# FLIXMIX

## Design Document

11/20/20

Version 2

**TeamBYR**

Brandon Christensen, Roshani Shiwakoti, Yuta Mogi

# TABLE OF CONTENTS

# I. Introduction

The main purpose of this application is to connect movie lovers, who can discuss movies, share recommendations and create reviews. The user can enter movies that they have watched, into a database and tag them per genre, ratings, and other tags that they may feel is necessary. The user can create a 'watch list' for movies that they are anticipating to watch, movies they have already watched. The system will also generate movies by recommendation, based on the user's preferences. Additionally, the user is able to share movies in a discussion forum where many other users get together to discuss, review and recommend movies. The user can add movies to their 'watch list' based on another user's post. We hope to use this application to connect with avid movie watchers, critics, and general cinema enthusiasts.

This document will discuss various aspects of the software design process:

- Section II includes Architecture Design.
- Section III includes Design Details and Implementations
- Section IV includes Testing Plans
- Section V includes any references that were used for this document.

## Document Revision History

Rev 1.0 2020-11-02 Initial Version (Iteration 1)
Rev 2.0 2020-11-20 Second Version (Iteration 2)

# II. Architecture Design

## II.1. Overview

Model:
The database 'movies.db' stores the states and data for our movies. This includes the movies genres, description, poster, and title. The database will also store information about which users have watched, or plan to watch these movies (many-to-one relationship). Movies.db will also store information about the users, including their names, password, and email. There will be a page dedicated to allowing the user to change that information. There will also be a page to allow new users to register an account, which will then be stored in movies.db. There will be a page to add movies to the database as well. The 'My Movies' page will access and display all the movies in the database. There will be a search bar that will display a given movie from the database. There will also be a sort function in the 'My Movies' section that will access the database and then sort the movies to be displayed by either the title or genres. Whenever any changes are made, a popup at the top of the page (flash message) will alert the user if the change was successful.

View:
We have spent a lot of time considering the way our application will look, including the color scheme (green and blue), as well as the general layout. Most of our program can be reached through the nav bar. There are tabs for "My Movie", "Search", "Add Movie", and "User Name". The user can reach all the main features through these options. When you click "My Movie", the user is sent to a page that displays all the movies in the database. "Search" sends the user to a page that allows the user to look up a specific movie, which then sends the user to that movie's specific page. "Add Movie" sends the user to a page where they can input a movie's details, and then add it to the database. "User Name" which will

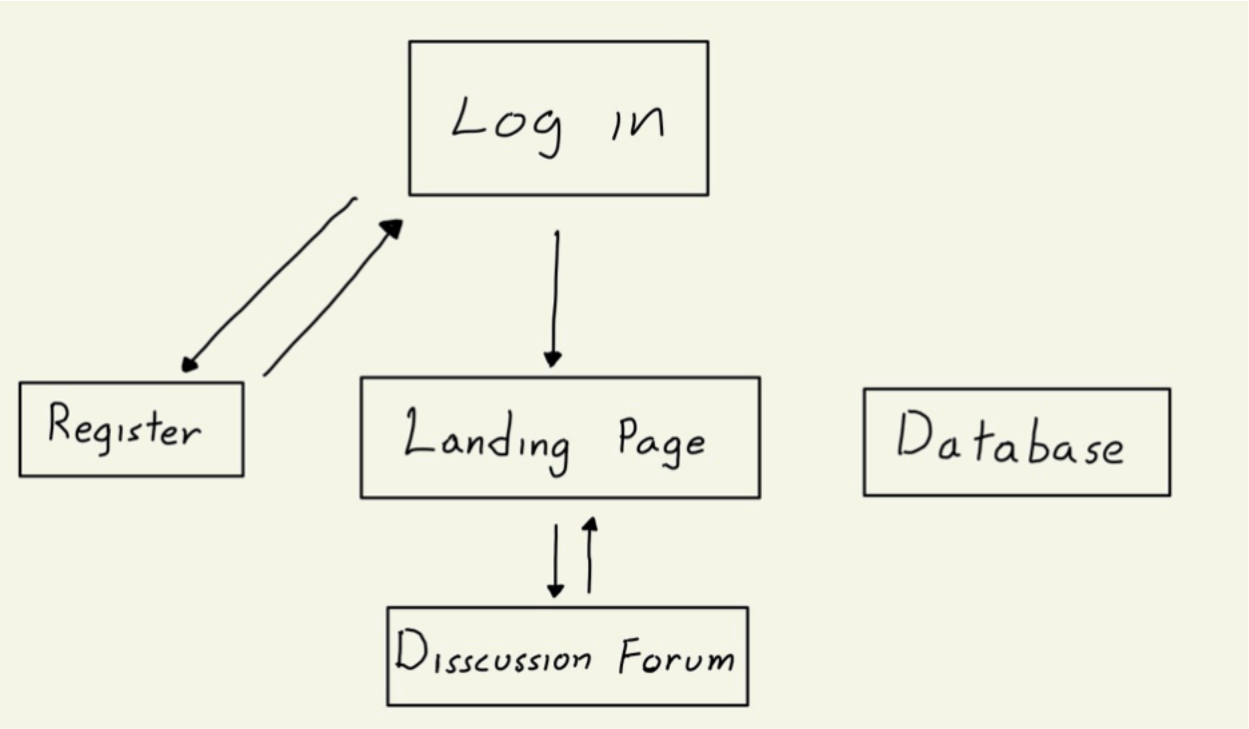
actually display the username of the logged in user, will take the user to an edit page, where they can update their profile's information. We have all of this set up as a pull model.

Controller:
Whenever a button on the nav bar is pressed, the user is using a hyperlink from "base.html" to go to a specific page. Any sort of add or search button is actually a submit request, which "routes.py" checks for in order to either check a database, or add to the database. When sorting the movies, when a user sorts by a specific item, after they click search, the program will update the page. There might be more than one page worth of movies on "My Movies", if so then when the user scrolls down, more movies will be displayed.

Our program is made up of website pages and a database which contains users and their movies (movie ratings, watched movies, etc.). The structure will be very similar to the smile project that we have completed earlier this semester. We will have a login and register page, and once you login, you can look at your movies. Similar to the smile project, you can also see other people's reviews, like smiles functionality to look at other people's posts.

