

Movie Mood

By:

Ethan Berk - EthanBerk

Henry Makman - Hsmakman5

Walid Kyambade - WalidKyambade

Nicolas Post - nicopost04

Project Description:

For people who enjoy movies, who struggle with choosing a movie. Movie Mood is a web tool that allows users to receive accurate movie recommendations based on their preferences and the weather. Movie Mood prompts a user to make an account, with which they can ask Movie Mood to generate movies for them based on a given set of parameters.

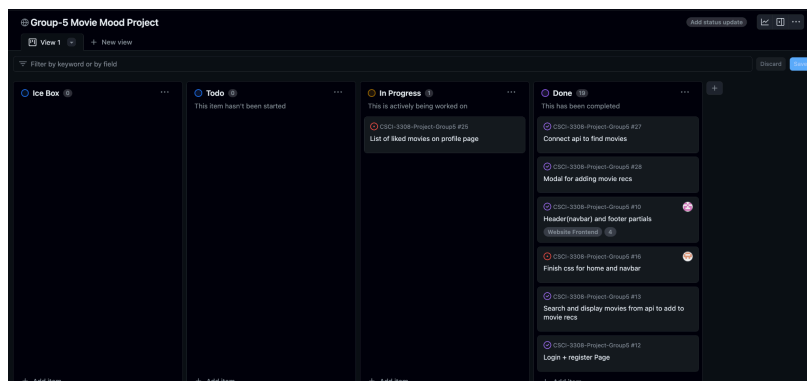
Movie Mood also prompts users to give data about which mood and weather a movie they have watched corresponds to. Movie Mood does this to learn *what* weather and mood corresponds to each movie. Therefore, the database grows as the website is used.

Movie mood can also keep track of what movies have been watched and the date they were watched. There is a whole page dedicated to this functionality that also provides the current weather and mood. If a movie has been watched twice but at different times, the mood and weather selected for both times watched will appear under the same movie.

In the future, we would like to implement more of a social media aspect to our movie watching website. We would like to add a like feature and an option to write what you thought about the movie and if it fit the mood/weather provided. On top of that, we would like to also add a friend feature, that way you can see other peoples watched movies and even share movies under different moods.

Project Tracker:

<https://github.com/users/Hsmakman5/projects/1/views/1>



VIDEO:

https://drive.google.com/file/d/1Ib43wPqn2tlxPKdR69dh7CyPa1_tnWNb/view?usp=sharing

VCS:

<https://github.com/Hsmakman5/CSCI-3308-Project-Group5>

ADD VIDEO TO THIS ^^

Contributions:

Nicolas Post:

Nico mainly worked on creating and populating the database that was used to store the movies, recommendations, and users. He also started several rough outlines of pages that were then further developed by other team members. He also created the log out functionality along with helping create the login process with the username and password bars. Nico helped start to create the modal used to add movies to the watched movies page and helped with some API calls used to navigate through the website.

Henry Makman:

Henry did a lot of the front end design work for designing the layout and scheme for the website. He used css, and html to build card groups and carousels for mainly the Home, Login, and Register page and built css styles to use on every page for uniformity throughout the website. Along with also building certain API routes for navigation through the header, and with making sql queries for the home page recommendation feature. Henry also helped along with all group members, diagnosing and fixing bugs throughout the website such as the login/register api.

Ethan Berk:

Ethan worked on the main functionality and structure of the app. He created the pages for finding movies, showing watched movies and adding watched movies. The add watched movies page required connecting the TMDB api and dynamically serving results based on a search. In addition there were dynamic (JS) modals that allowed the user to enter parameters then save the movie through a query. The watched movies page required another query to the database along with a dynamic page made with handlebars. The finding movies page required a query to get movies based on the user requests from a form.

