

**CSC 394 Senior  
Capstone**

**Movie  
Matcher  
Project  
Proposal**

# The Green Muskoxen

## Contents

Overview .....	5
Mission .....	5
Business Plan.....	5
Market Need.....	5
The Green Muskoxen Team .....	6
Business Plan .....	7
Operations .....	8
Production .....	8
Office Space.....	8
Future Expansions.....	8
Requirements Document .....	9
1.0 - Introduction .....	9
1.1 - Overview.....	9
1.2 - Business Objective .....	9
1.3 - Benefits.....	10
2.0 - Technical Requirements.....	10
3.0 - Functional & Testing Requirements.....	10
4.0 - Project Planning Requirements .....	10
5.0 - Project Roles Specification Requirements .....	11
5.1 Project Manager.....	11
5.2 Remote Collaboration Manager.....	12
5.3 Video Manager.....	12
5.4 Requirements Manager.....	12
5.5 Planning Manager.....	13
5.6 Design Manager.....	13
5.7 Implementation Manager.....	13
5.8 Webmaster .....	13
5.9 Documentation Manager.....	13
5.10 Interface Design & Testing .....	14
5.11 Testing Manager .....	14
5.12 Presentation Manager.....	14
5.13 Systems Programmer .....	14
6.0 - Assumptions.....	15

# The Green Muskoxen

7.0 - Constraints .....	15
7.1 - Limited Budget.....	15
7.2 - Limited Time .....	15
8.0 - Stakeholders .....	16
9.0 - Software Used.....	16
Functional Requirements matrix.....	16
Requirements Testing Matrix .....	23
Design .....	27
Original Design.....	27
Original Website Using Wordpress .....	32
Login.....	32
Original Swipe Page .....	33
Original Search Page .....	34
Original Search Result Page .....	34
Original Watchlist Page.....	36
Final Product .....	37
Login Page.....	37
Swipe Page.....	38
Old Profile Page.....	39
New Profile Page .....	40
Search Page .....	41
Search Result Page .....	42
Contact Us Page.....	43
Schemas .....	44
Scoring Schema.....	44
Database Schema .....	44
Level 0 Data Flow Schema .....	45
Finite State Machine Schema .....	46
Functional Relationship Between Modules Schema.....	47
Backend/Frontend code .....	48
Login.php .....	48
Login.php Continued .....	49
SearchBar.php .....	50
SearchBar.php Continued .....	51
SearchBar.php Continued .....	52

# The Green Muskoxen

SearchBar.php Continued .....	53
SearchBar.php Continued .....	54
SearchBar.php Continued .....	55
SearchBar.php Continued .....	56
SearchBar.php Continued .....	57
SearchBar.php Continued .....	58
Profile.php .....	59
Profile.php Continued .....	60
Profile.php Continued .....	61
Index.php .....	62
Movie.php.....	62
Movie.php Continued.....	63
Movie.php Continued.....	64
Movie.php Continued.....	65
Movie.php Continued.....	66
Movie.php Continued.....	67
Movie.php Continued.....	68
Movie.php Continued.....	69
Search.php .....	70
Register.php .....	71
Register.php Continued .....	72
Register.php Continued .....	73
Contact.php .....	74
Logout.php .....	74
CSS.....	75
Stylesheet.css.....	75
Stylesheet.css Continued.....	76
Stylesheet.css Continued.....	77
Stylesheet.css Continued.....	78
Stylesheet.css Continued.....	79
Stylesheet.css Continued.....	80
Stylesheet.css Continued.....	81
Stylesheet.css Continued.....	82
Stylesheet.css Continued.....	83
Register.css .....	83

# The Green Muskoxen

Searchbar.css.....	83
Planning and Workflow .....	84
ClickUP .....	84
GANT View.....	84
Calendar View .....	84
Discord .....	85
Session Planning Channel.....	85
Meeting Notes .....	86
Github.....	94
Commit History .....	94
Timelogs .....	95
Robert Kaszubski .....	95
Richard Watson.....	97
Stacey Reichard.....	99
Dan Kaminski .....	103
Nathan Hester .....	105
Casey Berg .....	107
Zaid Akel .....	109

# The Green Muskoxen

## MOVIE MATCHER

For Cinephiles Everywhere

### OVERVIEW

Movie Matcher is an easy to use, seamless movie recommending service that is here to add much needed time to your life. Our service is here to bring people together through the lens of film enjoyment, without all the time spent in the menus.

#### Mission

We want you to take your life back during the most important time: with family and friends. Life is busy and convoluted, so why should your personal time feel the same? Here at Movie Matcher, let us take the burden of finding the media that you like off of your shoulders and get to relaxing.

#### Business Plan

Using and integrating social media into Movie Matcher would allow us to gain an exponential userbase volume. With investor funding and support from ad revenue, our potential is limitless and our functionality harmonious.

#### Market Need

There are a few applications that currently do something similar to what Movie Matcher does, but none of them integrate the social features like we would, given the opportunity. Chat messaging with friends, integrated social media, and push notifications that remind the user to get back to swiping for recommendations.

