Experiment No.7

Date:13.03.25

Aim: Implementation of real world problem based on servlet concept

CO Mapping – CO 3

Objective:

1. To write servlet programs
2. To connect servlet programs with the help of JSP with database

Code:

Pom.xml  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>org.example</groupId>  
 <artifactId>servlet\_withDB</artifactId>  
 <packaging>war</packaging>  
 <version>1.0-SNAPSHOT</version>  
 <name>servlet\_withDB Maven Webapp</name>  
 <url>http://maven.apache.org</url>  
 <dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>3.8.1</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>javax.servlet-api</artifactId>  
 <version>4.0.1</version>  
 <scope>provided</scope>  
 </dependency>  
 <dependency>  
 <groupId>com.mysql</groupId>  
 <artifactId>mysql-connector-j</artifactId>  
 <version>8.0.33</version>  
 <scope>compile</scope>  
 </dependency>  
 <dependency>  
 <groupId>com.google.code.gson</groupId>  
 <artifactId>gson</artifactId>  
 <version>2.12.1</version>  
 </dependency>  
  
 </dependencies>  
 <build>  
 <finalName>servlet\_withDB</finalName>  
 </build>  
</project>

DBConnection.java  
package com.example;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
  
public class DBConnection {  
 private static final String *URL* = "jdbc:mysql://127.0.0.1:3306/userdb";  
 private static final String *USER* = "root";  
 private static final String *PASSWORD* = "Athanee@08";  
  
 public static Connection getConnection() {  
 Connection conn = null;  
 try {  
 Class.*forName*("com.mysql.cj.jdbc.Driver"); // Ensure driver is loaded  
 conn = DriverManager.*getConnection*(*URL*, *USER*, *PASSWORD*);  
 } catch (ClassNotFoundException e) {  
 System.*out*.println("MySQL JDBC Driver not found!");  
 e.printStackTrace();  
 } catch (SQLException e) {  
 System.*out*.println("Database connection failed!");  
 e.printStackTrace();  
 }  
 return conn;  
 }  
}

UserDao.java  
package com.example;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.List;  
  
public class UserDao {  
  
 public static boolean registerUser(String name, String email, String password) {  
 System.*out*.println("Attempting to register user: " + email);  
 String sql = "INSERT INTO users (name, email, password) VALUES (?, ?, ?)";  
  
 try (Connection conn = DBConnection.*getConnection*()) {  
 if (conn == null) {  
 System.*out*.println("Connection is null, cannot register user");  
 return false;  
 }  
  
 PreparedStatement pstmt = conn.prepareStatement(sql);  
 pstmt.setString(1, name);  
 pstmt.setString(2, email);  
 pstmt.setString(3, password); // In a real app, you should hash this password  
  
 int rowsAffected = pstmt.executeUpdate();  
 System.*out*.println("User registration result, rows affected: " + rowsAffected);  
 pstmt.close();  
 return rowsAffected > 0;  
  
 } catch (SQLException e) {  
 System.*out*.println("SQL Error during registration: " + e.getMessage());  
 e.printStackTrace();  
 return false;  
 }  
 }  
  
 public static List<User> getAllUsers() {  
 List<User> users = new ArrayList<>();  
 String sql = "SELECT name, email, password FROM users";  
  
 try (Connection conn = DBConnection.*getConnection*()) {  
 if (conn == null) {  
 System.*out*.println("Connection is null, cannot get users");  
 return users;  
 }  
  
 PreparedStatement pstmt = conn.prepareStatement(sql);  
 ResultSet rs = pstmt.executeQuery();  
  
 while (rs.next()) {  
 User user = new User();  
 user.setName(rs.getString("name"));  
 user.setEmail(rs.getString("email"));  
 user.setPassword(rs.getString("password"));  
 users.add(user);  
 }  
  
 rs.close();  
 pstmt.close();  
  
 } catch (SQLException e) {  
 System.*out*.println("Error getting users: " + e.getMessage());  
 e.printStackTrace();  
 }  
  
 return users;  
 }  
  
 public static boolean updateUser(String oldEmail, String newName, String newEmail) {  
 String sql = "UPDATE users SET name = ?, email = ? WHERE email = ?";  
  
 try (Connection conn = DBConnection.*getConnection*()) {  
 if (conn == null) {  
 System.*out*.println("Connection is null, cannot update user");  
 return false;  
 }  
  
 PreparedStatement pstmt = conn.prepareStatement(sql);  
 pstmt.setString(1, newName);  
 pstmt.setString(2, newEmail);  
 pstmt.setString(3, oldEmail);  
  
 int rowsAffected = pstmt.executeUpdate();  
 pstmt.close();  
 return rowsAffected > 0;  
  
 } catch (SQLException e) {  
 System.*out*.println("Error updating user: " + e.getMessage());  
 e.printStackTrace();  
 return false;  
 }  
 }  
  
 public static boolean deleteUser(String email) {  
 String sql = "DELETE FROM users WHERE email = ?";  
  
 try (Connection conn = DBConnection.*getConnection*()) {  
 if (conn == null) {  
 System.*out*.println("Connection is null, cannot delete user");  
 return false;  
 }  
  
 PreparedStatement pstmt = conn.prepareStatement(sql);  
 pstmt.setString(1, email);  
  
 int rowsAffected = pstmt.executeUpdate();  
 pstmt.close();  
 return rowsAffected > 0;  
  
 } catch (SQLException e) {  
 System.*out*.println("Error deleting user: " + e.getMessage());  
 e.printStackTrace();  
 return false;  
 }  
 }  
}

User.java  
package com.example;  
  
public class User {  
 private String name;  
 private String email;  
 private String password;  
  
 public User() {  
 }  
  
 public User(String name, String email, String password) {  
 this.name = name;  
 this.email = email;  
 this.password = password;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
}

UserServlet.java  
package com.example;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.util.List;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import com.google.gson.Gson;  
  
public class UserServlet extends HttpServlet {  
  
 protected void doPost(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
 System.*out*.println("doPost method called in UserServlet");  
 response.setContentType("application/json");  
 PrintWriter out = response.getWriter();  
  
 try {  
 String name = request.getParameter("name");  
 String email = request.getParameter("email");  
 String password = request.getParameter("password");  
  
 System.*out*.println("Received parameters - name: " + name + ", email: " + email);  
  
 // Validate inputs  
 if (name == null || email == null || password == null ||  
 name.trim().isEmpty() || email.trim().isEmpty() || password.trim().isEmpty()) {  
 out.println("{\"message\": \"All fields are required!\"}");  
 return;  
 }  
  
 boolean success = UserDao.*registerUser*(name, email, password);  
 String message = success ? "User registered successfully!" : "Registration failed!";  
 out.println("{\"message\": \"" + message + "\"}");  
 } catch (Exception e) {  
 System.*out*.println("Error in doPost: " + e.getMessage());  
 e.printStackTrace();  
 out.println("{\"message\": \"An error occurred: " + e.getMessage() + "\"}");  
 }  
 }  
  
 protected void doGet(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
 System.*out*.println("doGet method called in UserServlet");  
 response.setContentType("application/json");  
 PrintWriter out = response.getWriter();  
  
 try {  
 List<User> users = UserDao.*getAllUsers*();  
 Gson gson = new Gson();  
 String json = gson.toJson(users);  
 out.println(json);  
 } catch (Exception e) {  
 System.*out*.println("Error in doGet: " + e.getMessage());  
 e.printStackTrace();  
 out.println("{\"message\": \"An error occurred: " + e.getMessage() + "\"}");  
 }  
 }  
  
 protected void doPut(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
 System.*out*.println("doPut method called in UserServlet");  
 response.setContentType("application/json");  
 PrintWriter out = response.getWriter();  
  
 try {  
 String requestData = readRequestBody(request);  
 System.*out*.println("Request data: " + requestData);  
  
 String oldEmail = extractJsonValue(requestData, "oldEmail");  
 String newName = extractJsonValue(requestData, "name");  
 String newEmail = extractJsonValue(requestData, "email");  
  
 System.*out*.println("Updating user - oldEmail: " + oldEmail + ", newName: " + newName + ", newEmail: " + newEmail);  
  
 boolean updated = UserDao.*updateUser*(oldEmail, newName, newEmail);  
 String message = updated ? "User updated successfully!" : "User not found!";  
 out.println("{\"message\": \"" + message + "\"}");  
 } catch (Exception e) {  
 System.*out*.println("Error in doPut: " + e.getMessage());  
 e.printStackTrace();  
 out.println("{\"message\": \"An error occurred: " + e.getMessage() + "\"}");  
 }  
 }  
  
 protected void doDelete(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
 System.*out*.println("doDelete method called in UserServlet");  
 response.setContentType("application/json");  
 PrintWriter out = response.getWriter();  
  
 try {  
 String requestData = readRequestBody(request);  
 System.*out*.println("Request data: " + requestData);  
  
 String email = extractJsonValue(requestData, "email");  
 System.*out*.println("Deleting user with email: " + email);  
  
 boolean deleted = UserDao.*deleteUser*(email);  
 String message = deleted ? "User deleted successfully!" : "User not found!";  
 out.println("{\"message\": \"" + message + "\"}");  
 } catch (Exception e) {  
 System.*out*.println("Error in doDelete: " + e.getMessage());  
 e.printStackTrace();  
 out.println("{\"message\": \"An error occurred: " + e.getMessage() + "\"}");  
 }  
 }  
  
 private String readRequestBody(HttpServletRequest request) throws IOException {  
 BufferedReader reader = request.getReader();  
 StringBuilder sb = new StringBuilder();  
 String line;  
 while ((line = reader.readLine()) != null) {  
 sb.append(line);  
 }  
 return sb.toString().trim();  
 }  
  
 private String extractJsonValue(String json, String key) {  
 String searchKey = "\"" + key + "\":\"";  
 int startIndex = json.indexOf(searchKey);  
 if (startIndex == -1) {  
 return "";  
 }  
 startIndex += searchKey.length();  
 int endIndex = json.indexOf("\"", startIndex);  
 return endIndex == -1 ? "" : json.substring(startIndex, endIndex);  
 }  
}

web.xml  
<!DOCTYPE web-app PUBLIC  
 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  
 "http://java.sun.com/dtd/web-app\_2\_3.dtd" >  
  
