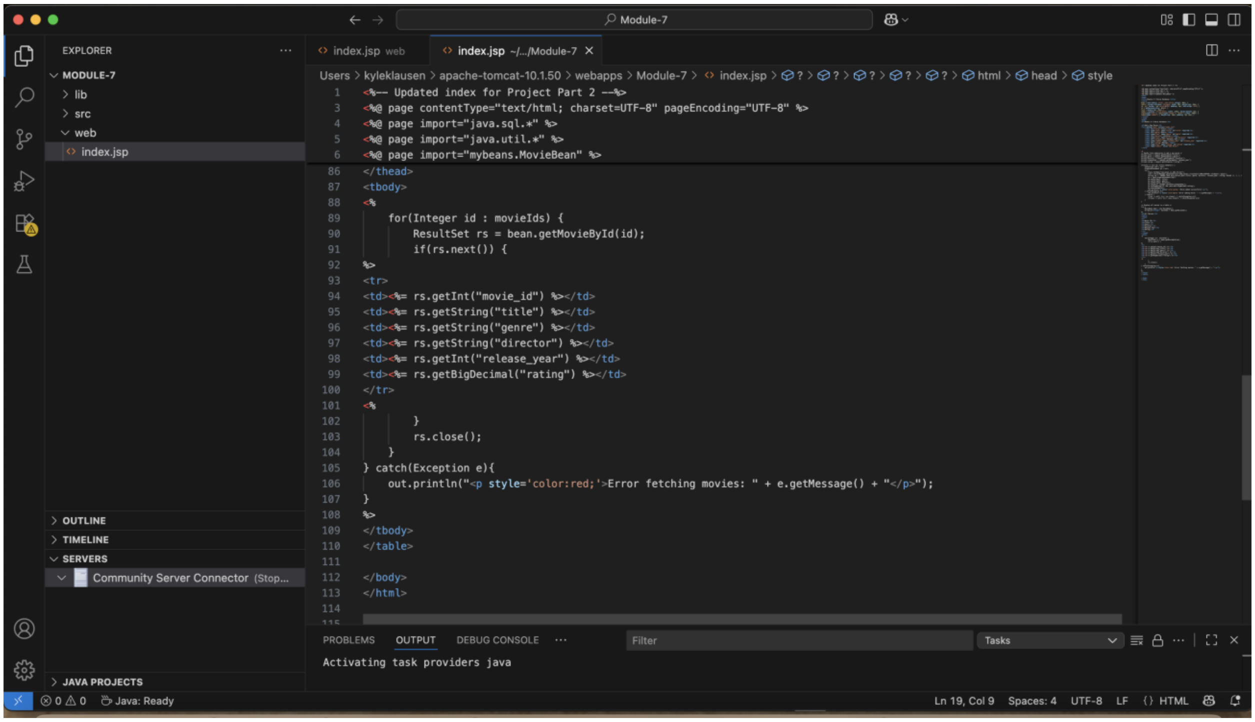
Kyle Klausen

CSD-430

Module 7

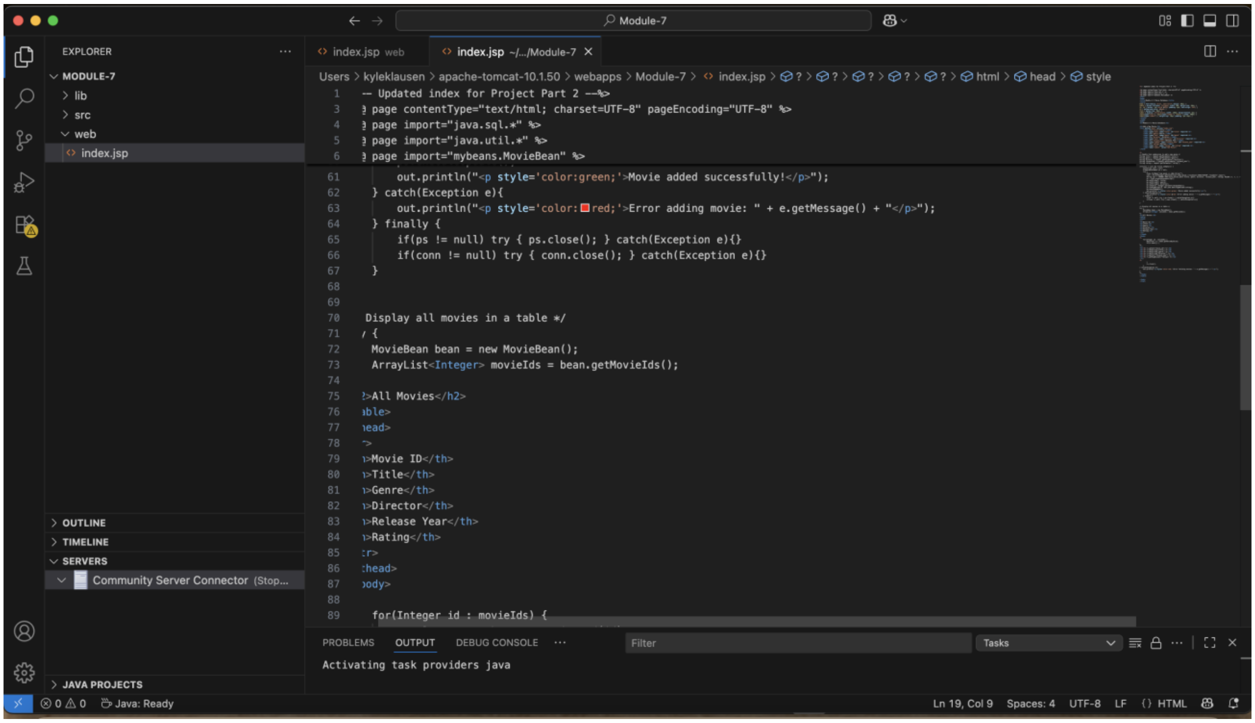
02/14/26

Index.jsp Code (Wouldn’t allow to paste without messing up the format):


Module-7
EXPLORER
MODULE-7
lib
src
web
index.jsp
OUTLINE
TIMELINE
SERVERS
Community Server Connector (Stop...
JAVA PROJECTS
index.jsp web
index.jsp ~/.../Module-7
Users > kyleklausen > apache-tomcat-10.1.50 > webapps > Module-7 > index.jsp > ? > ? > ? > ? > ? > html > head > style
PROBLEMS
OUTPUT
DEBUG CONSOLE
Filter
Tasks
0 0 Java: Ready
Ln 19, Col 9 Spaces: 4 UTF-8 LF {} HTML

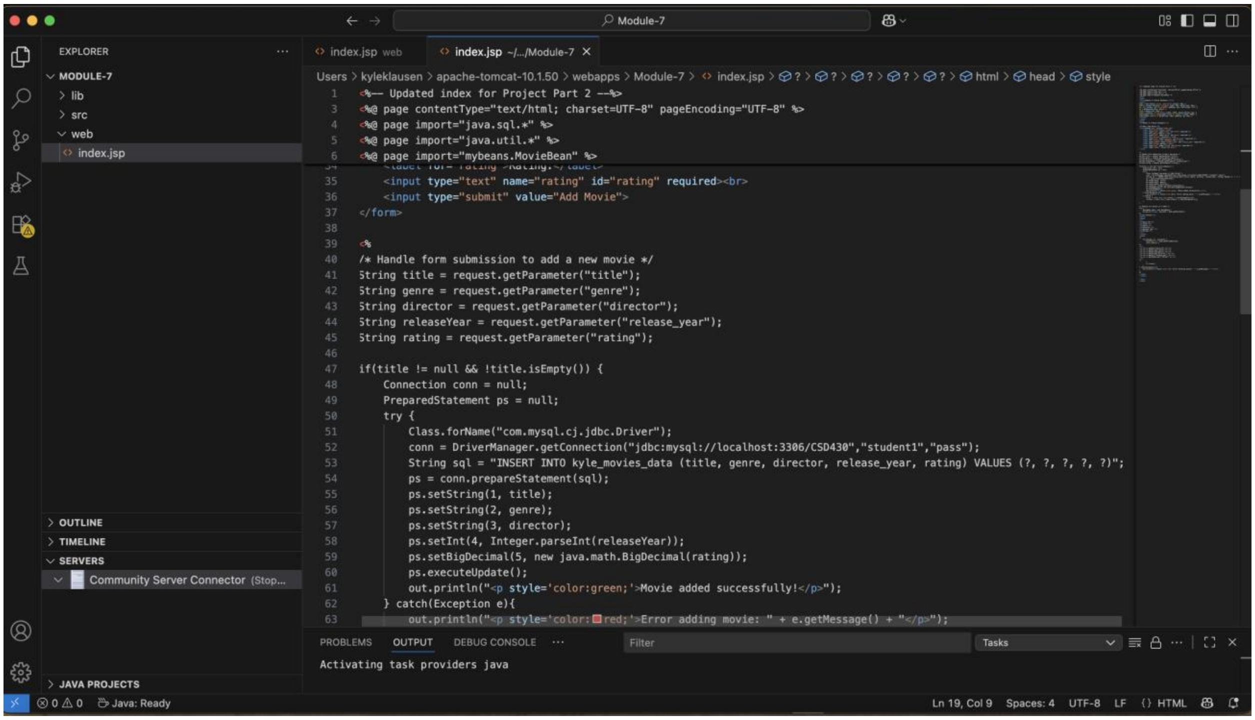
```
<%-- Updated index for Project Part 2 --%>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.*" %>
<%@ page import="mybeans.MovieBean" %>
</thead>
<tbody>
<%
    for(Integer id : movieIds) {
        ResultSet rs = bean.getMovieById(id);
        if(rs.next()) {
%>
<tr>
<td><%= rs.getInt("movie_id") %></td>
<td><%= rs.getString("title") %></td>
<td><%= rs.getString("genre") %></td>
<td><%= rs.getString("director") %></td>
<td><%= rs.getInt("release_year") %></td>
<td><%= rs.getBigDecimal("rating") %></td>
</tr>
<%
        }
        rs.close();
    }
} catch(Exception e){
    out.println("<p style='color:red;'>Error fetching movies: " + e.getMessage() + "</p>");
}
%>
</tbody>
</table>

</body>
</html>