# The Green Muskoxen

## The Green Muskoxen Team

**Project Manager, Planner**

**Stacey Reichard**

**Collaboration Software Manager, Web Designer/Master**

**Zaid Akel**

**Video/Media Manager, Presentation Manager**

**Robert Kaszubski**

**Requirements Manager**

**Casey Berg**

**Implementation Manager**

**Dan Kaminski**

**Interface/Security Manager**

**Nathan Hester**

**Systems Programmer, Documentation Manager**

**Richard Watson**

**Testing Manager**

**Mohammad Ali Kumail**

# The Green Muskoxen

## BUSINESS PLAN

### Summary

Movie Matcher is a premiere cinephile experience that is designed to keep the user out of the endlessly long menus inside the countless streaming applications that have crowded our lives. Our application presents a way out and a way back to enjoying the film and television that helps us unplug from the rest of the world.

Here at Movie Matcher, we want to change the way our users live every single day. Since the average American spends around four hours a day watching television, think about all the time lost to just *searching* for what to watch. We feel the societal need to give people their time back, with just a simple download. We also want the user to be able to share their movie watching experiences with friends and make plans to experience those viewing explorations together.

Our biggest competitor is an application called Taste. Although Taste does have some of the same functionality as Movie Matcher, they lack serious social media/social integration that is key to helping applications grow at an exponential rate. The average person does not have the time to go out of their way to recommend something by mouth unless they are forced into a small talk social situation. Why not recommend a movie with a couple of taps on your phone instead? This way, there is a digital receipt that can be viewed later.

As of right now, Movie Matcher is completely free to use. At our current state, we do not have the resources to implement all the features we would like. But with your investment and financial help from generating ad revenue, we believe that all of our entrepreneurial desires are possible.

# The Green Muskoxen

## OPERATIONS

### Production

As of now, Move Matcher has three developers working hard to bring a great product to fruition. A general overview of hardware and financial needs for our team includes: computers with processors with at least four cores and sixteen gigabytes of memory, proper programming and video editing software, high speed internet for streaming team collaboration, and capital for web hosting the Movie Matcher website (this price goes up as users increase). Team member salaries are also be included in this requirement.

### Office Space

As of now, Movie Matcher team members are currently all working from home. We do not see the need in the foreseeable future to purchase a lease for office space as most of our work is done from computers and collaboration done through video streaming software.

### Future Expansions

As our user count undoubtedly grows, more positions will need to be filled. Our main employee deficit starting out will be a financial team (consisting of one employee to start), marketing team, and additional developers. Once our employee count exceeds twenty, we will need an additional team manager per twenty employees with an additional implementation manager per fifty additional employees. We will also require a Human Resources Manager when our employee count exceeds twenty. When Movie Matcher takes off, we will of course need to hire an accountant and other third-party business consultants to help keep the ship on the right path.

# The Green Muskoxen

## REQUIREMENTS DOCUMENT

### 1.0 - Introduction

#### 1.1 - Overview

Movie Matcher is a secure application that provides users with the ability to build upon their previous movie knowledge and discover new flicks that pique their interest. With our user-oriented movie picking algorithm, our clients will never struggle again while searching for the perfect movie. Ultimately, Movie Matcher will allow for a stream of suggestions to make your next viewing experience as seamless as possible.

#### 1.2 - Business Objective

Movie Matcher currently doesn't have any source of revenue to expand our capabilities. Assuming we were to get hired in the future, we would use ad revenue to help offset some of the costs of what we would like to accomplish with the application. Some major accomplishments would include a friends list feature with chat messaging, a social media syncing system to share Movie Matcher results and features, along with push notifications to alert users of new updates about movies included on their watchlist.

Social media integration would further the user's voice about their experience with Movie Matcher. With social media being the widest used technology in society today, having a feature like this would draw more attention and popularity to our application, allowing for more new users and more ad revenue to come in. Users would also have the ability to use sharing functionality. We would use social media interactions to our advantage by allowing users to do the work for us by recommending a movie by word of mouth and using the applications main features.

At Movie Matcher, we hope users can spend less time on menus and more time enjoying film!

# The Green Muskoxen

## 1.3 - Benefits

Movie Matcher has a number of benefits to the consumers it serves. At its core, Movie Matcher is a completely free service that allows its users to freely and easily browse the movie catalogue, or swipe through a curated list of suggested movies. As the users add movies to their watchlist, the curated list of movies grows and adapts to the genres, actors and directors featured in the films. A user can later visit their watchlist and view all the movies they've added to the list to revisit them later.

## 2.0 - Technical Requirements

- 2.1 Host a live version of the application on WordPress using AWS (Amazon Web Service) Lightsail.
- 2.2 Switch hosting from AWS to GoDaddy at <https://staceyreichard.com/moviematcher>.
- 2.3 Setup and host a live backup of the site at <https://elonslilbackup.com>.

## 3.0 - Functional & Testing Requirements

### Functional Requirements

See “Functional Requirements” matrix.

