IMDB Like Website Project

March 4, 2023

Zhiyong Sui

Table of Contents

Introduction.....3

Relate Work.....4

System Architectural Design.....4

Relational table: user..... 5

Relational table: movies..... 6

Relational table: today.....6

Relational table: view.....6

Detailed Description of Components.....7

Main Page 7

Sign up 8

Log in 10

Search for actor..... 11

Search for producer..... 13

Search for comment 15

Search for today movies 17

Search for top 10 hottest movies 18

Search for movies view …..... 21

Screen of the response of each my servlet.....23

Conclusion 23

INTRODUCTION

System Overview

The IMDB like website project use 8 servlet to achieve the functions of user login, user registration, search movies show producer's name, search movies show actor/actress's name, show the top 10 hottest movies, movies of the day, show a specific movie summary, and show comments of a specific movie. Users can sign up and login in, after login the username will show on the top bar near the login icon.

The implementation of the project was designed as a three-tier web application completely separating the static content in apache server and servlet in Tomcat server. The application layer in apache server runs an apache server and concludes all static contents pictures and css files. And for the server layer, it concludes all servlet java files and mysql database which conclude all movies related tables and holds server-side logic, which manipulates the data system and returns requested data to the presentation layer where it is dynamically interpreted and presented to the user inside the browser. I also make a backup layer in Tomcat, this is for the situation if the apache server is down, and the user can access the website through the tomcat server ip with port 8080. This design is just like RAID-0 in system.

Design Overview

When the main page will show some movies and the top bar in the head of the html page will have sign up and login, and the top bar also concludes one part will show the name of the user after log in. This will retrieve name and password from the database to check the validation of the user. And the sign up icon will make the user sign up and insert user's information to the database.

On the right side of the MainPage1.html there is a select box that has two options for users, one is for top 10 hottest movies and another is today's video in the theater. All the function all achieved by servlet, and when servlet programs is running, the servlet will first return a webpage without any css style and javascript, because there is a situation that the there will be same name for different movies, and the singlemovie.html and singlemovie_1.html will return the webpage with result after user click the go forward.

The result will show the movie name, picture of the movies, and other required information. I put some "fake" information in the database just like "actor1" or "producor1". This is just for testing, and the administrator replaces this "fake" information to show the real information of the movies.

RELATE WORKS

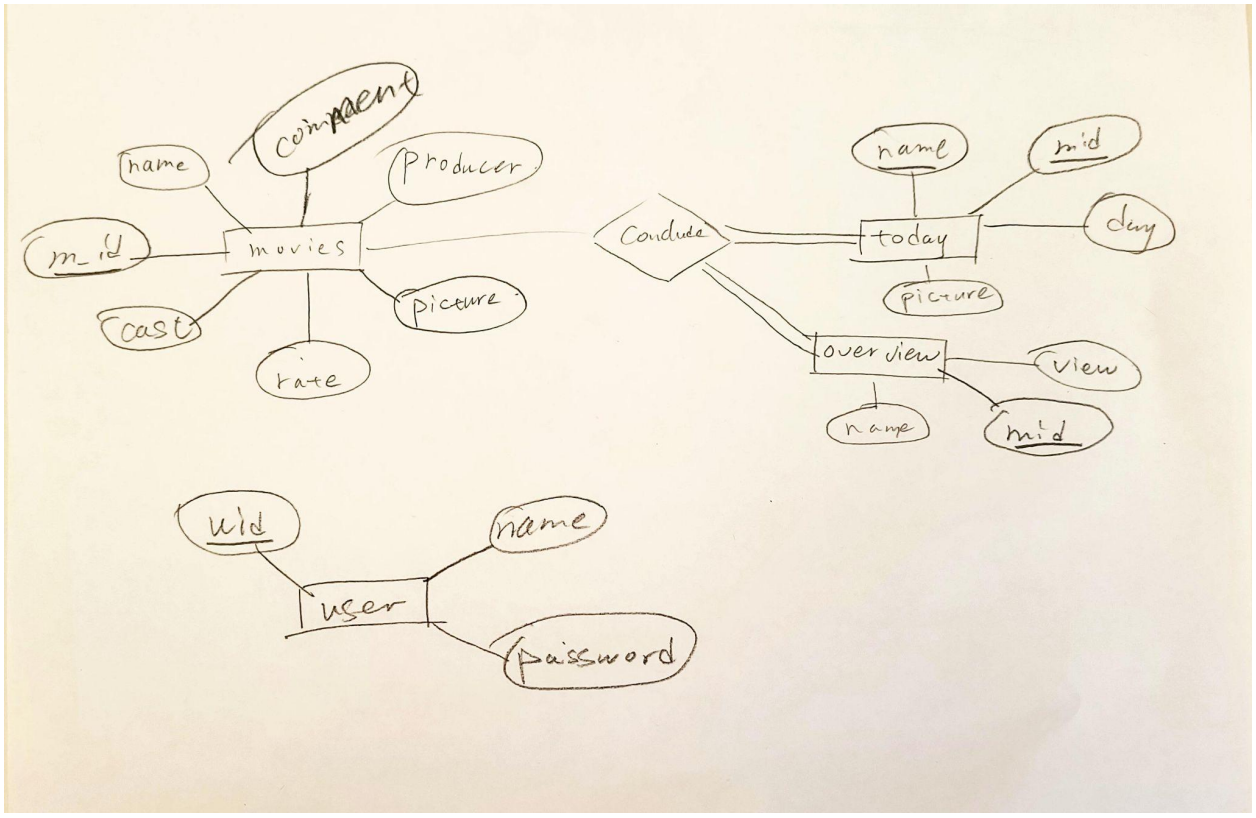
The project website seems like an IMDB website, but just achieves some parts of the function of the IMDB website. My project does not have the account page and can not add comments and views for movies. And other works may have more layers in the real world and also have more other functions for paying or buying the tickets. And in my project, the website does not allow users to upload their picture or movie resource to the database.

And for the IMDB website, they have more restrictive policies to check if the user is legal to login in , because there are some things related to bank count on the IMDB website. And they also separate users' levels, but my project does not have that function.

SYSTEM ARCHITECTURE DESIGN

The system architecture follows the general three-tier web application design described in the introduction. The application was deployed across two amazon EC2 instances, one containing the presentation layer and the other containing both logic and data layers. A special thing in my system design is I make a backup for the presentation layer, and if something happens in the presentation layer, we will restore the server for the presentation layer. The first instance runs Apache Web Server hosting the static web content at its root. The second instance runs Apache Tomcat Web Server containing JAVA servlets and MySQL for the relation database testDB storing movies and user data. Users can access the project by the public ip address for Apache Web Server hosting public ip. The Apache Web Server was connected to Tomcat Server with mod_jk through port 60 and the requests are sent to Tomcat server and call the servlet files to change the database, show something to the user, and link to another website. And all servlets are under the path /applicationLayer/. Each servlet is assigned to a specific path after the /applicationLayer/ path of the URL depending on the request action. Java servlets use Java Database Connectivity (JDBC) library to connect with the testDB MySQl database, and the database user name is "zhiyongsui", password is "1".

MySQL was chosen as the data system because of the general benefits of a Relational Database Management System for persistent storage and the ease of integration with the servlets using the JDBC library. The design and development process resulted in an entity-relationship model of the data needed for the IMDB-like website application as shown below.



This model was then transformed into relational tables for use in a MySQL database that could be queried and updated by the servlets called submit requests from the client. The headers of the relational tables are shown below along with brief descriptions of their use in the developed application.

Relational table: user

The user stores the data of user accounts that have registered. The primary key is (U_id) code to distinguish different users who have the same name. And when a user login, the database will first to check the name and find if there is a name with the identity password that can login in and after login there will be a username shown on the top bar.

U_id (Primary key)	Name VARCHAR(20)	Password VARCHAR(20)

Relational table: movies

The movies table stores the data of movies, such as name, rate, cast, producer, and also contains the path of the movie's picture that used to show to the user. The primary key is (M_id) code to distinguish different movies who have the same name. And users can retrieve information that they demand from the right-side submit form.

M_id (Primary key)	Name VARCHAR(20)	cast VARCHAR(20)	comment VARCHAR(100)	producer VARCHAR(100)	picture VARCHAR(20)	Rate Integer

Relational table: today

The today table stores the data of movies, such as name, day, and also contains the path of the movie's picture that used to show to the user. The primary key is (M_id) code to distinguish different movies who have the same name. The day value is generated randomly, and use this number mod by the day of the week from 1-7 to get the today's movies in the theater, today movies in theater may have one or more than one.

M_id (Primary key)(FK)	Name VARCHAR(20)(Primary key)	picture VARCHAR(20)

Relational table: view

The view table stores the data of movies, such as name, view, and also contains the path of the movie's picture that used to show to the user. The primary key is (M_id) code to distinguish different movies who have the same name. I do not have too many real word views, so I put the data like "viewX" in the database for the user to retrieve.