```

```
Activating task providers java
```


Module-7
EXPLORER
MODULE-7
lib
src
web
index.jsp
OUTLINE
TIMELINE
SERVERS
Community Server Connector (Stop...
JAVA PROJECTS
index.jsp web
index.jsp ~/.../Module-7
Users > kyleklausen > apache-tomcat-10.1.50 > webapps > Module-7 > index.jsp > ? > ? > ? > ? > ? > html > head > style
PROBLEMS
OUTPUT
DEBUG CONSOLE
Filter
Tasks
0 0 Java: Ready
Ln 19, Col 9 Spaces: 4 UTF-8 LF {} HTML

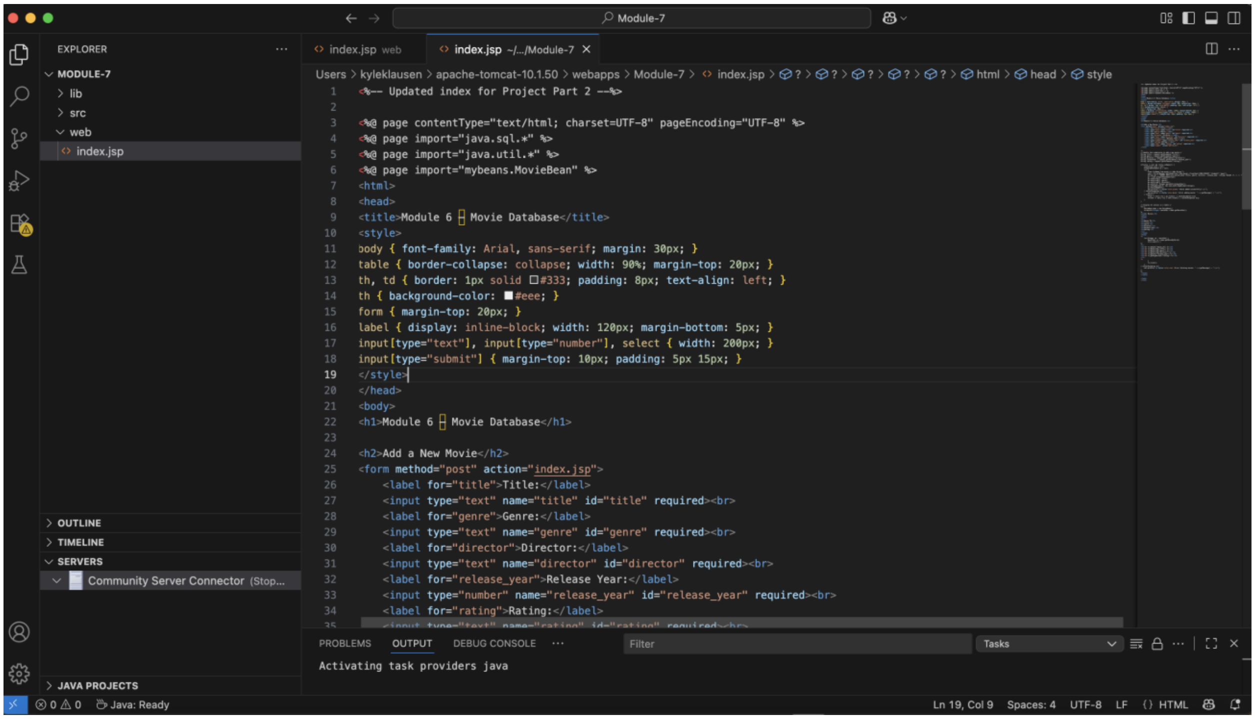
```
-- Updated index for Project Part 2 --%>
) page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
) page import="java.sql.*" %>
) page import="java.util.*" %>
) page import="mybeans.MovieBean" %>
      out.println("<p style='color:green;'>Movie added successfully!</p>");
  } catch(Exception e){
      out.println("<p style='color:red;'>Error adding movie: " + e.getMessage() + "</p>");
  } finally {
      if(ps != null) try { ps.close(); } catch(Exception e){}
      if(conn != null) try { conn.close(); } catch(Exception e){}
  }


 Display all movies in a table */
/ {
  MovieBean bean = new MovieBean();
  ArrayList<Integer> movieIds = bean.getMovieIds();

2>All Movies</h2>
able>
head>
->
1>Movie ID</th>
1>Title</th>
1>Genre</th>
1>Director</th>
1>Release Year</th>
1>Rating</th>
tr>
thead>
body>

  for(Integer id : movieIds) {
```

```
Activating task providers java
```



```
<%-- Updated index for Project Part 2 --%>
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.*" %>
<%@ page import="mybeans.MovieBean" %>
    <input type="text" name="rating" id="rating" required><br>
    <input type="submit" value="Add Movie">
</form>

<%
/* Handle form submission to add a new movie */
String title = request.getParameter("title");
String genre = request.getParameter("genre");
String director = request.getParameter("director");
String releaseYear = request.getParameter("release_year");
String rating = request.getParameter("rating");

if(title != null && !title.isEmpty()) {
    Connection conn = null;
    PreparedStatement ps = null;