### Testing Requirements

See “Testing Requirements” matrix.

## 4.0 - Project Planning Requirements

- 4.1 Setup a Discord channel for all Green Muskoxen team members to be able to touch base and keep on track with the project.

- 4.2 Setup and add tasks to Jira.

The Green Muskoxen chose not to continue with Jira after 9/29/20. This was due to the cost of various integrations that the group needed to be able to implement for this project.

# The Green Muskoxen

4.3 Switch/import all tasks into ClickUp (from Jira) as of 9/29/20.

- The Green Muskoxen collectively made the decision to import all tasks from Jira into ClickUp. This decision was made due to the cost effectiveness of the web-based software.
- Implement Discord integration with ClickUp.
- Implement GitHub integration with ClickUp.
- Create a critical path through the project using ClickUp's Gantt chart view.

4.4 Implement a system for updating and hosting files on GitHub for Movie Matcher.

## 5.0 - Project Roles Specification Requirements

Name	Project Role
Stacey Reichard	Project Manager, Planner
Zaid Akel	Collaboration Software Manager, Web Designer/Master
Robert Kaszubski	Video/Media Manager, Presentation Manager
Casey Berg	Requirements Manager
Dan Kaminski	Implementation Manager
Nathan Hester	Interface Manager, Security Manager
Richard Watson	Systems Programmer, Documentation Manager
Mohammad Ali Kumail	Testing Manager

### 5.1 Project Manager

Responsible for getting human, technical, and other resources to team members so that they can do their jobs.

Resolves people problems when they arise and encourages team members and shows appreciation for a job well done!

Makes tough decisions for the group when necessary and shows a strong sense of leadership along the way.

# The Green Muskoxen

## 5.2 Remote Collaboration Manager

Responsible for engaging the team when meeting or talking through the Green Muskoxen's Discord channel.

Reviews all documents in the team's Google Drive folder and makes sure that everything is listed in an organized fashion.

Documents the repositories used in GitHub and makes sure that all team members are aware of how to create, update, and merge branches with the master repository.

## 5.3 Video Manager

Responsible for pulling together group presentations and collaborating with various team members to make sure all parts of a given presentation make sense and are uploaded to Google Drive in a timely manner.

Trains team members on the software and formats video clips are to be submitted in. The Green Muskoxen uses .mp4 videos recorded in OBS/Quick Time Player.

The Video Manager also edits and uploads all completed presentations to the Green Muskoxen's YouTube and Panopto channels. They make the group look good!

## 5.4 Requirements Manager

Defines and records all requirements for the Movie Matcher application that drives the design and plan for the group. This includes capturing the complex relationship between the client and the developers of the application.

Provides clear documentation to make sure the Green Muskoxen's goals are met.

# The Green Muskoxen

Keeps track of all changes made to the application and updates documentation based on those changes.

## 5.5 Planning Manager

Responsible for creating the calendar, schedule, due dates, tasks, breakdowns, dependencies, deadlines, and critical path in ClickUp (the Green Muskoxen's designated project management software).

## 5.6 Design Manager

Responsible for the overall design of the Movie Matcher application along with changes that are made on it along the way.

Responsible for verifying that the application is working well on both desktop and mobile devices.

## 5.7 Implementation Manager

Oversees and supports the implementation of the Movie Matcher application. This includes the design of the movie scoring algorithm as well as technical systems, software, hardware or solutions.

Responsible for making sure that the remote implementation of the application is a success.

## 5.8 Webmaster

Responsible for coordinating the design and deployment of the Movie Matcher application.

Maintains and manages both the main and backup website running the Movie Matcher application. This includes verifying that the most current and up-to-date version of the site is running on both the main and backup servers.

## 5.9 Documentation Manager

Responsible for timely delivery of documentation on the project including but not limited to organizational policies, objectives, and budget.

Keeps track of all changes throughout the project and keeps accurate documentation on said changes as well.

# The Green Muskoxen

Responsible for getting time log documentation from team members and making them available to the group.

Responsible for creating, and presenting the Green Muskoxen's full manual for the Movie Matcher application.

## 5.10 Interface Design & Testing

Responsible for making sure that users are able to easily walk through the application without error.

Creates adequate tests on various aspects of the site based on recommendations from the team and works closely with the Testing Manager to document said testing along with requirements for the application.

## 5.11 Testing Manager

Responsible for working alongside the Interface Design & Testing Manager and for planning of tests, long with their execution.

Responsible for finding the best testing techniques to use on the application and ensuring that the application is suitable for deployment.

## 5.12 Presentation Manager

Responsible for presenting the most interesting aspects of the Movie Matcher application to the client. This responsibility is the face of the Green Muskoxen.

## 5.13 Systems Programmer

Responsible for working alongside the Implementation Manager to code the application's main algorithm to specification. They are required to make sure that there is always a working version of the code loaded to the application's website and pass information on updated versions of the code to the Webmaster.

