CODES

## DATABASE

CREATE DATABASE MovieRental;

USE MovieRental;

-- Create 'movies' table

CREATE TABLE movies (

movie\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255) NOT NULL,

genre VARCHAR(100) NOT NULL,

release\_year INT,

director VARCHAR(255),

rating DECIMAL(3, 1)

);

-- Create 'users' table

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL UNIQUE,

role ENUM('user', 'admin') DEFAULT 'user'

);

-- Create 'rentals' table

CREATE TABLE rentals (

rental\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

movie\_id INT,

rental\_date DATE,

return\_date DATE,

FOREIGN KEY (user\_id) REFERENCES users(user\_id),

FOREIGN KEY (movie\_id) REFERENCES movies(movie\_id)

);

## .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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>rental</groupId>

<artifactId>MovieRental</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<dependencies>

<dependency>

<groupId>javax.ws.rs</groupId>

<artifactId>jsr311-api</artifactId>

<version>1.1.1</version>

</dependency>

<dependency>

<groupId>com.sun.jersey</groupId>

<artifactId>jersey-servlet</artifactId>

<version>1.17.1</version>

</dependency>

<dependency>

<groupId>org.glassfish.jaxb</groupId>

<artifactId>jaxb-runtime</artifactId>

<version>2.3.2</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.25</version>

</dependency>

<dependency>

<groupId>javax.json</groupId>

<artifactId>javax.json-api</artifactId>

<version>1.1.4</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.4.0</version>

<scope>system</scope>

<systemPath>C:/Users/trini/eclipse-workspace/Rental/mysql-connector-j-8.4.0.jar</systemPath>

</dependency>

</dependencies>

</project>

<?**xml** version=*"1.0"* encoding=*"UTF-8"*?>

<**web-app** xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* version=*"2.5"*>

<**servlet**>

<**servlet-name**>Jersey REST Service</**servlet-name**>

<**servlet-class**>com.sun.jersey.spi.container.servlet.ServletContainer</**servlet-class**>

<**init-param**>

<**param-name**>com.sun.jersey.config.property.packages</**param-name**>

<**param-value**>rental</**param-value**>

</**init-param**>

</**servlet**>

<**servlet-mapping**>

<**servlet-name**>Jersey REST Service</**servlet-name**>

<**url-pattern**>/block/\*</**url-pattern**>

</**servlet-mapping**>

</**web-app**>

## JAVA

package rental;

public class loginresponse {

private boolean success;

private String message;

private user user;

public loginresponse() {

}

public loginresponse(boolean success, String message, user user) {

this.success = success;

this.message = message;

this.user = user;

}

public boolean isSuccess() {

return success;

}

public void setSuccess(boolean success) {

this.success = success;

}

public String getMessage() {

return message;

}

public void setMessage(String message) {

this.message = message;

}

public user getUser() {

return user;

}

public void setUser(user user) {

this.user = user;

}

}

package rental;

import rental.util.DBUtil;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.Consumes;

import javax.ws.rs.core.MediaType;

import javax.json.Json;

import javax.json.JsonObject;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

@Path("/login")

public class loginServlet extends HttpServlet {

@POST

@Consumes(MediaType.APPLICATION\_JSON)

@Produces(MediaType.APPLICATION\_JSON)

public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {

JsonObject jsonObject = Json.createReader(request.getReader()).readObject();

String username = jsonObject.getString("username");

String password = jsonObject.getString("password");

JsonObject responseJson;

try (Connection conn = DBUtil.getConnection()) {

String query = "SELECT \* FROM users WHERE username = ? AND password = ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1, username);

stmt.setString(2, password);

ResultSet rs = stmt.executeQuery();

if (rs.next()) {

user user = new user();

user.setUserId(rs.getInt("user\_id"));

user.setUsername(rs.getString("username"));

user.setEmail(rs.getString("email"));

responseJson = Json.createObjectBuilder()

.add("success", true)

.add("message", "Login successful")

