su zatim dostupni svim korisnicima (Čak I neprijavljenim) putem treće funkcionalnosti. Pretraga recepata:



Slika 19 – Brza pretraga recepata

Na stranici recepti korisnici mogu pretraživati recepte po želji, mogu odabrati dijetalnu restrikciju I pisati ključne reči kako bi suzili pretragu. Klikom na “Pretraga” bez ikakvih selekcija će ispisati sve recepte koji se nalaze u bazi.



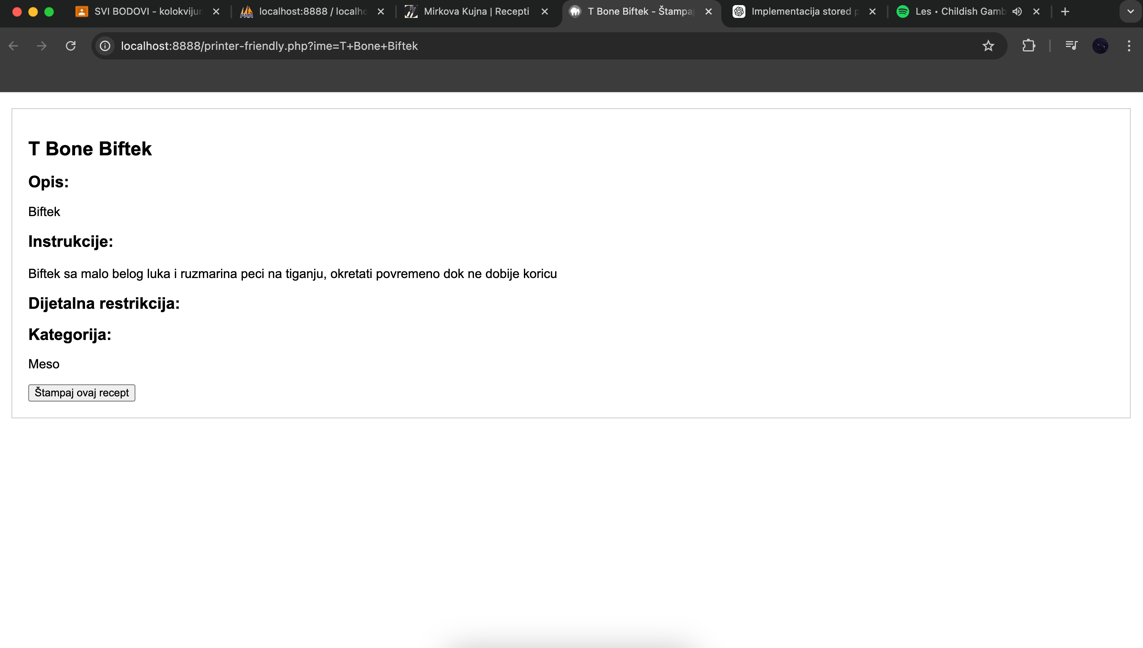
Slika 20 – Tabelarni prikaz svih recepata

Možemo izfiltrirati specifičan rezultat tako što smo odabrali kategoriju, dijetalnu restrikciju ili jednostavno ukucali ključnu reč koja se nalazi u imenu recepta.



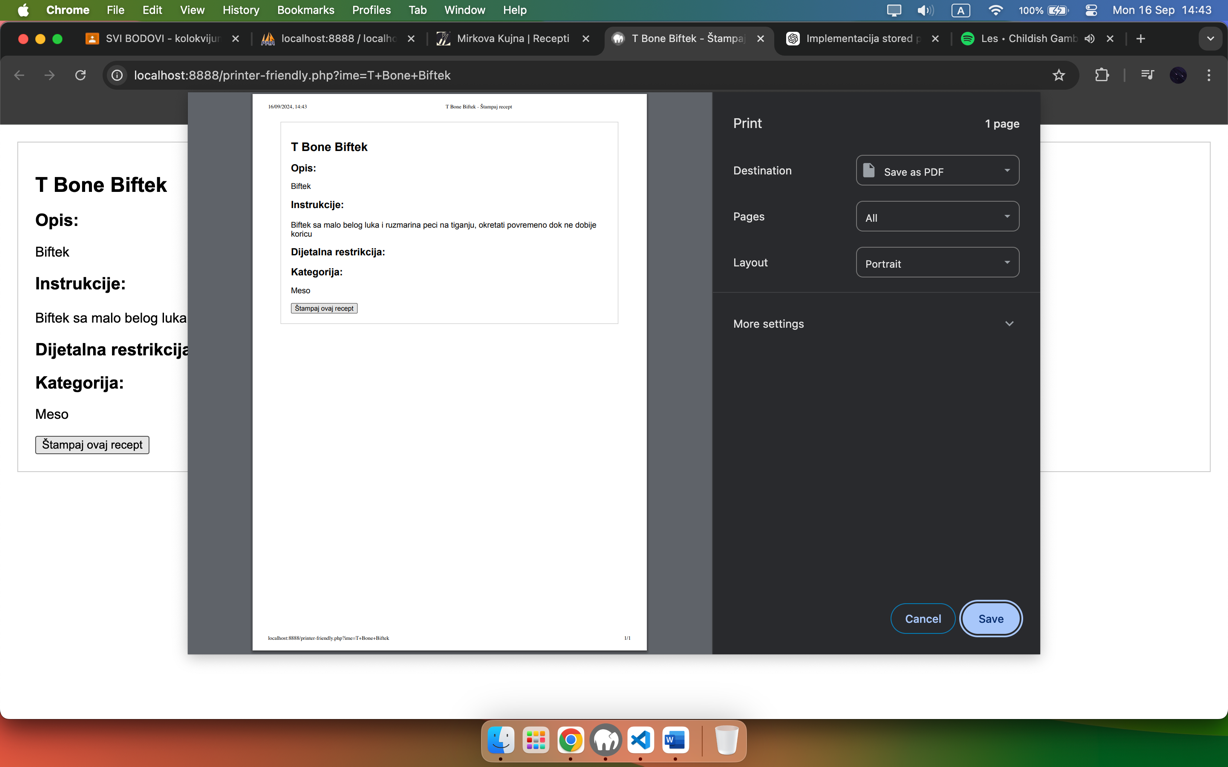
Slika 21 – Filter pretraga, tabelarni prikaz recepta