# The Green Muskoxen

## 6.0 - Assumptions

- The team has the skills or experience needed to succeed from skills taught at DePaul, to skills learned at home and through jobs or internships. This makes the Green Muskoxen capable of succeeding in software projects.
- The biggest factors that could hold our group back is health during pandemic, time management, and family emergencies.
- There are worries that the Movie Matcher idea could be copied, however, we feel we are unique based on how our algorithm is designed and built.
- As Movie Matcher is further developed, we will need to focus more on the security of the application to limit the risk of getting hacked.
- application to limit the risk of getting hacked.

## 7.0 - Constraints

### 7.1 - Limited Budget

- Since we have a limited budget, we had to be careful what we chose to integrate into the site. This also included the use of planning software like Jira and ClickUp.
- We initially chose to host the site using AWS Lightsail and later chose to host it through GoDaddy (pre-paid for by one of our teammates).
- Used lite and or free versions of various apps to help cut down on cost.

### 7.2 - Limited Time

- Everyone in the group is a student and is busy with other classes. Therefore, our time is restricted.
- We made sure to use our time wisely. This required us to focus on more important aspects of the application and limit our time spent on those tasks that were less important.
- We focused mainly on making the application run to our specifications and then focused on the design of it later.
- were less important.
- We focused mainly on making the application run to our specifications and then focused on the design of it later.

# The Green Muskoxen

## 8.0 - Stakeholders

- The Green Muskoxen (Zaid Akel, Casey Berg, Nathan Hester, Daniel Kaminski, Robert Kaszubski, Mohammad (Ali) Kumail, Stacey Reichard, and Richard Watson)
- Acme Corporation

## 9.0 - Software Used

- ClickUp
- Discord
- GitHub
- GoDaddy
- Google Drive

## FUNCTIONAL REQUIREMENTS MATRIX

Number	Revision	Requirement Description	Bad Dev. Attack	Bad Client Attack	Approval	Comments	Sign-Off
3.1.0	0	Set up the initial AWS/Lightsail solution to host our wordpress site.	N/A	N/A	YES	We eventually pivoted away from a AWS/Lightsail solution.	APPROVED by Zaid Akel 9/16/20
3.1.1	0	Implement a backup schedule to prevent loss of progress with the UpdraftPlus plugin.	N/A	N/A	YES	Backups created daily through Updraft Plus.	APPROVED by Zaid Akel 9/18/20
3.1.2	0	Mock up a general architecture and aesthetic of the site with the Elementor plugin.	N/A	N/A	YES		APPROVED by Zaid Akel 9/20/20

# The Green Muskoxen

3.1.3	0	Require all users to register for an account on the Movie Matcher site.	N/A	N/A	YES		APPROVED by Mohammad Ali Kumail 9/28/20
3.1.4	0	Require users to create a secure password on the site when creating an account.	Developers have access to users account credentials.	N/A	NO	Passwords need to be hashed for security.	NOT APPROVED
3.1.4	1	Require users to create a secure password on the site and then use hashing on the password to hide it from everyone but the user.	N/A	N/A	YES		APPROVED by Nathan Hester 10/6/20
3.1.5	0	Allow users to login to their account.	Developers could incorrectly implement an account login function.		NO		NOT APPROVED
3.1.5	1	Allow users to login after creating their account.	N/A	N/A	YES	After creating their account, all users are able to login.	APPROVED by Nathan Hester 10/9/20
3.1.6	0	Upon logging out of the application, users should be redirected to the login page.	Developers could incorrectly implement an account logout function.	N/A	NO	Users are unable to logout of an account.	NOT APPROVED
3.1.6	1	Upon logging out of the application, users should be redirected to the login page and their session should be cleared out.	Incorrectly coded session data could open the application up to a session hijacking attack.	N/A	NO	Sessions must be closed out properly when the user logs out of the application.	NOT APPROVED
3.1.6	2	Upon logging out of the application, users should be redirected to the login page and their session is	N/A	N/A	YES		APPROVED by Mohammad (Ali) Kumail 10/5/20

# The Green Muskoxen

		completely cleared out.					
3.2.1	0	Allow for logged in users to swipe "watch" or "pass" on a randomly generated movie.	Developers could incorrectly implement the functionality of the "watch" and "pass" buttons.	User could incorrectly swipe, leading to erroneous "watch" and "pass" entries added to their account. This would in turn affect the way the algorithm is recording scores for a given movie.	NO	The user can only swipe through 5 static movies.	NOT APPROVED
3.2.1	1	Users that swipe "watch" or "pass" are only able to review a set number of movie titles.	Developers could incorrectly setup the database resulting in a static number of titles which would in turn limit the number of times a user could "swipe" or "pass" on a given title.	N/A	NO	Further implementation on the movie page needed.	NOT APPROVED
3.2.1	2	Implement the swipe "watch" or "pass" on a procedurally generated list rather than a static number of movies.	N/A	N/A	YES	The user can swipe through an unlimited procedurally generated list of movies.	APPROVED by Nathan Hester 10/9/20
3.2.2	0	Populate data from OMDb into the swiping page.	Developers could incorrectly implement the OMDb shortcode into the swiping area.	N/A	NO	Swiping page now shows the full movie poster as well as details about the film.	NOT APPROVED
3.2.2	1	Populate the swiping page with data from OMDb that allows users to swipe through movies for as long as they'd like.	N/A	N/A	YES	Swiping page now shows the full movie poster as well as details about the film.	APPROVED by Mohammad (Ali) Kumail 10/9/20
3.2.3	0	Create and finalize a concept of the Movie	N/A	N/A	YES	Concept approved by	APPROVED by Stacey