.add("user", Json.createObjectBuilder()

.add("userId", user.getUserId())

.add("username", user.getUsername())

.add("email", user.getEmail()))

.build();

} else {

responseJson = Json.createObjectBuilder()

.add("success", false)

.add("message", "Invalid username or password")

.build();

}

} catch (SQLException e) {

responseJson = Json.createObjectBuilder()

.add("success", false)

.add("message", "An error occurred: " + e.getMessage())

.build();

}

response.setContentType("application/json");

response.getWriter().write(responseJson.toString());

}

}

package rental;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

import javax.json.Json;

import javax.json.JsonObject;

import java.io.IOException;

@Path("/home")

public class mainservlet extends HttpServlet {

@GET

@Produces(MediaType.APPLICATION\_JSON)

public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {

JsonObject responseJson = Json.createObjectBuilder()

.add("message", "Welcome to the Movie Rental Platform")

.build();

response.setContentType("application/json");

response.getWriter().write(responseJson.toString());

}

}

package rental;

public class movies {

private int movieId;

private String title;

private String genre;

private int releaseYear;

public movies() {

}

public movies(int movieId, String title, String genre, int releaseYear) {

this.movieId = movieId;

this.title = title;

this.genre = genre;

this.releaseYear = releaseYear;

}

public int getMovieId() {

return movieId;

}

public void setMovieId(int movieId) {

this.movieId = movieId;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getGenre() {

return genre;

}

public void setGenre(String genre) {

this.genre = genre;

}

public int getReleaseYear() {

return releaseYear;

}

public void setReleaseYear(int releaseYear) {

this.releaseYear = releaseYear;

}

}

package rental;

import rental.util.DBUtil;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.Consumes;

import javax.ws.rs.core.MediaType;

import javax.json.Json;

import javax.json.JsonObject;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

@Path("/register")

public class registrationServlet extends HttpServlet {

@POST

@Consumes(MediaType.APPLICATION\_JSON)

@Produces(MediaType.APPLICATION\_JSON)

public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {

JsonObject jsonObject = Json.createReader(request.getReader()).readObject();

String username = jsonObject.getString("username");

String password = jsonObject.getString("password");

String email = jsonObject.getString("email");

JsonObject responseJson;

try (Connection conn = DBUtil.getConnection()) {

String query = "INSERT INTO users (username, password, email) VALUES (?, ?, ?)";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1, username);

stmt.setString(2, password);

stmt.setString(3, email);

int result = stmt.executeUpdate();

if (result > 0) {

responseJson = Json.createObjectBuilder()

.add("success", true)

.add("message", "Registration successful")

.build();

} else {

responseJson = Json.createObjectBuilder()

.add("success", false)

.add("message", "Registration failed")

.build();

}

} catch (SQLException e) {

responseJson = Json.createObjectBuilder()

.add("success", false)

.add("message", "An error occurred: " + e.getMessage())

.build();

}

response.setContentType("application/json");

response.getWriter().write(responseJson.toString());

}

}

package rental;

import rental.util.DBUtil;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.Consumes;

import javax.ws.rs.core.MediaType;

import javax.json.Json;

import javax.json.JsonObject;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

*@Path*("/rentMovie")

public class rentmoviesservlet extends HttpServlet {

*@POST*

*@Consumes*(MediaType.***APPLICATION\_JSON***)

*@Produces*(MediaType.***APPLICATION\_JSON***)

public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {

JsonObject jsonObject = Json.*createReader*(request.getReader()).readObject();

int userId = jsonObject.getInt("userId");

int movieId = jsonObject.getInt("movieId");

JsonObject responseJson;

try (Connection conn = DBUtil.*getConnection*()) {

String query = "INSERT INTO rentals (user\_id, movie\_id, rental\_date) VALUES (?, ?, NOW())";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setInt(1, userId);

stmt.setInt(2, movieId);

int result = stmt.executeUpdate();

if (result > 0) {

responseJson = Json.*createObjectBuilder*()