Novo-otvorena tabela pruža nam recept ukratko, kada neko poželi da nabrzaka pogleda kako se nešto pravi. Recept se takođe može štampati kako u vidu “printer-friendly” stranice:



Slika 22 – Printer-friendly stranica

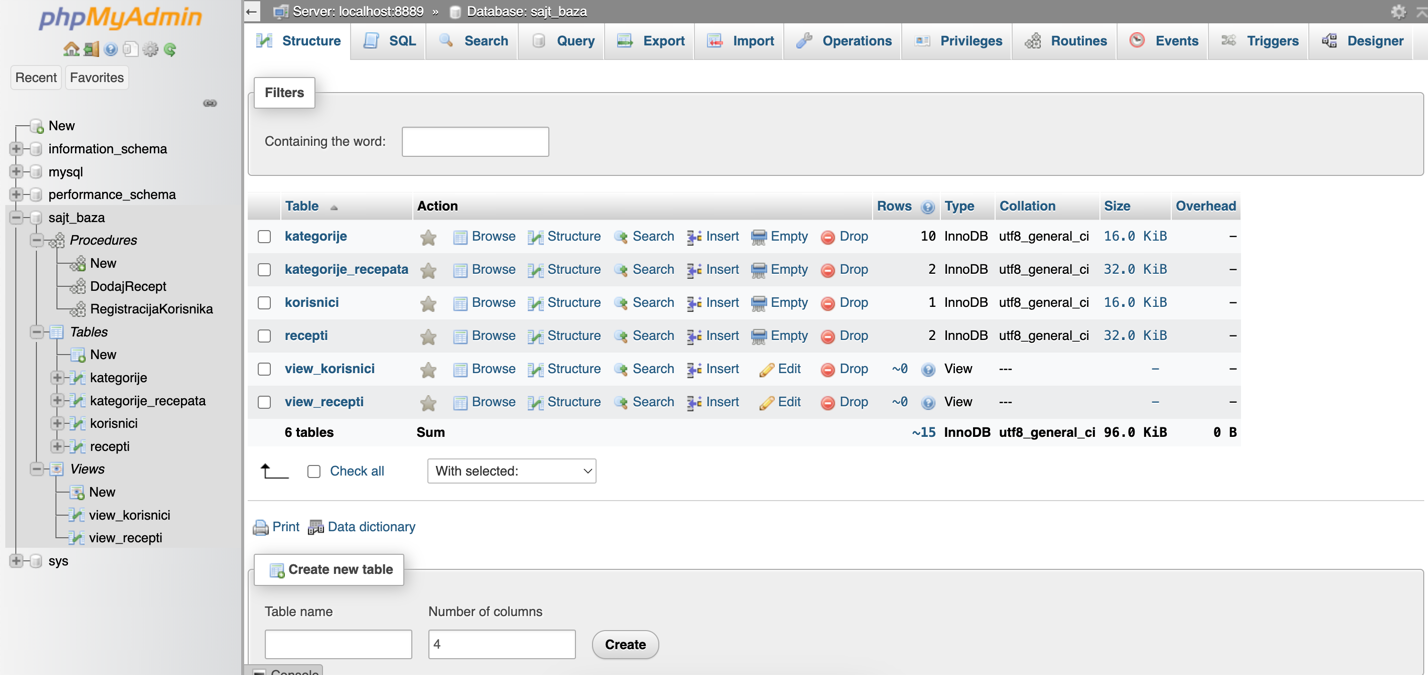
Tako se može i preuzeti tabela za kasniju fizičku štampu:



Slika 23 – PDF štampa

# SQL SKRIPT KREIRANJA RELACIONE BAZE PODATAKA

Nakon što smo pokrenuli veb server putem MAMP-a možemo otići na WebStart –> Tools -> phpMyAdmin



Slika 24 – phpMyAdmin

Baza I tabele su pravljene upotrebom SQL kodova ali I ugrađenog sistema za kreiranje tabela, nakon što je sajt završen u potpunosti kao I naša baza podataka klikom na “Export” smo preuzeli SQL skript za našu bazu I tabele:

|  |
| --- |
| -- phpMyAdmin SQL Dump  -- version 5.2.0  -- https://www.phpmyadmin.net/  --  -- Host: localhost:8889  -- Generation Time: Sep 17, 2024 at 12:53 PM  -- Server version: 5.7.39  -- PHP Version: 7.4.33  SET  SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";  START TRANSACTION;  SET  time\_zone = "+00:00";  --  -- Database: `sajt\_baza`  --  DELIMITER $$ --  -- Procedures  --  CREATE DEFINER = `root` @`localhost` PROCEDURE `DodajRecept` (  IN `p\_ime` VARCHAR(255),  IN `p\_opis` TEXT,  IN `p\_instrukcije` TEXT,  IN `p\_dijeta` VARCHAR(255),  IN `p\_napravio` VARCHAR(255),  IN `p\_kategorija` VARCHAR(255)  ) BEGIN DECLARE v\_recept\_id INT;  DECLARE v\_kategorija\_id INT;  -- Ubacivanje u recepti tabelu sa kategorijom  INSERT INTO recepti (  ime, opis, instrukcije, dijeta, napravio,  kategorija  )  VALUES  (  p\_ime, p\_opis, p\_instrukcije, p\_dijeta,  p\_napravio, p\_kategorija  );  -- Dodano p\_kategorija  -- Uzima id od recepta  SET  v\_recept\_id = LAST\_INSERT\_ID();  -- Poklapanje id kategorije  SELECT  id INTO v\_kategorija\_id  FROM  kategorije  WHERE  naziv = p\_kategorija;  -- Ako se poklope ID onda ubacuje recept i u drugu tabelu  IF v\_kategorija\_id IS NOT NULL THEN INSERT INTO kategorije\_recepata (recept\_id, kategorija\_id)  VALUES  (v\_recept\_id, v\_kategorija\_id);  END IF;  END$$ CREATE DEFINER = `root` @`localhost` PROCEDURE `RegistracijaKorisnika` (  IN `korisnickoIme` VARCHAR(255),  IN `lozinka` VARCHAR(255)  ) BEGIN DECLARE korisnikPostoji INT;  -- Proveri da li korisničko ime već postoji  SELECT  COUNT(\*) INTO korisnikPostoji  FROM  korisnici  WHERE  korisnicko\_ime = korisnickoIme;  IF korisnikPostoji > 0 THEN SIGNAL SQLSTATE '45000'  SET  MESSAGE\_TEXT = 'Korisničko ime je zauzeto.';  ELSE INSERT INTO korisnici (korisnicko\_ime, lozinka)  VALUES  (korisnickoIme, lozinka);  END IF;  END$$ DELIMITER;  -- --------------------------------------------------------  --  -- Table structure for table `kategorije`  --  CREATE TABLE `kategorije` (  `id` int(11) NOT NULL,  `naziv` varchar(255) NOT NULL  ) ENGINE = InnoDB DEFAULT CHARSET = utf8;  --  -- Dumping data for table `kategorije`  --  INSERT INTO `kategorije` (`id`, `naziv`)  VALUES  (1, 'Meso'),  (2, 'Dezerti'),  (3, 'Pasta'),  (4, 'Testo'),  (5, 'Piće'),  (6, 'Riba'),  (7, 'Salate'),  (8, 'Grickalice'),  (9, 'Sosevi'),  (10, 'Kuvano');  -- --------------------------------------------------------  --  -- Table structure for table `kategorije\_recepata`  --  CREATE TABLE `kategorije\_recepata` (  `recept\_id` int(11) NOT NULL,  `kategorija\_id` int(11) NOT NULL  ) ENGINE = InnoDB DEFAULT CHARSET = utf8;  --  -- Dumping data for table `kategorije\_recepata`  --  INSERT INTO `kategorije\_recepata` (`recept\_id`, `kategorija\_id`)  VALUES  (14, 1),  (13, 3),  (7, 10);  -- --------------------------------------------------------  --  -- Table structure for table `korisnici`  --  CREATE TABLE `korisnici` (  `id` int(11) NOT NULL,  `korisnicko\_ime` varchar(50) NOT NULL,  `lozinka` varchar(255) NOT NULL  ) ENGINE = InnoDB DEFAULT CHARSET = utf8;  --  -- Dumping data for table `korisnici`  --  INSERT INTO `korisnici` (  `id`, `korisnicko\_ime`, `lozinka`  )  VALUES  (9, 'Mirko', '12345'),  (  10, 'TestZaNasledjivanje', 'registracija'  );  -- --------------------------------------------------------  --  -- Table structure for table `recepti`  --  CREATE TABLE `recepti` (  `id` int(11) NOT NULL,  `ime` varchar(255) NOT NULL,  `opis` text,  `kategorija` varchar(50) DEFAULT NULL,  `dijeta` varchar(50) DEFAULT NULL,  `instrukcije` text,  `napravio` varchar(50) NOT NULL  ) ENGINE = InnoDB DEFAULT CHARSET = utf8;  --  -- Dumping data for table `recepti`  --  INSERT INTO `recepti` (  `id`, `ime`, `opis`, `kategorija`,  `dijeta`, `instrukcije`, `napravio`  )  VALUES  (  7, 'Natašina čorba od krompira i jaja',  'Lagana čorbica', 'Kuvano', 'Vegeterijanac',  'Skuvati jaja u zasebnoj vodi, oguliti. U drugu vodu skuvati krompir, vratiti jaja, sipati vegetu i pasirani paradajz, dokuvati još malko',  'Mirko'  ),  (  13, 'Test za kategoriju', '123124',  'Pasta', 'Vegeterijanac', '21512351',  'Mirko'  ),  (  14, 'T Bone Biftek', 'Biftek', 'Meso',  '', 'Biftek sa malo belog luka i ruzmarina peci na tiganju, okretati povremeno dok ne dobije koricu',  'Mirko'  );  -- --------------------------------------------------------  --  -- Stand-in structure for view `view\_korisnici`  -- (See below for the actual view)  --  CREATE TABLE `view\_korisnici` (  `id` int(11),  `korisnicko\_ime` varchar(50),  `lozinka` varchar(255)  );  -- --------------------------------------------------------  --  -- Stand-in structure for view `view\_recepti`  -- (See below for the actual view)  --  CREATE TABLE `view\_recepti` (  `id` int(11),  `ime` varchar(255),  `opis` text,  `kategorija` varchar(50),  `dijeta` varchar(50),  `instrukcije` text,  `napravio` varchar(50)  );  -- --------------------------------------------------------  --  -- Structure for view `view\_korisnici`  --  DROP  TABLE IF EXISTS `view\_korisnici`;  CREATE ALGORITHM = UNDEFINED DEFINER = `root` @`localhost` SQL SECURITY DEFINER VIEW `view\_korisnici` AS  SELECT  `korisnici`.`id` AS `id`,  `korisnici`.`korisnicko\_ime` AS `korisnicko\_ime`,  `korisnici`.`lozinka` AS `lozinka`  FROM  `korisnici``korisnici`;  -- --------------------------------------------------------  --  -- Structure for view `view\_recepti`  --  DROP  TABLE IF EXISTS `view\_recepti`;  CREATE ALGORITHM = UNDEFINED DEFINER = `root` @`localhost` SQL SECURITY DEFINER VIEW `view\_recepti` AS  SELECT  `recepti`.`id` AS `id`,  `recepti`.`ime` AS `ime`,  `recepti`.`opis` AS `opis`,  `recepti`.`kategorija` AS `kategorija`,  `recepti`.`dijeta` AS `dijeta`,  `recepti`.`instrukcije` AS `instrukcije`,  `recepti`.`napravio` AS `napravio`  FROM  `recepti``recepti`;  --  -- Indexes for dumped tables  --  --  -- Indexes for table `kategorije`  --  ALTER TABLE  `kategorije`  ADD  PRIMARY KEY (`id`);  --  -- Indexes for table `kategorije\_recepata`  --  ALTER TABLE  `kategorije\_recepata`  ADD  PRIMARY KEY (`recept\_id`, `kategorija\_id`),  ADD  KEY `kategorija\_id` (`kategorija\_id`);  --  -- Indexes for table `korisnici`  --  ALTER TABLE  `korisnici`  ADD  PRIMARY KEY (`id`);  --  -- Indexes for table `recepti`  --  ALTER TABLE  `recepti`  ADD  PRIMARY KEY (`id`),  ADD  UNIQUE KEY `unique\_ime` (`ime`);  --  -- AUTO\_INCREMENT for dumped tables  --  --  -- AUTO\_INCREMENT for table `kategorije`  --  ALTER TABLE  `kategorije` MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT,  AUTO\_INCREMENT = 11;  --  -- AUTO\_INCREMENT for table `korisnici`  --  ALTER TABLE  `korisnici` MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT,  AUTO\_INCREMENT = 11;  --  -- AUTO\_INCREMENT for table `recepti`  --  ALTER TABLE  `recepti` MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT,  AUTO\_INCREMENT = 15;  --  -- Constraints for dumped tables  --  --  -- Constraints for table `kategorije\_recepata`  --  ALTER TABLE  `kategorije\_recepata`  ADD  CONSTRAINT `kategorije\_recepata\_ibfk\_1` FOREIGN KEY (`recept\_id`) REFERENCES `recepti` (`id`) ON DELETE CASCADE,  ADD  CONSTRAINT `kategorije\_recepata\_ibfk\_2` FOREIGN KEY (`kategorija\_id`) REFERENCES `kategorije` (`id`) ON DELETE CASCADE;  COMMIT; |