<web-app>  
 <display-name>Archetype Created Web Application</display-name>  
 <servlet>  
 <servlet-name>UserServlet</servlet-name>  
 <servlet-class>com.example.UserServlet</servlet-class>  
 </servlet>  
  
 <servlet-mapping>  
 <servlet-name>UserServlet</servlet-name>  
 <url-pattern>/register</url-pattern>  
 </servlet-mapping>  
  
 <servlet-mapping>  
 <servlet-name>UserServlet</servlet-name>  
 <url-pattern>/users</url-pattern>  
 </servlet-mapping>  
  
 <servlet-mapping>  
 <servlet-name>UserServlet</servlet-name>  
 <url-pattern>/updateUser</url-pattern>  
 </servlet-mapping>  
  
 <servlet-mapping>  
 <servlet-name>UserServlet</servlet-name>  
 <url-pattern>/deleteUser</url-pattern>  
 </servlet-mapping>  
  
 <welcome-file-list>  
 <welcome-file>index.jsp</welcome-file>  
 </welcome-file-list>  
</web-app>

index.jsp  
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>User Management System</title>  
 <style>  
 body {  
 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
 background-color: #f7f9fc;  
 margin: 0;  
 padding: 0;  
 }  
 .container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 20px;  
 }  
 .header {  
 background-color: #3498db;  
 color: white;  
 padding: 20px 0;  
 text-align: center;  
 border-radius: 8px 8px 0 0;  
 margin-bottom: 30px;  
 box-shadow: 0 4px 6px rgba(0,0,0,0.1);  
 }  
 .header h1 {  
 margin: 0;  
 }  
 .card-container {  
 display: flex;  
 flex-wrap: wrap;  
 justify-content: space-around;  
 gap: 20px;  
 }  
 .card {  
 background-color: white;  
 border-radius: 8px;  
 box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
 padding: 20px;  
 width: 220px;  
 text-align: center;  
 transition: transform 0.3s, box-shadow 0.3s;  
 }  
 .card:hover {  
 transform: translateY(-5px);  
 box-shadow: 0 6px 12px rgba(0,0,0,0.15);  
 }  
 .card-icon {  
 font-size: 64px;  
 margin-bottom: 15px;  
 color: #3498db;  
 }  
 .card h2 {  
 color: #2c3e50;  
 margin: 0 0 15px 0;  
 }  
 .card p {  
 color: #7f8c8d;  
 margin-bottom: 20px;  
 min-height: 60px;  
 }  
 .btn {  
 display: inline-block;  
 background-color: #3498db;  
 color: white;  
 padding: 10px 20px;  
 border-radius: 4px;  
 text-decoration: none;  
 transition: background-color 0.3s;  
 }  
 .btn:hover {  
 background-color: #2980b9;  
 }  
 .footer {  
 text-align: center;  
 margin-top: 40px;  
 padding: 20px;  
 color: #7f8c8d;  
 border-top: 1px solid #e0e0e0;  
 }  
 </style>  
</head>  
<body>  
 <div class="container">  
 <div class="header">  
 <h1>User Management System</h1>  
 </div>  
  
 <div class="card-container">  
 <div class="card">  
 <div class="card-icon">➕</div>  
 <h2>Register</h2>  
 <p>Add a new user to the system with name, email, and password.</p>  
 <a href="register.jsp" class="btn">Register User</a>  
 </div>  
  
 <div class="card">  
 <div class="card-icon">👁️</div>  
 <h2>View Users</h2>  
 <p>View all registered users and their information in the system.</p>  
 <a href="view.jsp" class="btn">View All Users</a>  
 </div>  
  
 <div class="card">  
 <div class="card-icon">🔄</div>  
 <h2>Update User</h2>  
 <p>Update existing user information such as name and email.</p>  
 <a href="update.jsp" class="btn">Update User</a>  
 </div>  
  
 <div class="card">  
 <div class="card-icon">❌</div>  
 <h2>Delete User</h2>  
 <p>Remove a user from the system using their email address.</p>  
 <a href="delete.jsp" class="btn">Delete User</a>  
 </div>  
 </div>  
  
 <div class="footer">  
 <p>Servlet-based User Management System &copy; 2025</p>  
 </div>  
 </div>  
</body>  
</html>

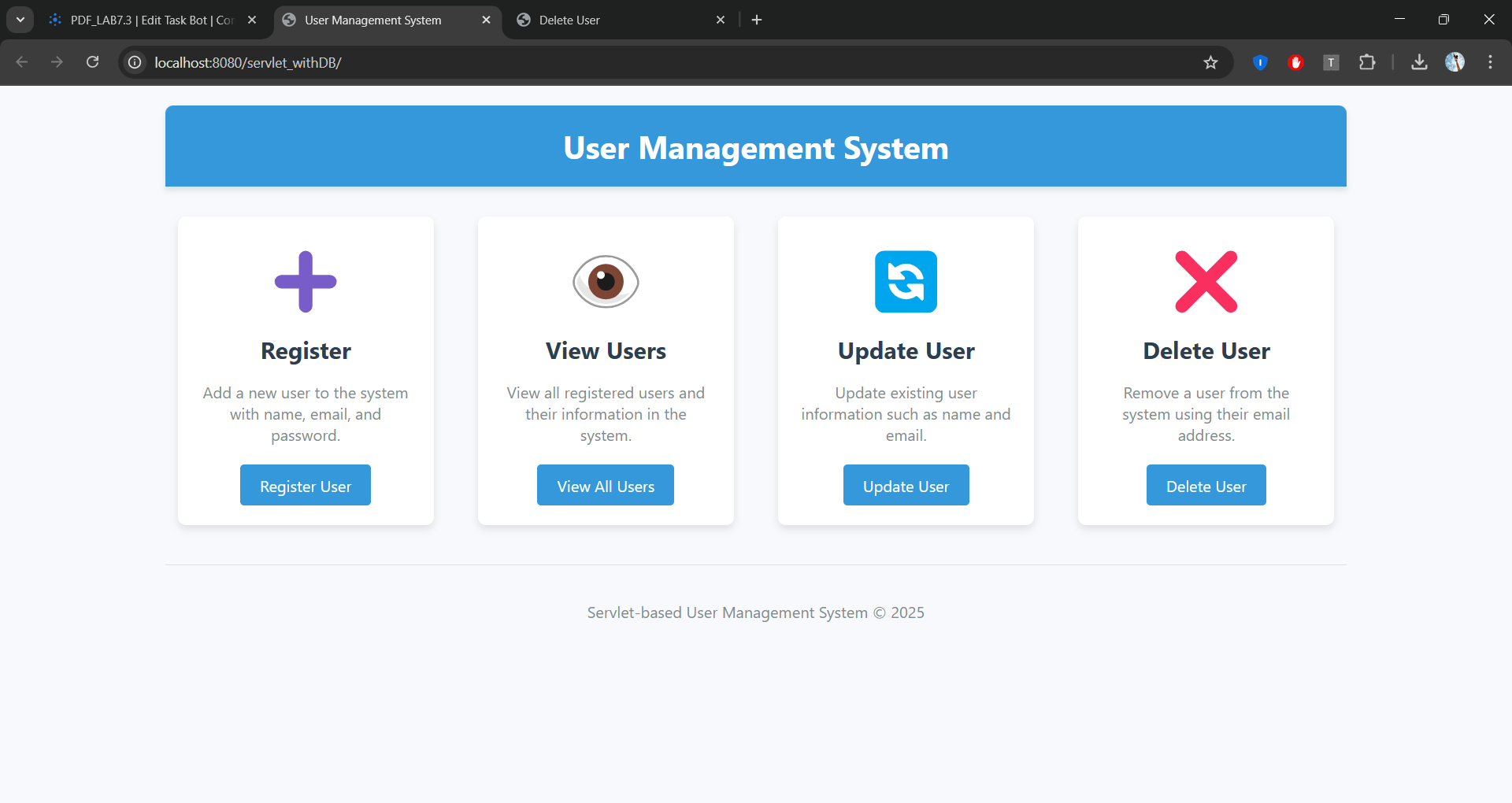
register.jsp  
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Register User</title>  
 <style>  
 body {  
 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
 background-color: #f7f9fc;  
 margin: 0;  
 padding: 0;  
 }  
 .container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 20px;  
 }  
 .header {  
 background-color: #3498db;  
 color: white;  
 padding: 20px 0;  
 text-align: center;  
 border-radius: 8px 8px 0 0;  
 margin-bottom: 30px;  
 box-shadow: 0 4px 6px rgba(0,0,0,0.1);  
 }  
 .header h1 {  
 margin: 0;  
 }  
 .nav {  
 display: flex;  
 justify-content: center;  
 margin-bottom: 30px;  
 flex-wrap: wrap;  
 }  
 .nav a {  
 display: inline-block;  
 padding: 10px 15px;  
 margin: 0 5px 10px 5px;  
 background-color: white;  
 color: #3498db;  
 text-decoration: none;  
 border-radius: 4px;  
 box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
 transition: all 0.3s;  
 }  
 .nav a:hover {  
 background-color: #3498db;  
 color: white;  
 transform: translateY(-2px);  
 box-shadow: 0 4px 6px rgba(0,0,0,0.15);  
 }  
 .card {  
 background-color: white;  
 border-radius: 8px;  
 box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
 padding: 30px;  
 max-width: 500px;  
 margin: 0 auto 30px auto;  
 text-align: center;  
 }  
 .card-icon {  
 font-size: 64px;  
 margin-bottom: 15px;  
 color: #3498db;  
 }  
 .card h2 {  
 color: #2c3e50;  
 margin: 0 0 25px 0;  
 }  
 .form-group {  
 margin-bottom: 20px;  
 text-align: left;  
 }  
 .form-group label {  
 display: block;  
 margin-bottom: 8px;  
 color: #2c3e50;  
 font-weight: 500;  
 }  
 .form-control {  
 width: 100%;  
 padding: 12px;  
 border: 1px solid #ddd;  
 border-radius: 4px;  
 box-sizing: border-box;  
 font-size: 16px;  
 transition: border-color 0.3s;  
 }  
 .form-control:focus {  
 outline: none;  
 border-color: #3498db;  
 box-shadow: 0 0 0 3px rgba(52, 152, 219, 0.2);  
 }  
 .btn {  
 display: inline-block;  
 background-color: #3498db;  
 color: white;  
 padding: 12px 30px;  
 border: none;  
 border-radius: 4px;  
 cursor: pointer;  
 font-size: 16px;  
 font-weight: 500;  
 transition: background-color 0.3s;  
 width: 100%;  
 }  
 .btn:hover {  
 background-color: #2980b9;  
 }  
 .alert {  
 padding: 15px;  
 border-radius: 4px;  
 margin-bottom: 20px;  
 color: white;  
 display: none;  
 }  
 .alert-success {  
 background-color: #2ecc71;  
 }  
 .alert-danger {  
 background-color: #e74c3c;  
 }  
 .footer {  
 text-align: center;  
 margin-top: 40px;  
 padding: 20px;  
 color: #7f8c8d;  
 border-top: 1px solid #e0e0e0;  
 }  
 </style>  
</head>  
<body>  
 <div class="container">  
 <div class="header">  
 <h1>User Management System</h1>  
 </div>  
  