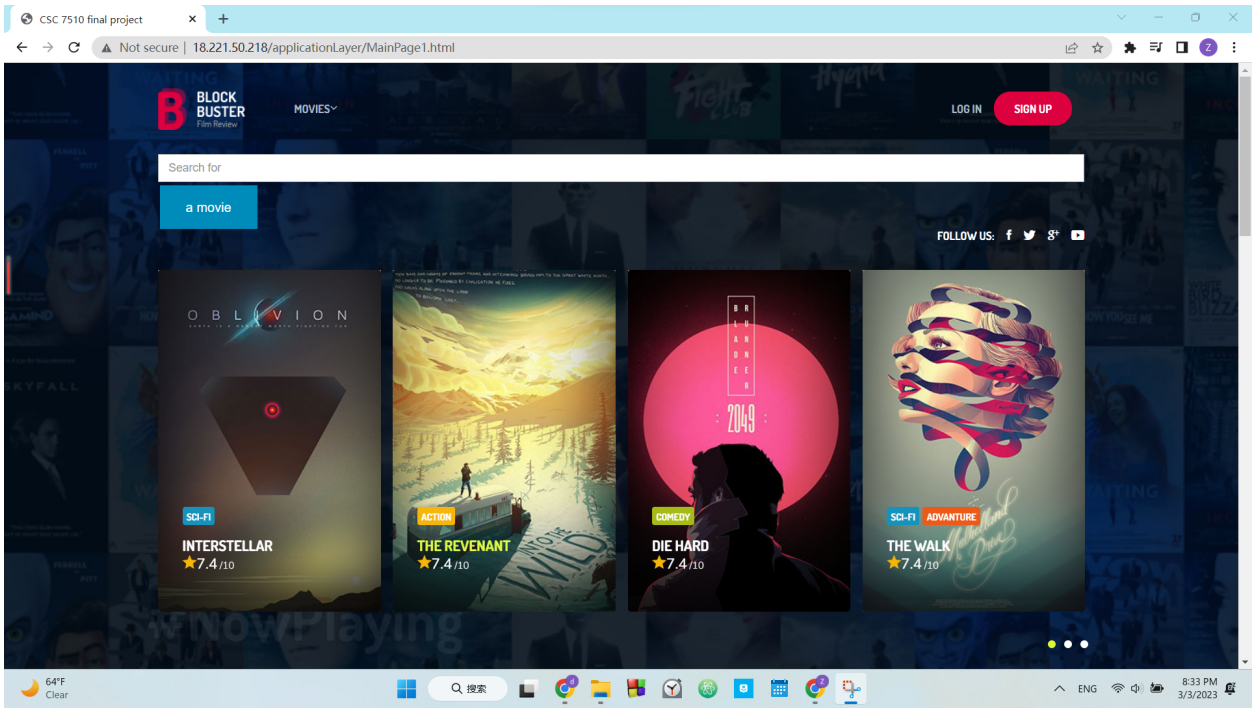
M_id (Primary key)(FK)	Name VARCHAR(20)(Primary key)	view VARCHAR(100)

Detailed Description of Components

The following is a description of each functional component of the ImDB-like application as a servlet in the Tomcat container with a screenshot of how the developed front-end presents the response of the servlet if applicable. All of the servlets forward POST requests to the doGet method. If a request doesn't specify a required parameter for the servlet, then the servlet will return with a response that contains a reminder message and a link back to the main page. JDBC is used for all communication with the database and prepared statements are used when handling user submitted data.

Main Page

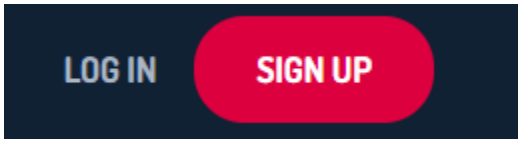
Main page will show some movie pictures in the gallery and on the op bar there will be three functions, one for user login, one for user sign up, and one for searching the view of the movie.



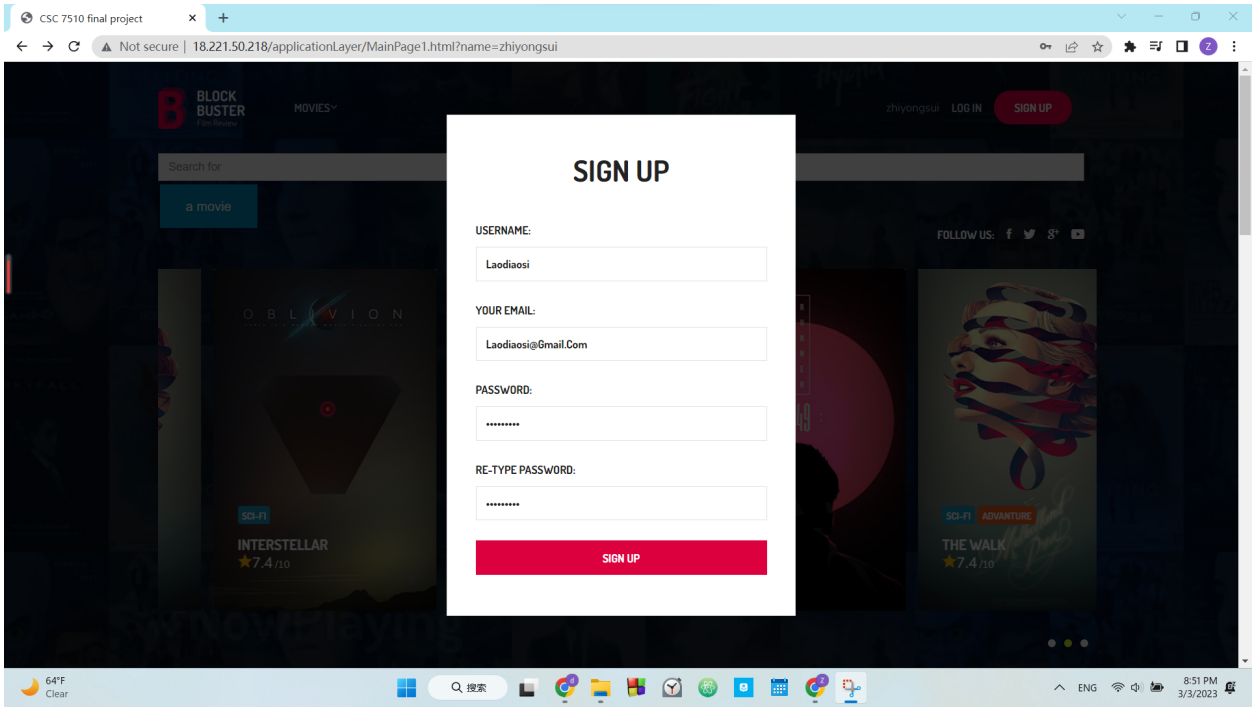
Sign up

The register servlet receives an name and password for registration of a new account. If the information user input passed the system' s policies checking, there will be redirected to a new page showing that it was successful. Then the servlet generates information needed to insert into the database. The database and servlet allow users to have the same name but with different U_id, and then insert a row into the database with the name,

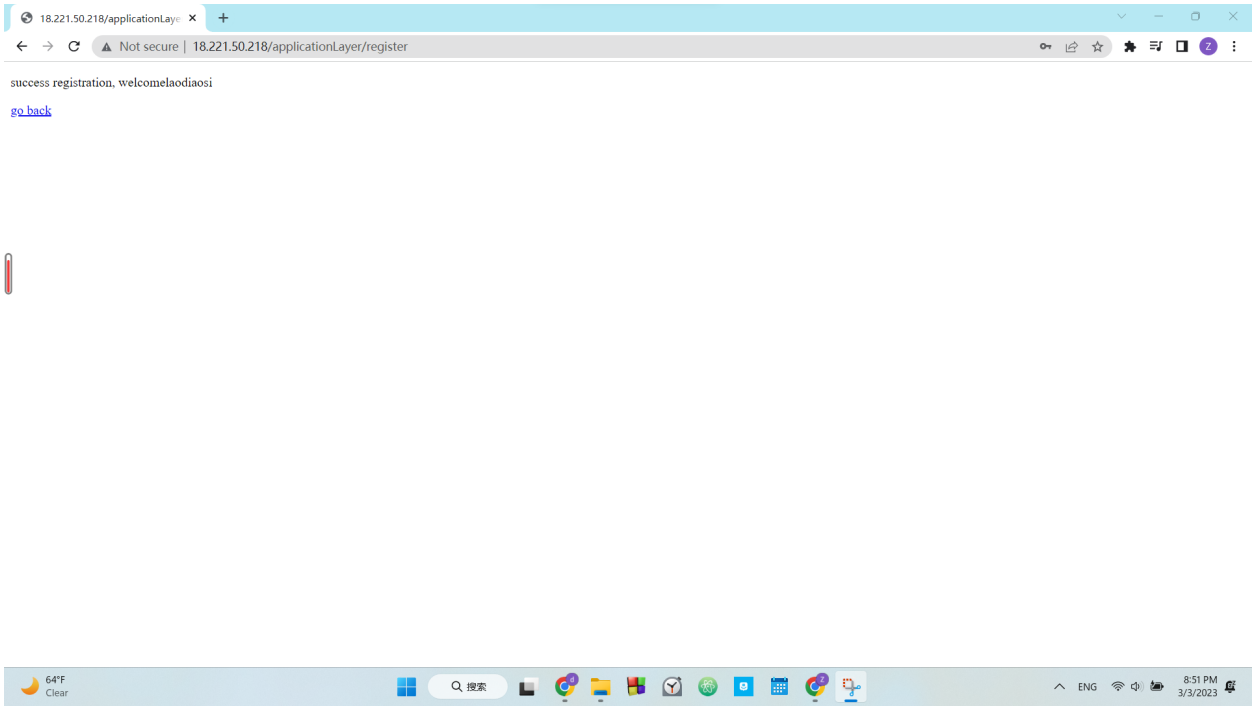
password, and U_id. The following screenshot is an example of an email sent by the servlet upon registration.