The rationale is the same that the smile project used. We want the program to flow in a logical manner, where you must be logged in to see content and to make posts. You must go to the page that you want (i.e. movie reviews, watch lists, etc.), and then the program will take you there.

Everything will be coupled to the users account (in the database). The users can look up how other people reviewed their movies, and then write their own. Other users can then comment on the reviews. These reviews and comments will all be tied to the posts, which will then be tied to the users account.



The subsystems are fairly self explanatory. There will be the login page, where you can login if you already are a user, or be redirected to a register page, where you can become a user. If you are already a user and you successfully login, then you are redirected to the landing page. If you successfully register as a new user, then you are redirected back to the login page. The landing page will have most of the main functionality of the program. This is where you can add movies to your watchlist, get movie recommendations, and search for new movies. There will also be a button to go to a discussion board page from here. This is where you can see other user's posts and reviews. This will all be stored in a database.

## III. Design Details

### III.1. Subsystem Design

| Methods | URL | Description |
|---|---|---|
| **GET, POST** | /login | Login to account from this page. If user data exists in the database, then the user will be able to successfully log in, otherwise user will have to register their information. |
| **GET, POST** | /register | Creating account for users who don't have an account yet. Then the user will be redirected to the login page, where they can sign in to their account. |
| **GET, POST** | /logout | Logging out from the account, will be redirected to the login page. |
| **GET, POST** | /index | Where the user's personal "my movies" will be displayed. User can access search engine, and filter through their movies. |
| **GET, POST** | /discussion | Where the discussion forum will be |
| **POST** | /addMovie | Adds movies to the database of movies. User can select genres, add a title, add a description of the movie to the database. (to-do: add a movie poster) |
| **GET** | /search | Searching implementation. User will search for a movie title and genre and will be redirected to the results page where the new filtered movies of that title will be displayed. |
| **POST** | /results | Will display the filtered results of the movie title search which is a redirect from the /search route |
| **GET, POST** | /profile | This page contains user profile, displays the user's profile picture (to-do), their 'about-me' section, their username through the /user/<username> route which is linked on this page, as well as post personal posts |
| **GET, POST** | /user/<username> | Allows the user to edit personal information, upload and change the profile picture (to-do), change their username. Once the changes have been submitted, user will be redirected to their /profile and the changes will be displayed. |
| **GET, POST** | /discussion | Discussion forum here, users will be able to interact with one another, post movies, likes, comment etc. |
| **POST** | /comment | Posts a comment to a movie, once comment is made, will be redirected to the discussion post where the comment will be displayed and uploaded |

### III.1.1. Login Page

This will be the first page that the users using this application will be greeted with. It will display the name of the app along with text boxes for users to input their username/email and password. This will interact with the database as that will be the way to tell if the user and password matchup. There will also be a remember me checkbox, which will bypass this screen if checked next time the user comes to this app. This will also have to be remembered in the database. Finally, there will be a hyperlink for new users that don't have an account yet, and it will send you to the register page.

### III.1.2. Register Page

There will be four text boxes for users to enter information into. They will be as follows: (1) username (2) email (3) password (4) re-enter password. The app will check to make sure that the username and email have not been used before by consulting the database. The app will also make sure that the password and re-entered password are matching. If both of these tests are passed, then the user is created, entered into the database, and then redirected to the login page.

### III.1.3. Landing Page

This is the main section of the entire app. There will be a top navigation bar, where you can log out quickly, or search your own database for movies. There will be a button for you to enter new movies. This is something we need to consider further, but this will probably need to redirect the user to a new page that lets them enter the movie details. The movies that have been entered into the watched list will be displayed. The user will be able to scroll through the movies they have seen. There will also be a randomly generated recommendation list, set up in a similar fashion. We want all the movies that are displayed here to be clickable, and they will send you to that movies discussion forum page. There will also be a watch list that contains movies the user wants to watch next.

### III.1.4. Discussion Forum Page:

Finally, there will be the discussion forum page. This page will display a specific movie, and display that movie's details. There will be textboxes to make comments, and these comments will be displayed. There will also be buttons to add this to the users watch list. Tags, genres, etc. will be displayed here too. This will all be stored in a database.

### III.1.5. Profile Page

This page displays the user's profile, their profile picture, their about me section and any posts that the user decides to upload, in a blog format.