 <div class="nav">  
 <a href="index.jsp">Home</a>  
 <a href="register.jsp">Register</a>  
 <a href="view.jsp">View Users</a>  
 <a href="update.jsp">Update User</a>  
 <a href="delete.jsp">Delete User</a>  
 </div>  
  
 <div class="card">  
 <div class="card-icon">➕</div>  
 <h2>Register New User</h2>  
  
 <div id="message" class="alert"></div>  
  
 <form id="registerForm">  
 <div class="form-group">  
 <label for="name">Full Name</label>  
 <input type="text" id="name" name="name" class="form-control" placeholder="Enter your full name" required>  
 </div>  
  
 <div class="form-group">  
 <label for="email">Email Address</label>  
 <input type="email" id="email" name="email" class="form-control" placeholder="Enter your email address" required>  
 </div>  
  
 <div class="form-group">  
 <label for="password">Password</label>  
 <input type="password" id="password" name="password" class="form-control" placeholder="Create a password" required>  
 </div>  
  
 <button type="submit" class="btn">Register User</button>  
 </form>  
 </div>  
  
 <div class="footer">  
 <p>Servlet-based User Management System &copy; 2025</p>  
 </div>  
 </div>  
  
 <script>  
 // Check if there's a message parameter in the URL  
 const urlParams = new URLSearchParams(window.location.search);  
 const message = urlParams.get('message');  
 const messageDiv = document.getElementById('message');  
  
 if (message) {  
 messageDiv.style.display = 'block';  
 if (message.includes('successfully')) {  
 messageDiv.className = 'alert alert-success';  
 } else {  
 messageDiv.className = 'alert alert-danger';  
 }  
 messageDiv.textContent = decodeURIComponent(message);  
 }  
  
 // For AJAX form submission  
 document.getElementById('registerForm').addEventListener('submit', function(event) {  
 event.preventDefault();  
  
 console.log("Form submission started");  
  
 // Clear previous messages  
 messageDiv.style.display = 'none';  
  
 // Get form values  
 const name = document.getElementById('name').value.trim();  
 const email = document.getElementById('email').value.trim();  
 const password = document.getElementById('password').value;  
  
 // Simple validation  
 if (!name || !email || !password) {  
 messageDiv.style.display = 'block';  
 messageDiv.className = 'alert alert-danger';  
 messageDiv.textContent = 'All fields are required!';  
 return;  
 }  
  
 console.log("Sending data - name: " + name + ", email: " + email);  
  
 // Create a URL-encoded string from the form data  
 const formData = new FormData(this);  
 const urlEncodedData = new URLSearchParams(formData).toString();  
 console.log("Form data: " + urlEncodedData);  
  
 // Show loading state  
 const submitBtn = this.querySelector('button[type="submit"]');  
 const originalBtnText = submitBtn.textContent;  
 submitBtn.textContent = 'Processing...';  
 submitBtn.disabled = true;  
  
 fetch('register', {  
 method: 'POST',  
 headers: {  
 'Content-Type': 'application/x-www-form-urlencoded',  
 'Accept': 'application/json'  
 },  
 body: urlEncodedData  
 })  
 .then(response => {  
 // Log response status for debugging  
 console.log('Response status:', response.status);  
 console.log('Response headers:', response.headers);  
 return response.json();  
 })  
 .then(data => {  
 // Log data for debugging  
 console.log('Response data:', data);  
  
 messageDiv.style.display = 'block';  
 if (data.message && data.message.includes('successfully')) {  
 messageDiv.className = 'alert alert-success';  
 document.getElementById('registerForm').reset();  
 } else {  
 messageDiv.className = 'alert alert-danger';  
 }  
 messageDiv.textContent = data.message || 'Unknown error occurred';  
 })  
 .catch(error => {  
 console.error('Error details:', error);  
 messageDiv.style.display = 'block';  
 messageDiv.className = 'alert alert-danger';  
 messageDiv.textContent = 'An error occurred. Please try again.';  
 })  
 .finally(() => {  
 // Restore button state  
 submitBtn.textContent = originalBtnText;  
 submitBtn.disabled = false;  
 console.log("Form submission complete");  
 });  
 });  
 </script>  
</body>  
</html>

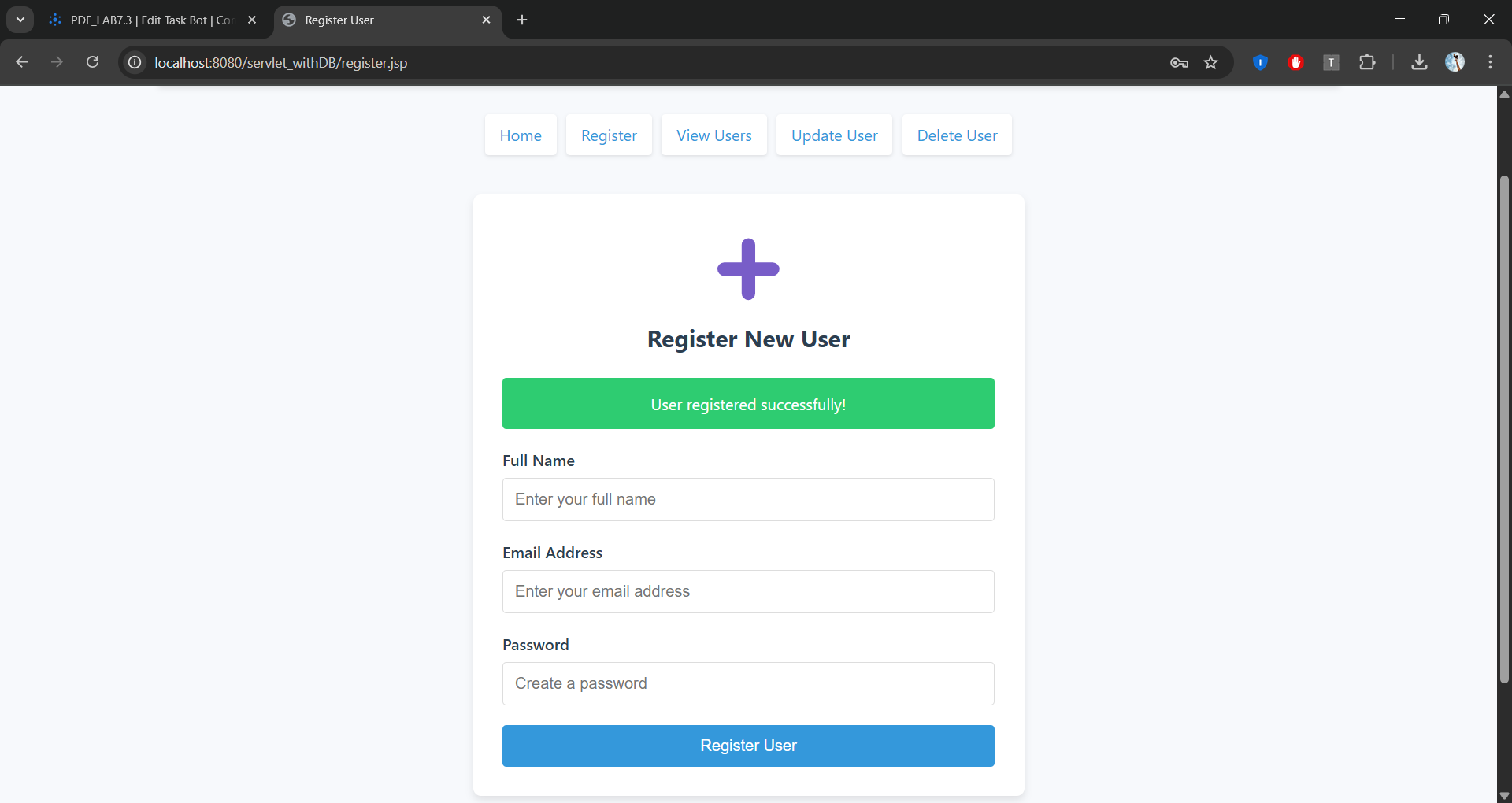
view.jsp  
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>View Users</title>  
 <style>  
 body {  
 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
 background-color: #f7f9fc;  
 margin: 0;  
 padding: 0;  
 }  
 .container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 20px;  
 }  
 .header {  
 background-color: #3498db;  
 color: white;  
 padding: 20px 0;  
 text-align: center;  
 border-radius: 8px 8px 0 0;  
 margin-bottom: 30px;  
 box-shadow: 0 4px 6px rgba(0,0,0,0.1);  
 }  
 .header h1 {  
 margin: 0;  
 }  
 .nav {  
 display: flex;  
 justify-content: center;  
 margin-bottom: 30px;  
 flex-wrap: wrap;  
 }  
 .nav a {  
 display: inline-block;  
 padding: 10px 15px;  
 margin: 0 5px 10px 5px;  
 background-color: white;  
 color: #3498db;  
 text-decoration: none;  
 border-radius: 4px;  
 box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
 transition: all 0.3s;  
 }  
 .nav a:hover {  
 background-color: #3498db;  
 color: white;  
 transform: translateY(-2px);  
 box-shadow: 0 4px 6px rgba(0,0,0,0.15);  
 }  
 .card {  
 background-color: white;  
 border-radius: 8px;  
 box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
 padding: 30px;  
 margin: 0 auto 30px auto;  
 text-align: center;  
 }  
 .card-icon {  
 font-size: 64px;  
 margin-bottom: 15px;  
 color: #3498db;  
 }  
 .card h2 {  
 color: #2c3e50;  
 margin: 0 0 25px 0;  
 }  
 .btn {  
 display: inline-block;  
 background-color: #3498db;  
 color: white;  
 padding: 12px 30px;  
 border: none;  
 border-radius: 4px;  
 cursor: pointer;  
 font-size: 16px;  
 font-weight: 500;  
 transition: background-color 0.3s;  
 margin-bottom: 20px;  
 }  
 .btn:hover {  
 background-color: #2980b9;  
 }  
 .user-table {  
 width: 100%;  
 border-collapse: collapse;  
 margin-top: 20px;  
 }  
 .user-table th, .user-table td {  
 padding: 12px 15px;  
 text-align: left;  
 border-bottom: 1px solid #ddd;  
 }  
 .user-table th {  
 background-color: #f2f2f2;  
 color: #2c3e50;  
 font-weight: 600;  
 }  
 .user-table tr:hover {  
 background-color: #f5f5f5;  
 }  
 .empty-state {  
 padding: 40px 20px;  
 color: #7f8c8d;  
 }  
 .footer {  
 text-align: center;  
 margin-top: 40px;  
 padding: 20px;  
 color: #7f8c8d;  
 border-top: 1px solid #e0e0e0;  
 }  
 </style>  
</head>  
<body>  
 <div class="container">  
 <div class="header">  
 <h1>User Management System</h1>  
 </div>  
  