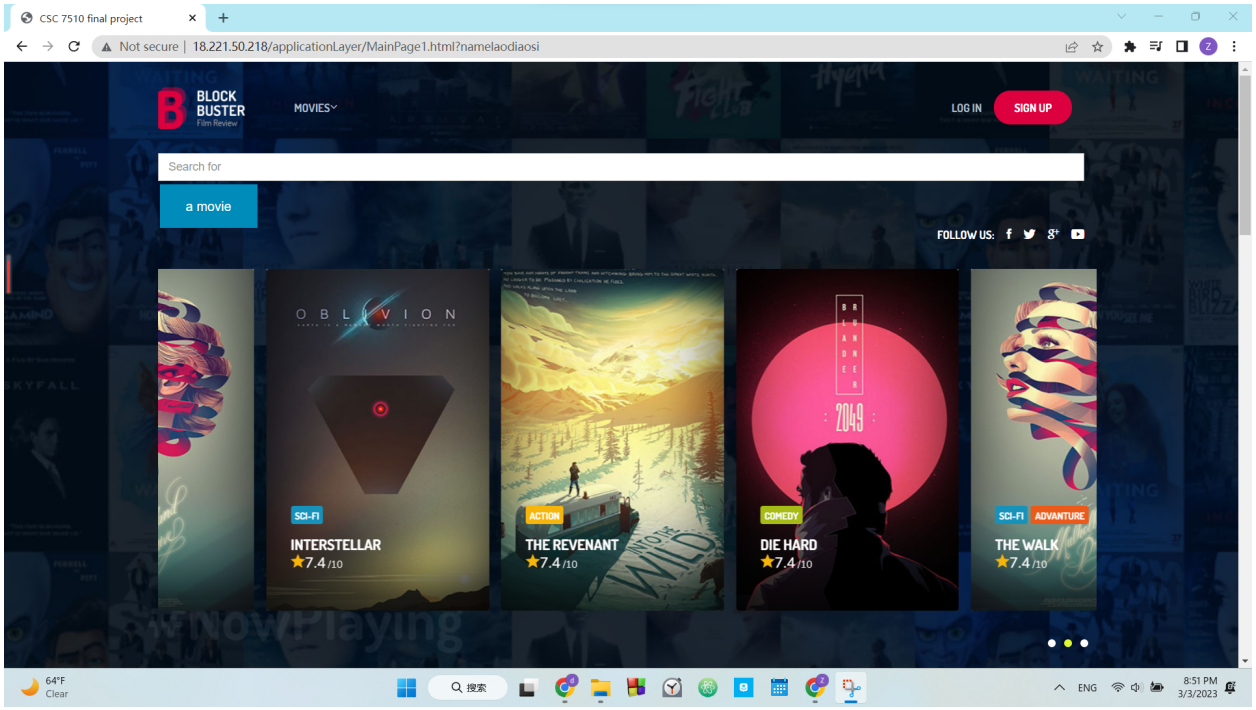
Picture 1, Sign up icon



Picture 2, popup sign up window



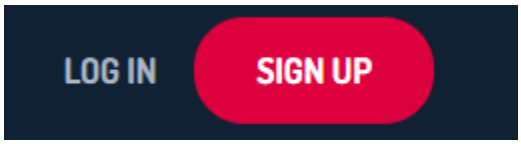
Picture 3, sign up success middle page.



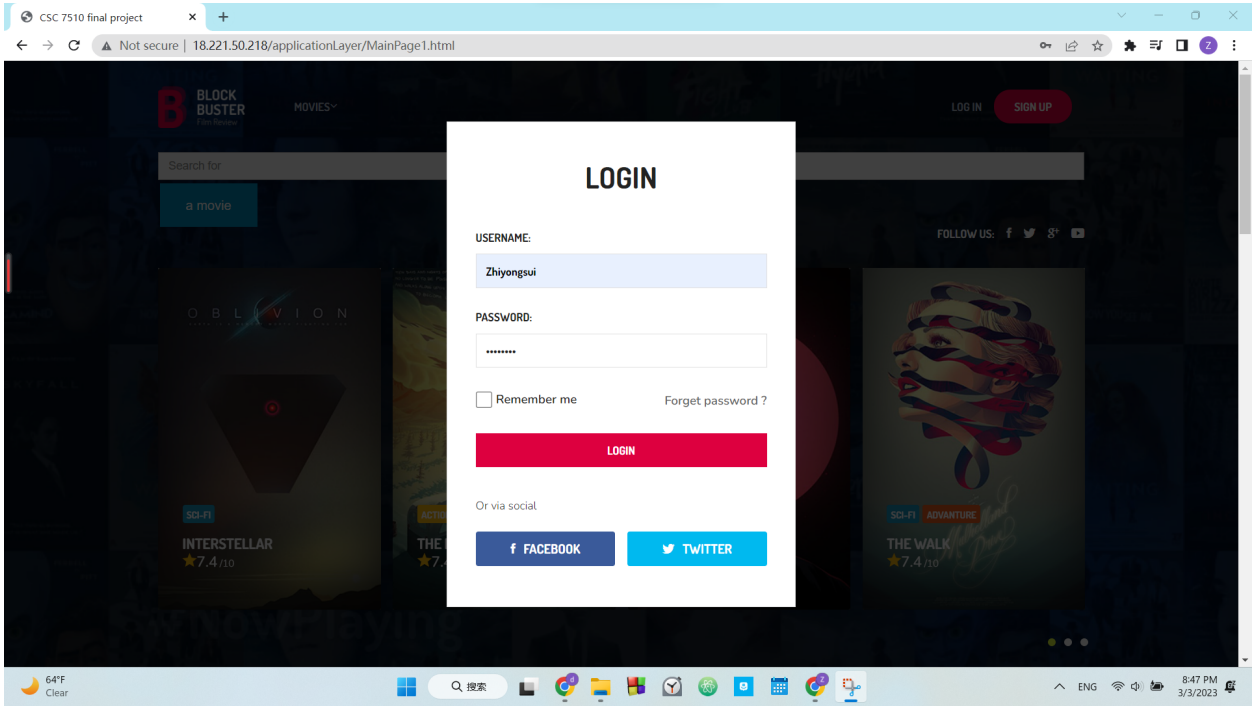
Picture 4, return to main page after sign up

Log in

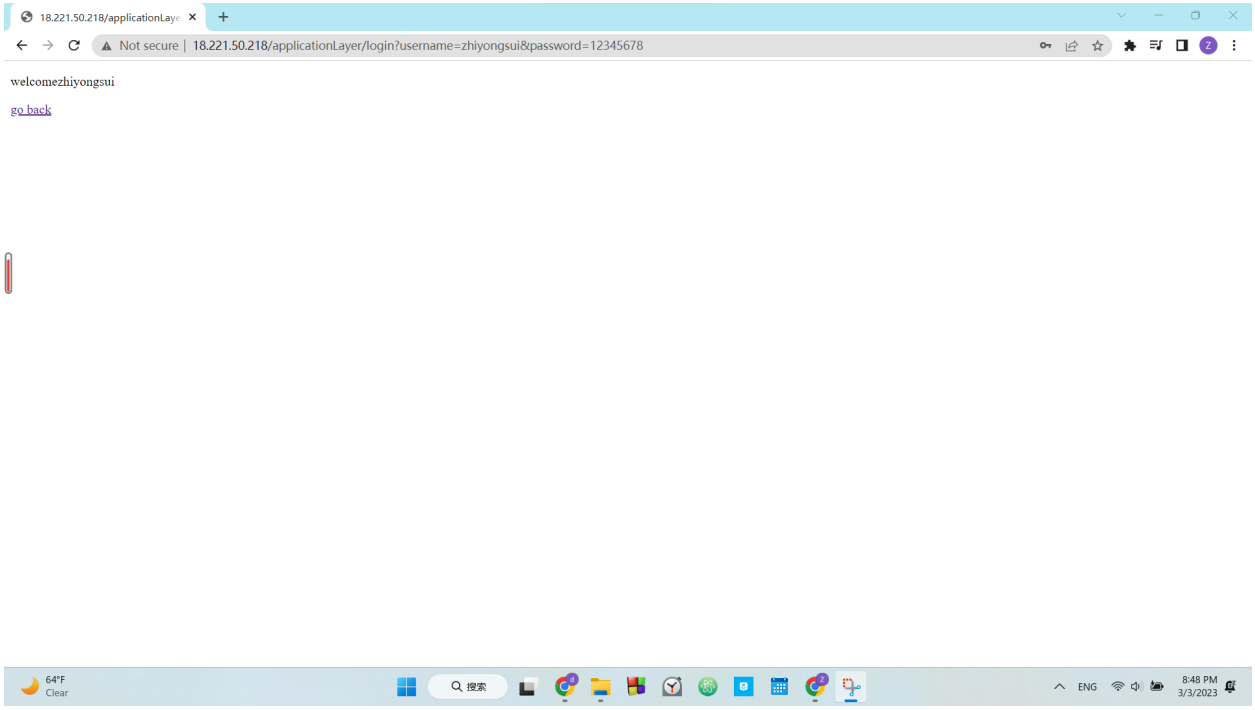
The login servlet receives a name and password for an account. If the information user input passed the system' s policies checking, it will be redirected to a new page showing that it was successful. Then the servlet checks the name and password of the database to see if the user is legal to login. The database and servlet allow users to have the same name but with different U_id, and then insert a row into the database with the email, password, and U_id. The following screenshot is an example of an email sent by the servlet upon registration.