# The Green Muskoxen

		Suggestion Algorithm.				the Green Muskoxen.	Reichard 10/9/20
3.2.31	0	Create database schema.	Developers could not properly implement the database schema with primary and foreign keys.	N/A	NO	Refine schema to include more information from OMDb.	NOT APPROVED
3.2.31	1	Refine schema to better suit the algorithm and to pull data from the OMDb API.	N/A	N/A	YES	Schema has been finalized.	APPROVED by Nathan Hester 10/19/20
3.2.32	0	Create swiping algorithm.	Algorithm could not be as in depth as originally intended.	Client could claim that the algorithm isn't producing impactful results based on recommendations on the user's profile page.	NO	Some movies were seen twice during Swipe feature.	NOT APPROVED
3.2.32	1	Ensure duplicate suggestions do not show up in the swiping feature of the application.	N/A	N/A	YES	Algorithm adequately suggests movies to users based upon swipe preference.	APPROVED by Nathan Hester 10/27/20
3.3.1	0	Create a searchbar where users can input a movie title to search for it.	Searchbar could be incorrectly made to return the wrong movie or that the movie could not be found.	User could accidentally search for an unintended film	NO		NOT APPROVED
3.3.1	1	Create a searchbar where users can input a movie title to search for the closest match.	N/A	N/A	YES	Closest film shows up that is searched for.	APPROVED by Mohammad (Ali) Kumail 11/2/20
3.3.2	0	Return data from the OMDb API to the searchbar page.	SQLite database might not be able to handle multiple threads at a time. Write operations (which include opening transactions that are about to do an INSERT/UPDATE/DELETE) because the other threads	N/A	NO	Switch to a more robust MySQL database.	NOT APPROVED

# The Green Muskoxen

			are blocked until the operation is complete. This could potentially break the application.				
3.3.2	1	Replace SQLite database with the same database using MySQL (PHPMyAdmin).	N/A	N/A	YES	MySQL allows for multiple threads and a resolution to the SQLite problem.	APPROVED by Nathan Hester 11/5/20
3.3.3	0	Add a "Watch" button to the searchbar so users can add titles to their watchlist.	The "Watch" button could easily be clicked accidentally forcing the title to be added to a user's watchlist.	N/A	NO	Code is based upon the existing movie page.	NOT APPROVED
3.3.3	1	Add a "Watch" button to the searchbar and modify the algorithm to be able to handle movie scoring.	Users still have the ability to add titles to their watchlist but no way to formally delete them if they want to modify said watchlist.	N/A	NO		NOT APPROVED
3.3.3	2	Add a "Watch" button to the searchbar resulting in a title being added to the user's watchlist. Users also have the ability to remove a title from their watchlist if they accidentally added a movie.		N/A	YES		APPROVED by Mohammad (Ali) Kumail 11/2/20
3.4.1	0	Show users 4 recommendations based on their use of the application.	Developers could program to show a random assortment of 4 movies rather than recommending 4 movies based on the user's watchlist.	Client claims users aren't happy with the recommendations the application is presenting them with.	NO	Refine algorithm to appease the client.	NOT APPROVED.
3.4.1	1	Show users 4 recommendations based on their use of the application. The algorithm sorts from least to	N/A	N/A	YES		APPROVED by Nathan Hester 11/5/20

# The Green Muskoxen

		greatest and then recommends 4 movies at random from those 4 highest scoring categories.					
3.4.2	0	Show the users watchlist based on what they've added to their watchlist.	The developer could easily randomize from all movies that exist from OMDB and random sort without actually implementing the algorithm for the users' watchlist.	N/A	NO	Doesn't work to add from search function.	NOT APPROVED
3.4.2	1	Show the users watchlist based on what they've added to their watchlist from both the movie and searchbar pages.	N/A	N/A	YES	APPROVED by Nathan Hester 11/4/20	
3.4.3	0	Create a way for users to Delete titles from their watchlist on the profile page.	Deletion could delete the entire watchlist. This could also potentially decrement the scoring system for the individual user who deleted a given title from their watchlist.	User claims that deleting movies does not remove the movie from their watchlist.	NO	This will allow the user to customize their profile on demand in a scenario where the user does not wish to view this movie anymore, they may have the option to remove it whenever they want.	NOT APPROVED
3.4.3	1	Users have a way to remove titles from their watchlist in addition to adding new titles.	N/A	N/A	YES	Users have full control over their watchlist.	APPROVED by Mohammad (Ali) Kumail 11/8/20
3.5.1	0	Create a contact page where users can contact Movie Matcher through a web form.	Developers could incorrectly implement the 'contact us' form, meaning the users message never reaches the dev team.	N/A	NO	A separate page that will allow the user to fill in required boxes (name, email, and feedback text).	NOT APPROVED.