 <div class="nav">  
 <a href="index.jsp">Home</a>  
 <a href="register.jsp">Register</a>  
 <a href="view.jsp">View Users</a>  
 <a href="update.jsp">Update User</a>  
 <a href="delete.jsp">Delete User</a>  
 </div>  
  
 <div class="card">  
 <div class="card-icon">👁️</div>  
 <h2>View All Users</h2>  
  
 <button class="btn" onclick="viewUsers()">Refresh Users List</button>  
  
 <div id="usersList">  
 <!-- User list will be populated here -->  
 </div>  
 </div>  
  
 <div class="footer">  
 <p>Servlet-based User Management System &copy; 2025</p>  
 </div>  
 </div>  
  
 <script>  
 // Load users when the page loads  
 window.onload = viewUsers;  
  
 function viewUsers() {  
 fetch('register', { method: 'GET' })  
 .then(response => response.json())  
 .then(data => {  
 let usersList = document.getElementById('usersList');  
  
 if (data.length === 0) {  
 usersList.innerHTML = '<div class="empty-state">' +  
 '<h3>No Users Found</h3>' +  
 '<p>There are no users registered in the system yet.</p>' +  
 '<p>Go to the Register page to add new users.</p>' +  
 '</div>';  
 } else {  
 let tableHTML = '<table class="user-table">' +  
 '<thead>' +  
 '<tr>' +  
 '<th>Name</th>' +  
 '<th>Email</th>' +  
 '</tr>' +  
 '</thead>' +  
 '<tbody>';  
  
 data.forEach(user => {  
 tableHTML += `<tr>  
 <td>${user.name}</td>  
 <td>${user.email}</td>  
 </tr>`;  
 });  
  
 tableHTML += '</tbody></table>';  
 usersList.innerHTML = tableHTML;  
 }  
 })  
 .catch(error => {  
 console.error('Error:', error);  
 document.getElementById('usersList').innerHTML =  
 '<div class="empty-state">' +  
 '<h3>Error</h3>' +  
 '<p>Failed to load users. Please try again later.</p>' +  
 '</div>';  
 });  
 }  
 </script>  
</body>  
</html>

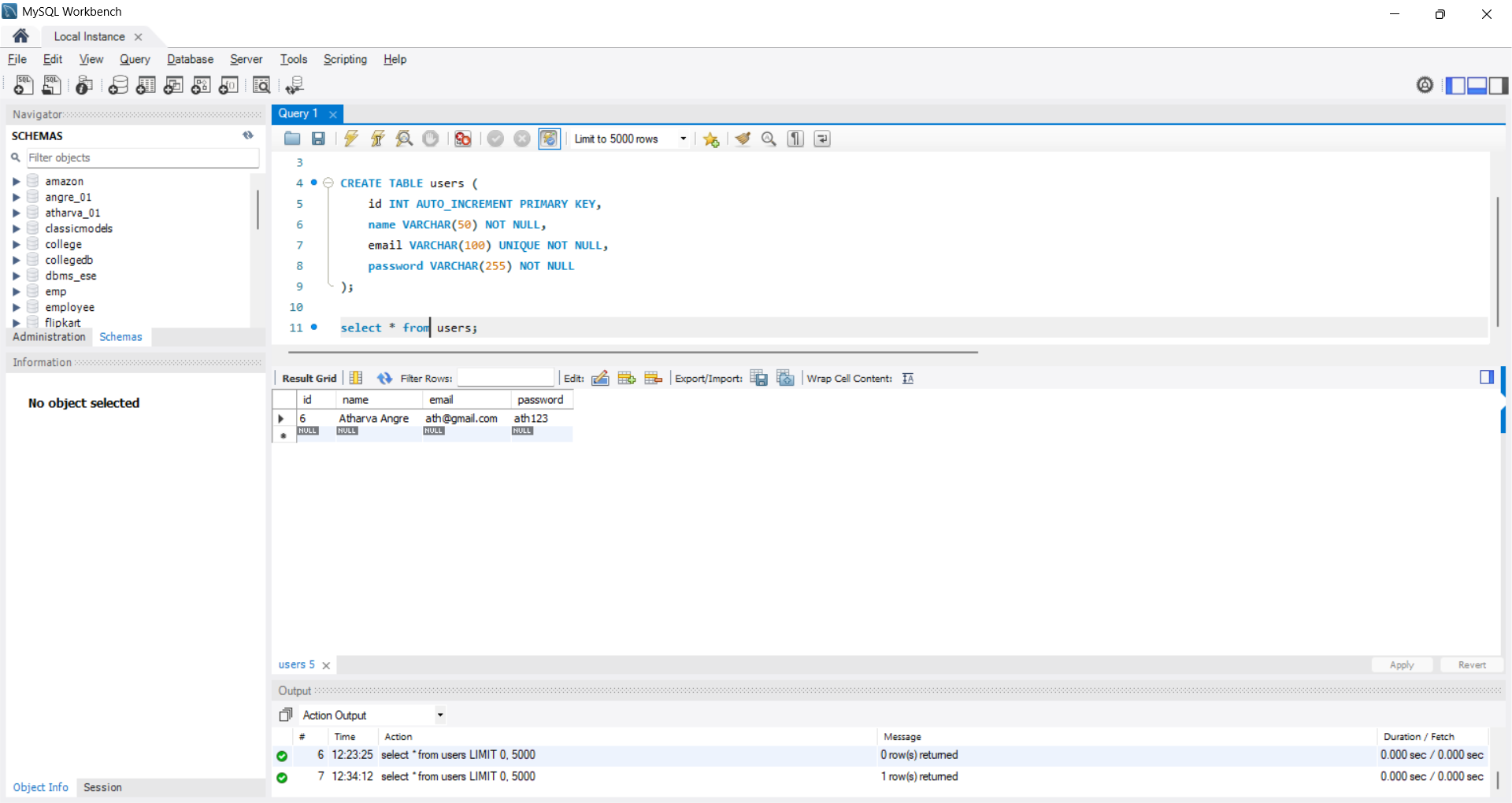
update.jsp  
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Update User</title>  
 <style>  
 body {  
 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
 background-color: #f7f9fc;  
 margin: 0;  
 padding: 0;  
 }  
 .container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 20px;  
 }  
 .header {  
 background-color: #3498db;  
 color: white;  
 padding: 20px 0;  
 text-align: center;  
 border-radius: 8px 8px 0 0;  
 margin-bottom: 30px;  
 box-shadow: 0 4px 6px rgba(0,0,0,0.1);  
 }  
 .header h1 {  
 margin: 0;  
 }  
 .nav {  
 display: flex;  
 justify-content: center;  
 margin-bottom: 30px;  
 flex-wrap: wrap;  
 }  
 .nav a {  
 display: inline-block;  
 padding: 10px 15px;  
 margin: 0 5px 10px 5px;  
 background-color: white;  
 color: #3498db;  
 text-decoration: none;  
 border-radius: 4px;  
 box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
 transition: all 0.3s;  
 }  
 .nav a:hover {  
 background-color: #3498db;  
 color: white;  
 transform: translateY(-2px);  
 box-shadow: 0 4px 6px rgba(0,0,0,0.15);  
 }  
 .card {  
 background-color: white;  
 border-radius: 8px;  
 box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
 padding: 30px;  
 max-width: 500px;  
 margin: 0 auto 30px auto;  
 text-align: center;  
 }  
 .card-icon {  
 font-size: 64px;  
 margin-bottom: 15px;  
 color: #3498db;  
 }  
 .card h2 {  
 color: #2c3e50;  
 margin: 0 0 25px 0;  
 }  
 .form-group {  
 margin-bottom: 20px;  
 text-align: left;  
 }  
 .form-group label {  
 display: block;  
 margin-bottom: 8px;  
 color: #2c3e50;  
 font-weight: 500;  
 }  
 .form-control {  
 width: 100%;  
 padding: 12px;  
 border: 1px solid #ddd;  
 border-radius: 4px;  
 box-sizing: border-box;  
 font-size: 16px;  
 transition: border-color 0.3s;  
 }  
 .form-control:focus {  
 outline: none;  
 border-color: #3498db;  
 box-shadow: 0 0 0 3px rgba(52, 152, 219, 0.2);  
 }  
 .btn {  
 display: inline-block;  
 background-color: #3498db;  
 color: white;  
 padding: 12px 30px;  
 border: none;  
 border-radius: 4px;  
 cursor: pointer;  
 font-size: 16px;  
 font-weight: 500;  
 transition: background-color 0.3s;  
 width: 100%;  
 }  
 .btn:hover {  
 background-color: #2980b9;  
 }  
 .alert {  
 padding: 15px;  
 border-radius: 4px;  
 margin-bottom: 20px;  
 color: white;  
 display: none;  
 }  
 .alert-success {  
 background-color: #2ecc71;  
 }  
 .alert-danger {  
 background-color: #e74c3c;  
 }  
 .footer {  
 text-align: center;  
 margin-top: 40px;  
 padding: 20px;  
 color: #7f8c8d;  
 border-top: 1px solid #e0e0e0;  
 }  
 </style>  
</head>  
<body>  
 <div class="container">  
 <div class="header">  
 <h1>User Management System</h1>  
 </div>  
  