.add("success", true)

.add("message", "Movie rented successfully")

.build();

} else {

responseJson = Json.*createObjectBuilder*()

.add("success", false)

.add("message", "Failed to rent movie")

.build();

}

} catch (SQLException e) {

responseJson = Json.*createObjectBuilder*()

.add("success", false)

.add("message", "An error occurred: " + e.getMessage())

.build();

}

response.setContentType("application/json");

response.getWriter().write(responseJson.toString());

}

}

package rental;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.ws.rs.GET;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.Consumes;

import javax.ws.rs.core.MediaType;

import javax.json.Json;

import javax.json.JsonArrayBuilder;

import javax.json.JsonObject;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

@Path("/search")

public class searchmoviesservlet extends HttpServlet {

@POST

@Consumes(MediaType.APPLICATION\_JSON)

@Produces(MediaType.APPLICATION\_JSON)

public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {

JsonObject jsonObject = Json.createReader(request.getReader()).readObject();

String searchTerm = jsonObject.getString("searchTerm");

JsonArrayBuilder arrayBuilder = Json.createArrayBuilder();

try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/MovieRental", "root", "password");

String query = "SELECT \* FROM movies WHERE title LIKE ? OR genre LIKE ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setString(1, "%" + searchTerm + "%");

stmt.setString(2, "%" + searchTerm + "%");

ResultSet rs = stmt.executeQuery();

while (rs.next()) {

JsonObject movieJson = Json.createObjectBuilder()

.add("movieId", rs.getInt("movie\_id"))

.add("title", rs.getString("title"))

.add("genre", rs.getString("genre"))

.add("releaseYear", rs.getInt("release\_year"))

.build();

arrayBuilder.add(movieJson);

}

conn.close();

} catch (Exception e) {

e.printStackTrace();

}

response.setContentType("application/json");

response.getWriter().write(arrayBuilder.build().toString());

}

}

package rental;

public class user {

private int userId;

private String username;

private String password;

private String email;

public user() {

}

public user(int userId, String username, String password, String email) {

this.userId = userId;

this.username = username;

this.password = password;

this.email = email;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

}

package rental;

import rental.util.DBUtil;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

import javax.json.Json;

import javax.json.JsonArrayBuilder;

import javax.json.JsonObject;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

*@Path*("/viewAllMovies")

public class viewallservlet extends HttpServlet {

*@GET*

*@Produces*(MediaType.***APPLICATION\_JSON***)

public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {

JsonArrayBuilder moviesArrayBuilder = Json.*createArrayBuilder*();

JsonObject responseJson;

try (Connection conn = DBUtil.*getConnection*()) {

String query = "SELECT \* FROM movies";

PreparedStatement stmt = conn.prepareStatement(query);

ResultSet rs = stmt.executeQuery();

while (rs.next()) {

JsonObject movieJson = Json.*createObjectBuilder*()

.add("movieId", rs.getInt("movie\_id"))

.add("title", rs.getString("title"))

.add("genre", rs.getString("genre"))

.add("releaseYear", rs.getInt("release\_year"))

.build();

moviesArrayBuilder.add(movieJson);

}

responseJson = Json.*createObjectBuilder*()

.add("success", true)

.add("movies", moviesArrayBuilder)

.build();

} catch (SQLException e) {

responseJson = Json.*createObjectBuilder*()

.add("success", false)

.add("message", "An error occurred: " + e.getMessage())

.build();

}

response.setContentType("application/json");

response.getWriter().write(responseJson.toString());

}

}

package rental;

import rental.util.DBUtil;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.ws.rs.GET;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

import javax.json.Json;

import javax.json.JsonArrayBuilder;

import javax.json.JsonObject;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

*@Path*("/viewRentedMovies")

public class viewrentedmoviesservlet extends HttpServlet {

*@GET*

*@Produces*(MediaType.***APPLICATION\_JSON***)

public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {

int userId = Integer.*parseInt*(request.getParameter("userId"));