Gore je navedena cela SQL skripta preuzeta sa phpMyAdmin platforme.

# KLJUČNI DELOVI PROGRAMSKOG KODA

## 4.1 *Obavezni deo*

### 4.1.1 Login

Zarad potrebe ovog seminarskog u segmentima za kod navodiću samo delove koji su bitni za rad samih funkcionalnosti. Prva na redu je funkcionalnost za Log-In:

|  |
| --- |
| <div class="gazda-kontejner">  <div class="forma-kontejner">  <form action="login.php" method="POST">  <div class="forma-grupa">  <label for="username">Korisničko ime:</label>  <input type="text" id="username" name="username" required>  </div>  <div class="forma-grupa lozinka-kontejner">  <label for="password">Lozinka:</label>  <input type="password" id="password" name="password" required>  <span class="toggle-lozinka" onclick="toggleLozinku('password')">👁️</span>  </div>  <div class="forma-grupa">  <input type="submit" value="Uloguj se" class="promeni-dugme">  </div>  </form>  <div class="forma-grupa">  <a href="kreiraj-nalog.html">Nemaš nalog? Registruj se!</a>  </div>  </div>  </div> |

Navedena gore je forma za prijavu korisnika, ista struktura je i kod forme za registraciju. Kada korisnik klikne dugme „Uloguj se“ pokreće se kod „login.php“:

|  |
| --- |
| <?php  session\_start();  $servername = "localhost";  $username = "root";  $password = "root";  $dbname = "sajt\_baza";  class BazaPodataka  {  protected $konekcija;  public function \_\_construct($servername, $username, $password, $dbname)  {  $this->konekcija = new mysqli(  $servername,  $username,  $password,  $dbname  );  if ($this->konekcija->connect\_error) {  die(  "Greška prilikom povezivanja: " .  $this->konekcija->connect\_error  );  }  }  public function dohvatiKonekciju()  {  return $this->konekcija;  }  public function zatvoriKonekciju()  {  $this->konekcija->close();  }  }  class Autentifikator extends BazaPodataka  {  private $korisnickoIme;  private $lozinka;  public function \_\_construct(  $servername,  $username,  $password,  $dbname,  $korisnickoIme,  $lozinka  ) {  parent::\_\_construct($servername, $username, $password, $dbname);  $this->korisnickoIme = $korisnickoIme;  $this->lozinka = $lozinka;  }  public function autentifikuj()  {  $konekcija = $this->dohvatiKonekciju();  $upit = $konekcija->prepare(  "SELECT lozinka FROM korisnici WHERE korisnicko\_ime = ?"  );  $upit->bind\_param("s", $this->korisnickoIme);  $upit->execute();  $upit->store\_result();  if ($upit->num\_rows > 0) {  $upit->bind\_result($sacuvanaLozinka);  $upit->fetch();  if ($this->lozinka === $sacuvanaLozinka) {  $\_SESSION["username"] = $this->korisnickoIme;  header("Location: index2.php");  exit();  } else {  return "Pogrešna lozinka.";  }  } else {  return "Korisničko ime ne postoji.";  }  $upit->close();  $this->zatvoriKonekciju();  }  }  // Kreiranje instance autentifikatora  if ($\_SERVER["REQUEST\_METHOD"] === "POST") {  $korisnickoIme = $\_POST["username"];  $lozinka = $\_POST["password"];  // Kreiranje instance baze podataka  $baza = new BazaPodataka($servername, $username, $password, $dbname);  // Kreiranje instance autentifikatora  $autentifikator = new Autentifikator(  $servername,  $username,  $password,  $dbname,  $korisnickoIme,  $lozinka  );  $rezultat = $autentifikator->autentifikuj();  if ($rezultat) {  echo $rezultat;  }  }  ?> |