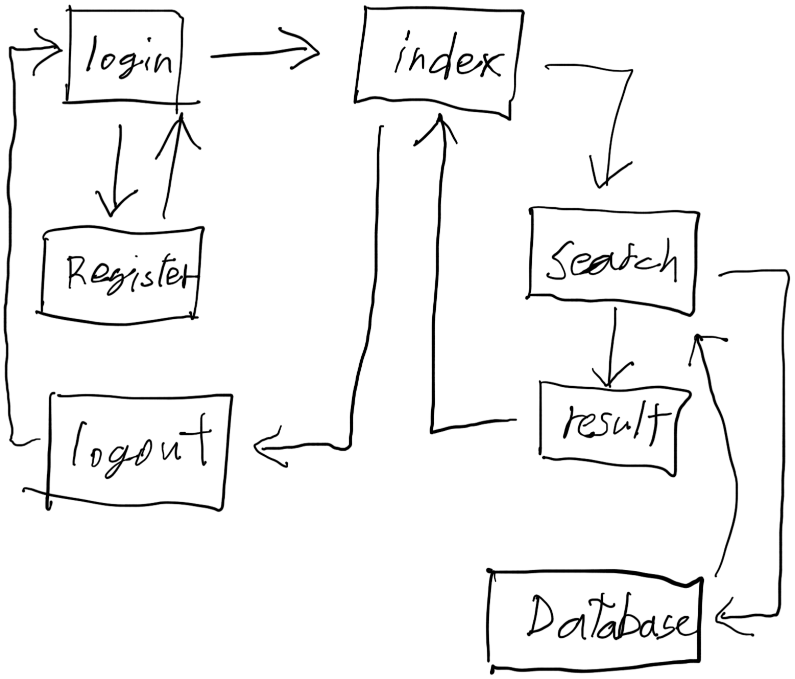
### III.1.6. Results

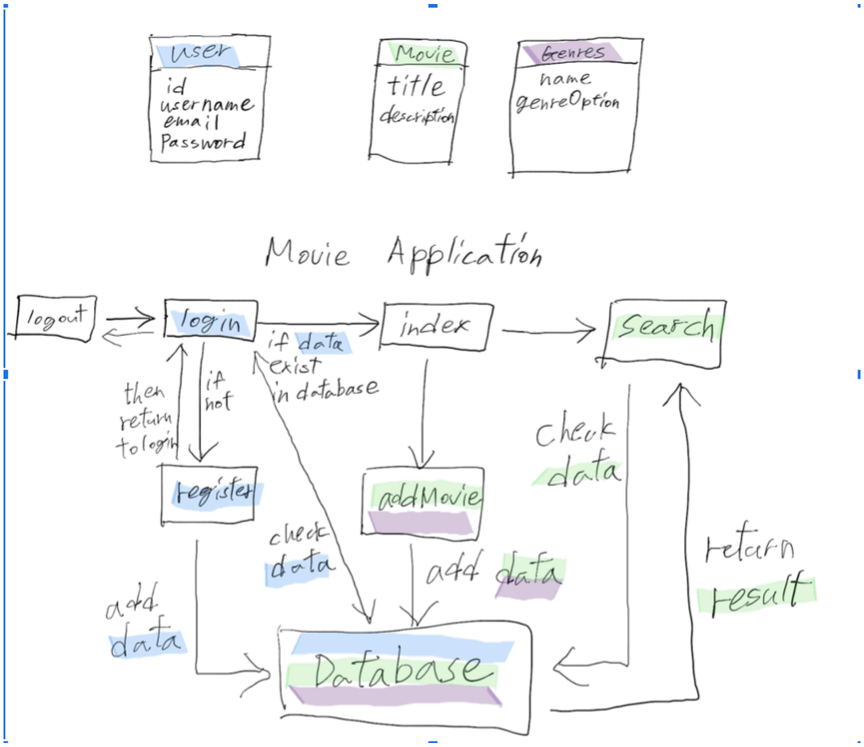
This page will upload the filtered results from the search. It will filter the user's search requests and display only the posts that are being searched for from the database.

### III.1.7. User Page

This page allows the user to edit personal information about themselves, and will redirect the page to the profile page where the changes will be shown automatically.

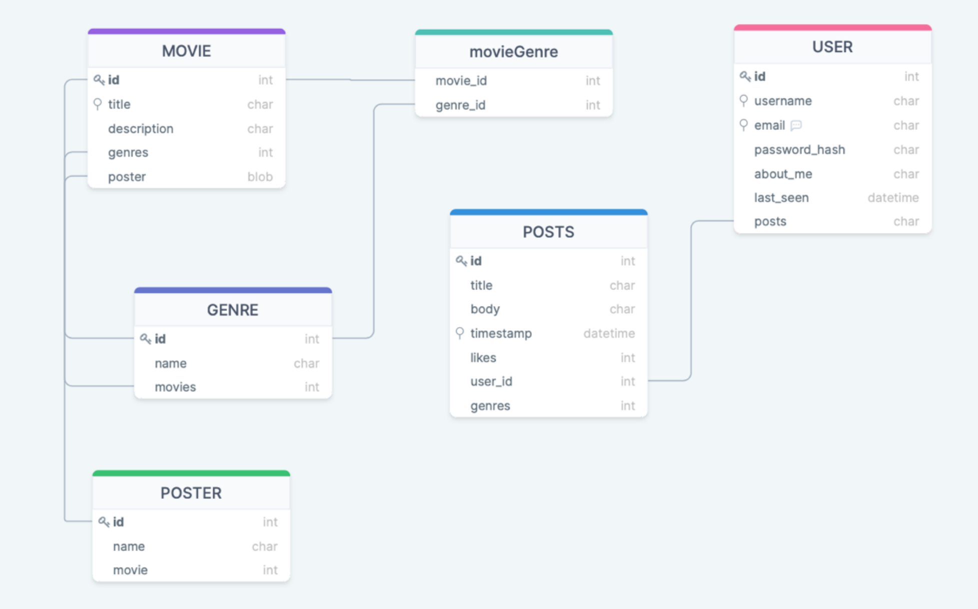
UML Class Diagram:





## III.2. Data design

Schema



| MOVIE | |
|---|---|
| id | int |
| title | char |
| description | char |
| genres | int |
| poster | blob |

| movieGenre | |
|---|---|
| movie_id | int |
| genre_id | int |

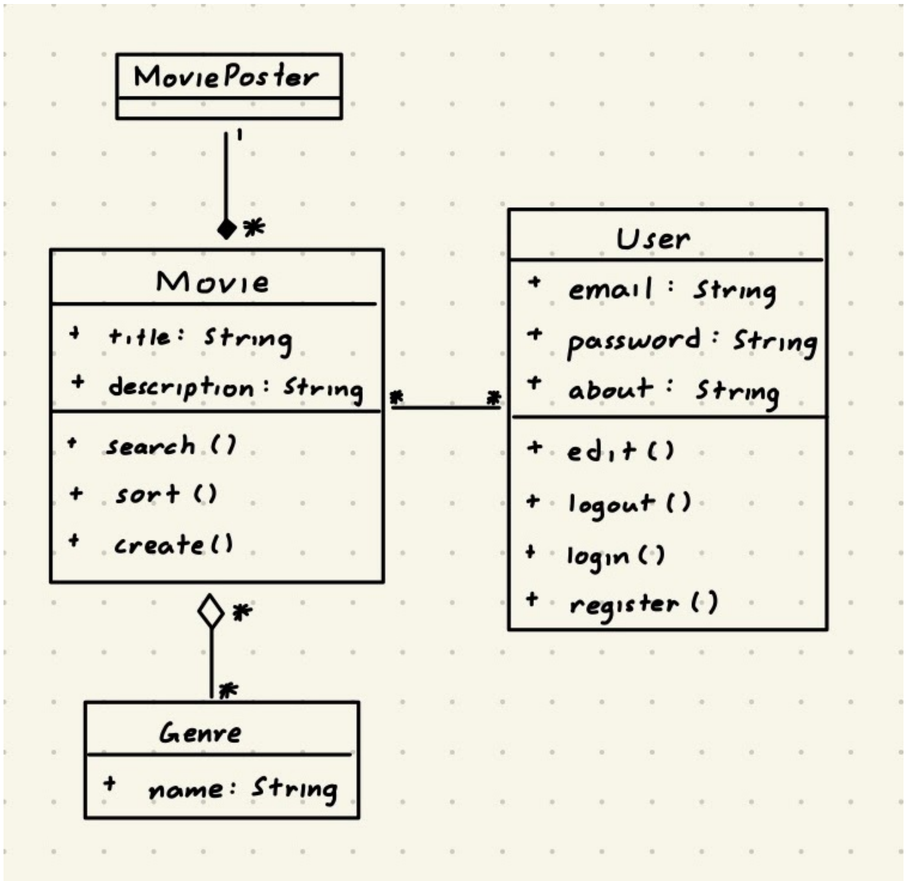
| USER | |
|---|---|
| id | int |
| username | char |
| email | char |
| password_hash | char |
| about_me | char |
| last_seen | datetime |
| posts | char |