# The Green Muskoxen

3.5.1	1	Connect contact page to Google Forms.	N/A	Client claims Google Forms direct users away from the main Movie Matcher site.	NO	Have the contact form from the contact page be sent to google forms allowing the team to view the feedback in a clear and organized manner.	NOT APPROVED.
3.5.1	2	Create link pointing back to Movie Matcher application from Google Form submission page.	N/A	N/A	YES	Implemented a hyperlink the user can click on to navigate back to Movie Matcher for a cleaner way back to the website rather than using the "Click to go back" button.	APPROVED by Stacey Reichard 10/15/20

# The Green Muskoxen

## REQUIREMENTS TESTING MATRIX

Number	Revision	Requirement Test	Result	Approval	Test Date	Comments	Client Sign-Off
3.1.3	0	5 users try to register an account on the site.	All 5 users were able to register an account on the site.	YES	09/28/20	Test succeeded.	APPROVED by CEO Elon Musk 9/29/20
3.1.4, 3.1.5	0	5 users will create accounts, and an administrator will check to see if their passwords are correctly hashed and other user information is securely stored.	All 5 users were able to create an account, but without a secured password via hashing	NO	10/05/20	Test failed.	NOT APPROVED BY CLIENT
3.1.4, 3.1.5	1	5 users will create accounts, and an administrator will verify that they are unable to see users passwords due to security with password hashing.	All 5 users were able to successfully create accounts with hashed passwords on the administrator end.	YES	10/7/20	Passwords created on the site are secure with hashing.	APPROVED by CIO Helen Hunt 10/7/20
3.1.6	0	2 users tested that when logging out of the application they were redirected back to the main Movie Matcher login page.	Both users that tested logging out of the application were redirected to the login page when logging out. They were able to logout from the site but session data wasn't fully cleared out. This left the site vulnerable to attack.	NO	10/6/20	Implementation team needs to verify that session data is being cleared out completely when a users logs out of the application.	NOT APPROVED BY CLIENT.
3.1.6	1	Verify that 5 users' sessions are cleared out upon clicking the	All 5 users' sessions are cleared out upon clicking the	YES	10/9/20	Test succeeded.	APPROVED by CEO Elon Musk 10/9/20

# The Green Muskoxen

		upon clicking the logout button.	"Logout" button on the application and they were redirected to the main login page on the site.				
3.2.1	0	5 users can swipe through the movie page.	The list appeared to be limited to a static number of 5 total movie entries.	NO	10/15/20	Having the list be limited is a flaw needed to be fixed.	NOT APPROVED BY CLIENT
3.2.1, 3.2.2	1	5 users can swipe through the movie page for as long as they'd like.	All users could swipe freely through their unlimited list.	YES	10/17/20	Users can swipe through an unlimited, randomly generated list of movies. Test Succeeded.	APPROVED by CEO Elon Musk 10/17/20
3.2.2	0	5 users are given information regarding 5 different movies on the swiping page.	All 5 users were given some information on 25 different movies; however, not all information was shown.	NO	10/20/20	Not all the information for movies shows up, (ex: genre, rating, and/or release date missing).	NOT APPROVED BY CLIENT
3.2.2	1	5 users are given all information regarding 5 different movies on the swiping page.	All 5 users were given various information on 25 different movies.	YES	10/22/20	All information is shown for movies displaying on the swipe page.	APPROVED by CEO Elon Musk 10/22/20
3.2.31	0	5 different users match and pass on the same 10 movies; are they recommended the same movies?	All 5 separate users get recommended the same movies after passing and matching on the same 10 movies.	YES	11/6/20	Test succeeded.	APPROVED by CEO Elon Musk 11/7/20
3.2.32	0	Test the Pass and Watch buttons for Swipe page to make sure that if passed, then the film is never suggested to that user again. If the user chooses	Some movies were seen as duplicates during Swipe testing	NO	10/26/20	Duplicate movies shown even though they were selected as pass/watch.	NOT APPROVED BY CLIENT