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/CSD430","student1","pass");
        String sql = "INSERT INTO kyle_movies_data (title, genre, director, release_year, rating) VALUES (?, ?, ?, ?, ?)";
        ps = conn.prepareStatement(sql);
        ps.setString(1, title);
        ps.setString(2, genre);
        ps.setString(3, director);
        ps.setInt(4, Integer.parseInt(releaseYear));
        ps.setBigDecimal(5, new java.math.BigDecimal(rating));
        ps.executeUpdate();
        out.println("<p style='color:green;'>Movie added successfully!</p>");
    } catch(Exception e){
        out.println("<p style='color:red;'>Error adding movie: " + e.getMessage() + "</p>");
```

```
Activating task providers java
```



```
<%-- Updated index for Project Part 2 --%>

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ page import="java.sql.*" %>
<%@ page import="java.util.*" %>
<%@ page import="mybeans.MovieBean" %>
<html>
<head>
<title>Module 6 ▯ Movie Database</title>
<style>
body { font-family: Arial, sans-serif; margin: 30px; }
table { border-collapse: collapse; width: 90%; margin-top: 20px; }
th, td { border: 1px solid #333; padding: 8px; text-align: left; }
th { background-color: #eee; }
form { margin-top: 20px; }
label { display: inline-block; width: 120px; margin-bottom: 5px; }
input[type="text"], input[type="number"], select { width: 200px; }
input[type="submit"] { margin-top: 10px; padding: 5px 15px; }
</style>
</head>
<body>
<h1>Module 6 ▯ Movie Database</h1>

<h2>Add a New Movie</h2>
<form method="post" action="index.jsp">
    <label for="title">Title:</label>
    <input type="text" name="title" id="title" required><br>
    <label for="genre">Genre:</label>
    <input type="text" name="genre" id="genre" required><br>
    <label for="director">Director:</label>
    <input type="text" name="director" id="director" required><br>
    <label for="release_year">Release Year:</label>
    <input type="number" name="release_year" id="release_year" required><br>
    <label for="rating">Rating:</label>
```

```
Activating task providers java
```