| POSTS | |
|---|---|
| id | int |
| title | char |
| body | char |
| timestamp | datetime |
| likes | int |
| user_id | int |
| genres | int |

| GENRE | |
|---|---|
| id | int |
| name | char |
| movies | int |

| POSTER | |
|---|---|
| id | int |
| name | char |
| movie | int |

UML diagram showing the associations:



Here we list our database schemas.

User:

- Id – this field contains an auto generated id which acts as an identifier for the user
- Username – this field contains the user’s username which they will use to log-in to the application
- Email – this field contains the user’s email, which will be used to register an account with the application
- Password_hash – this field will contain the user’s password, which will allow the user to log-in to their account, and is also required to register an account.
- about_me – this field stores information the user uploads to the ‘user’ page
- last_seen – this datetime field contains the date that the user was last seen on the app
- posts – this contains the users posts

Movie:

- id – contains the id which acts as the identifier for the movie
- title – contains the title of the movie
- description – contains the description of the movie, that the user manually uploads
- genres – contains genre tags for the movies
- poster – will link the uploaded movie poster to the movie

Genre:

- id – this field contains the id for the genre tags
- name – contains the name of the genre
- movie – links to the movie which the genre pertains to

Posts:

- id – an identifier id for the post
- title – the title of the post
- body – the body of the post
- timestamp – time that the post was updated on the discussion forum
- like – the likes on the post by other users
- user_id – connects the post to the user database to
- genres – contains the genre tags

Poster:

- id – id of the movie poster that the user will update to the database
- name – contains the name of the poster
- movie – links the poster to the movie

### III.3. User Interface Design

For iteration 2, our team has retained the interface design from iteration 1. This includes a login, and a register (sign-up) page. This user interface implements use-case number seven from the Project Requirements Document. The user can successfully register an account with a username, email and password, which in turn gets registered into the database. After a valid account is registered into the database, the user is then able to log-in to their account and view their feed, and the forum.

The username and password fields in the log-in page are checked for validity, and are cross checked to ensure that it exists in the database as valid log-in credentials. Similarly, the sign-up page, where the user creates an account will be checked for field validity. All input fields must be completed, and a valid email must be used. Once the user is logged in, the 'logout' button will appear in the navigation bar, where the user will be able to log out of the account.

Additionally, the 'remember me' field in the log-in page allows for the system to remember the user's username and password so that they can log-in easily the next time around.

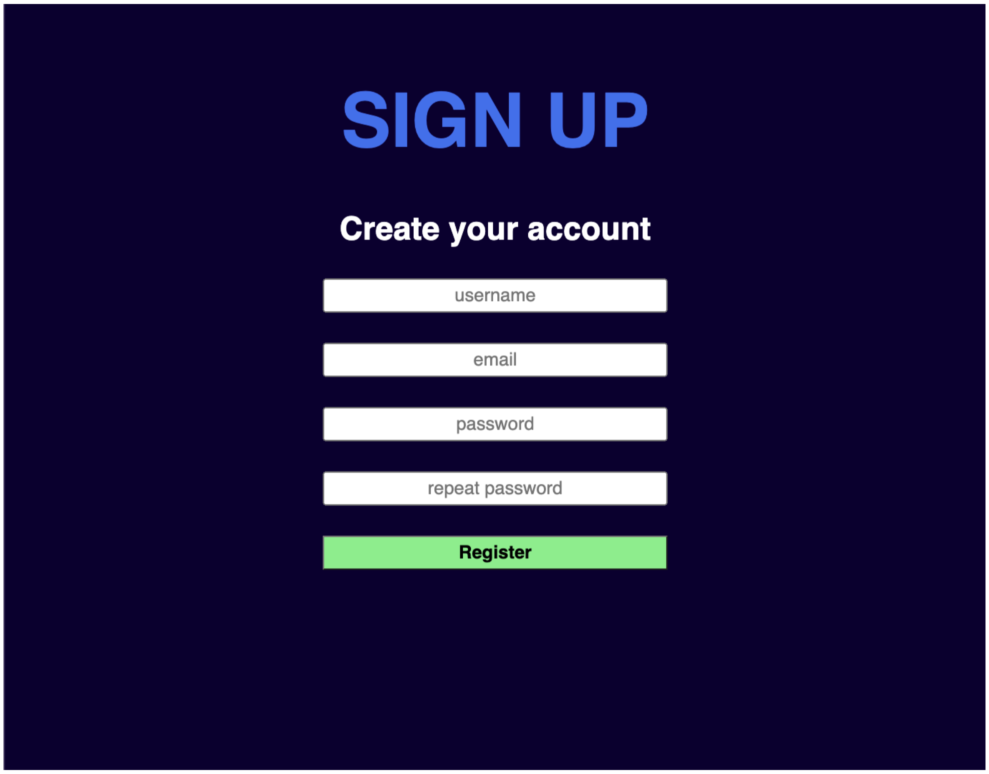
Visually, we decided that the color scheme for this application should reflect our logo—various shades of blues. Our color scheme reflects this. Additionally, we have chosen a very simple, user friendly system for the input boxes. The minimalistic, yet concise layout should make the website less busy and overwhelming. For future iterations, we are planning on including slight changes to the visual aspect of this interface. Such as a proper background, and a more design focused navigation bar. Additionally, we are going to implement an 'about us' section which explains to the user the purpose of this application.