Kod proverava da li korisnik postoji u tabeli korisnici. Ukoliko korisnik ne postoji ili je lozinka pogrešna daje nam upozorenje. Ovaj kod kao I svi ostali delovi koda za rad sa bazom podataka su rađeni po principu objektno orijentisanog programiranja upotrebom klasa. Proces za registraciju je jednostavniji:

|  |
| --- |
| <?php  class BazaPodataka  {  private $conn;  public function \_\_construct($servername, $username, $password, $dbname)  {  $this->conn = new mysqli($servername, $username, $password, $dbname);  if ($this->conn->connect\_error) {  die("Veza neuspešna: " . $this->conn->connect\_error);  }  }  public function getConnection()  {  return $this->conn;  }  public function \_\_destruct()  {  $this->conn->close();  }  }  class Korisnik extends BazaPodataka  {  public function \_\_construct($servername, $username, $password, $dbname)  {  parent::\_\_construct($servername, $username, $password, $dbname);  }  public function registrujKorisnika($korisnickoIme, $lozinka)  {  $stmt = $this->getConnection()->prepare(  "**CALL RegistracijaKorisnika**(?, ?)"  );  if ($stmt) {  $stmt->bind\_param("ss", $korisnickoIme, $lozinka);  if ($stmt->execute()) {  header("Location: index2.php");  exit();  } else {  echo "Greška: " . $stmt->error;  }  $stmt->close();  } else {  echo "Greška: " . $this->getConnection()->error;  }  }  }  // Korisnički podaci  $servername = "localhost";  $username = "root";  $password = "root";  $dbname = "sajt\_baza";  $korisnik = new Korisnik($servername, $username, $password, $dbname);  // Pribavljanje podataka iz forme  if ($\_SERVER["REQUEST\_METHOD"] === "POST") {  $korisnicko\_ime = $\_POST["username"];  $lozinka = $\_POST["lozinka"];  // Registracija korisnika  $korisnik->registrujKorisnika($korisnicko\_ime, $lozinka);  }  ?> |

Kod proverava da li je korisničko ime zauzeto pre no što ga uopšte ubaci u tabelu. Ubacivanje korisnika u tabelu je odrađeno upotrebom procedure.

### 4.1.2 Unos

Unos podataka je moguć na tabeli “recepti” putem našeg kreatora recepata, unos je urađen takođe upotrebom procedure:

|  |
| --- |
| <h2 style="text-align: left; margin-left: 15%;">Kreator recepata</h2>  <br><br>  <hr class="separator">  <br><br>  <section class="recipe-form-section">  <form action="submit-recept.php" method="POST">  <label for="ime">Ime:</label>  <input type="text" id="ime" name="ime" required>  <label for="opis">Opis:</label>  <textarea id="opis" name="opis" required></textarea>  <label for="instrukcije">Instrukcije:</label>  <textarea id="instrukcije" name="instrukcije" required></textarea>  <label for="kategorija">Kategorija:</label>  <select id="kategorija" name="kategorija" required>  <option value="Meso">Meso</option>  <option value="Dezerti">Dezerti</option>  <option value="Pasta">Pasta</option>  <option value="Testo">Testo</option>  <option value="Pice">Piće</option>  <option value="Riba">Riba</option>  <option value="Salate">Salate</option>  <option value="Grickalice">Grickalice</option>  <option value="Sosevi">Sosevi</option>  <option value="Kuvano">Čorbe i Kuvano</option>  </select>  <label>Ograničenje ishrane:</label>  <div>  <input type="radio" id="sve" name="dijeta" value="Sve" checked>  <label for="sve">Sve</label>  <input type="radio" id="vegan" name="dijeta" value="Vegan">  <label for="vegan">Vegan</label>  <input type="radio" id="vegeterijanac" name="dijeta" value="Vegeterijanac">  <label for="vegeterijanac">Vegeterijanac</label>  </div>  <button type="submit">Kreiraj Recept</button>  </form>  </section>  </main> |

Gore je navedena struktura forme za kreaciju tj. Unos podataka u tabelu, klikom na dugme pokreće se kod “submit-recept.php”:

|  |
| --- |
| <?php  session\_start();  // Klasa za rad sa bazom podataka  class BazaPodataka {  private $conn;  public function \_\_construct($servername, $username, $password, $dbname) {  $this->conn = new mysqli($servername, $username, $password, $dbname);  if ($this->conn->connect\_error) {  die("Veza neuspešna: " . $this->conn->connect\_error);  }  }  public function getConnection() {  return $this->conn;  }  public function \_\_destruct() {  $this->conn->close();  }  }  // Klasa za rad sa receptima  class Recept {  private $db;  public function \_\_construct($db) {  $this->db = $db;  }  public function dodajRecept($ime, $opis, $instrukcije, $dijeta, $napravio, $kategorija) {  $sql = "CALL DodajRecept(?, ?, ?, ?, ?, ?)";  $stmt = $this->db->getConnection()->prepare($sql);  if ($stmt) {  $stmt->bind\_param("ssssss", $ime, $opis, $instrukcije, $dijeta, $napravio, $kategorija);  if ($stmt->execute()) {  return "Novi recept je uspešno dodat!";  } else {  return "Greška: " . $stmt->error;  }  $stmt->close();  } else {  return "Greška sa pripremom izjave: " . $this->db->getConnection()->error;  }  }  }  // Provera da li je korisnik ulogovan  if (!isset($\_SESSION['username'])) {  header("Location: login.html");  exit();  }  // Korisnički podaci  $servername = "localhost";  $username = "root";  $password = "root";  $dbname = "sajt\_baza";  // Uzimanje korisničkog imena iz sesije  $napravio = $\_SESSION['username'];  $baza = new BazaPodataka($servername, $username, $password, $dbname);  $recept = new Recept($baza);  // Obrada podataka iz forme  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  $ime = $\_POST['ime'];  $opis = $\_POST['opis'];  $instrukcije = $\_POST['instrukcije'];  $kategorija = $\_POST['kategorija'];  $dijeta = $\_POST['dijeta'];  if ($dijeta == "Sve") {  $dijeta = "";  }  // Dodaj recept  $poruka = $recept->dodajRecept($ime, $opis, $instrukcije, $dijeta, $napravio, $kategorija);  echo $poruka;  }  ?> |

### 4.1.3 Brisanje

Opcija za brisanje objekata je implementirana na stranici nalog, tj. opcija za brisanje korisničkog naloga.

|  |
| --- |
| <?php  session\_start();  class Korisnik  {  private $conn;  private $username;  public function \_\_construct($servername, $username, $password, $dbname)  {  $this->conn = new mysqli($servername, $username, $password, $dbname);    if ($this->conn->connect\_error) {  die("Konekcija nije uspela: " . $this->conn->connect\_error);  }  }  public function setUsername($username)  {  $this->username = $username;  }  public function obrisiNalog()  {  $stmt = $this->conn->prepare("DELETE FROM korisnici WHERE korisnicko\_ime = ?");  $stmt->bind\_param("s", $this->username);  if ($stmt->execute()) {  // Uništavanje sesije nakon brisanja naloga  session\_destroy();  echo '  <script>alert("Vaš nalog je uspešno obrisan."); window.location.href = "login.html";</script>';  exit();  } else {  echo "Greška pri brisanju: " . $this->conn->error;  }  $stmt->close();  }  public function zatvoriKonekciju()  {  // Zatvaranje konekcije  $this->conn->close();  }  }  // Provera da li je korisnik ulogovan  if (!isset($\_SESSION['username'])) {  header("Location: login.html");  exit();  }  // Kreiranje instance klase Korisnik  $servername = "localhost";  $username = "root";  $password = "root";  $dbname = "sajt\_baza";  $korisnik = new Korisnik($servername, $username, $password, $dbname);  $korisnik->setUsername($\_SESSION['username']);  // Provera da li je došlo do zahteva za brisanje  if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['delete'])) {  // Brisanje naloga  $korisnik->obrisiNalog();  // Zatvaranje konekcije  $korisnik->zatvoriKonekciju();  exit();  }  ?>  <!DOCTYPE html>  <html lang="sr">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Brisanje naloga</title>  </head>  <body>  <h1>Brisanje naloga</h1>  <p>Da li ste sigurni? Brisanje naloga je trajno!</p>  <form action="" method="POST">  <input type="hidden" name="delete" value="1">  <button type="submit">Obriši nalog</button>  <a href="nalog.php">Odustani</a>  </form>  </body>  </html> |

### 4.1.4 Izmena

Mogućnost izmene je prikazana menjanjem korisničke lozinke:

|  |
| --- |
| <?php  session\_start();  if (!isset($\_SESSION['username'])) {  header("Location: login.html");  exit();  }  class BazaPodataka {  private $servername = "localhost";  private $username = "root";  private $password = "root";  private $dbname = "sajt\_baza";  protected $conn;  public function \_\_construct() {  $this->conn = new mysqli($this->servername, $this->username, $this->password, $this->dbname);  if ($this->conn->connect\_error) {  die("Veza neuspešna: " . $this->conn->connect\_error);  }  }  public function close() {  $this->conn->close();  }  }  class Korisnik extends BazaPodataka {  public function changePassword($username, $newPassword) {  // Ažuriraj lozinku u bazi  $stmt = $this->conn->prepare("UPDATE korisnici SET lozinka = ? WHERE korisnicko\_ime = ?");  $stmt->bind\_param("ss", $newPassword, $username);  if ($stmt->execute()) {  return true;  } else {  return false;  }  }  }  $username = $\_SESSION['username'];  $new\_password = $\_POST['new\_password'];  $korisnik = new Korisnik();  if ($korisnik->changePassword($username, $new\_password)) {  echo '  <script>alert("Lozinka je uspešno promenjena."); window.location.href = "nalog.php";</script>';  } else {  echo "Greška pri menjanju lozinke: " . $korisnik->conn->error;  }  $korisnik->close();  ?> |

### 4.1.5 Štampa

Štampa se vrši pretragom recepata, recepti se takođe mogu filtrirati radi specifičnije pretrage. Korisnik takođe može „bukvalno“ štampati recept kao posebnu „printer-friendly“ stranicu ali i takođe može preuzeti PDF fajl recepta na svoj računar.