Walid Kyambade:

Walid worked on bug fixes and overall testing for the app. He worked with Nico and Ethan to conceptualize and implement the database, as well as tweaks to the login and register functionality. He also worked on the base functionality for several integral API routes in the index.js file, such as the /register, /login, and /logout routes. He also worked on navbar functionality, and user authentication. He also worked on CSS and HTML for the profile page. Testing the app and making sure it ran on Azure was also one of his main responsibilities.

v0.3

Latest



What's Changed

- home page layout Merge by [@Hsmakman5](#) in [#17](#)
- Added the logout function by [@nicopost04](#) in [#20](#)
- add Movie rec Page with API by [@EthanBerk](#) in [#22](#)
- Add search for add movie rec by [@EthanBerk](#) in [#23](#)
- Finished modal template by [@EthanBerk](#) in [#26](#)
- Working on Home Page by [@Hsmakman5](#) in [#29](#)
- 14 query to find movies recs related to mood and weather for home page by [@nicopost04](#) in [#30](#)

New Contributors

- [@Hsmakman5](#) made their first contribution in [#17](#)
- [@nicopost04](#) made their first contribution in [#20](#)
- [@EthanBerk](#) made their first contribution in [#22](#)

Full Changelog: [v0.1...v0.3](#)

Contributors



EthanBerk, Hsmakman5, and nicopost04

▼ Assets 2

- | | |
|--------------------------------------|-------------|
| Source code (zip) | 2 weeks ago |
| Source code (tar.gz) | 2 weeks ago |



```
1  Notes for and from Meeting 3:
2
3  Commits from throughout the week can be seen here:
4  https://github.com/Hsmakman5/CSCI-3308-Project-Group5/commits/v0.2
5
6  Whats new:
7
8  FRONT END
9  - Build frameworks for pages once signed in
10     - Home / landing page (Ethan , Henry)
11     - Profile page (Nico)
12  - Added HTML and CSS to Homepage (Henry)
13     - Bootstrap card to be updated with info from SQL tables (Henry)
14  - Built a header for all pages (Ethan, Henry)
15  - Handlebars prefabs (Ethan)
16
17  BACKEND
18  - Fully Working Docker (Ethan)
19  - Added test cases from lab (Ethan)
20  - Finalized functionality of Login and Register pages (Henry, Nico, Walid)
21  - Included AUTH so users must sign in to use website (Ethan)
22     -May add a "use as guest" feature
23  - Built API get and Post calls for Login and Register with hashed passwords (Walid, Ethan)
24  - Built SQL tables for all important information including (Walid, Nico)
25     - users table
26     - Movies table
27     - Table link movies to weather
28     - Table to link users to moview
29  - Built navigational api calls (Nico, Walid)
30
31
32  Known Bugs
33  - Images are not displaying correctly (Henry)
34  - Can crash when creating a new user with matching credentials (Ethan)
35
36  -----
37  Upcoming
38  - API of Weather implement (Ethan, Henry)
39  - INSERT sql file for new movies (Nico)
40  - Updating movies based off of weather api (Henry)
41  - Finish back and front end of additional pages (Frontend: Henry Backend: Nico, Walid )
42  - Finish Logout function (Ethan)
```

```
1
2 Notes for and from Meeting 4:
3
4 Commits from throughout the week can be seen here:
5 https://github.com/Hsmakman5/CSCI-3308-Project-Group5/commits/v0.2
6
7 What's new:
8
9 FRONT END
10 - Finished Modal Templates (Ethan)
11   - Allows user to input data based on the movies (Ethan, Henry)
12 - Added HTML and CSS to Homepage (Henry)
13 - worked on header display if user is not logged in (Walid)
14 - Worked further on profile page (Walid, Nico)
15 - added logout function (Nico)
16 - Started modal used to submit information from movies search (Nico)
17 - Started working on dropdown menus for Home page (Henry)
18
19 BACKEND
20 - Added movie search function (Ethan)
21 - Connected to external Movie Database (Ethan)
22 - fixed nav bar redirect to profile (Walid)
23 - Finalized functionality of Login and Register pages (Henry, Nico, Walid)
24 - Improved AUTH so users must sign in to use website (Ethan, Henry, Walid)
25
26
27 Known Bugs
28 - Images are not displaying correctly (Henry)
29 - Can crash when unknown user logs in (Henry)
30
31 -----
32 Upcoming
33 - Finish back and front end of additional pages (Frontend: Henry Backend: Nico, Walid )
34 - Refine Profile page
35 - Have it so user can "Like" a movie
36 - Refine HTML and CSS on main page (Henry)
37 - Fix Log in API Route (Henry)
```