New features we have implemented include the My Movies page, where it will display the user's personal movies, and a search bar and filter system which will eventually display the filtered and searched movies to the user. The title being inputted into the search bar must be an identical match to the titles stored in the database. The use case being satisfied is "the user will be able to add movies to their watchlist".

Another feature we have implemented is the Profile page. This page will eventually display the profile picture of the user, once we get that going. So far, the user is able to edit their username, which will immediately update the username to the database. Additionally, the user can add an about-me section where they can write a paragraph about themselves. Lastly, the date-time will be continuously updated to show the last time that the user was active on the website.

In the future, we will be working to figure out how to upload images to the website, this includes a movie poster which will be displayed on the My Movies page, and a profile picture which will be displayed on the user profile.

Our interfaces:



SIGN UP

Create your account

username

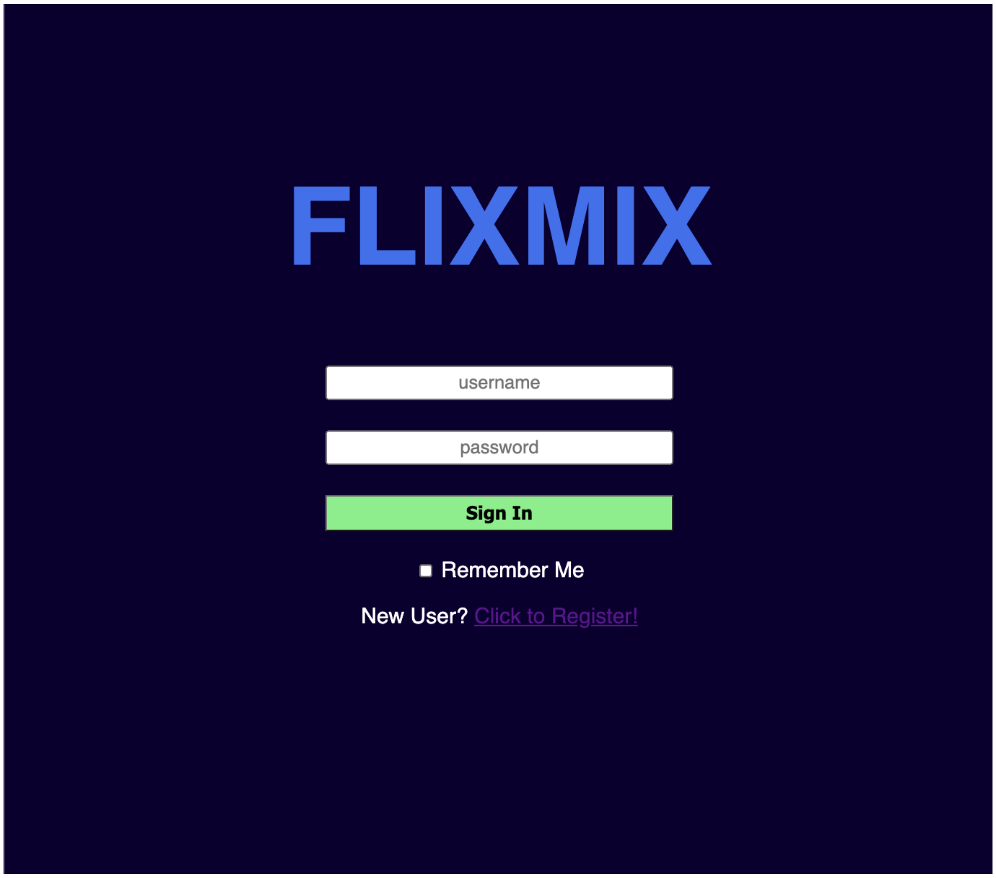
email

password

repeat password

Register

Sign Up page



FLIXMIX

username

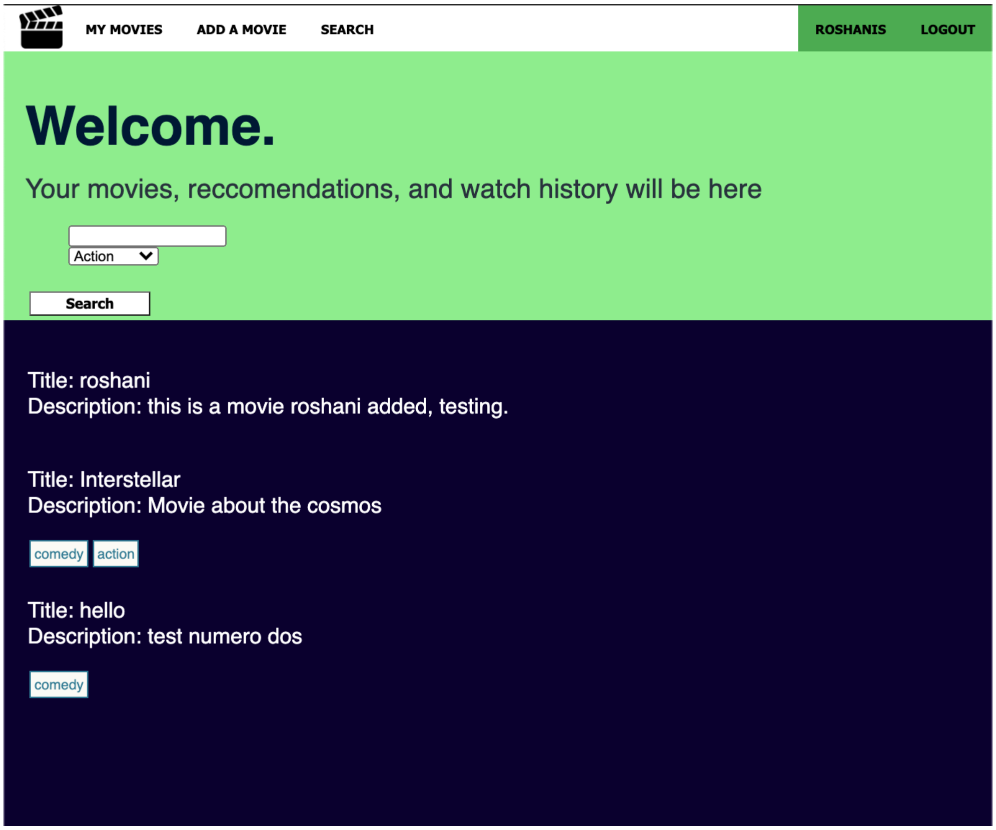
password

Sign In

■ Remember Me

New User? Click to Register!

Register an account



Welcome.

Your movies, reccomendations, and watch history will be here

Action

Search

Title: roshani
Description: this is a movie roshani added, testing.

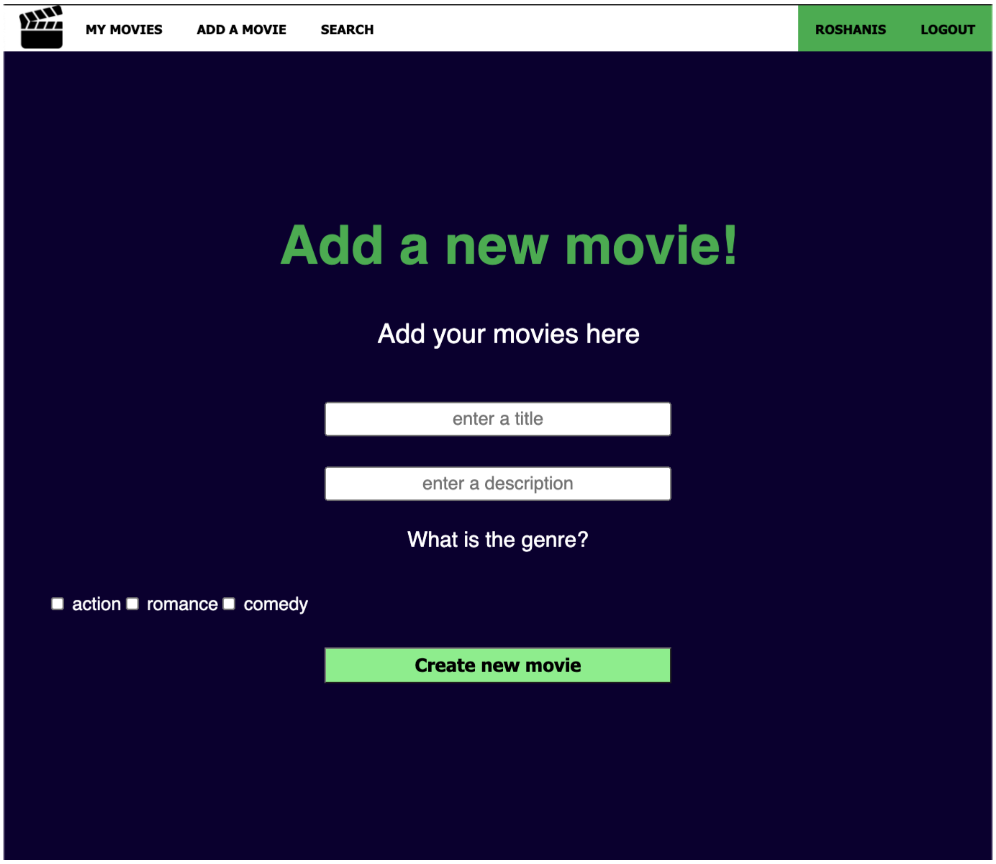
Title: Interstellar
Description: Movie about the cosmos

comedy action

Title: hello
Description: test numero dos

comedy

Landing page. Needs to be styled better. Fix grouping, get search engine and sorting to display results correctly. Search/sort function does not work as of now. We want to be able to display poster images here as well. The movies will be displayed based on watched movies (archived), Want to watch, and recommendations, with a horizontal scroll.



Add a new movie!