|  |
| --- |
| <h2 style="text-align: left; margin-left: 15%;">Daj nešto nabrzaka!</h2>  <br><br>  <hr class="separator">  <br><br>  <div class="search-container">  <form method="GET" action="recepti2.php">  <input type="text" name="search" placeholder="Pretraži recepte...">  <label>  <input type="checkbox" name="vegan"> Vegan  </label>  <label>  <input type="checkbox" name="vegeterijanac"> Vegeterijanac  </label>  <label>  <input type="checkbox" name="sve"> Sve  </label>  <!-- Dropdown lista za kategorije -->  <label for="kategorija">Izaberi kategoriju:</label>  <select name="kategorija" id="kategorija">  <option value="">Sve kategorije</option>  <option value="Meso">Meso</option>  <option value="Dezerti">Dezerti</option>  <option value="Pasta">Pasta</option>  <option value="Testo">Testo</option>  <option value="Piće">Piće</option>  <option value="Riba">Riba</option>  <option value="Salate">Salate</option>  <option value="Grickalice">Grickalice</option>  <option value="Sosevi">Sosevi</option>  <option value="Kuvano">Kuvano</option>  </select>  <button type="submit">Pretraga</button>  <button type="button" class="clear-button" onclick="clearResults()">Očisti rezultate</button>  </form>  </div>  <!-- PHP povezuje se sa bazom da prikaze rezultate -->  <div id="searchResults">  <?php  // Klasa za rad sa bazom podataka  class BazaPodataka  {  private $conn;  public function \_\_construct($servername, $username, $password, $dbname)  {  $this->conn = new mysqli(  $servername,  $username,  $password,  $dbname  );  if ($this->conn->connect\_error) {  die("Veza neuspešna: " . $this->conn->connect\_error);  }  }  public function getConnection()  {  return $this->conn;  }  public function \_\_destruct()  {  $this->conn->close();  }  }  // Klasa za pretragu recepata koja nasledjuje BazaPodataka  class PretragaRecepata extends BazaPodataka  {  public function \_\_construct($servername, $username, $password, $dbname)  {  parent::\_\_construct($servername, $username, $password, $dbname);  }  public function pretraziRecepte($searchTerm, $filter, $kategorija)  {  $conn = $this->getConnection();  $sql = "SELECT \* FROM recepti WHERE ime LIKE ?";  $params = ["%$searchTerm%"];  $types = "s";  if ($filter === "Vegan") {  $sql .= " AND dijeta = 'Vegan'";  } elseif ($filter === "Vegeterijanac") {  $sql .= " AND dijeta = 'Vegeterijanac'";  }  if (!empty($kategorija)) {  $sql .= " AND kategorija = ?";  $params[] = $kategorija;  $types .= "s";  }  $stmt = $conn->prepare($sql);  if ($stmt === false) {  die("Greška sa pripremom upita: " . $conn->error);  }  $stmt->bind\_param($types, ...$params);  $stmt->execute();  return $stmt->get\_result();  }  }  // Povezivanje sa bazom  $baza = new PretragaRecepata("localhost", "root", "root", "sajt\_baza");  if ($\_SERVER["REQUEST\_METHOD"] == "GET" && isset($\_GET["search"])) {  $searchTerm = $\_GET["search"];  $filter = isset($\_GET["vegan"])  ? "Vegan"  : (isset($\_GET["vegeterijanac"])  ? "Vegeterijanac"  : "Sve");  $kategorija = $\_GET["kategorija"] ?? "";  $result = $baza->pretraziRecepte($searchTerm, $filter, $kategorija);  if ($result->num\_rows > 0) {  echo "<br><br>";  echo "<table border='1' cellpadding='5' cellspacing='0'>";  echo "<tr><th>Ime</th><th>Opis</th><th>Instrukcije</th><th>Dijetalna restrikcija</th><th>Kategorija</th><th>Štampaj recept</th></tr>";  while ($row = $result->fetch\_assoc()) {  echo "<tr>";  echo "<td><a href='#' class='recipe-link' data-ime='" .  htmlspecialchars($row["ime"]) .  "' data-opis='" .  htmlspecialchars($row["opis"]) .  "' data-instrukcije='" .  htmlspecialchars($row["instrukcije"]) .  "'>";  echo htmlspecialchars($row["ime"]) . "</a></td>";  echo "<td>" . htmlspecialchars($row["opis"]) . "</td>";  echo "<td>" . htmlspecialchars($row["instrukcije"]) . "</td>";  echo "<td>" . htmlspecialchars($row["dijeta"]) . "</td>";  echo "<td>" . htmlspecialchars($row["kategorija"]) . "</td>";  echo "<td><a href='printer-friendly.php?ime=" .  urlencode($row["ime"]) .  "' target='\_blank'>Štampaj</a></td>"; // Link za štampanje  echo "</tr>";  }  echo "</table>";  } else {  echo "<p>Nema rezultata za upit: " .  htmlspecialchars($searchTerm) .  "</p>";  }  }  ?>  </div> |

## 4.2 *Opcioni deo*

### 4.2.1 Parametarska štampa

Recepti se pronalaze upotrebom ključnih reči ili ponuđenih filter opcija. Pronađeni recepti se daju u tabelarnom prikazu, gde se potom za svaki recept mogu pojedinačno otvarati posebni „printer-friendly“ prozori kako bi korisnici dobili jasan prikaz jednog recepta koji se kasnije može fizički štampati