What's Changed

- updated the Database by [@nicopost04](#) in [#31](#)
 - Insert.sql additions
- finished modal api + fix nav auth by [@EthanBerk](#) in [#32](#)
 - Backend work with API calls and movie api
- add watched movie page with query and api by [@EthanBerk](#) in [#33](#)
 - Built outline of Movies Watched Page
 - Built multiple new API routes and sql queries
- started adding some information to the databases by [@nicopost04](#) in [#34](#)
 - Included editing and finalizing Create.sql and queries
 - Finished foundation and started adding data to Insert.sql
- 35 update html and css by [@Hsmakman5](#) in [#36](#)
 - Rebuilt Login page
 - Rebuilt Register page
 - Uniform color scheme for all pages
 - Home Page Layout
- Made changes to options in DB and also added 2 more movies by [@nicopost04](#) in [#37](#)
 - Changed create.sql so table are better formatted for important information and queries
- Update db by [@nicopost04](#) [@WalidKyambade](#) in [#38](#)
 - Small bug fixes and getting live working demo
- Fixed Login & Register Bugs by [@WalidKyambade](#) [@Hsmakman5](#) [@nicopost04](#)
 - Login and Register now fully work
- Presentation Plan [@nicopost04](#) [@WalidKyambade](#) [@Hsmakman5](#)

Upcoming & Finalization:

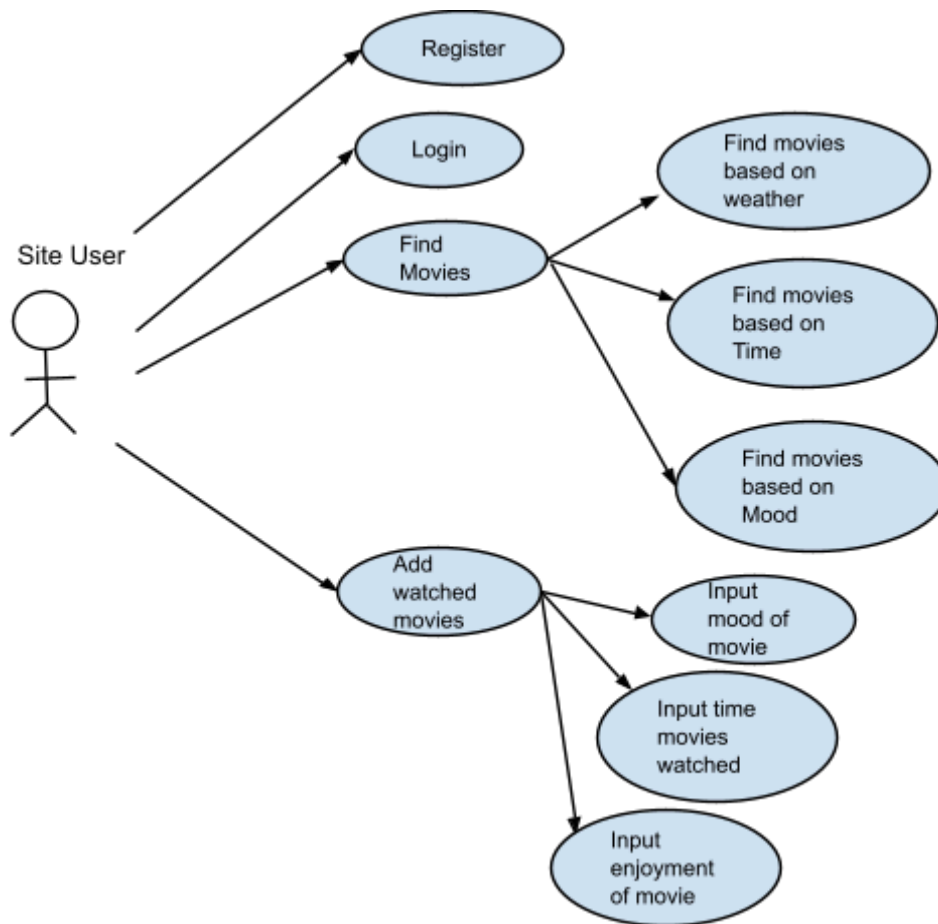
- Finish profile page [@WalidKyambade](#) [@Hsmakman5](#)
- Header bugs with redirects [@WalidKyambade](#)
- Finish home page functionality with api route and sql query [@EthanBerk](#)
- Logout page [@Hsmakman5](#)
- finish adding inserts of prebuilt movie recommendations [@nicopost04](#)