JsonArrayBuilder rentedMoviesArrayBuilder = Json.*createArrayBuilder*();

JsonObject responseJson;

try (Connection conn = DBUtil.*getConnection*()) {

String query = "SELECT m.movie\_id, m.title, m.genre, m.release\_year, r.rental\_date " +

"FROM rentals r JOIN movies m ON r.movie\_id = m.movie\_id WHERE r.user\_id = ?";

PreparedStatement stmt = conn.prepareStatement(query);

stmt.setInt(1, userId);

ResultSet rs = stmt.executeQuery();

while (rs.next()) {

JsonObject movieJson = Json.*createObjectBuilder*()

.add("movieId", rs.getInt("movie\_id"))

.add("title", rs.getString("title"))

.add("genre", rs.getString("genre"))

.add("releaseYear", rs.getInt("release\_year"))

.add("rentalDate", rs.getDate("rental\_date").toString())

.build();

rentedMoviesArrayBuilder.add(movieJson);

}

responseJson = Json.*createObjectBuilder*()

.add("success", true)

.add("rentedMovies", rentedMoviesArrayBuilder)

.build();

} catch (SQLException e) {

responseJson = Json.*createObjectBuilder*()

.add("success", false)

.add("message", "An error occurred: " + e.getMessage())

.build();

}

response.setContentType("application/json");

response.getWriter().write(responseJson.toString());

}

}

package rental.util;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBUtil {

private static final String URL = "jdbc:mysql://localhost:3306/MovierRental";

private static final String USER = "root";

private static final String PASSWORD = "nth@bi";

public static Connection getConnection() throws SQLException {

try {

Class.forName("com.mysql.cj.jdbc.Driver");

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

return DriverManager.getConnection(URL, USER, PASSWORD);

}

}

## HTML

<!**DOCTYPE** html>

<**html** lang=*"en"*>

<**head**>

<**meta** charset=*"UTF-8"*>

<**meta** name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>

<**title**>Movie Rental</**title**>

<**style**>

body {

font-family: *Arial, sans-serif*;

margin: *20px*;

}

h1 {

color: *#333*;

}

form {

max-width: *300px*;

margin: *0 auto*;

}

label {

display: *block*;

margin-bottom: *5px*;

}

input[type="text"], input[type="password"] {

width: *100%*;

padding: *8px*;

margin-bottom: *10px*;

box-sizing: *border-box*;

}

button {

padding: *10px 20px*;

background-color: *#007bff*;

color: *#fff*;

border: *none*;

cursor: *pointer*;

}

button:hover {

background-color: *#0056b3*;

}

</**style**>

</**head**>

<**body**>

<**h1**>Login</**h1**>

<**form** id=*"loginForm"*>

<**label** for=*"username"*>Username:</**label**>

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

<**label** for=*"password"*>Password:</**label**>

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

<**button** type=*"submit"*>Login</**button**>

</**form**>

<**a** href=*"main.html"*>Back to Main</**a**>

<**div** id=*"message"*></**div**>

<**script**>

document.getElementById('loginForm').addEventListener('submit', **function**(event) {

event.preventDefault();

**const** formData = {

username: document.getElementById('username').value,

password: document.getElementById('password').value

};

fetch('/rental/block/login', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(formData)

})

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

.then(data => {

**if** (data.success) {

window.location.href = '/rental/main.html';

} **else** {

document.getElementById('message').textContent = data.message;

}

})

.**catch**(error => console.error('Error logging in:', error));

});

</**script**>

</**body**>

</**html**>

<!**DOCTYPE** html>

<**html** lang=*"en"*>

<**head**>

<**meta** charset=*"UTF-8"*>

<**meta** name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>

<**title**>Movie Rental</**title**>

<**style**>

body {

font-family: *Arial, sans-serif*;

margin: *20px*;

}

h1 {

color: *#333*;

}

nav {

margin-bottom: *20px*;