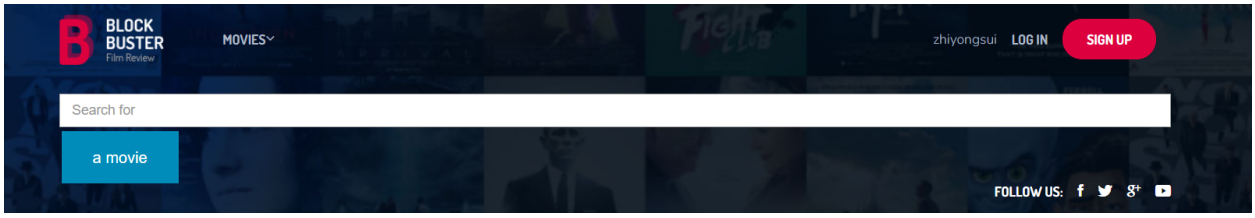
Picture 5, login icon



Picture 6, login pop up



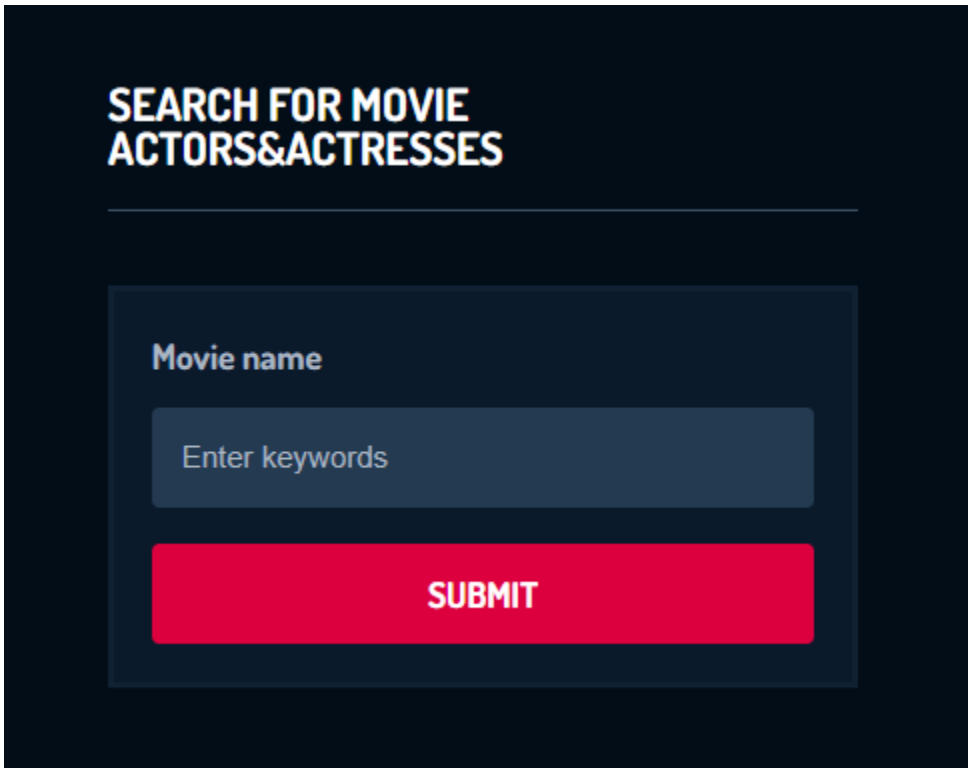
Picture 7, login middle page



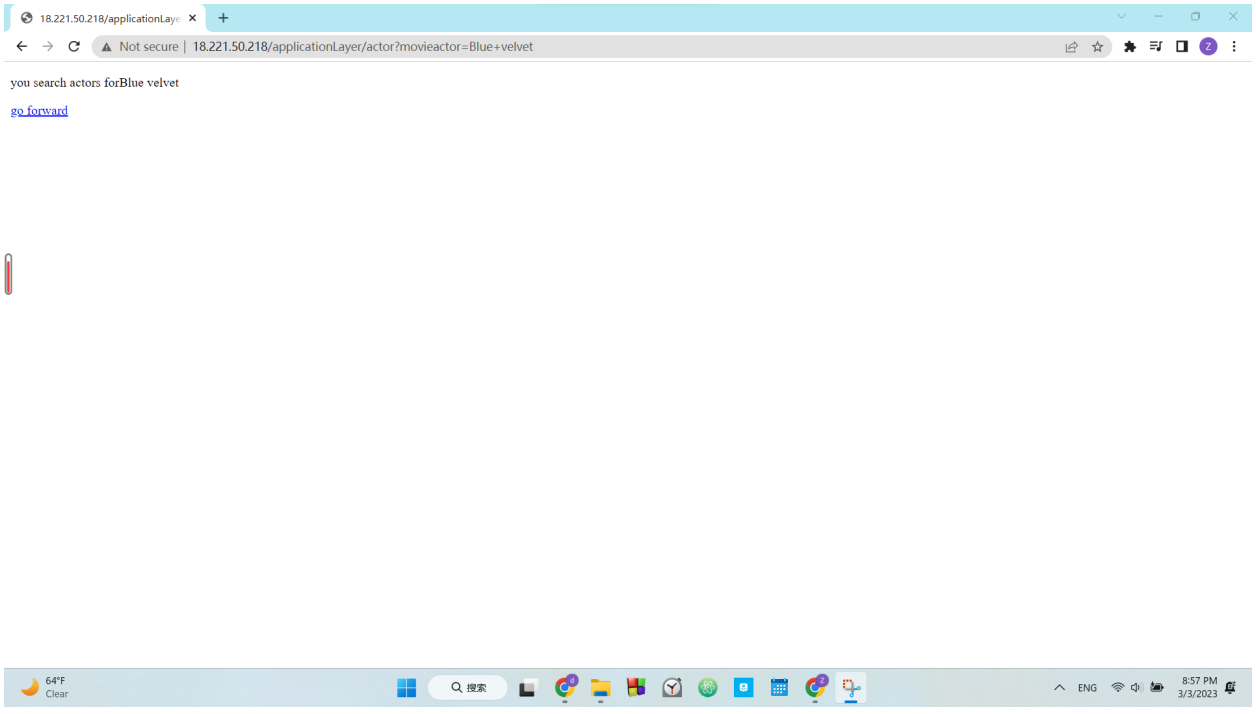
Picture 8, name show in the top bar

Search for actor

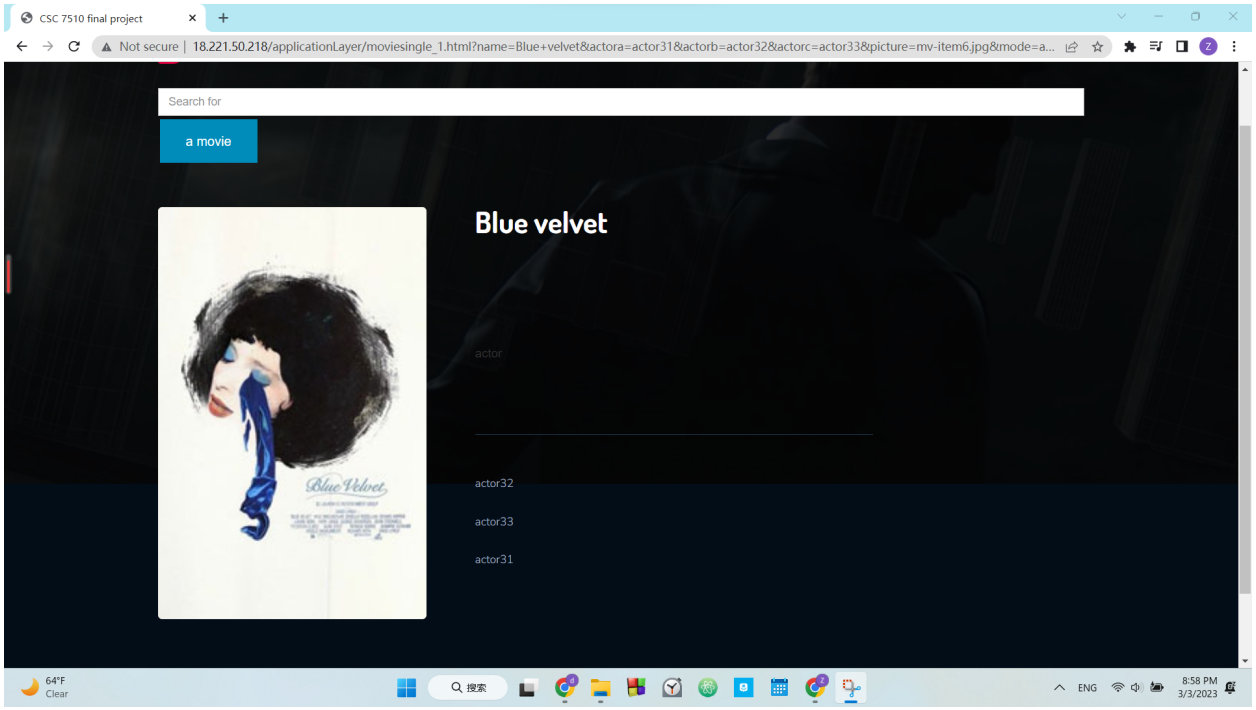
The actor servlet receives a name for the movie and retrieves the actor's name from the database. If the movie name is really in the database and the actor's name is not null. Application will redirect to page tell user success and have a go forward link to go to the result page. And there will be a new page showing the movie name, actors and picture.