# The Green Muskoxen

		watch, then that movie is added to their watchlist and also never suggested to the user again.					
3.2.32	1	5 users swiped on titles consecutively for 15 minutes each without receiving a duplicate.	Movies that are selected to be pass/watch are now removed from the swipe suggestion pool.	YES	10/27/20	No more duplicates to be seen when using Swipe feature.	APPROVED by CEO Elon Musk 10/27/20
3.3.1	0	5 different films can be searched for, with their closest match being presented after the search.	All 5 searches returned their closest movie entries.	YES	11/3/20	Test succeeded.	APPROVED by CEO Elon Musk 11/4/20
3.3.2	0	3 users are able to return results on the searchbar page simultaneously without running into error with the application.	SQLite database was not able to handle search functionality for multiple users at the same time.	NO	11/2/20	Test failed.	NOT APPROVED BY CLIENT
3.3.2	1	5 simultaneous users were able to perform INSERTS, UPDATES, and DELETES within the application without breaking it.	MySQL database through PHPMyAdmin was able to handle multiple simultaneous user interactions at once.	YES	11/5/20	Test succeeded.	APPROVED by Data Analyst Bob Murray 11/5/20
3.3.3	0	3 users are able to add movie titles to their watchlist from the searchbar page.	Users were able to add titles to their watchlist from the searchbar page but were unable to modify their watchlist from profile afterwards.	NO	10/20/20	Test failed.	NOT APPROVED BY CLIENT
3.3.3	1	3 users are able to add movie titles to their watchlist from the	All testing users were able to add and remove titles from their	YES	11/2/20	Test succeeded.	APPROVED by Data Analyst Bob Murray 11/2/20

# The Green Muskoxen

		searchbar or swipe page and are then able to modify their watchlist from the profile page.	watchlist successfully.				
3.4.1	0	5 users are able to interact with the application and get 4 solid movie recommendations.	Users were unable to obtain good movie recommendations based on their taste.	NO	When a user passes or watches a movie, that should determine whether the score is incremented or decremented from the application's scoring system. This should in turn affect the user's scoring information attached to the application overall. This will also affect future movies appearing on the swipe page as well as their movie recommendations.	Test failed.	NOT APPROVED BY CLIENT
3.4.1	1	Based on the modified movie scoring algorithm, 5 users are able to interact with the application and get 4 solid movie recommendations viewable on their profile page.	Users were able to obtain solid movie recommendations based on their taste and interaction with the Movie Matcher application.	YES	11/5/20	Test succeeded!	APPROVED by CEO Elon Musk 11/5/20
3.4.2	0	5 users were able to interact with their watchlist and then confirm that the titles they added to their watchlist appeared in it.	All 5 users were able to confirm that the titles from their interactions with the application (and only those films) were indeed added to their watchlist.	YES	11/4/20	Test succeeded!	APROVED by Data Analyst, Bob Murray 11/5/20

# The Green Muskoxen

3.3.3, 3.4.3	0	3 users were able to add 5 films to their watchlist and then choose 2 of them to remove from their watchlist.	All 3 users were able to add 5 films to their watchlist and remove 2 which resulted in a total of 3 leftover.	YES	10/28/20	Films were removed, working as intended. Test succeeded!	APPROVED by CEO Elon Musk 10/28/20
3.5.1	0	Verify that form submission is working on the contact page.	Form submission did not work.	NO	10/13/20	Implementation team needs to add/modify a Google Form to be used in place of the existing form.	NOT APPROVED
3.5.1	1	Verify that form submission is submitting to a Google Sheet and is powered by Google Forms.	Form submission successfully submits to Google Sheets. Google Forms directs users to a proprietary Google page that directs users away from the Movie Matcher application.	NO	10/14/20	Implementation team needs to redirect from the Google Form submission page back to the Movie Matcher application. Test failed.	NOT APPROVED
3.5.1	2	Verify that the Google Form submission directs users back to the Movie Matcher application in some way.	Form submission works and successfully redirects users back to Movie Matcher.	YES	10/15/20	Test succeeded!	APPROVED by CEO Elon Musk 10/15/20