}

nav a:hover {

background-color: *#0056b3*;

}

</**style**>

</**head**>

<**body**>

<**h1**>Welcome to the Movie Rental Platform</**h1**>

<**nav**>

<**ul**>

<**li**><**a** href=*"registration.html"*>Register</**a**></**li**>

<**li**><**a** href=*"login.html"*>Login</**a**></**li**>

<**li**><**a** href=*"ViewAllMovies.html"*>View All Movies</**a**></**li**>

<**li**><**a** href=*"RentMovie.html"*>Rent a Movie</**a**></**li**>

<**li**><**a** href=*"ViewRentedMovies.html"*>View Rented Movies</**a**></**li**>

<**li**><**a** href=*"SearchMovies.html"*>Search Movies</**a**></**li**>

</**ul**>

</**nav**>

<**p**>Welcome to the Movie Rental Platform. You can navigate using the links above.</**p**>

</**body**>

</**html**>

<!**DOCTYPE** html>

<**html** lang=*"en"*>

<**head**>

<**meta** charset=*"UTF-8"*>

<**meta** name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>

<**title**>Movie Rental</**title**>

<**style**>

body {

font-family: *Arial, sans-serif*;

margin: *20px*;

}

h1 {

color: *#333*;

}

form {

max-width: *300px*;

margin: *0 auto*;

}

label {

display: *block*;

margin-bottom: *5px*;

}

input[type="text"], input[type="password"], input[type="email"] {

width: *100%*;

padding: *8px*;

margin-bottom: *10px*;

box-sizing: *border-box*;

}

button {

padding: *10px 20px*;

background-color: *#007bff*;

color: *#fff*;

border: *none*;

cursor: *pointer*;

}

button:hover {

background-color: *#0056b3*;

}

</**style**>

</**head**>

<**body**>

<**h1**>Registration</**h1**>

<**form** id=*"registrationForm"*>

<**label** for=*"username"*>Username:</**label**>

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

<**label** for=*"password"*>Password:</**label**>

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

<**label** for=*"email"*>Email:</**label**>

<**input** type=*"email"* id=*"email"* name=*"email"* required>

<**button** type=*"submit"*>Register</**button**>

</**form**>

<**div** id=*"message"*></**div**>

<**a** href=*"main.html"*>Back to Main</**a**>

<**script**>

document.getElementById('registrationForm').addEventListener('submit', **function**(event) {

event.preventDefault();

**const** formData = {

username: document.getElementById('username').value,

password: document.getElementById('password').value,

email: document.getElementById('email').value

};

fetch('/rental/block/register', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(formData)

})

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

.then(data => {

document.getElementById('message').textContent = data.message;

})

.**catch**(error => console.error('Error registering:', error));

});

</**script**>

</**body**>

</**html**>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"UTF-8"*>

<**title**>Movie Rental</**title**>

<**style**>

body {

font-family: *Arial, sans-serif*;

margin: *20px*;

}

h1 {

color: *#333*;

}

form {

max-width: *400px*;

margin: *0 auto*;

}

label {

display: *block*;

margin-bottom: *5px*;

}

input[type="text"], input[type="number"] {

width: *100%*;

padding: *8px*;

margin-bottom: *10px*;

box-sizing: *border-box*;

}

button {

padding: *10px 20px*;

background-color: *#007bff*;

color: *#fff*;

border: *none*;

cursor: *pointer*;

}

button:hover {

background-color: *#0056b3*;

}

</**style**>

</**head**>

<**body**>

<**h1**>Rent a Movie</**h1**>

<**form** id=*"rentMovieForm"*>

<**label** for=*"userId"*>User ID:</**label**>

<**input** type=*"number"* id=*"userId"* name=*"userId"* required>

<**label** for=*"movieId"*>Movie ID:</**label**>

<**input** type=*"number"* id=*"movieId"* name=*"movieId"* required>

<**button** type=*"submit"*>Rent Movie</**button**>

</**form**>

<**a** href=*"main.html"*>Back to Main</**a**>

<**div** id=*"message"*></**div**>

<**script**>

document.getElementById('rentMovieForm').addEventListener('submit', **function**(event) {

event.preventDefault();