Picture 9, search for actor



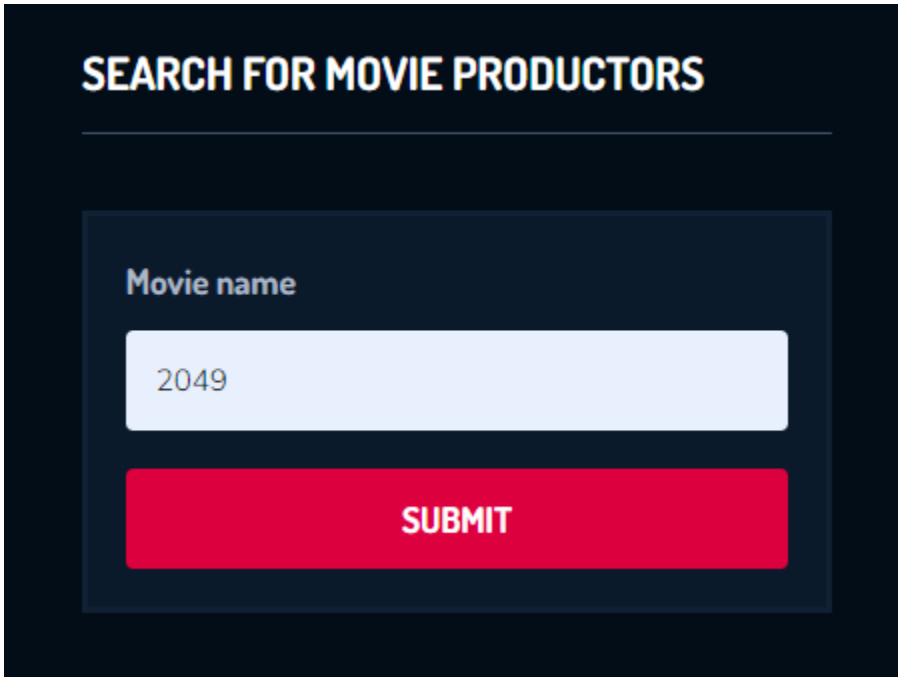
Picture 10, search for actor middle page



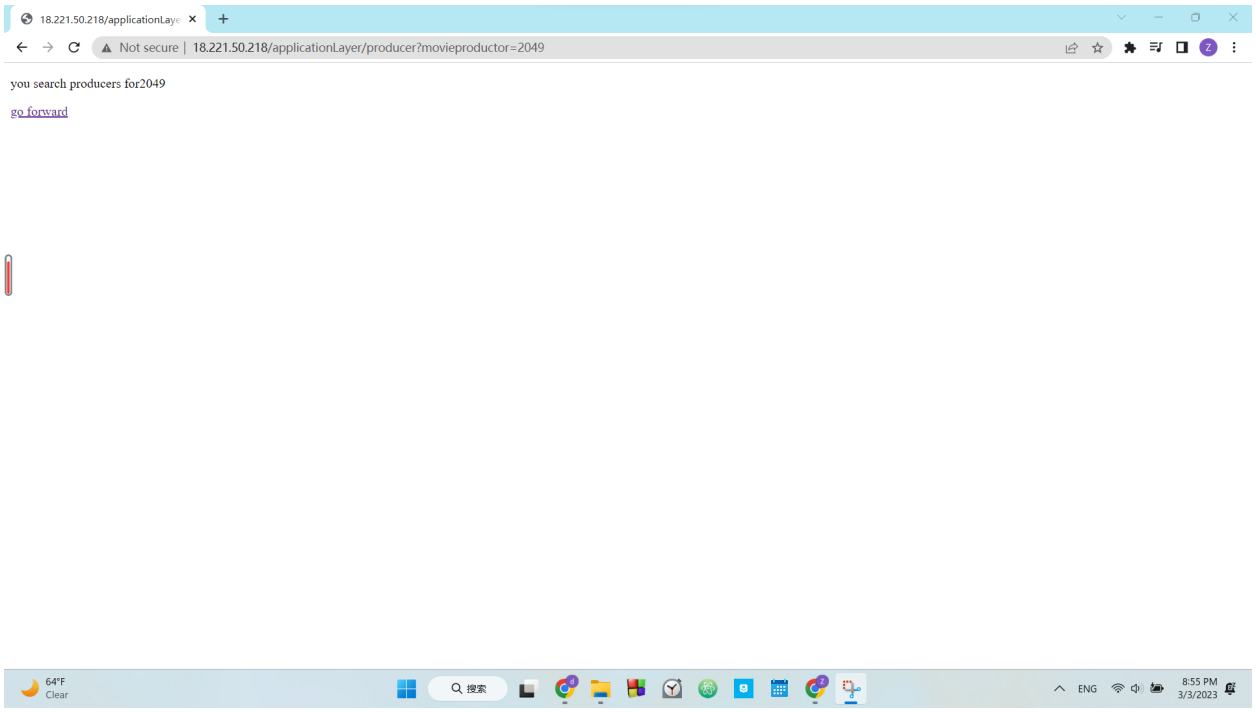
Picture 11, search for actor result page

Search for movie's producers

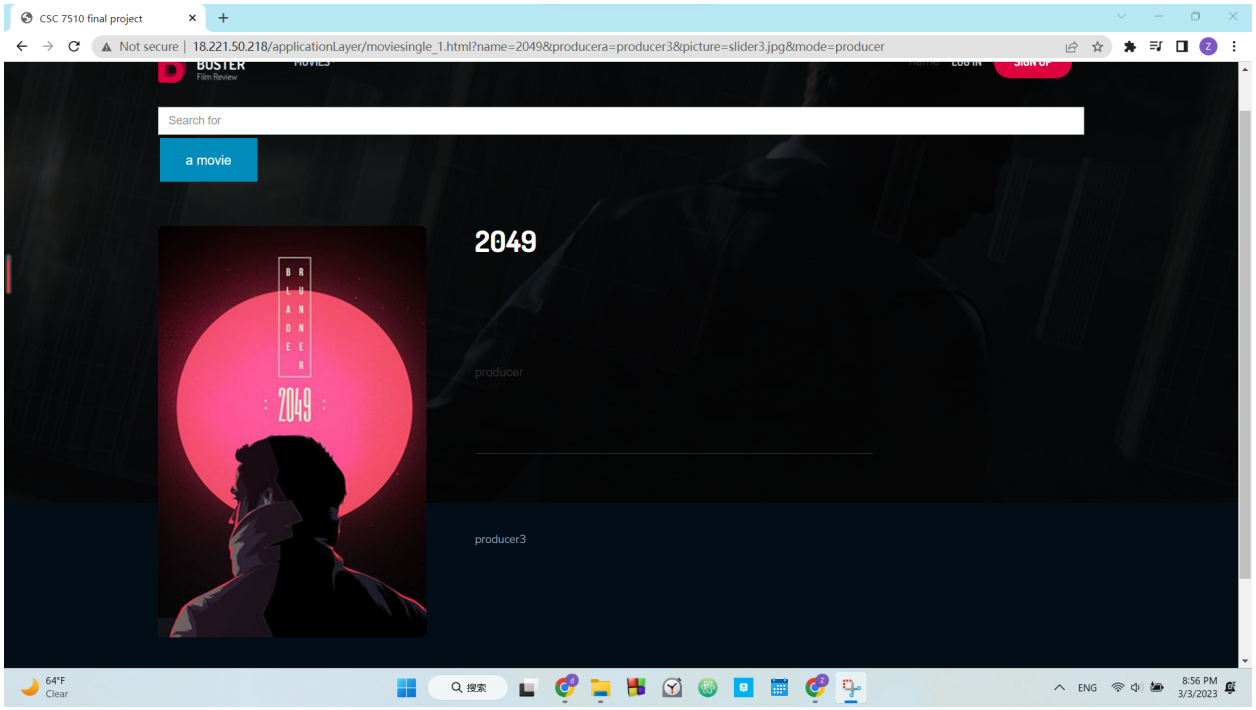
The producer servlet receives a name for the movie and retrieves the producers' name from the database. If the movie name is really in the database and the producers' name is not null. Application will redirect to page tell user success and have a go forward link to go to the result page. And there will be a new page showing the movie name, producers and picture.