|  |  |
| --- | --- |
| |  | | --- | | <h2 style="text-align: left; margin-left: 15%;">Daj nešto nabrzaka!</h2>  <br>  <br>  <hr class="separator">  <br>  <br>  <div class="search-container">  <form method="GET" action="recepti.php">  <input type="text" name="search" placeholder="Pretraži recepte...">  <label>  <input type="checkbox" name="vegan"> Vegan </label>  <label>  <input type="checkbox" name="vegeterijanac"> Vegeterijanac </label>  <label>  <input type="checkbox" name="sve"> Sve </label>  <!-- Dropdown lista za kategorije -->  <label for="kategorija">Izaberi kategoriju:</label>  <select name="kategorija" id="kategorija">  <option value="">Sve kategorije</option>  <option value="Meso">Meso</option>  <option value="Dezerti">Dezerti</option>  <option value="Pasta">Pasta</option>  <option value="Testo">Testo</option>  <option value="Piće">Piće</option>  <option value="Riba">Riba</option>  <option value="Salate">Salate</option>  <option value="Grickalice">Grickalice</option>  <option value="Sosevi">Sosevi</option>  <option value="Kuvano">Kuvano</option>  </select>  <button type="submit">Pretraga</button>  <button type="button" class="clear-button" onclick="clearResults()">Očisti rezultate</button>  </form>  </div>  <!-- PHP povezuje se sa bazom da prikaze rezultate -->  <div id="searchResults"> <?php  // Klasa za rad sa bazom podataka  class BazaPodataka {  private $conn;  public function \_\_construct($servername, $username, $password, $dbname) {  $this->conn = new mysqli($servername, $username, $password, $dbname);  if ($this->conn->connect\_error) {  die("Veza neuspesna: " . $this->conn->connect\_error);  }  }  public function getConnection() {  return $this->conn;  }  public function \_\_destruct() {  $this->conn->close();  }  }  // Klasa za pretragu recepata  class PretragaRecepata {  private $db;  public function \_\_construct($db) {  $this->db = $db;  }  public function pretraziRecepte($searchTerm, $filter, $kategorija) {  $conn = $this->db->getConnection();  $sql = "SELECT \* FROM recepti WHERE ime LIKE ?";  $params = ["%$searchTerm%"];  $types = "s";  if ($filter === 'Vegan') {  $sql .= " AND dijeta = 'Vegan'";  } elseif ($filter === 'Vegeterijanac') {  $sql .= " AND dijeta = 'Vegeterijanac'";  }  // Filtriranje po kategoriji  if (!empty($kategorija)) {  $sql .= " AND kategorija = ?";  $params[] = $kategorija;  $types .= "s"; // dodajemo još jedan string tip  }  $stmt = $conn->prepare($sql);  if ($stmt === false) {  die("Greška sa pripremom upita: " . $conn->error);  }  $stmt->bind\_param($types, ...$params);  $stmt->execute();  return $stmt->get\_result();  }  }  // Povezivanje sa bazom  $baza = new BazaPodataka("localhost", "root", "root", "sajt\_baza");  $pretraga = new PretragaRecepata($baza);  if ($\_SERVER["REQUEST\_METHOD"] == "GET" && isset($\_GET['search'])) {  $searchTerm = $\_GET['search'];  // Filtriranje na osnovu dijetalnih restrikcija  $filter = "";  if (isset($\_GET['vegan'])) {  $filter = 'Vegan';  } elseif (isset($\_GET['vegeterijanac'])) {  $filter = 'Vegeterijanac';  } elseif (isset($\_GET['sve'])) {  $filter = 'Sve';  }  // Uzmi kategoriju iz GET zahteva  $kategorija = $\_GET['kategorija'] ?? '';  // Pretraga recepata  $result = $pretraga->pretraziRecepte($searchTerm, $filter, $kategorija);  if ($result->num\_rows > 0) {  echo "  <div class='search-results'>";  while ($row = $result->fetch\_assoc()) {  echo "  <a href='#' class='recipe-link' data-ime='" . htmlspecialchars($row['ime']) . "' data-opis='" . htmlspecialchars($row['opis']) . "' data-instrukcije='" . htmlspecialchars($row['instrukcije']) . "'>";  echo "  <h3>" . htmlspecialchars($row['ime']) . "</h3>";  echo "  </a>";  }  echo "  </div>";  } else {  echo "  <p>Nema rezultata za upit: " . htmlspecialchars($searchTerm) . "</p>";  }  }  ?> </div>  <!-- js skripte za ciscenje rezultata i js koji prikazuje stranicu kao dinamicku pop-up stranicu --> | |

Gore naveden kod predstavlja ceo segment za pretragu recepata sa JavaScript funkcionalnošću za čišćenje rezultata kao i php delom sekcije koji je zaslužan za uzimanje podataka iz tabele u bazi podataka.

# ZAKLJUČAK

Kroz rad na ovoj web aplikaciji upoznati smo alatima i jezicima koji su potrebni da bi se ona izradila, kao što su PHP i MySQL. Takođe smo se upoznali smo sa terminom lokalnog hosting-a sajta putem aplikacije MAMP. Upotrebom alata i veština koje smo stekli tokom rada smo upotrebili u izradi funkcionalnog sajta za deljenje i kreiranje recepata. Sajt se dalje može unaprediti tako što se može implementirati opcija za čuvanje recepata, funkcija za lajkove se može izmeniti u funkciju za davanje ocena i te ocene se mogu dinamično potom prikazivati na stranicama, može se implementirati kompleksniji sistem filtriranja i recepti se mogu čuvati na stranicama kategorija koje su zabeležene na stranici „Recepti“.

# LITERATURA

1. <https://classroom.google.com/u/1/c/NjYyNDYwNDk0MjY5/a/NzEyMzE5MzE3NjQ4/details> - Šablon za kolokvijum iz Web Programiranja (Prof. Dr. Ljubica Kazi)
2. <https://www.w3schools.com/php/php_intro.asp> - PHP kurs (W3Schools)
3. <https://laracasts.com/series/php-for-beginners-2023-edition> - PHP kurs (Laracasts)
4. <https://www.w3schools.com/sql/> - SQL kurs (W3Schools)
5. <https://github.com/m3Mza/Veb-Programiranje.git> - Link ka GitHub repozitorijumu