ONLINE- MOVIE-TICKET BOOKING-APPLICATION USING JAVA

A simple web-based ticket booking application built using **Java Servlets** and **MySQL**. Users can fill out a form to book tickets, and the backend processes the data, checks availability, stores the booking, and provides a confirmation. This project is a web-based application that allows users to book movie tickets online. The system is built using Java Servlets for server-side logic, JSP for the view layer, and SQL for database management.

**Overview**

The Online Movie Ticket Booking System provides a simple interface for users to:

* View available movies.
* Book tickets by selecting a movie and specifying the number of seats.
* Login or register to manage their bookings.
* Receive a confirmation upon successful booking.

The system integrates with a MySQL database to store user details, movie information, and ticket bookings.

**Technologies Used**

* **Frontend:** HTML, CSS, JSP
* **Backend:** Java Servlets, JSP
* **Database:** MySQL (accessed via JDBC)
* **Server:** Apache Tomcat

**JSP Pages & Code**

**Home Page (home.jsp)**

This page displays the list of available movies and provides a form for users to book tickets.

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Home - Movie Booking</title>

<%@ page import="java.sql.Connection, java.sql.PreparedStatement, java.sql.ResultSet" %>

<%@ page import="tickt.DatabaseConnection" %>

</head>

<body style="background-color: lightblue;">

<h2>Available Movies</h2>

<table border="1">

<tr>

<th>Movie Name</th>

<th>Genre</th>

<th>Duration (mins)</th>

<th>Book</th>

</tr>

<%

Connection conn = null;

PreparedStatement ps = null;

ResultSet rs = null;

try {

conn = DatabaseConnection.getConnection();

ps = conn.prepareStatement("SELECT \* FROM movies");

rs = ps.executeQuery();

while (rs.next()) {

%>

<tr>

<td><%= rs.getString("title") %></td>

<td><%= rs.getString("genre") %></td>

<td><%= rs.getInt("duration") %></td>

<td>

<form action="BookTicketServlet" method="post">

<input type="hidden" name="movie\_id" value="<%= rs.getInt("id") %>">

<label>Seats:</label>

<input type="number" name="seats" min="1" required>

<input type="submit" value="Book Now">

</form>

</td>

</tr>

<%

}

} catch (Exception e) {

out.println("<p>Error: " + e.getMessage() + "</p>");

e.printStackTrace();

} finally {

if (rs != null) rs.close();

if (ps != null) ps.close();

if (conn != null) conn.close();

}

%>

</table>

</body>

</html>

**Login Page (login.jsp)**

The login page authenticates existing users.

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

<link rel="stylesheet" type="text/css" href="css/styles.css">

</head>

<body style="background-color: lightblue;">

<h2>Login</h2>

<% String errorMessage = (String) request.getAttribute("errorMessage"); %>

<% if (errorMessage != null) { %>

<p style="color: red;"><%= errorMessage %></p>

<% } %>

<form action="LoginServlet" method="post">

<label>User name:</label>

<input type="text" name="username" required>

<br><br>

<label>Password:</label>

<input type="password" name="password" required>

<br><br>

<input type="submit" value="Login">

</form>

</body>

</html>

**Registration Page (register.jsp)**

New users can register on this page.

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Register</title>

</head>

<body style="background-color: lightblue;">

<h2>User Registration</h2>

<form action="RegisterServlet" method="post">

<label>User name:</label>

<input type="text" name="username" required>

<br>

<label>Password:</label>

<input type="password" name="password" required>

<br>

<input type="submit" value="Register">

</form>

<p>Already have an account? <a href="login.jsp">Login here</a></p>

</body>

</html>

**Booking Confirmation (confirmation.jsp)**

This page displays a confirmation message once the booking is successful.

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<title>Booking Confirmation!</title>

</head>

<body style="background-color: lightblue;">

<h2>Booking Confirmed!</h2>

<p>Your movie tickets have been successfully booked. ENJOY YOUR DAY!</p>

<a href="home.jsp">Back to Home</a>

</body>

</html>

**Database Setup**

Below is the SQL script used to set up the database for the project. Create a database named moviedb and run the following commands:

-- Create database and use it

create database moviedb;

use moviedb;

-- User table (for login)

create table users(

id int auto\_increment primary key,

username varchar(50),

password varchar(255)

);

-- Movie table (list of movies)

create table movie(

id int auto\_increment primary key,

movie\_name varchar(100),

show\_time varchar(50),

total\_seats int

);

-- Ticket table (booking record)

create table tickets(

id int auto\_increment primary key,

user\_id int,

movie\_id int,

seat\_number int,

foreign key(user\_id) references users(id),

foreign key(movie\_id) references movie(id)

);

-- Insert sample data into users table

INSERT INTO userses (username, password)