Known Bugs

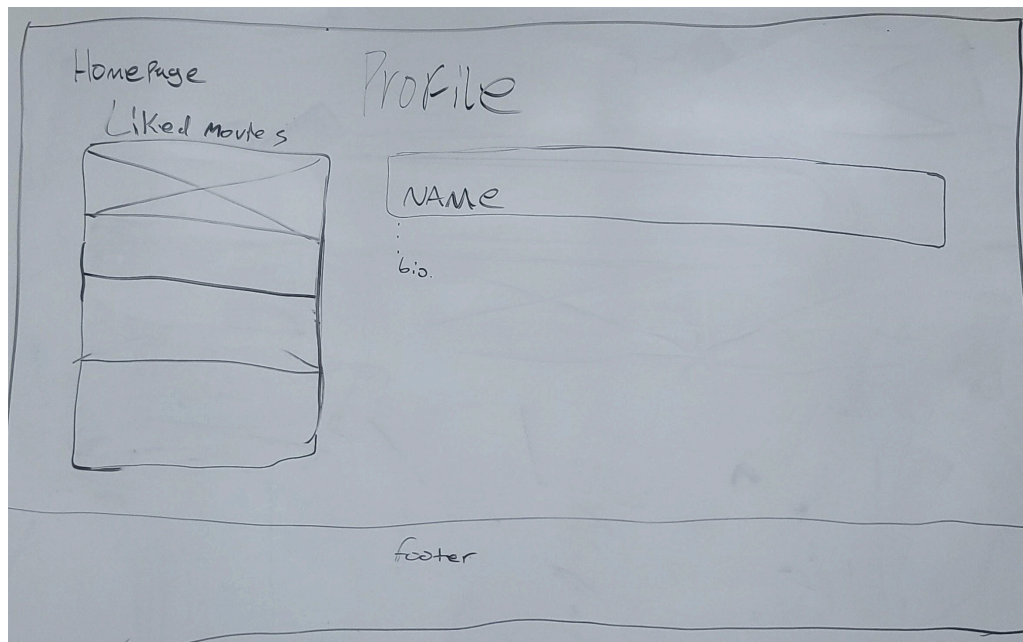
- Crash on redirect to Watched Movies page
- Profile page instant logout

Full Changelog: [v0.3...v0.4](#)

Use Case Diagram:



Wireframes:



Login

footer

Liked Movies

profile

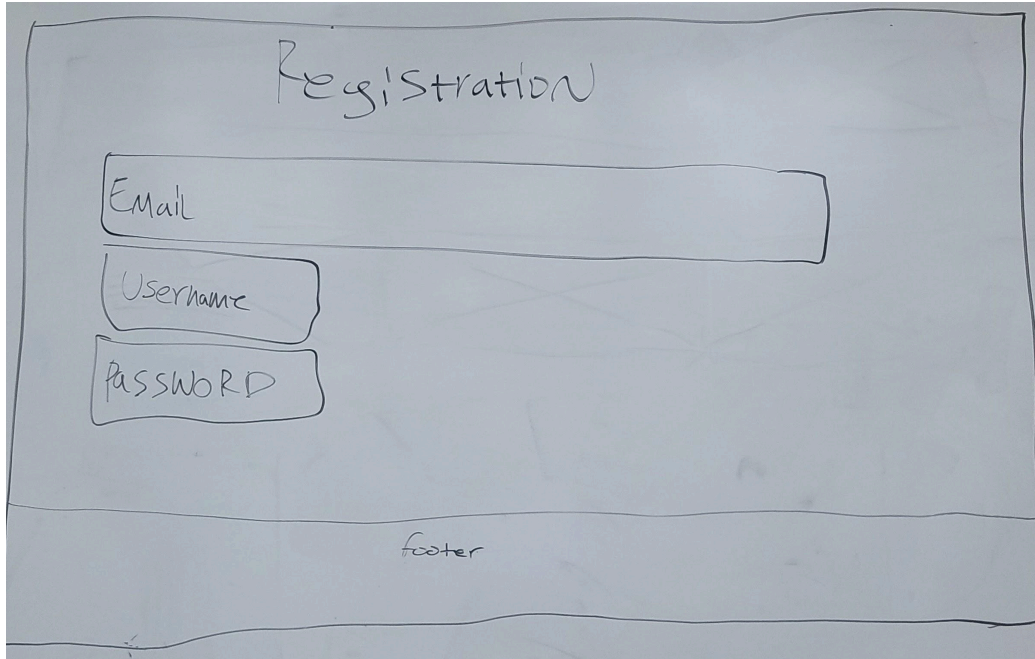
Login

Weather

Time of day?

Mood?

footer



Test Results:

Feature #1: Login to the website

Test Cases:

- User will be redirected to the login page when not logged in
- Users should be able to login with correct credentials.
- User authentication fails when the user provides invalid credentials.
- The form provides the user with specific feedback about the error.
- The User can navigate to the register page
- User can Create register and credentials will be saved for future logins

The test data will be using the users table. It keeps track of a username and encrypted password string. This will be updated with the get and post routes when logging in and registering.

Acceptance Criteria:

- A user cannot submit a Registration form without completing all of the mandatory fields. Mandatory fields include:
 - User Name
 - Password
 - Confirm Password

- Information from the form will be stored in the user table in the users_db database.
- A user cannot submit a Login form without completing all of the mandatory fields. Mandatory fields include:
 - User Name
 - Password
- Information from the form will be compared against the user table in the users_db database. Returning either a positive or negative login request

When tested, on the cloud a user should be able to register, sign in, and then leave the site and come back and still be able to sign in with the original credentials they used to register. A successful test response will be a valid response for the get sign in route.

Feature #2: Get a movie recommendation

Test Cases:

- Once logged in the user will be redirected to the home page
- Api calls should return local weather information which will be reflected on home page
- When user asks for "generate"
- Will Generate an error message if no movie was able to be generated
- Mood specifics will be cleared allowing the user to generate again
- after "Generate" user will be able to like the movie if they choose to

The test data that will be used to test the feature will be the local weather, time of day, and overall mood of the user.

The tests will be done on the local host because we will be using our database to check in with our apis.

The test results will be used to see if the tests are functioning correctly. If the criteria is met for the movie recommendation given based on the specifics entered then we know everything is running properly.

The user acceptance testers should be able to have an output movie based on the criteria.

Feature #3: User can update their profile on account page

Test Cases:

- Users can view their liked movies
- Users can log out on this page
- Users can unlike any movie they've previously liked

The test data that will be used are the users profiles since they will be doing modifications to their page.

The tests will be done on the local host since the users information will be on our local pc.

Deployment:

The way to deploy our website is to go to our github repository and download the files onto your local computer. Once this is done you need to have docker installed to run the servers locally. Once you have everything installed, you must navigate into the folder named project source code. From there you can use the command 'docker compose up' to start the server. From there you can go onto google chrome and type in "<http://localhost:3000/login>" to reach the login page to our website.