Picture 12, search for producer



Picture 13, search for actor producer page

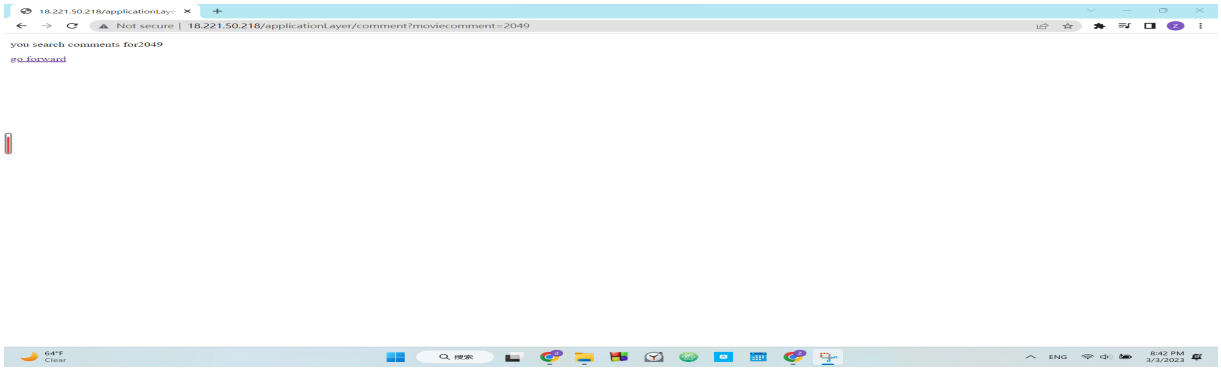


Picture 14, search for actor producer page

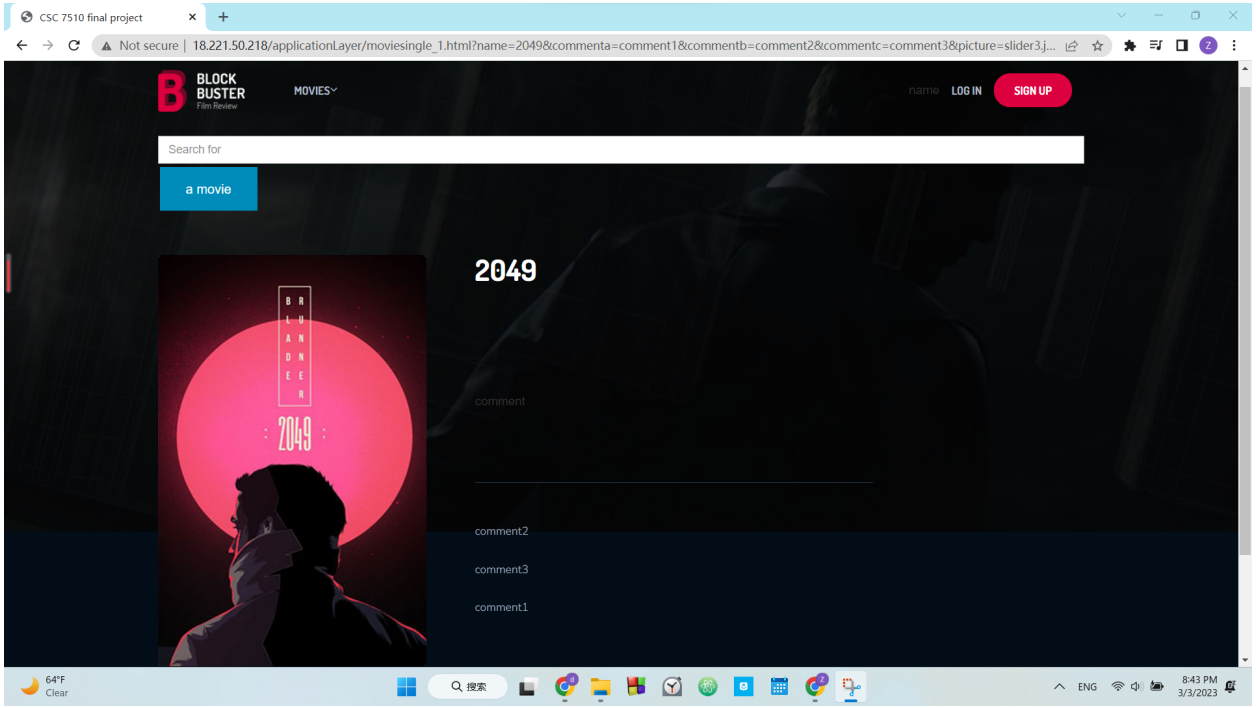
Search for movie comments

The comment servlet receives a name for the movie and retrieves the movie comments from the database. If the movie name is really in the database and the movie comments are not null. Application will redirect to page tell user success and have a go forward link to go to the result page. And there will be a new page showing the movie name, movie comments and picture.

Picture 15, search for movie comments



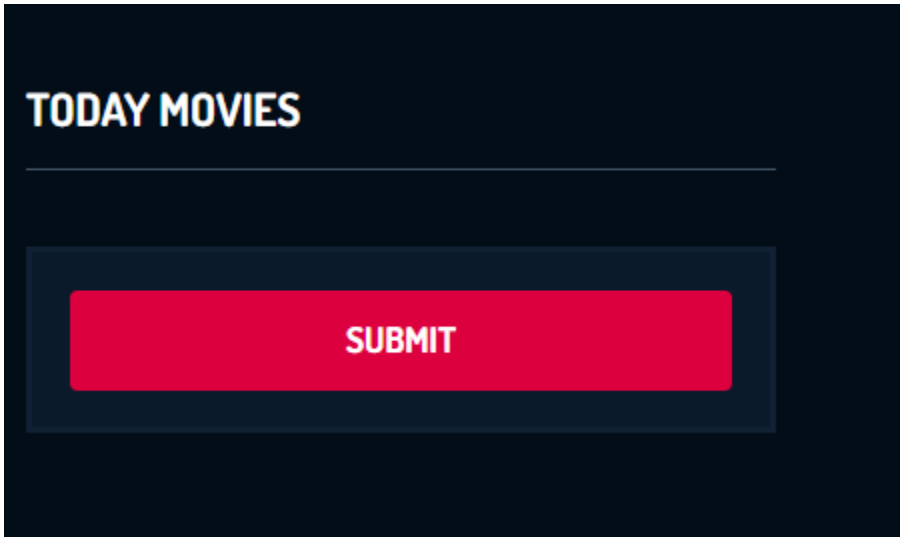
Picture 16, search for movie comments middle page



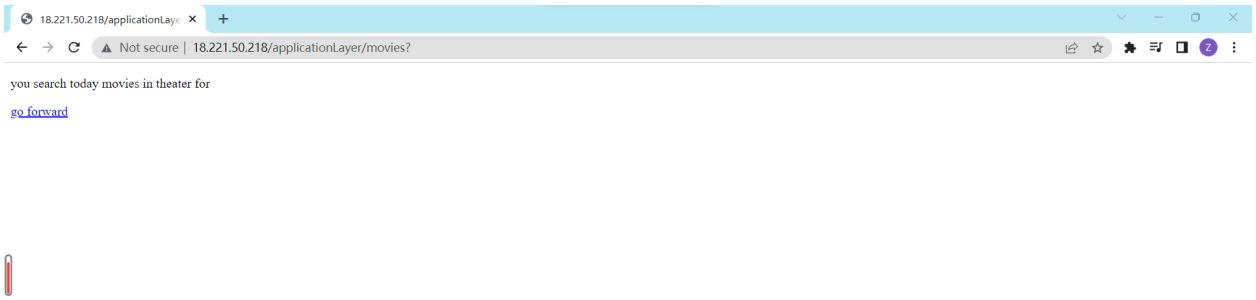
Picture 17, search for movie comments result page

Search for today movies

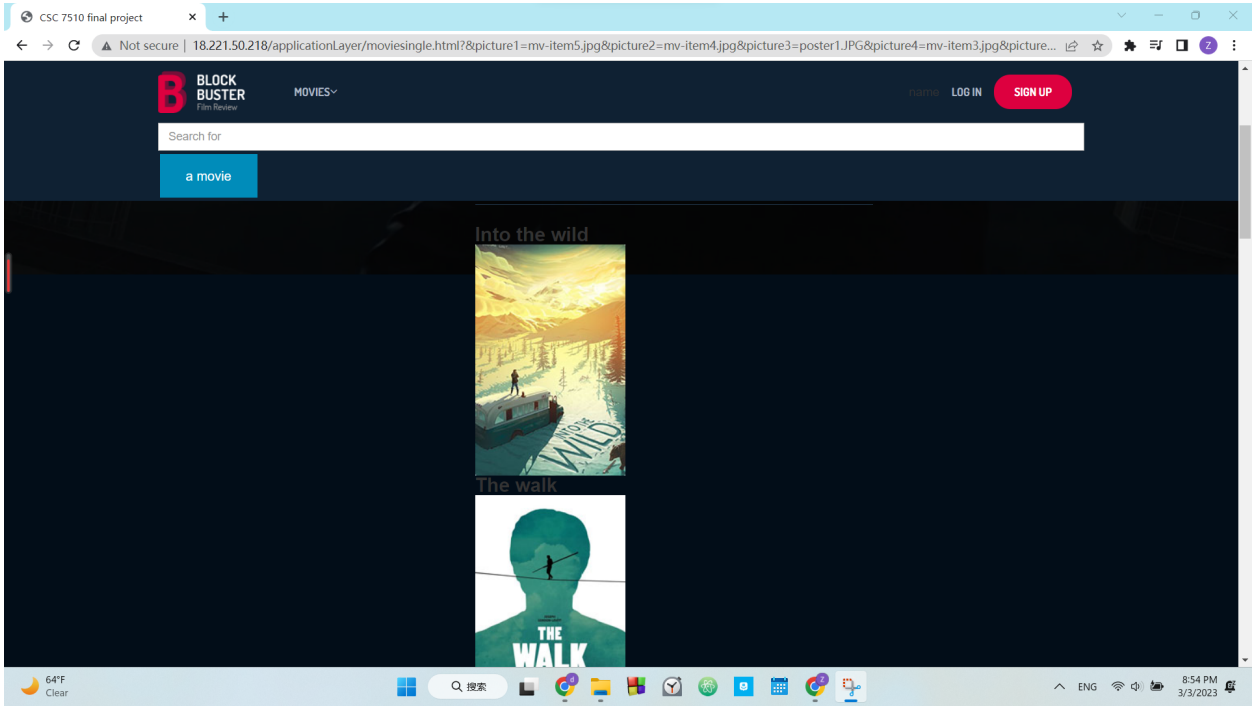
The today servlet receives a name for the movie and retrieves the today's movies from the database. If the movie name is really in the database and the movies are not null. Application will redirect to page tell user success and have a go forward link to go to the result page. And there will be a new page showing the movie name, picture.