Same MovieBean.java as before:

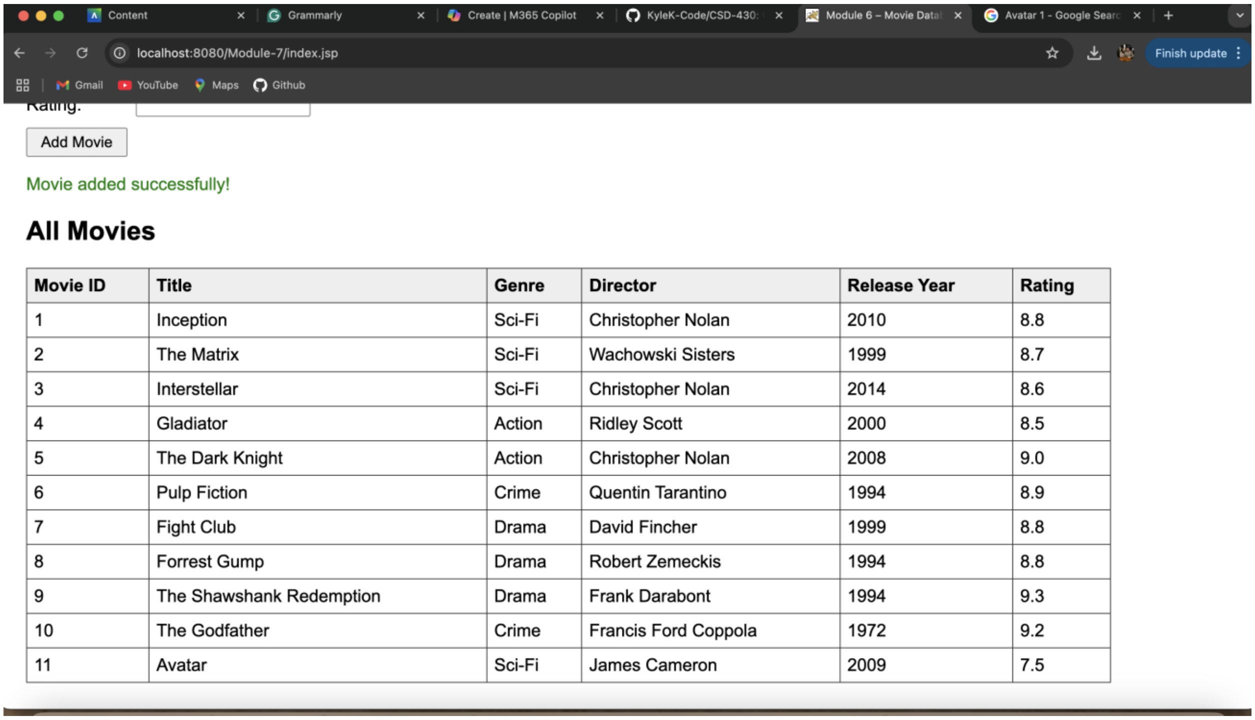
package mybeans;

import java.sql.*;

```
    }

import java.util.ArrayList;

public class MovieBean {

    private Connection conn;

    public MovieBean() throws Exception {
        Class.forName("com.mysql.cj.jdbc.Driver");
        conn = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/CSD430",
            "student1",
            "pass"
        );
    }

    public ArrayList<Integer> getMovieIds() throws SQLException {
        ArrayList<Integer> ids = new ArrayList<>();
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT movie_id FROM kyle_movies_data");

        while (rs.next()) {
            ids.add(rs.getInt("movie_id"));
        }
        return ids;
    }
```

```
  public ResultSet getMovieById(int id) throws SQLException {
    PreparedStatement ps = conn.prepareStatement(
      "SELECT * FROM kyle_movies_data WHERE movie_id = ?"
    );
    ps.setInt(1, id);
    return ps.executeQuery();
  }
}
```

Screenshots of successful run:



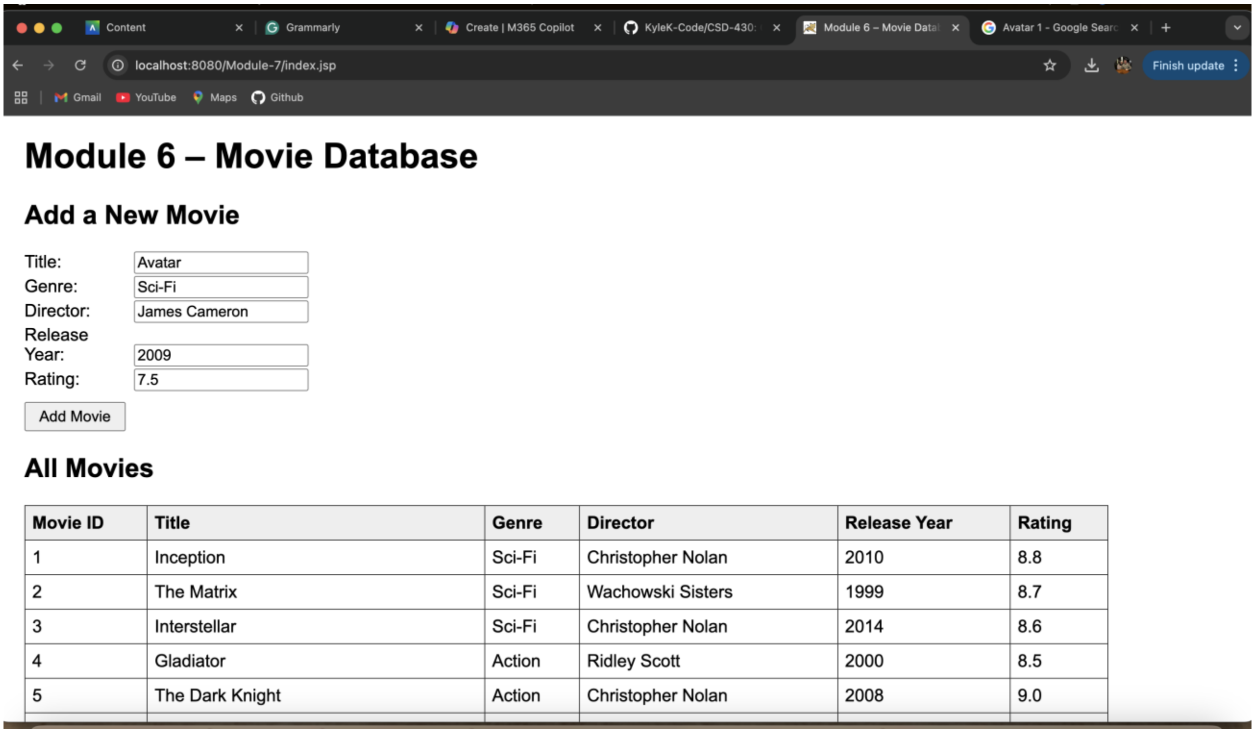
Rating:

Add Movie

Movie added successfully!

All Movies

Movie ID	Title	Genre	Director	Release Year	Rating
1	Inception	Sci-Fi	Christopher Nolan	2010	8.8
2	The Matrix	Sci-Fi	Wachowski Sisters	1999	8.7
3	Interstellar	Sci-Fi	Christopher Nolan	2014	8.6
4	Gladiator	Action	Ridley Scott	2000	8.5
5	The Dark Knight	Action	Christopher Nolan	2008	9.0
6	Pulp Fiction	Crime	Quentin Tarantino	1994	8.9
7	Fight Club	Drama	David Fincher	1999	8.8
8	Forrest Gump	Drama	Robert Zemeckis	1994	8.8
9	The Shawshank Redemption	Drama	Frank Darabont	1994	9.3
10	The Godfather	Crime	Francis Ford Coppola	1972	9.2
11	Avatar	Sci-Fi	James Cameron	2009	7.5