**const** formData = {

userId: document.getElementById('userId').value,

movieId: document.getElementById('movieId').value

};

fetch('/rental/block/rent', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(formData)

})

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

.then(data => {

document.getElementById('message').innerHTML = data.message;

})

.**catch**(error => console.error('Error renting movie:', error));

});

</**script**>

</**body**>

</**html**>

<!**DOCTYPE** html>

<**html** lang=*"en"*>

<**head**>

<**meta** charset=*"UTF-8"*>

<**meta** name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>

<**title**>Movie Rental</**title**>

<**style**>

body {

font-family: *Arial, sans-serif*;

margin: *20px*;

}

h1 {

color: *#333*;

}

form {

max-width: *400px*;

margin: *0 auto*;

}

label {

display: *block*;

margin-bottom: *5px*;

}

input[type="text"] {

width: *100%*;

padding: *8px*;

margin-bottom: *10px*;

box-sizing: *border-box*;

}

button {

padding: *10px 20px*;

background-color: *#007bff*;

color: *#fff*;

border: *none*;

cursor: *pointer*;

}

button:hover {

background-color: *#0056b3*;

}

</**style**>

</**head**>

<**body**>

<**h1**>Search Movies</**h1**>

<**form** id=*"searchMovieForm"*>

<**label** for=*"searchTerm"*>Search Term:</**label**>

<**input** type=*"text"* id=*"searchTerm"* name=*"searchTerm"* required>

<**button** type=*"submit"*>Search</**button**>

</**form**>

<**ul** id=*"searchResults"*>

</**ul**>

<**a** href=*"main.html"*>Back to Main</**a**>

<**script**>

document.getElementById('searchMovieForm').addEventListener('submit', **function**(event) {

event.preventDefault();

**const** searchTerm = document.getElementById('searchTerm').value;

fetch(`/rental/block/search`, {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ searchTerm: searchTerm })

})

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

.then(data => {

**const** searchResults = document.getElementById('searchResults');

searchResults.innerHTML = '';

data.forEach(movie => {

**const** li = document.createElement('li');

li.textContent = `${movie.title} (${movie.genre}, ${movie.releaseYear})`;

searchResults.appendChild(li);

});

})

.**catch**(error => console.error('Error searching movies:', error));

});

</**script**>

</**body**>

</**html**>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"UTF-8"*>

<**title**>View All Movies</**title**>

<**style**>

body {

font-family: *Arial, sans-serif*;

margin: *20px*;

}

h1 {

color: *#333*;

}

.movie-list {

list-style-type: *none*;

padding: *0*;

}

.movie-item {

margin-bottom: *10px*;

padding: *10px*;

border: *1px solid #ccc*;

background-color: *#f9f9f9*;

}

.movie-item h2 {

margin: *0*;

}

.movie-item p {

margin-top: *5px*;

}

</**style**>

</**head**>

<**body**>

<**h1**>View All Movies</**h1**>

<**ul** class=*"movie-list"*>

</**ul**>

<**a** href=*"main.html"*>Back to Main</**a**>

<**script**>

fetch('/rental/block/movies', {

method: 'GET',

headers: {

'Content-Type': 'application/json'

}

})

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

.then(data => {

**const** movieList = document.querySelector('.movie-list');

movieList.innerHTML = '';

data.forEach(movie => {

**const** movieItem = document.createElement('li');

movieItem.classList.add('movie-item');

movieItem.innerHTML = `

<h2>${movie.title}</h2>

<p>Genre: ${movie.genre}</p>

<p>Release Year: ${movie.releaseYear}</p>

`;

movieList.appendChild(movieItem);

});

})