Add your movies here

enter a title

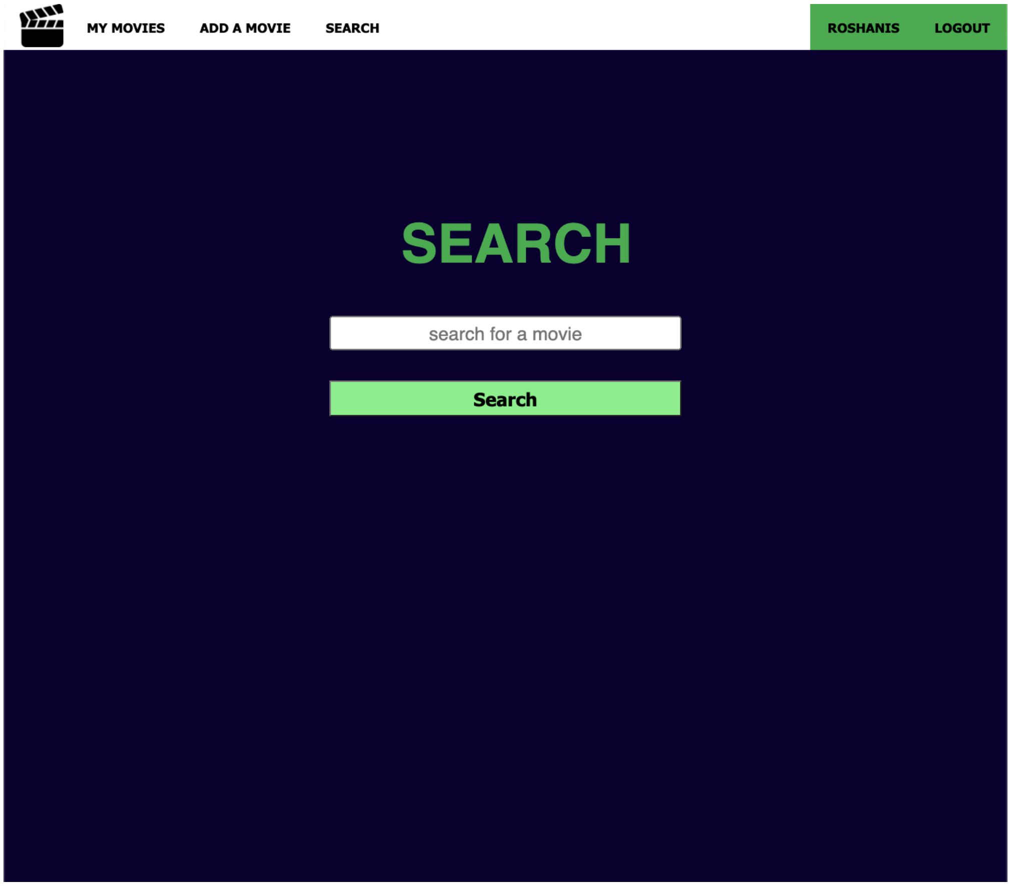
enter a description

What is the genre?

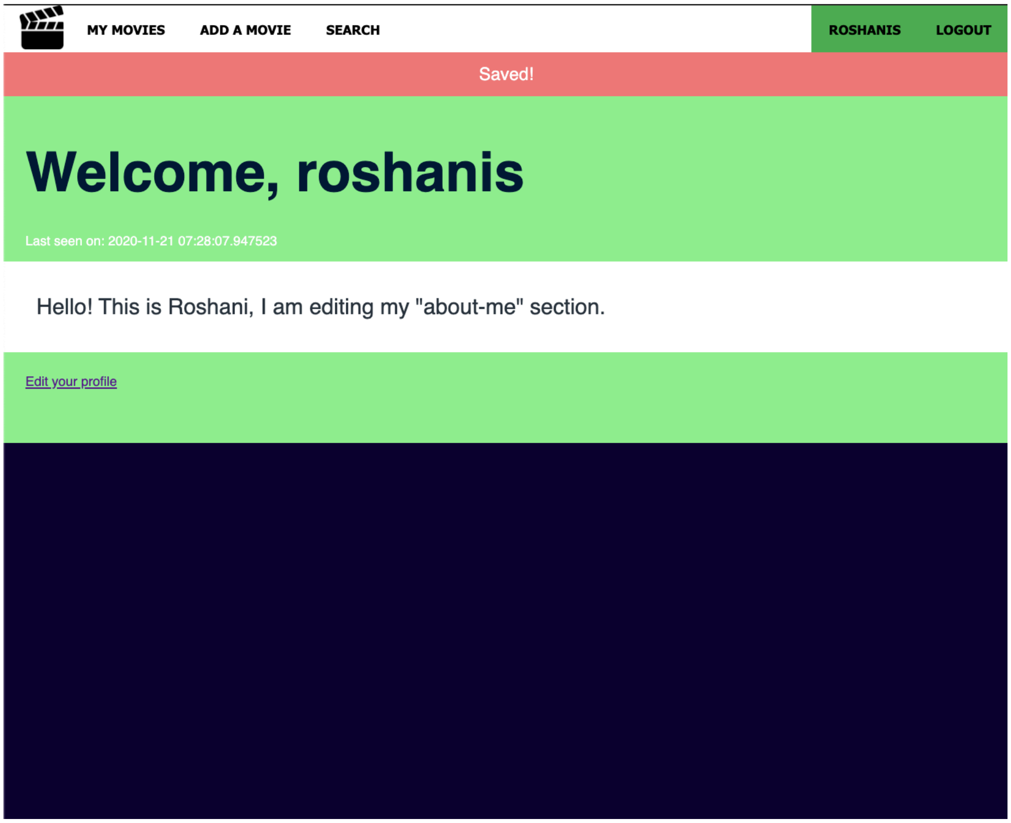
□ action □ romance □ comedy

Create new movie

Add new movie page. Allows user to add movie title, description and genre. Don't understand why the genre checklist will not center and align with the rest of the form. Need to fix that. We also want to highlight the page we are on in the navbar.



Search page. This does the same thing the landing ‘My Movie’ sort/search functionality does. We will either stick with this page, the search/sort function on the landing page, or we will implement the search bar up in the navigation bar (most likely possibility)



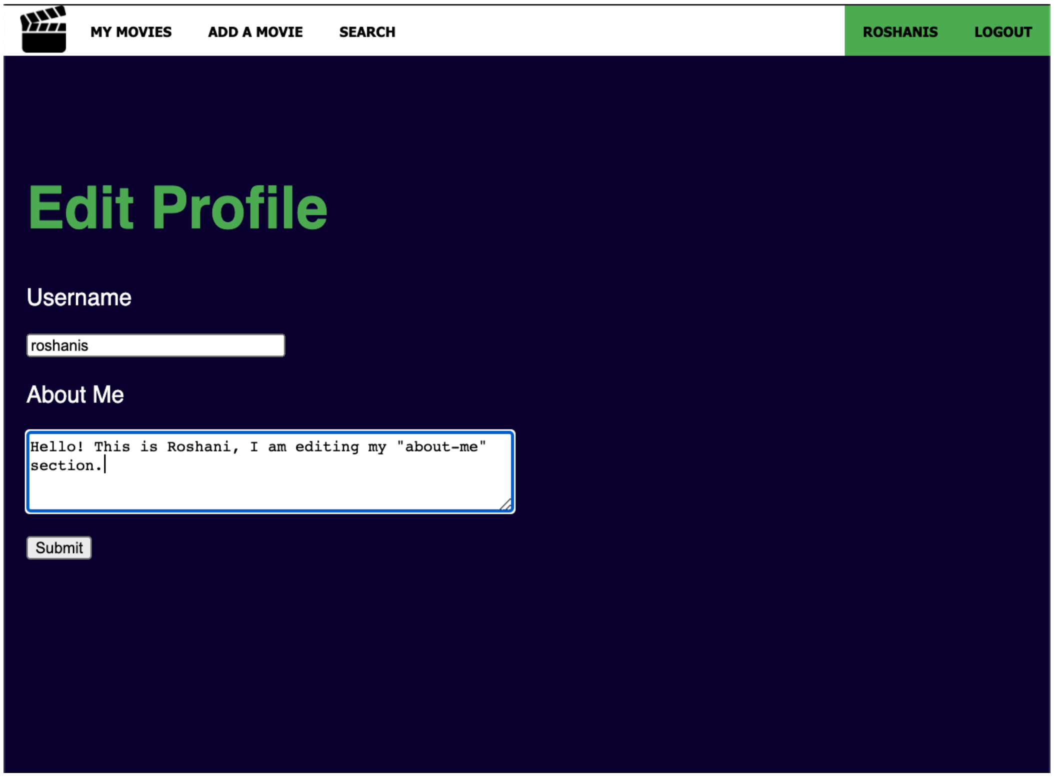
Saved!