Module 6 – Movie Database

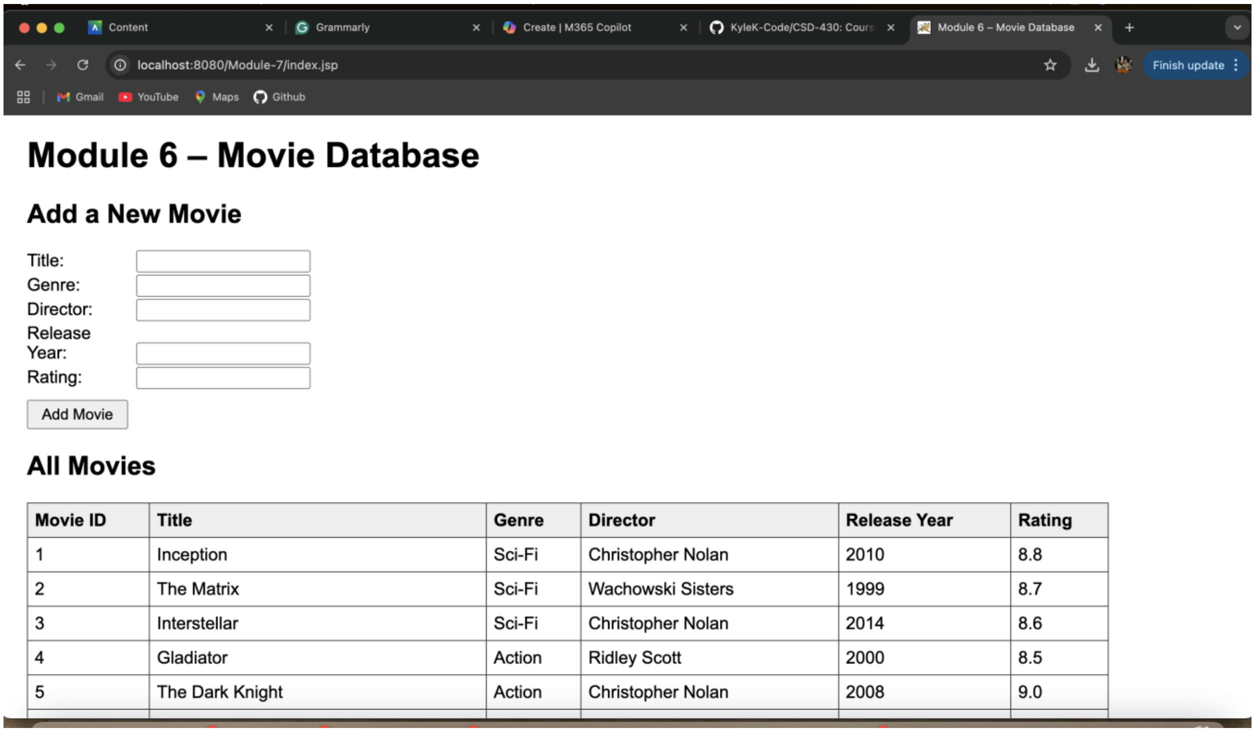
Add a New Movie

Title: Avatar
Genre: Sci-Fi
Director: James Cameron
Release Year: 2009
Rating: 7.5

Add Movie

All Movies

Movie ID	Title	Genre	Director	Release Year	Rating
1	Inception	Sci-Fi	Christopher Nolan	2010	8.8
2	The Matrix	Sci-Fi	Wachowski Sisters	1999	8.7
3	Interstellar	Sci-Fi	Christopher Nolan	2014	8.6
4	Gladiator	Action	Ridley Scott	2000	8.5
5	The Dark Knight	Action	Christopher Nolan	2008	9.0


Content
Grammarly
Create | M365 Copilot
KyleK-Code/CSD-430: Cours
Module 6 – Movie Database
localhost:8080/Module-7/index.jsp
Finish update
Gmail
YouTube
Maps
Github