 <div class="nav">  
 <a href="index.jsp">Home</a>  
 <a href="register.jsp">Register</a>  
 <a href="view.jsp">View Users</a>  
 <a href="update.jsp">Update User</a>  
 <a href="delete.jsp">Delete User</a>  
 </div>  
  
 <div class="card">  
 <div class="card-icon">🔄</div>  
 <h2>Update User</h2>  
  
 <div id="message" class="alert"></div>  
  
 <form id="updateForm">  
 <div class="form-group">  
 <label for="oldEmail">Current Email</label>  
 <input type="email" id="oldEmail" class="form-control" placeholder="Enter current email address" required>  
 </div>  
  
 <div class="form-group">  
 <label for="newName">New Name</label>  
 <input type="text" id="newName" class="form-control" placeholder="Enter new name" required>  
 </div>  
  
 <div class="form-group">  
 <label for="newEmail">New Email</label>  
 <input type="email" id="newEmail" class="form-control" placeholder="Enter new email address" required>  
 </div>  
  
 <button type="submit" class="btn">Update User</button>  
 </form>  
 </div>  
  
 <div class="footer">  
 <p>Servlet-based User Management System &copy; 2025</p>  
 </div>  
 </div>  
  
 <script>  
 document.getElementById('updateForm').addEventListener('submit', function(event) {  
 event.preventDefault();  
  
 const oldEmail = document.getElementById('oldEmail').value;  
 const newName = document.getElementById('newName').value;  
 const newEmail = document.getElementById('newEmail').value;  
 const messageDiv = document.getElementById('message');  
  
 fetch('register', {  
 method: 'PUT',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({ oldEmail, name: newName, email: newEmail })  
 })  
 .then(response => response.json())  
 .then(data => {  
 messageDiv.style.display = 'block';  
  
 if (data.message.includes('successfully')) {  
 messageDiv.className = 'alert alert-success';  
 document.getElementById('updateForm').reset();  
 } else {  
 messageDiv.className = 'alert alert-danger';  
 }  
  
 messageDiv.textContent = data.message;  
 })  
 .catch(error => {  
 console.error('Error:', error);  
 messageDiv.style.display = 'block';  
 messageDiv.className = 'alert alert-danger';  
 messageDiv.textContent = 'An error occurred. Please try again.';  
 });  
 });  
 </script>  
</body>  
</html>

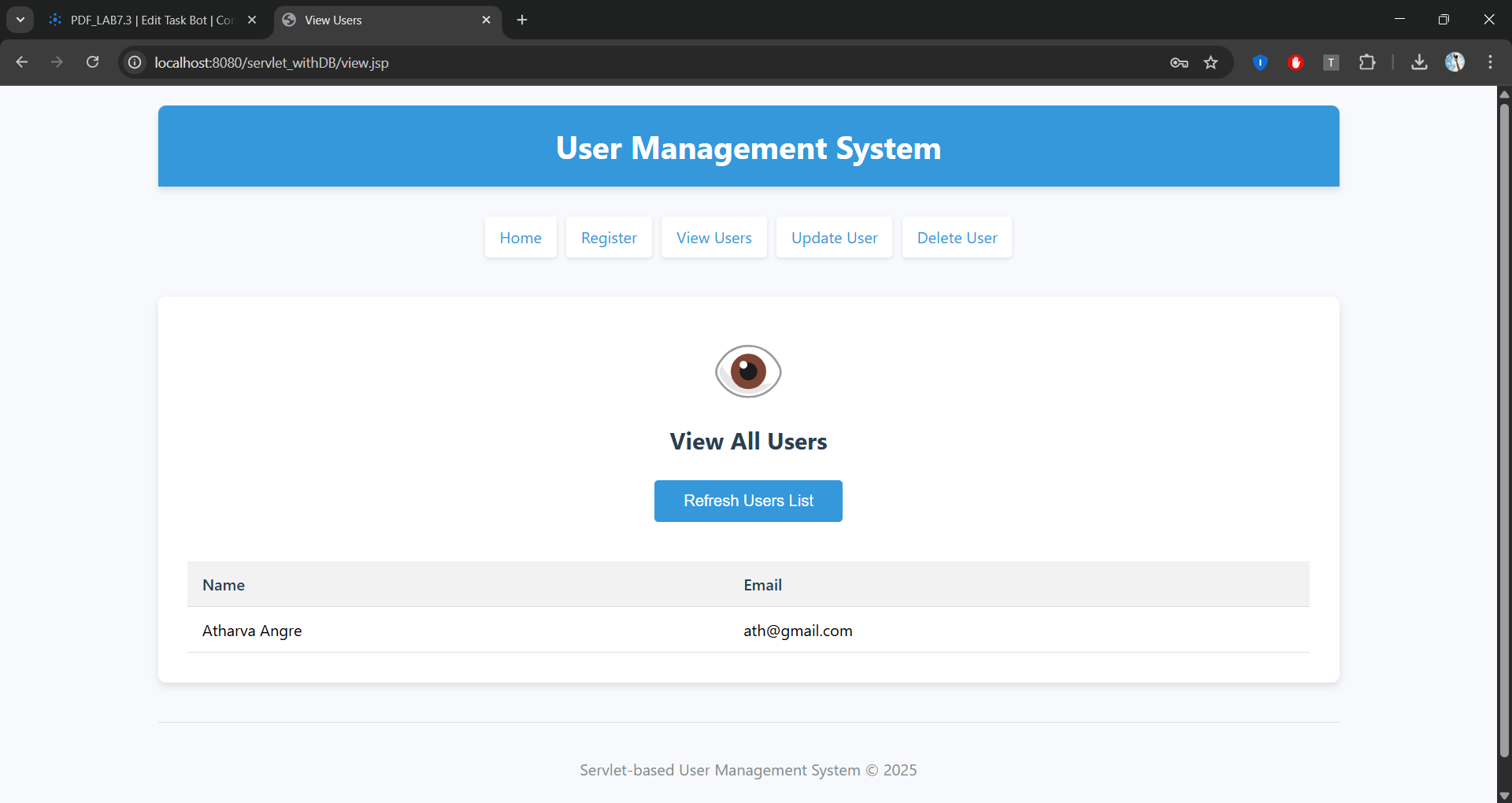
delete.jsp  
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Delete User</title>  
 <style>  
 body {  
 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
 background-color: #f7f9fc;  
 margin: 0;  
 padding: 0;  
 }  
 .container {  
 max-width: 1200px;  
 margin: 0 auto;  
 padding: 20px;  
 }  
 .header {  
 background-color: #3498db;  
 color: white;  
 padding: 20px 0;  
 text-align: center;  
 border-radius: 8px 8px 0 0;  
 margin-bottom: 30px;  
 box-shadow: 0 4px 6px rgba(0,0,0,0.1);  
 }  
 .header h1 {  
 margin: 0;  
 }  
 .nav {  
 display: flex;  
 justify-content: center;  
 margin-bottom: 30px;  
 flex-wrap: wrap;  
 }  
 .nav a {  
 display: inline-block;  
 padding: 10px 15px;  
 margin: 0 5px 10px 5px;  
 background-color: white;  
 color: #3498db;  
 text-decoration: none;  
 border-radius: 4px;  
 box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
 transition: all 0.3s;  
 }  
 .nav a:hover {  
 background-color: #3498db;  
 color: white;  
 transform: translateY(-2px);  
 box-shadow: 0 4px 6px rgba(0,0,0,0.15);  
 }  
 .card {  
 background-color: white;  
 border-radius: 8px;  
 box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
 padding: 30px;  
 max-width: 500px;  
 margin: 0 auto 30px auto;  
 text-align: center;  
 }  
 .card-icon {  
 font-size: 64px;  
 margin-bottom: 15px;  
 color: #3498db;  
 }  
 .card h2 {  
 color: #2c3e50;  
 margin: 0 0 25px 0;  
 }  
 .form-group {  
 margin-bottom: 20px;  
 text-align: left;  
 }  
 .form-group label {  
 display: block;  
 margin-bottom: 8px;  
 color: #2c3e50;  
 font-weight: 500;  
 }  
 .form-control {  
 width: 100%;  
 padding: 12px;  
 border: 1px solid #ddd;  
 border-radius: 4px;  
 box-sizing: border-box;  
 font-size: 16px;  
 transition: border-color 0.3s;  
 }  
 .form-control:focus {  
 outline: none;  
 border-color: #3498db;  
 box-shadow: 0 0 0 3px rgba(52, 152, 219, 0.2);  
 }  
 .btn {  
 display: inline-block;  
 background-color: #3498db;  
 color: white;  
 padding: 12px 30px;  
 border: none;  
 border-radius: 4px;  
 cursor: pointer;  
 font-size: 16px;  
 font-weight: 500;  
 transition: background-color 0.3s;  
 width: 100%;  
 }  
 .btn:hover {  
 background-color: #2980b9;  
 }  
 .delete-btn {  
 background-color: #e74c3c;  
 }  
 .delete-btn:hover {  
 background-color: #c0392b;  
 }  
 .alert {  
 padding: 15px;  
 border-radius: 4px;  
 margin-bottom: 20px;  
 color: white;  
 display: none;  
 }  
 .alert-success {  
 background-color: #2ecc71;  
 }  
 .alert-danger {  
 background-color: #e74c3c;  
 }  
 .footer {  
 text-align: center;  
 margin-top: 40px;  
 padding: 20px;  
 color: #7f8c8d;  
 border-top: 1px solid #e0e0e0;  
 }  
 </style>  
</head>  
<body>  
 <div class="container">  
 <div class="header">  
 <h1>User Management System</h1>  
 </div>  
  