Picture 18, search for today movies



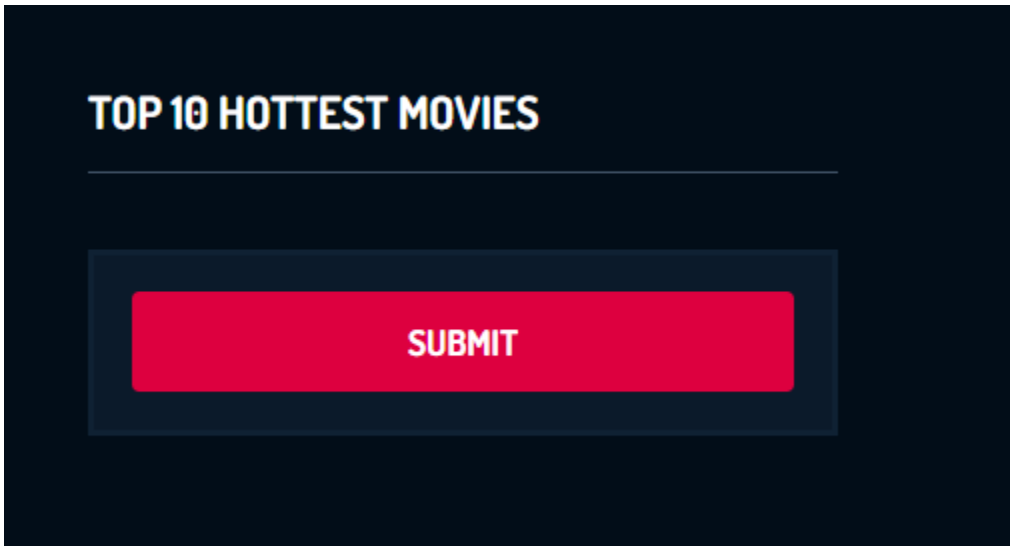
Picture 19, search for today movies middle page



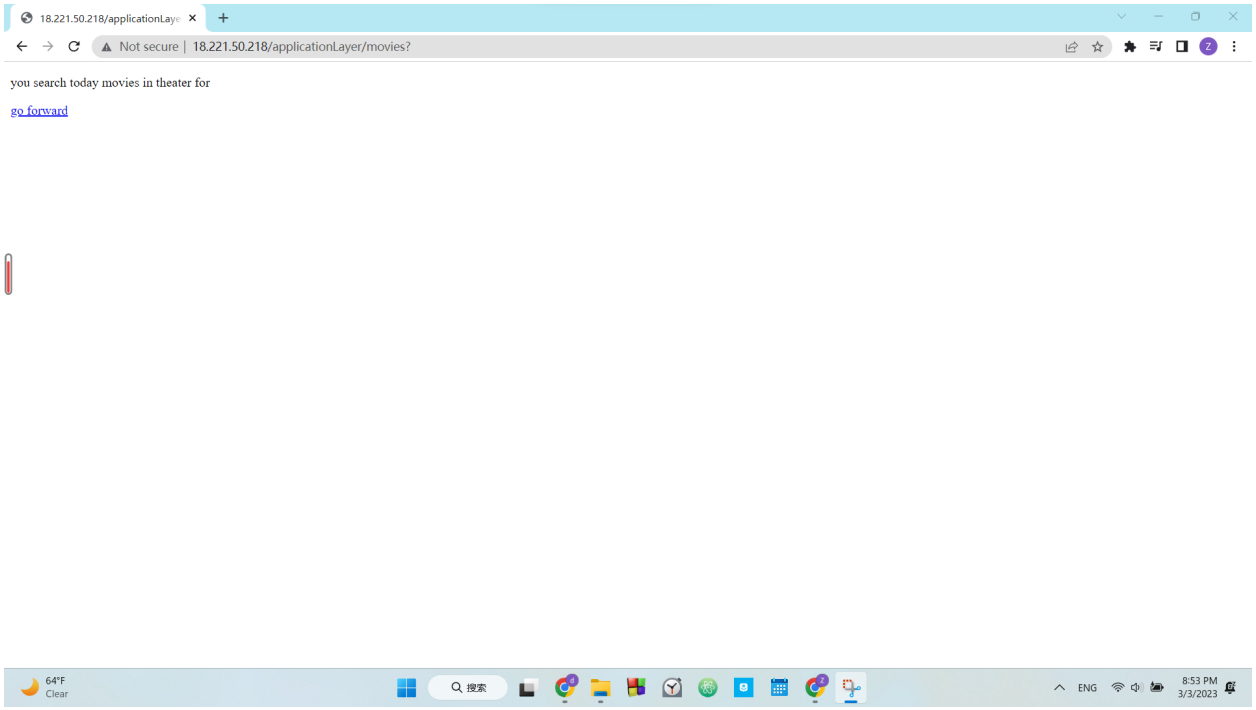
Picture 20, search for today movies result page

Search for top 10 hottest movies

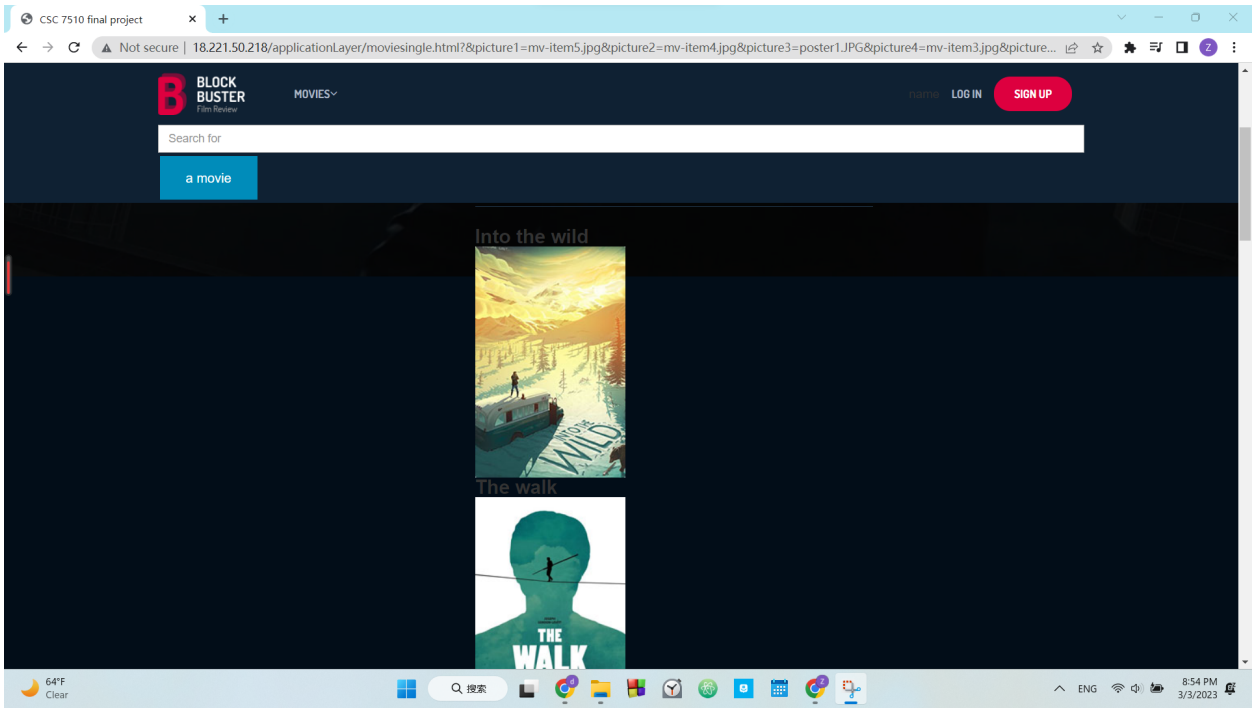
The movies servlet receives a name for the movie and retrieves the top 10 hottest movies from the database. If the movie name is really in the database and the movies are not null. Application will redirect to page tell user success and have a go forward link to go to the result page. And there will be a new page showing the movie name, picture.



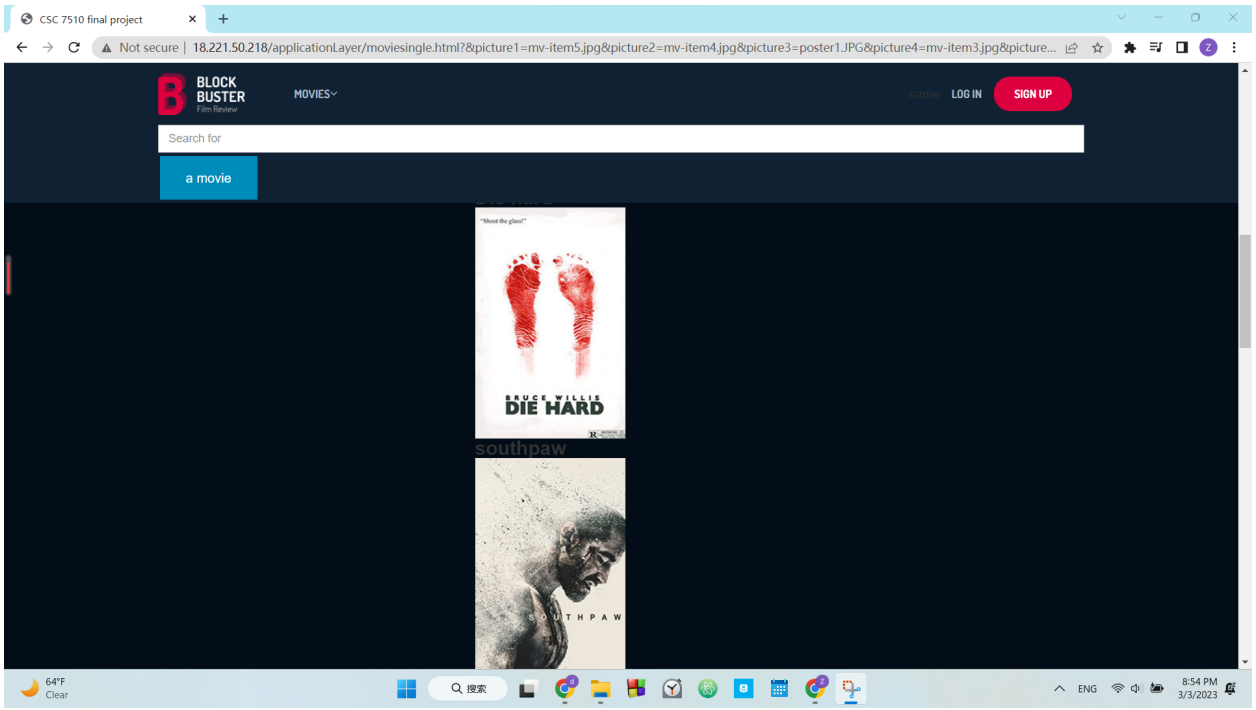
Picture 21, search for top 10 hottest movies