Module 6 – Movie Database

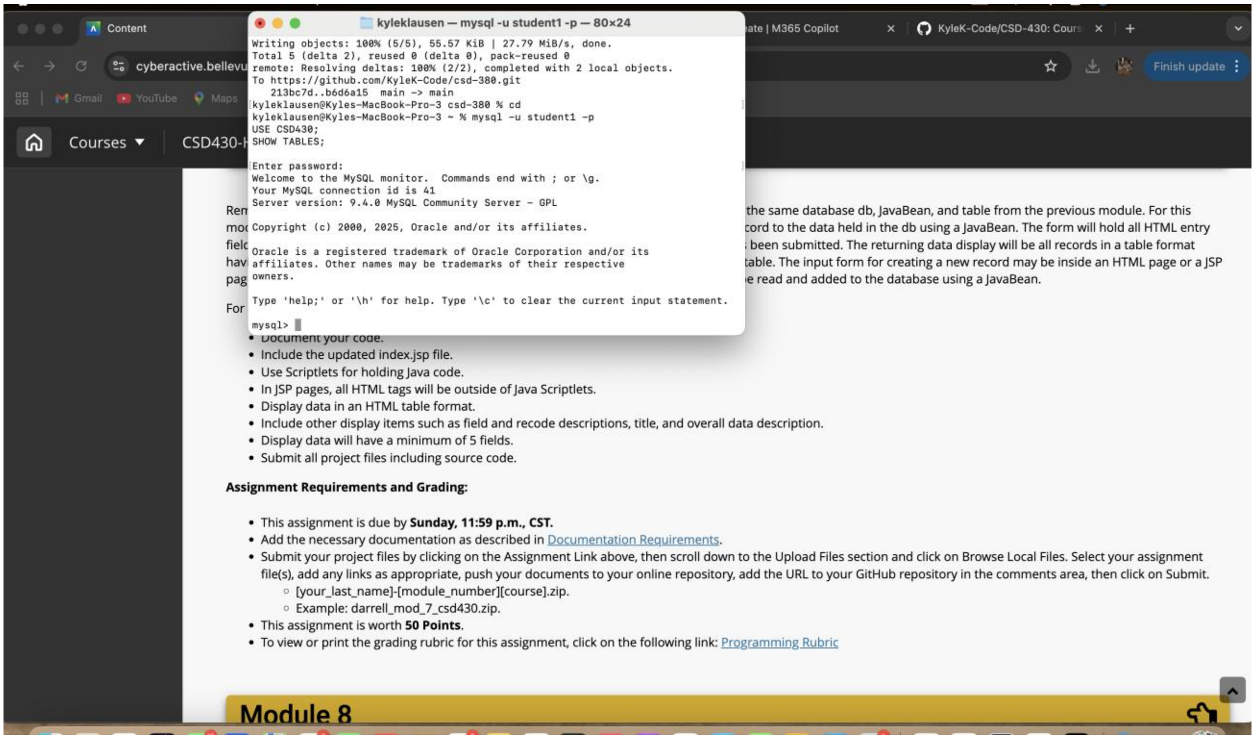
Add a New Movie

Title:	
Genre:	
Director:	
Release Year:	
Rating:	

Add Movie

All Movies

Movie ID	Title	Genre	Director	Release Year	Rating
1	Inception	Sci-Fi	Christopher Nolan	2010	8.8
2	The Matrix	Sci-Fi	Wachowski Sisters	1999	8.7
3	Interstellar	Sci-Fi	Christopher Nolan	2014	8.6
4	Gladiator	Action	Ridley Scott	2000	8.5
5	The Dark Knight	Action	Christopher Nolan	2008	9.0



```
Writing objects: 100% (5/5), 55.57 KiB | 27.79 MiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/KyleK-Code/csd-380.git
   213bc7d..b6d6a15  main -> main
kyleklausen@Kyles-MacBook-Pro-3 csd-380 % cd
kyleklausen@Kyles-MacBook-Pro-3 ~ % mysql -u student1 -p
USE CSD430;
SHOW TABLES;

Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 41
Server version: 9.4.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Rem the same database db, JavaBean, and table from the previous module. For this mod cord to the data held in the db using a JavaBean. The form will hold all HTML entry field been submitted. The returning data display will be all records in a table format hav table. The input form for creating a new record may be inside an HTML page or a JSP pag e read and added to the database using a JavaBean.

For

- Document your code.
- Include the updated index.jsp file.
- Use Scriptlets for holding Java code.
- In JSP pages, all HTML tags will be outside of Java Scriptlets.
- Display data in an HTML table format.
- Include other display items such as field and recode descriptions, title, and overall data description.
- Display data will have a minimum of 5 fields.
- Submit all project files including source code.

Assignment Requirements and Grading:

- This assignment is due by **Sunday, 11:59 p.m., CST.**
- Add the necessary documentation as described in Documentation Requirements.
- Submit your project files by clicking on the Assignment Link above, then scroll down to the Upload Files section and click on Browse Local Files. Select your assignment file(s), add any links as appropriate, push your documents to your online repository, add the URL to your GitHub repository in the comments area, then click on Submit.
  - [your_last_name]-[module_number][course].zip.
  - Example: darrell_mod_7_csd430.zip.
- This assignment is worth **50 Points.**
- To view or print the grading rubric for this assignment, click on the following link: Programming Rubric

Module 8