 <div class="nav">  
 <a href="index.jsp">Home</a>  
 <a href="register.jsp">Register</a>  
 <a href="view.jsp">View Users</a>  
 <a href="update.jsp">Update User</a>  
 <a href="delete.jsp">Delete User</a>  
 </div>  
  
 <div class="card">  
 <div class="card-icon">❌</div>  
 <h2>Delete User</h2>  
  
 <div id="message" class="alert"></div>  
  
 <form id="deleteForm">  
 <div class="form-group">  
 <label for="deleteEmail">Email Address</label>  
 <input type="email" id="deleteEmail" class="form-control" placeholder="Enter user email to delete" required>  
 </div>  
  
 <button type="submit" class="btn delete-btn">Delete User</button>  
 </form>  
 </div>  
  
 <div class="footer">  
 <p>Servlet-based User Management System &copy; 2025</p>  
 </div>  
 </div>  
  
 <script>  
 document.getElementById('deleteForm').addEventListener('submit', function(event) {  
 event.preventDefault();  
  
 const email = document.getElementById('deleteEmail').value;  
 const messageDiv = document.getElementById('message');  
  
 // Confirm deletion  
 if (!confirm('Are you sure you want to delete this user? This action cannot be undone.')) {  
 return;  
 }  
  
 fetch('register', {  
 method: 'DELETE',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({ email })  
 })  
 .then(response => response.json())  
 .then(data => {  
 messageDiv.style.display = 'block';  
  
 if (data.message.includes('successfully')) {  
 messageDiv.className = 'alert alert-success';  
 document.getElementById('deleteForm').reset();  
 } else {  
 messageDiv.className = 'alert alert-danger';  
 }  
  
 messageDiv.textContent = data.message;  
 })  
 .catch(error => {  
 console.error('Error:', error);  
 messageDiv.style.display = 'block';  
 messageDiv.className = 'alert alert-danger';  
 messageDiv.textContent = 'An error occurred. Please try again.';  
 });  
 });  
 </script>  
</body>  
</html>

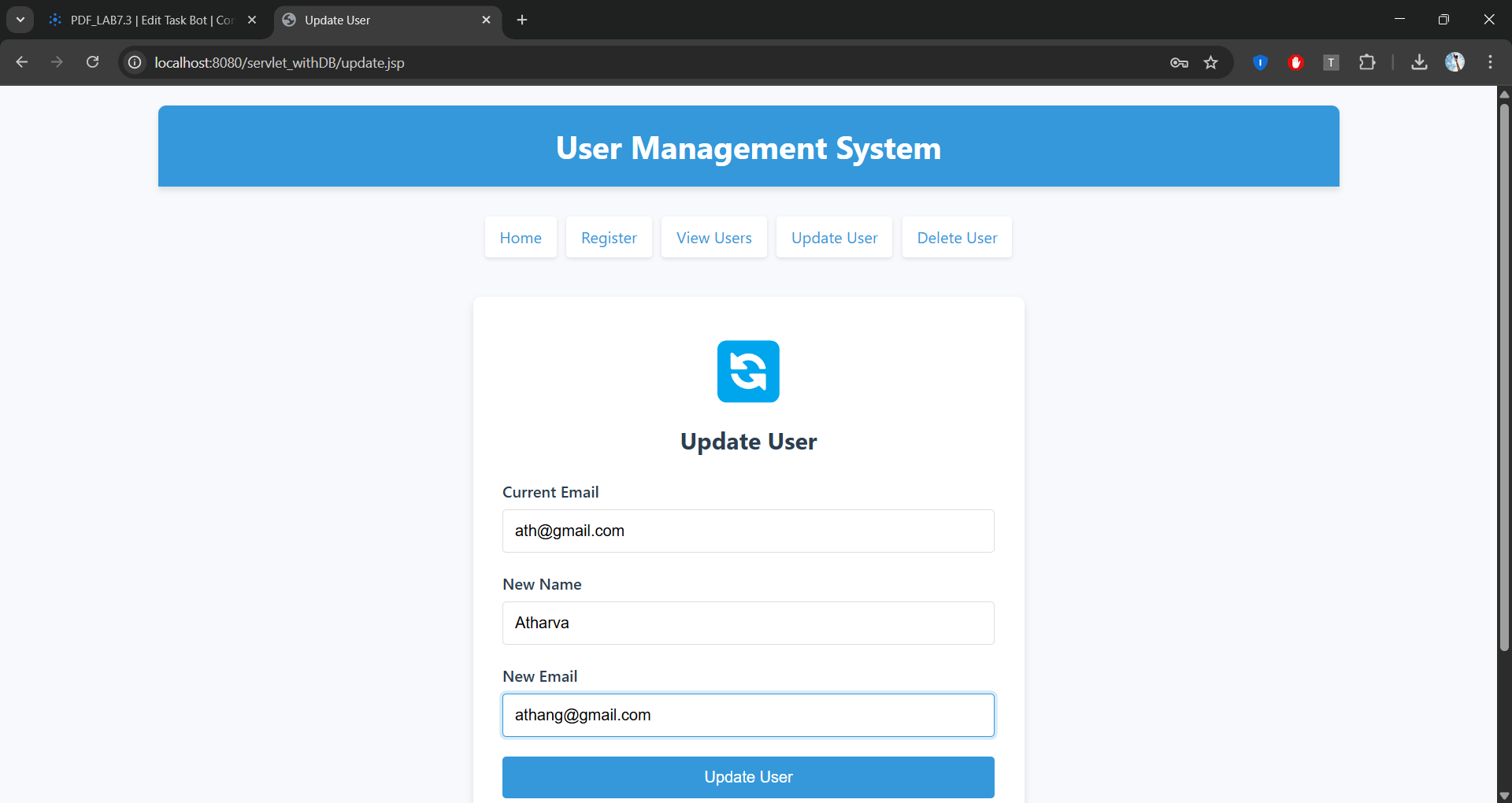
Output:

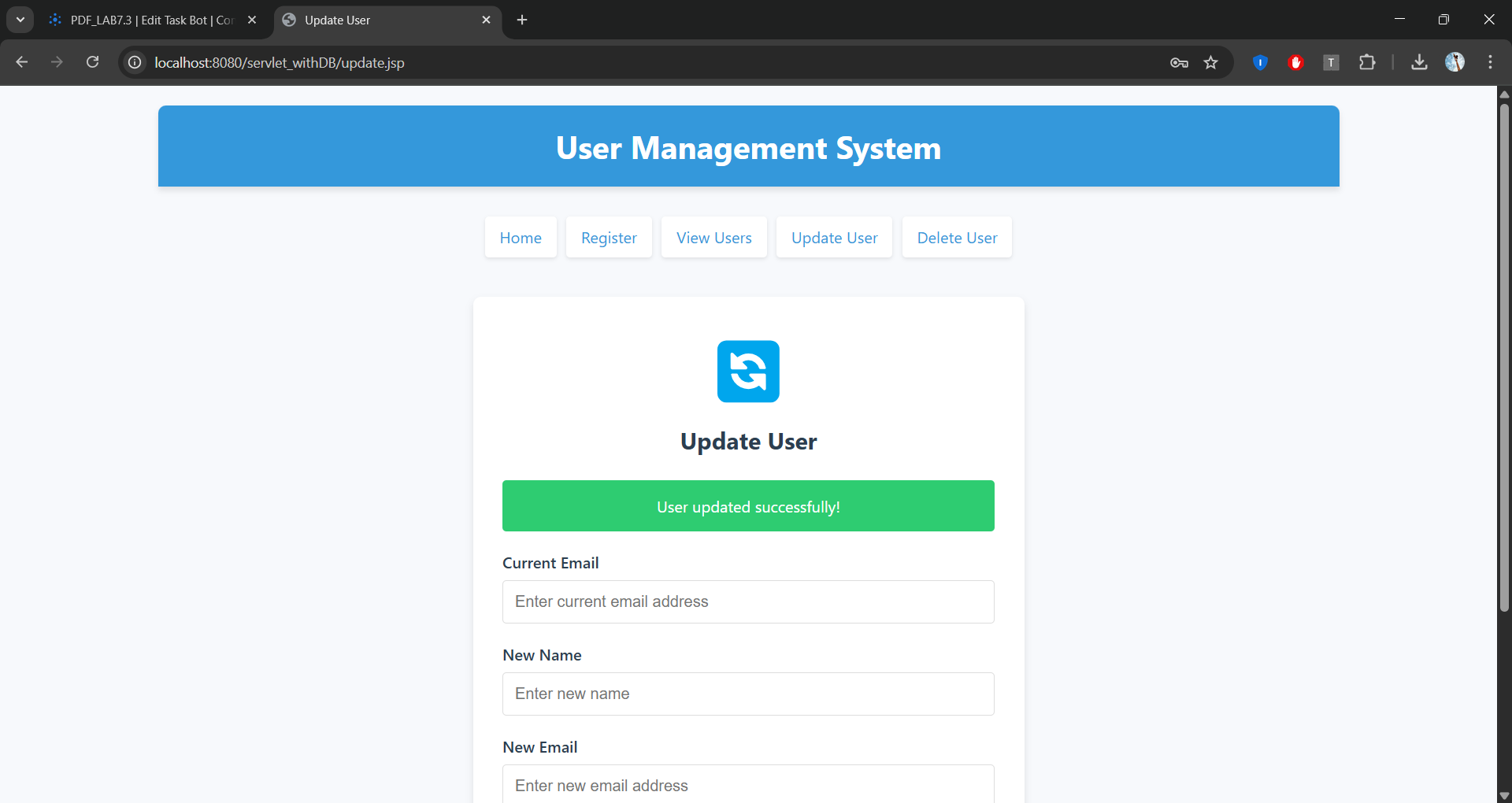


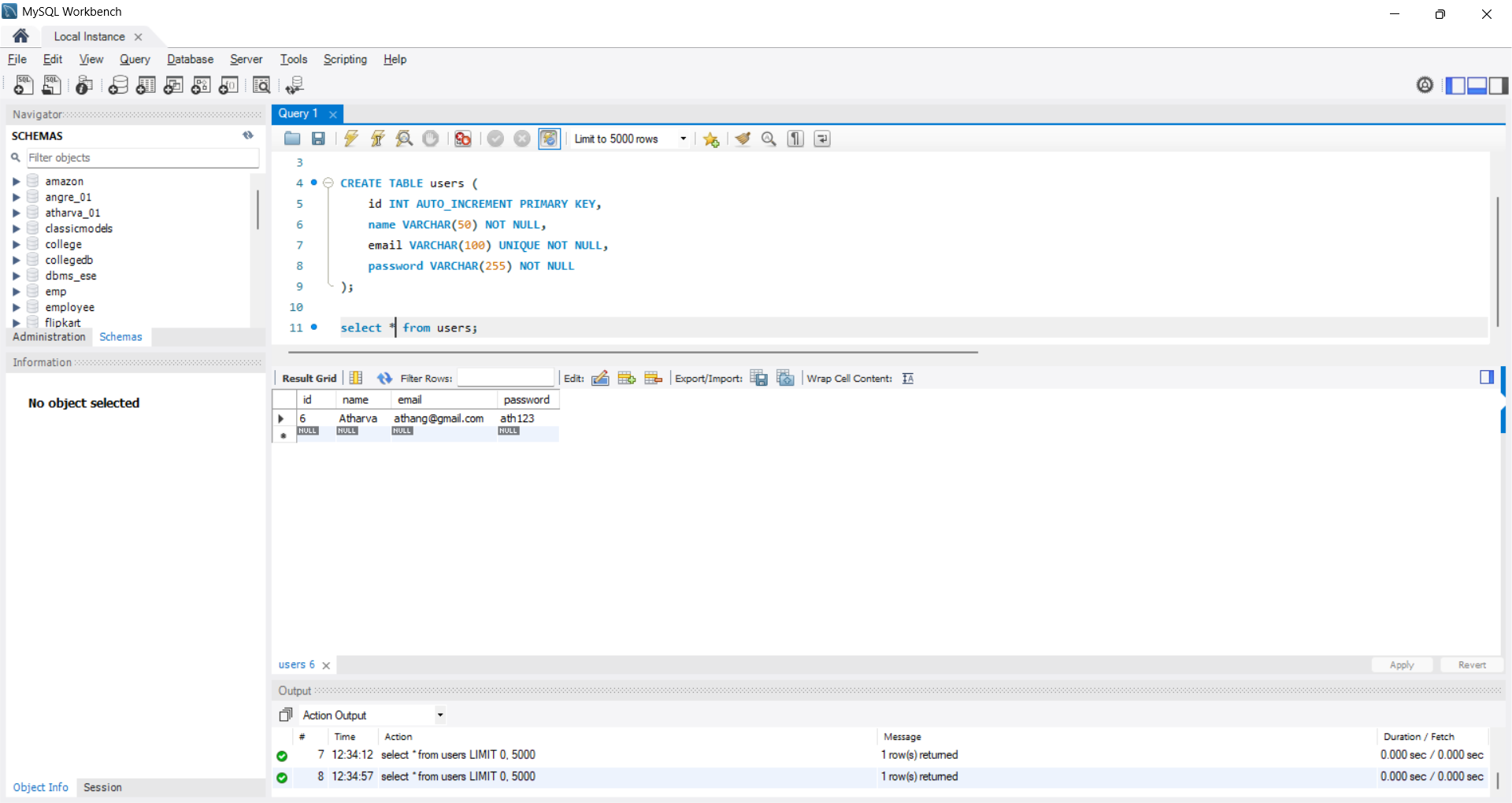


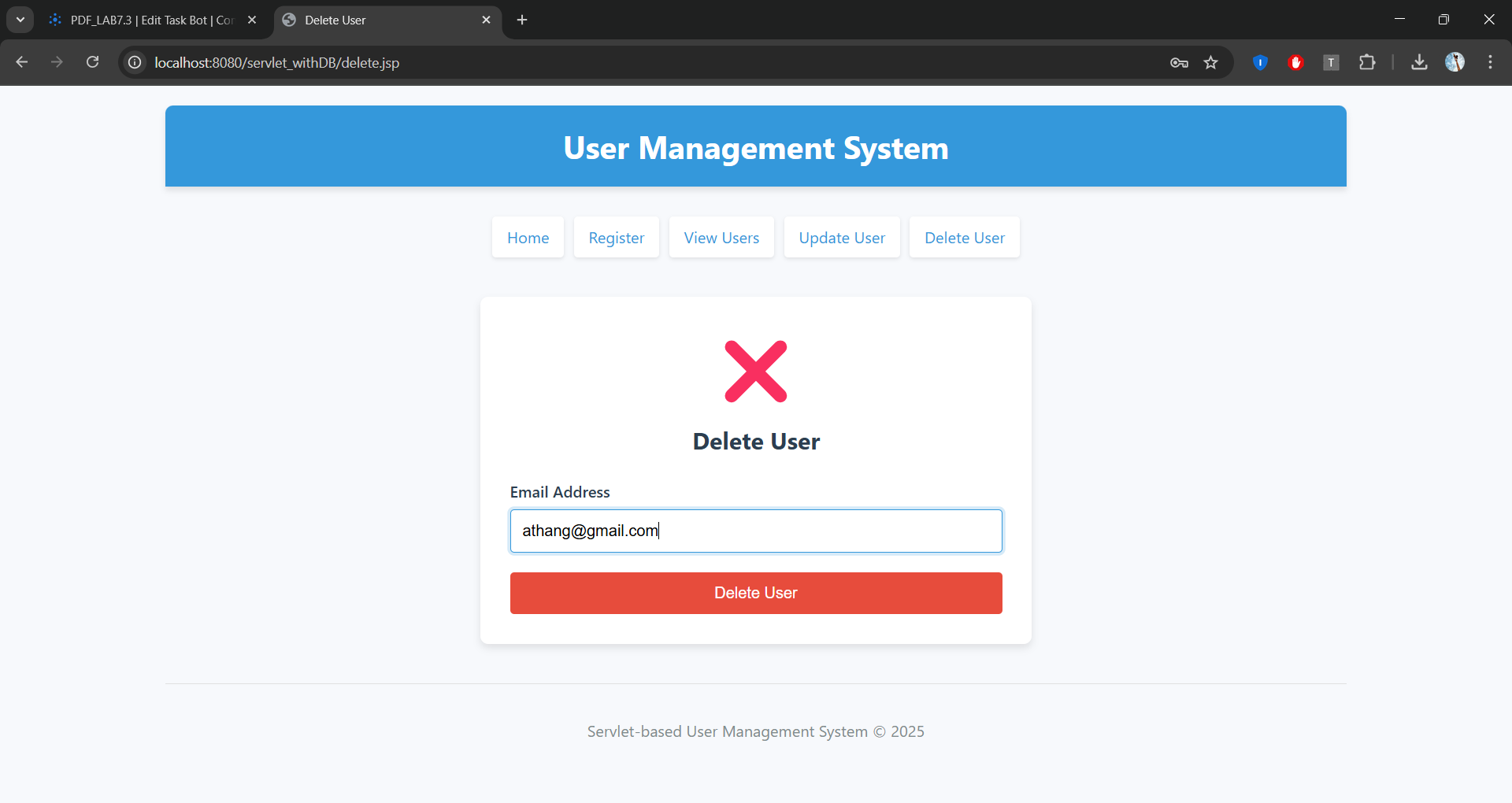


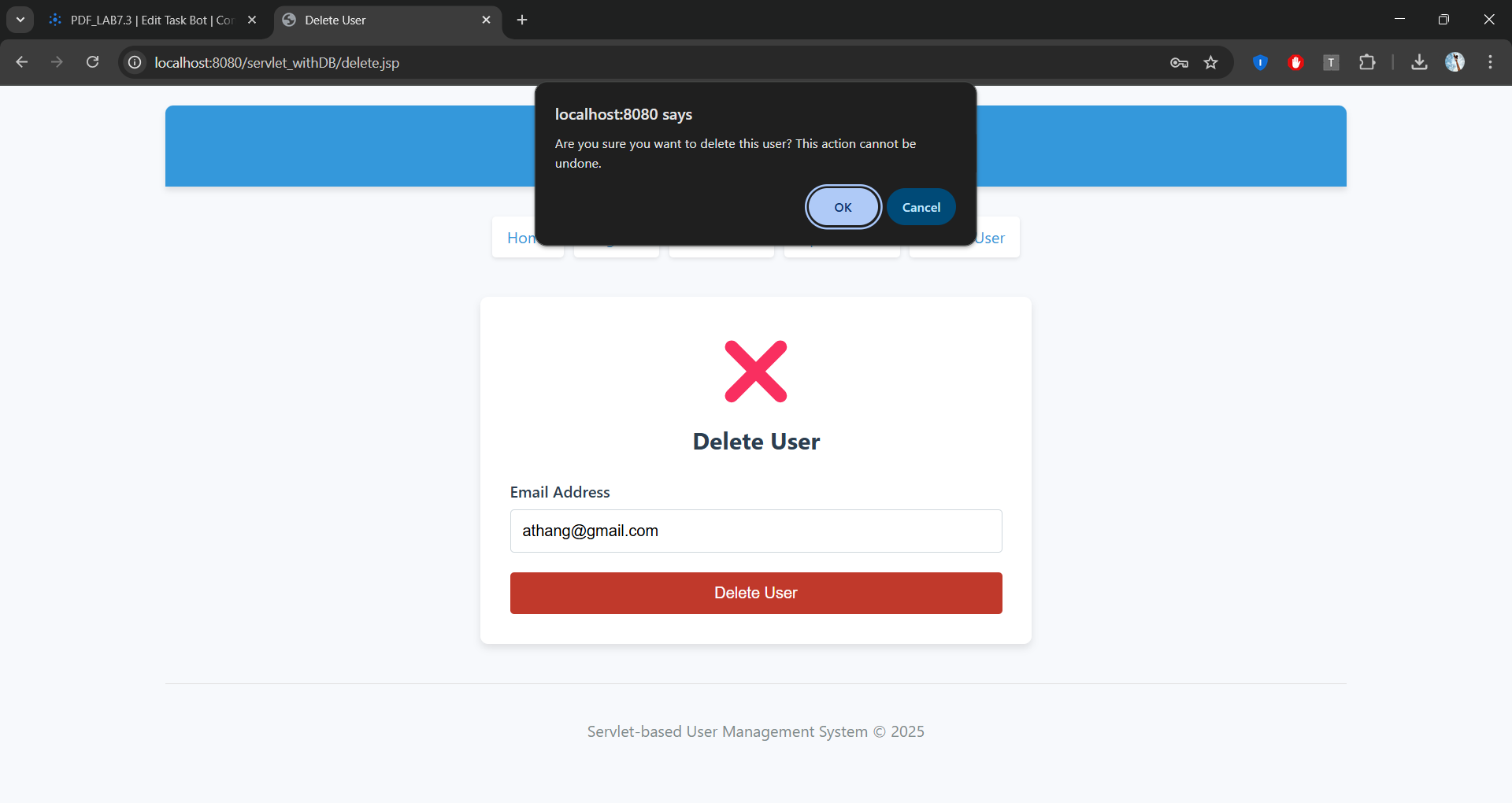


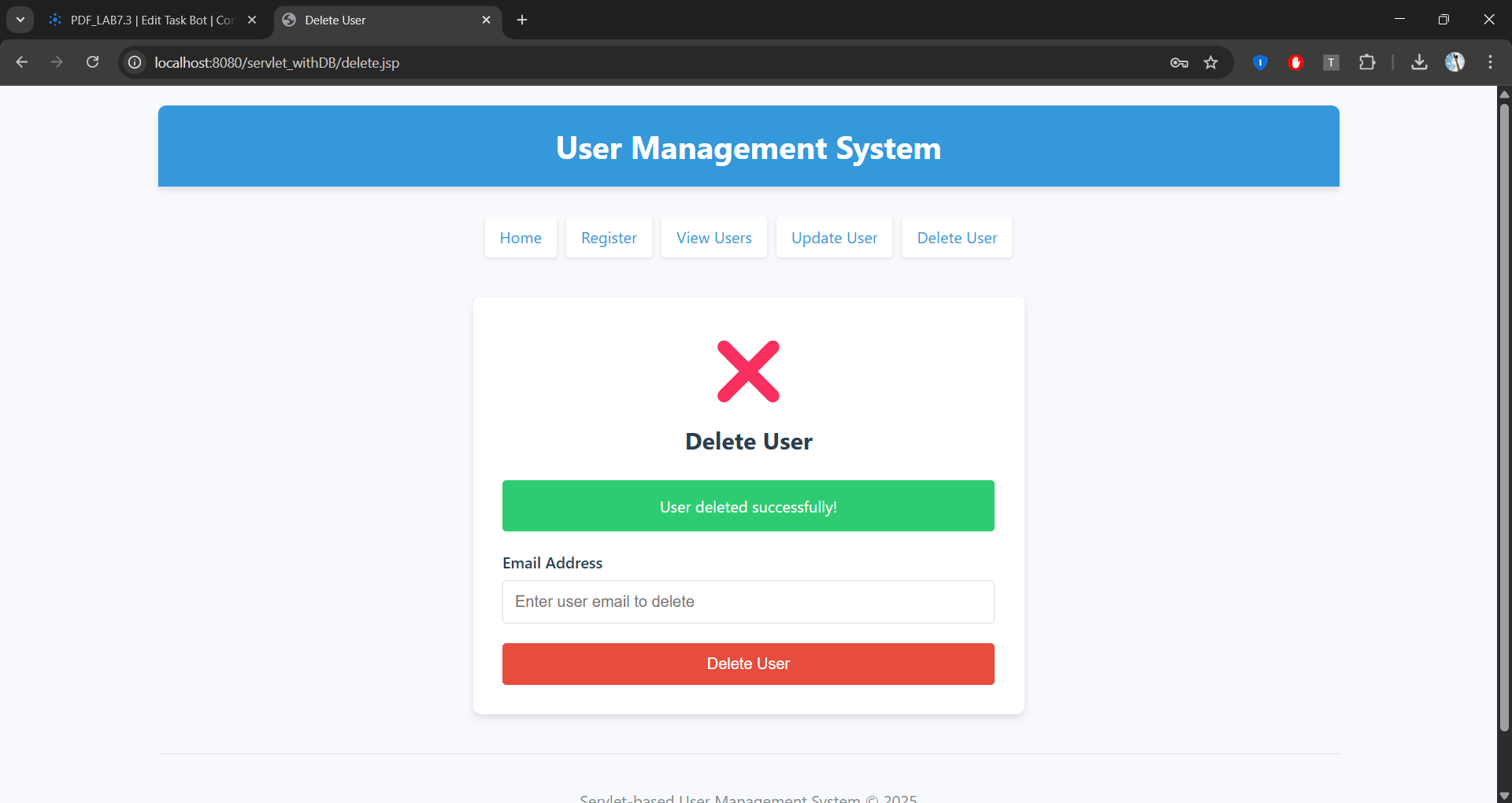


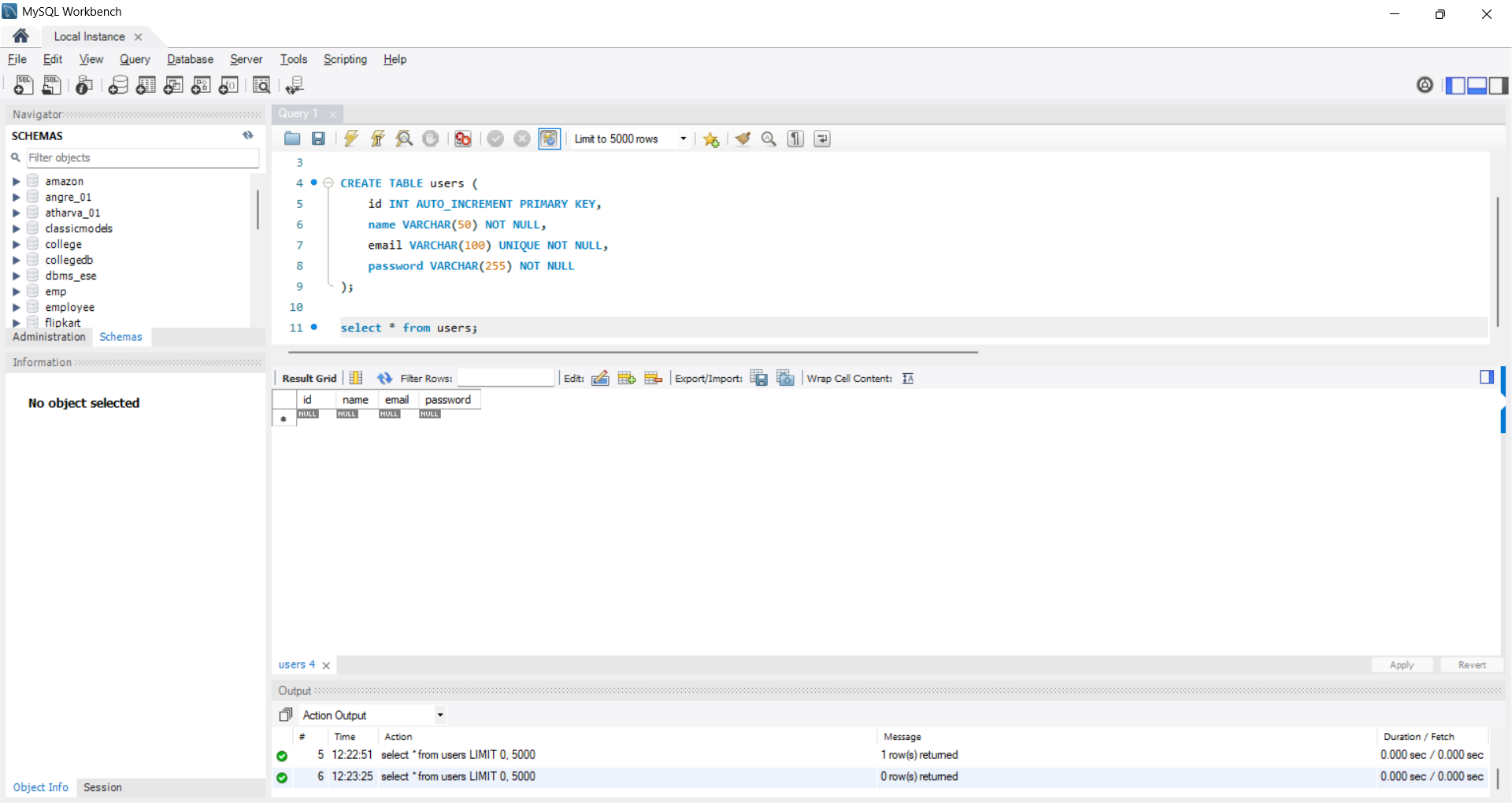












**Observation**:Through these practical sessions, I learned to develop a servlet-based web application using IntelliJ and Maven. I gained experience in integrating MySQL with JDBC for CRUD operations, managing project dependencies, setting up a DBConnectiion class for database connectivity, and mapping servlets correctly in the web.xml file.