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

</**script**>

</**body**>

</**html**>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"UTF-8"*>

<**title**>Search Movies</**title**>

<**style**>

body {

font-family: *Arial, sans-serif*;

margin: *20px*;

}

h1 {

color: *#333*;

}

.rented-movie-list {

list-style-type: *none*;

padding: *0*;

}

.rented-movie-item {

margin-bottom: *10px*;

padding: *10px*;

border: *1px solid #ccc*;

background-color: *#f9f9f9*;

}

.rented-movie-item h2 {

margin: *0*;

}

.rented-movie-item p {

margin-top: *5px*;

}

</**style**>

</**head**>

<**body**>

<**h1**>View Rented Movies</**h1**>

<**ul**

class=*"rented-movie-list"*>

</**ul**>

<**a** href=*"main.html"*>Back to Main</**a**>

<**script**>

fetch('/rental/block/rented', {

method: 'GET',

headers: {

'Content-Type': 'application/json'

}

})

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

.then(data => {

**const** rentedMovieList = document.querySelector('.rented-movie-list');

rentedMovieList.innerHTML = '';

data.forEach(rentedMovie => {

**const** rentedMovieItem = document.createElement('li');

rentedMovieItem.classList.add('rented-movie-item');

rentedMovieItem.innerHTML = `

<h2>${rentedMovie.title}</h2>

<p>Genre: ${rentedMovie.genre}</p>

<p>Release Year: ${rentedMovie.releaseYear}</p>

`;

rentedMovieList.appendChild(rentedMovieItem);

});

})

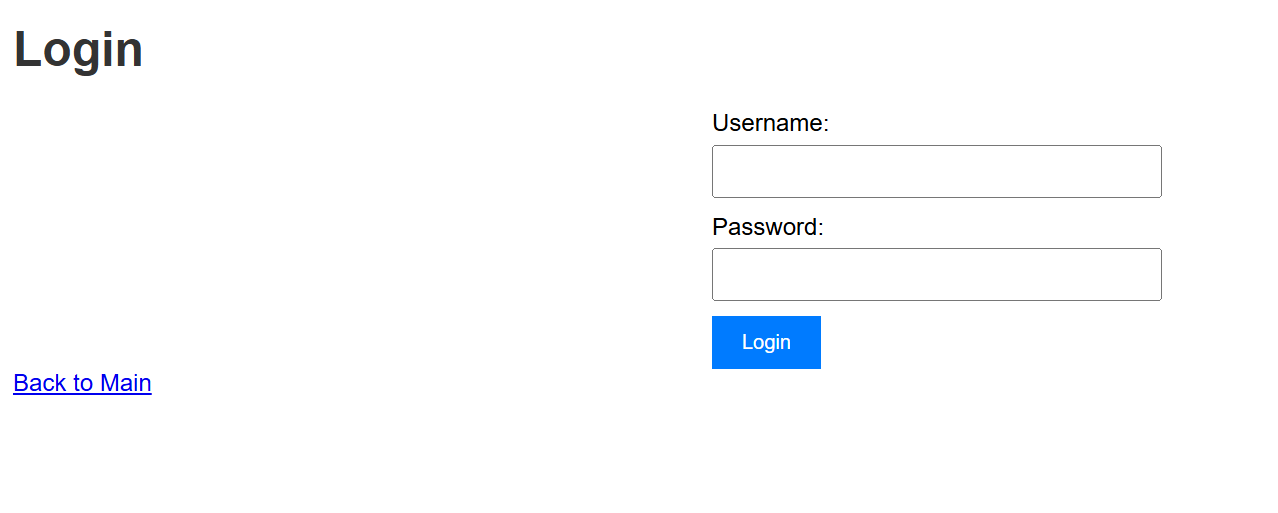
.**catch**(error => console.error('Error fetching rented movies:', error));

</**script**>

</**body**>

</**html**>

## PICTURES



A screen shot of a movie

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer login box

Description automatically generated

A computer screen shot of a blue box

Description automatically generated