VALUES

('admin', 'admin123'),

('john\_doe', 'john2025'),

('ajith', 'ajith123'),

('vijay', 'vijay@123'),

('priya\_s', 'priya456');

-- Insert sample data into movie table

insert into movie(movie\_name, show\_time, total\_seats)

values('AVENGERS: ENDGAME', '7:00 PM', 60),

('MERSAL', '9:00 PM', 20);

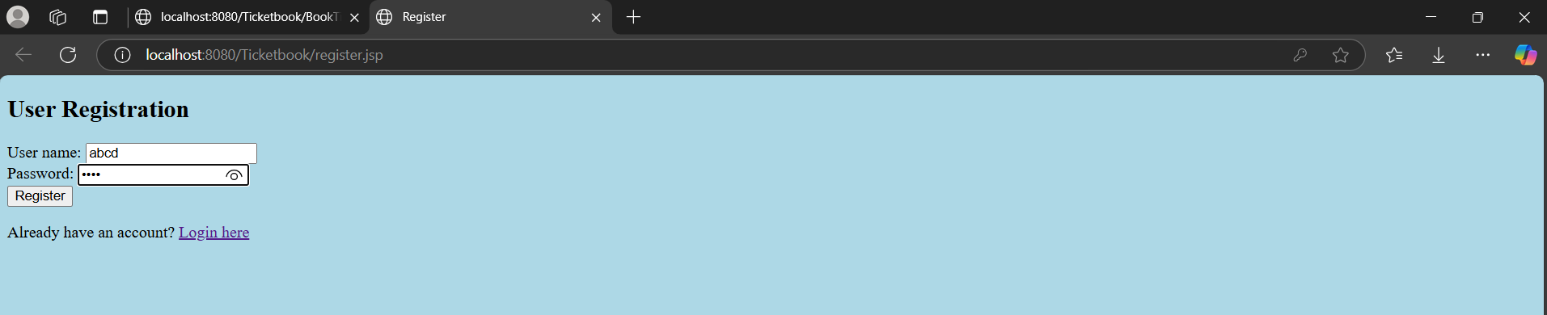
-- To view the inserted data:

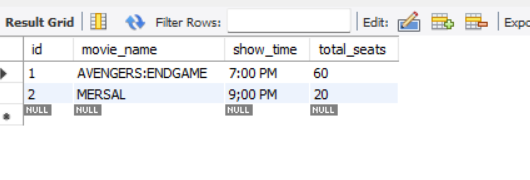
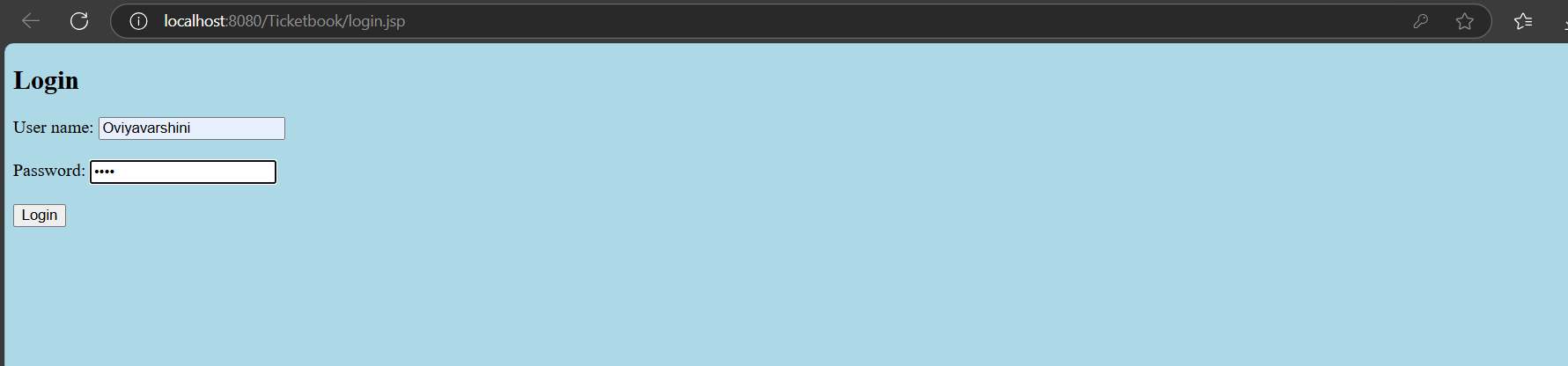
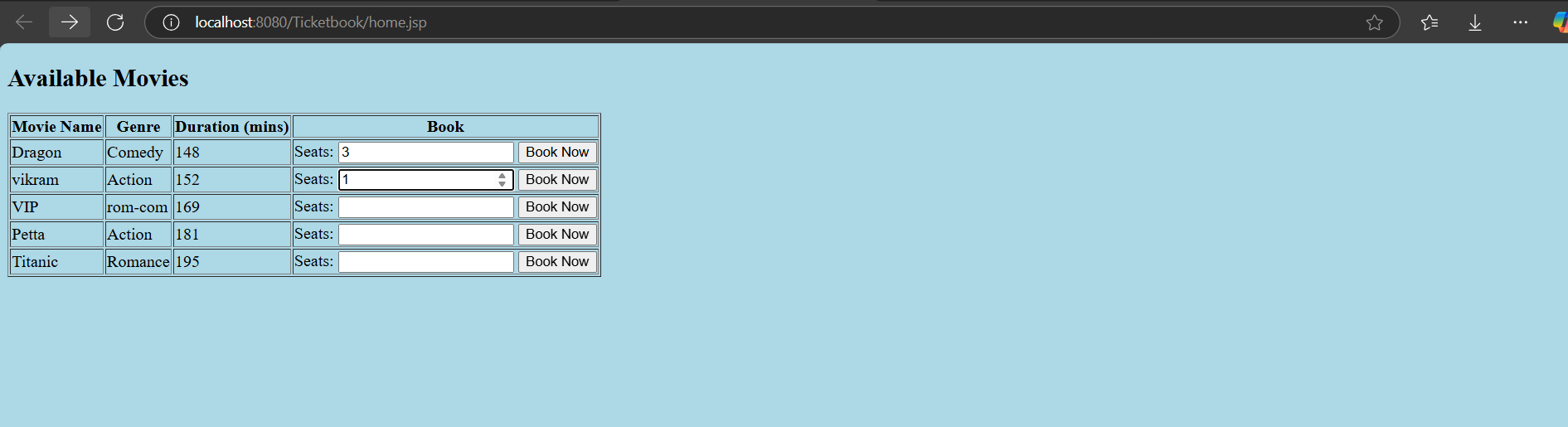
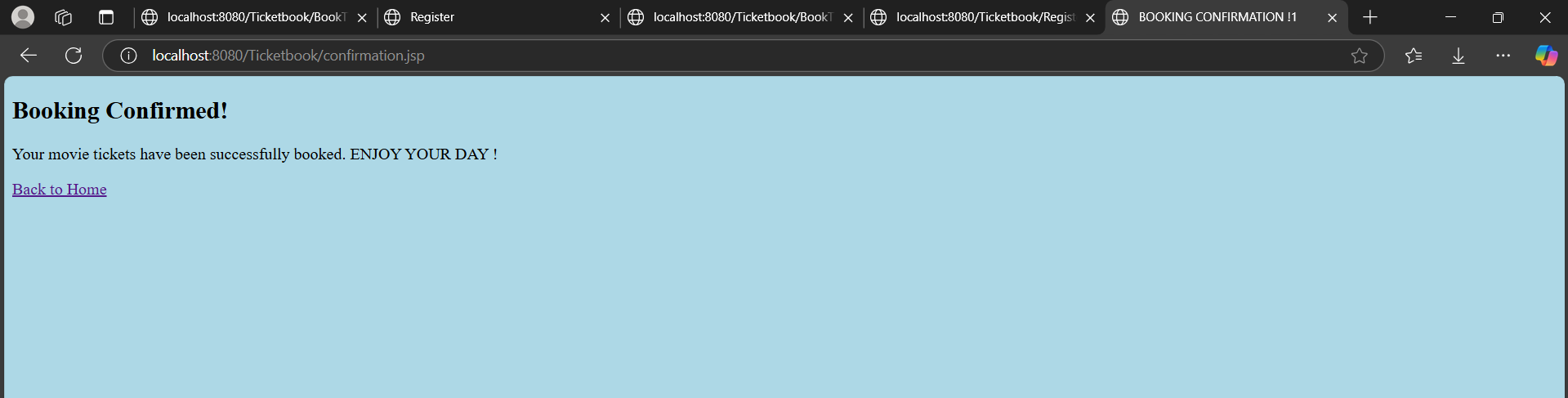
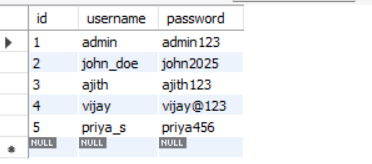
select \* from users;

select \* from movie;

-- To remove the database

-- drop database moviedb;

**PROJECT OUTPUT:**



**Conclusion:**

This project demonstrates the integration of Java Servlets with JSP and MySQL to create a dynamic, interactive ticket booking application. The application covers user registration, login, movie listing, booking, and confirmation functionalities, making it a comprehensive example of a full-stack Java web application.