Picture 22, search for top 10 hottest movies middle page



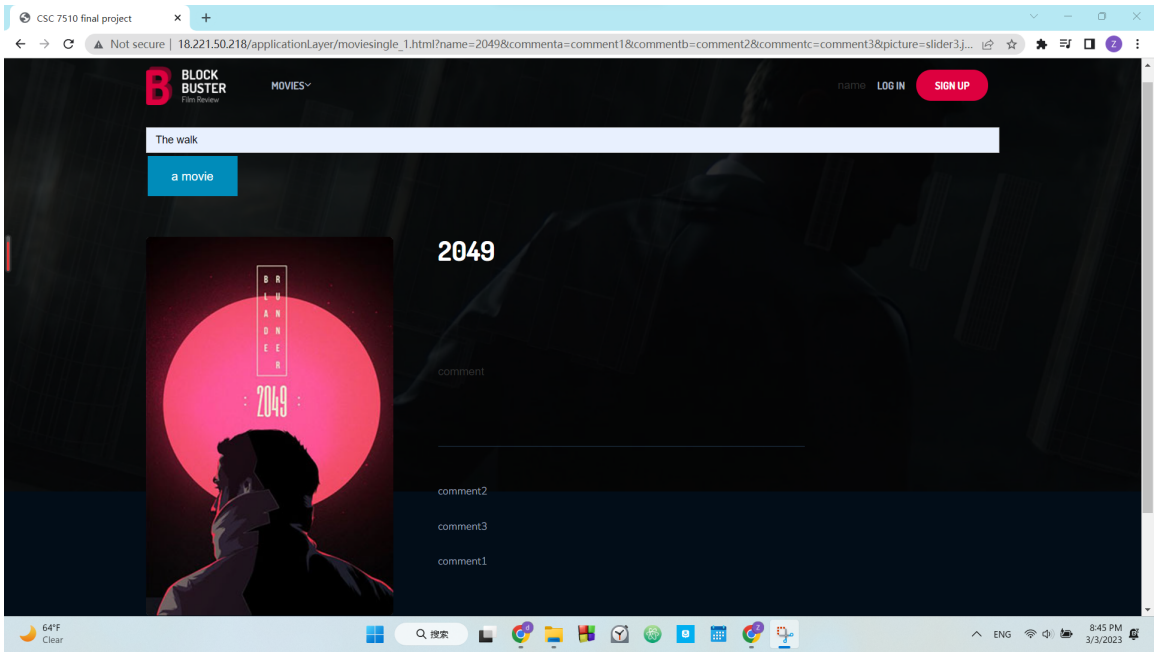
Picture 23, search for top 10 hottest movies result page



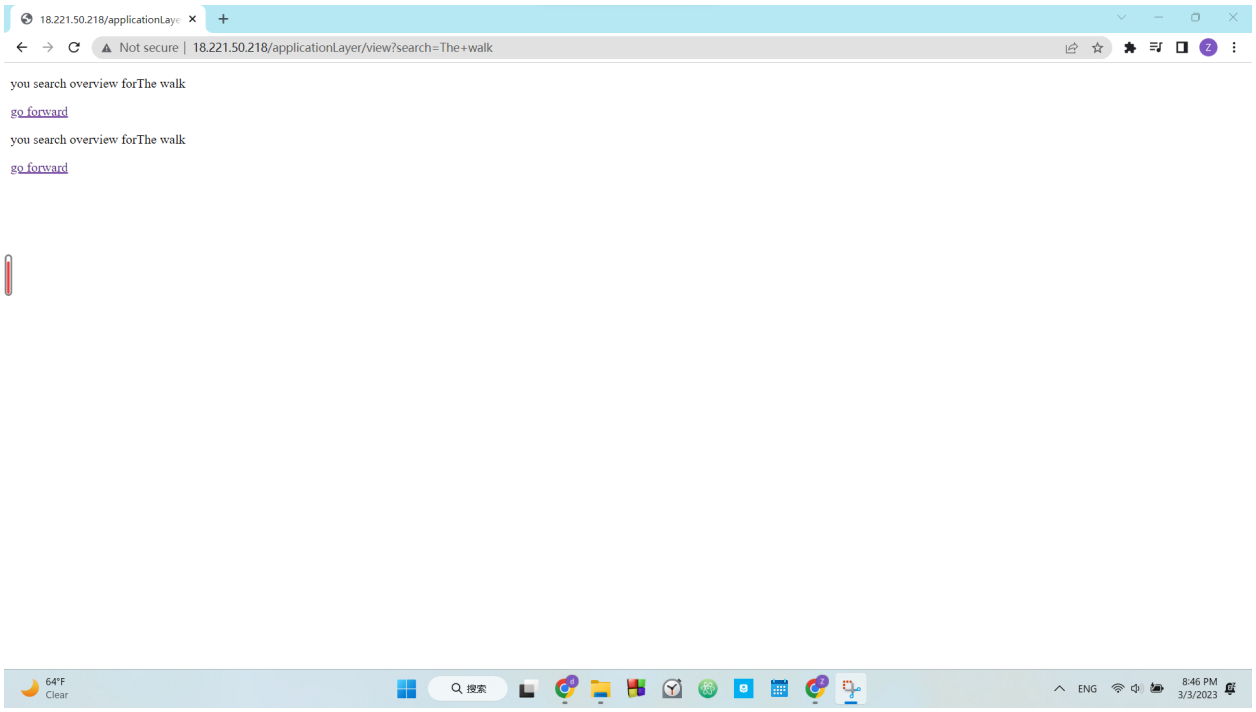
Picture 24, search for top 10 hottest movies result page

Search for movies view

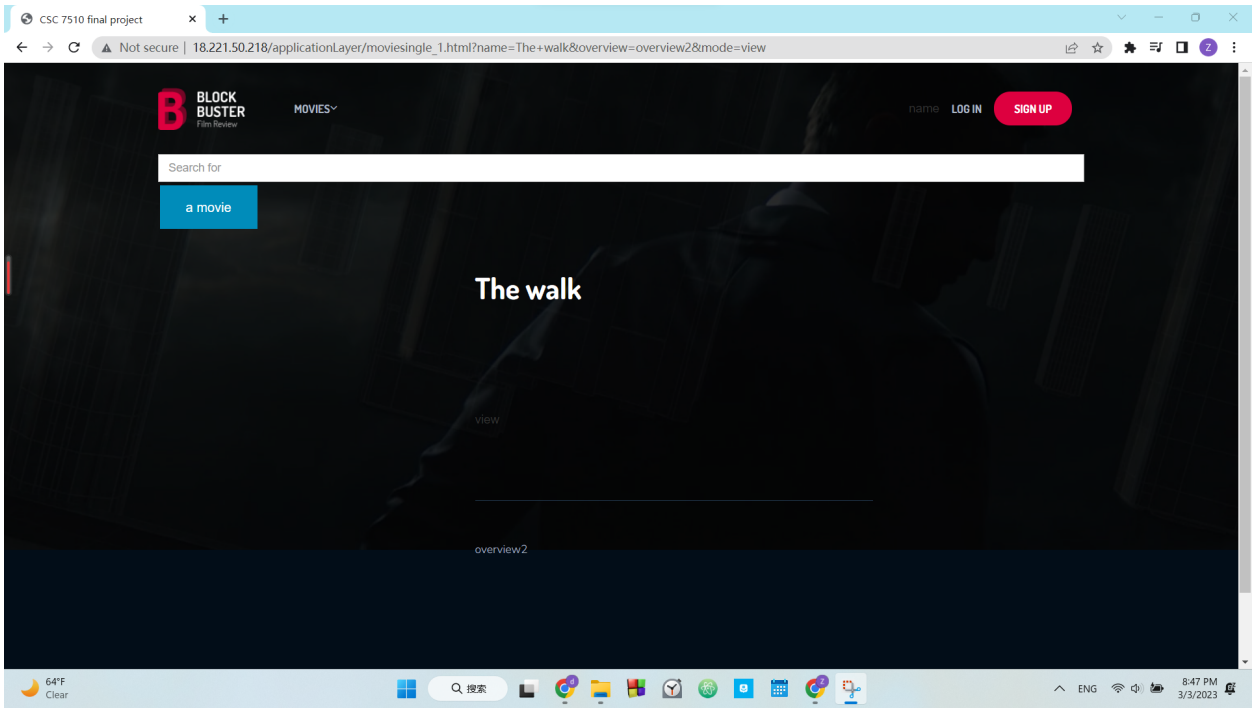
The view servlet receives a name for the movie and retrieves the view from the database. If the movie name is really in the database and the movie's views are not null. Application will redirect to page tell user success and have a go forward link to go to the result page. And there will be a new page showing the movie name and view.



Picture 25, search for movie view



Picture 26, search for movie view middle page



Picture 27, search for movie view result page

Screenshot of the response of each my servlet

The screenshots are concluded in the above section with explanations.

Conclusion

The final implementation of the IMDB-like website project delivers a fully functional three-tier web application hosted at http://18.221.50.218/applicationLayer/MainPage1.html that can provide the functions for user registration, login, and also other functions relatives to searching movies informations in database. I also made some styles for mobile or small screen devices, but the performance was not very good. All static contents are stored in Apache server and other dynamic contents are stored in Tomcat server. For the simplification of design and reducing the workload, I use some fake information just like “actor 1” or “ comment 1” to insert into the database and when after retrieving the information, this fake information will return.

For future work, I will add more real information to the database to replace the fake information and also add functions such as user account profile, add comments and views, and add movies.