Welcome, roshanis

Last seen on: 2020-11-21 07:28:07.947523

Hello! This is Roshani, I am editing my "about-me" section.

Profile Page. We need to edit the user interface. We want to be able to display a profile picture, and display the blog posts. The posts the user puts in the discussion post will also appear on the profile.



Edit Profile

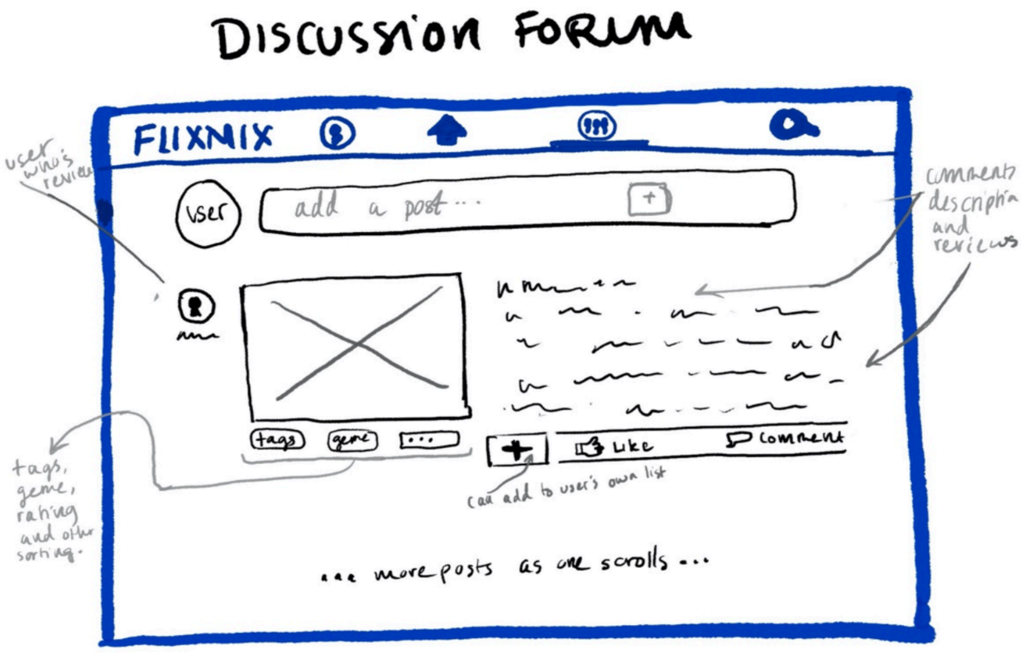
Username

roshanis

About Me

Hello! This is Roshani, I am editing my "about-me" section.

Allows user to edit their profile, including their username. We will be implementing a password and email change too


DISCUSSION FORUM
FLIXMIX
user who's review
user
add a post...
comments description and reviews
tags
genre
...
Like
Comment
tags, genre, rating and other sorting.
can add to user's own list
... more posts as one scrolls ...

Last interface we need to display by the end of iteration 3. Discussion forum, with this relative design/scheme

## IV. Progress Report

The main difficulty we faced during this iteration was getting a search bar implemented, and allowing the user to upload pictures and eventually being able to store them in the database. This is very difficult, and we still haven't found success. We hope that by the final iteration, we will be able to knock these two problems down. Additionally, we found that some User Interface implementations we had decided on in the beginning were not practical to implement in the way that we had intended. Therefore, for logical and user-experience reasons, we are re-organizing links and outlines. It was also difficult to organize and use the Git-space, we persevered through the conflicts on GitLab.

We completed the "add movie" feature that displays movies added to the database by the user. We can add the title, description and genre tags with this feature. In the future, we are considering implementing actor and director names as well. We are also hoping to create the ability to upload images which would be the movie poster, although we are finding this difficult. We have also implemented a basic user profile page where the user can edit their username, and write an 'about-me' section with whatever they desire on their profile, it will also display the time that the user was last seen. Likewise, we hope to implement a profile picture feature. One feature we tried to implement by this iteration but were unable to, is a basic text-input search bar where the user can input the title of the movie, and it would return the desired movies. Unfortunately, we were unable to implement this function due to wrong implementation methods, and are planning on completing it for the final iteration. Additionally, we are still working on styling and design for the final iteration.

For this iteration, we are using functional tests, and back-end database tests. We have not implemented unit tests for this iteration but will be working on them for the final iteration.

Here is the link to our GitLab iteration2 branch:

## V. Testing Plan

For the final iteration we are planning on implementing three methods of testing:

Unit Testing: We plan to write Unit Tests for adding new movies, adding posts to the discussion forum, deleting posts, and deleting images. For these methods of testing we plan on applying the unittest, doctest and pytest frameworks.

Functional Testing: We will be applying manual tests, creating a checklist for actions that must be completed and verified for each task on each interface, and whether the produce the desired output. This will ensure that the correct information is being displayed and altered on the profile.

UI testing: In order to test the User Interface, we will be employing usability testing with friends, family and peers. We will record their observations and get feedback on the user experience and user interface logic. With this, we will proceed to make the necessary adjustments to our code, and design.

## VI. References

N/A