## DESIGN

### Original Design

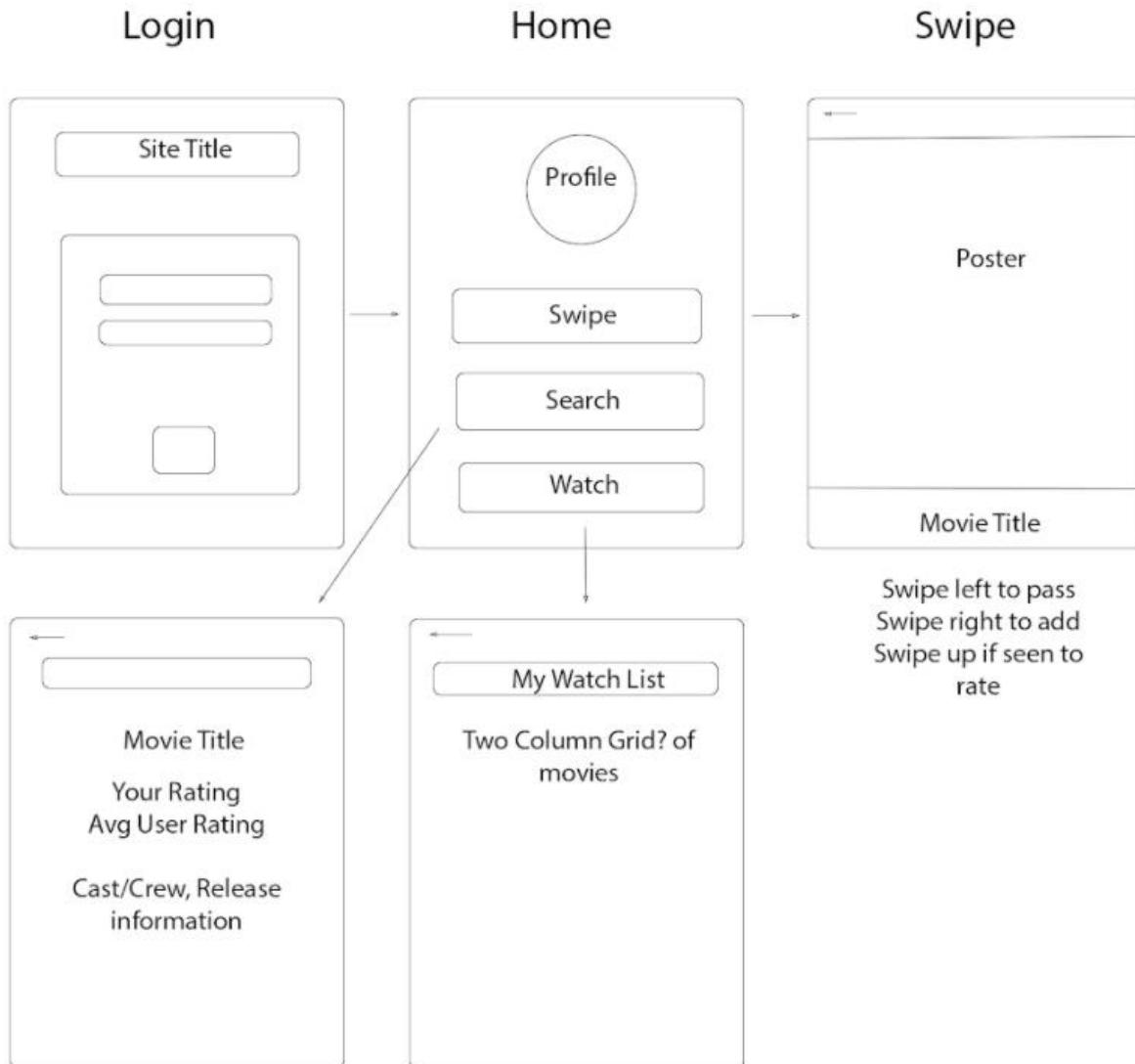
# The Green Muskoxen

Site Name

Login:

Sign in with  
Google

# The Green Muskoxen

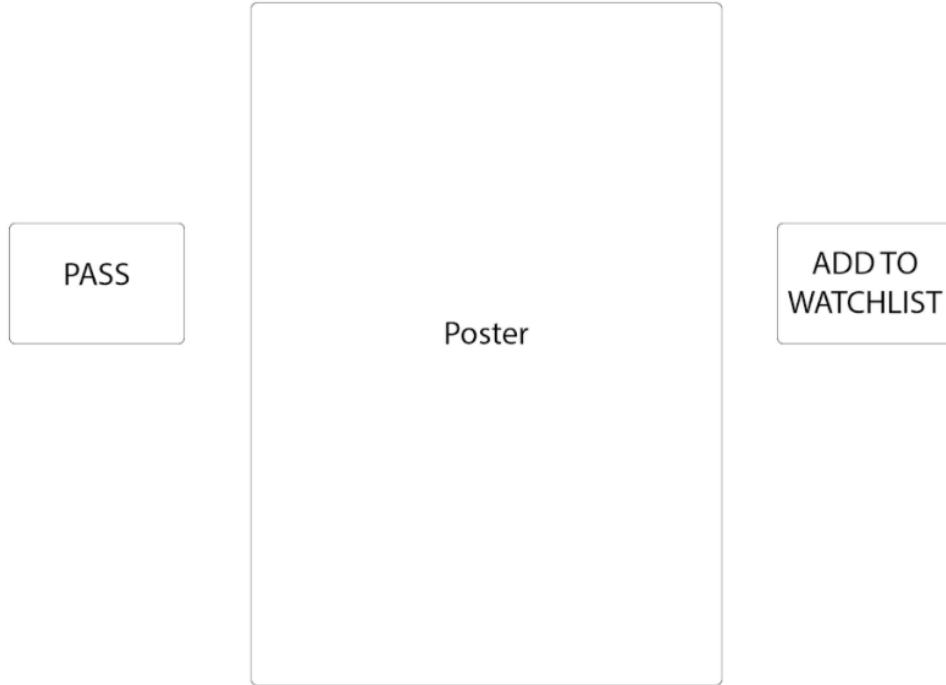


# The Green Muskoxen

SWIPE!

Navigation Bar (Replace with site title)

Search



I ALREADY  
SAW IT

# The Green Muskoxen

SWIPE!

Navigation Bar (Replace with site title)

Search



Poster

## Movie Title

Your Rating  5  
Avg User Rating  4.32

Available on / Now playing:

Netflix  
Hulu  
Fandango  
Amc

Studio:

Release:

Genre:

Cast / Crew Information:

Director

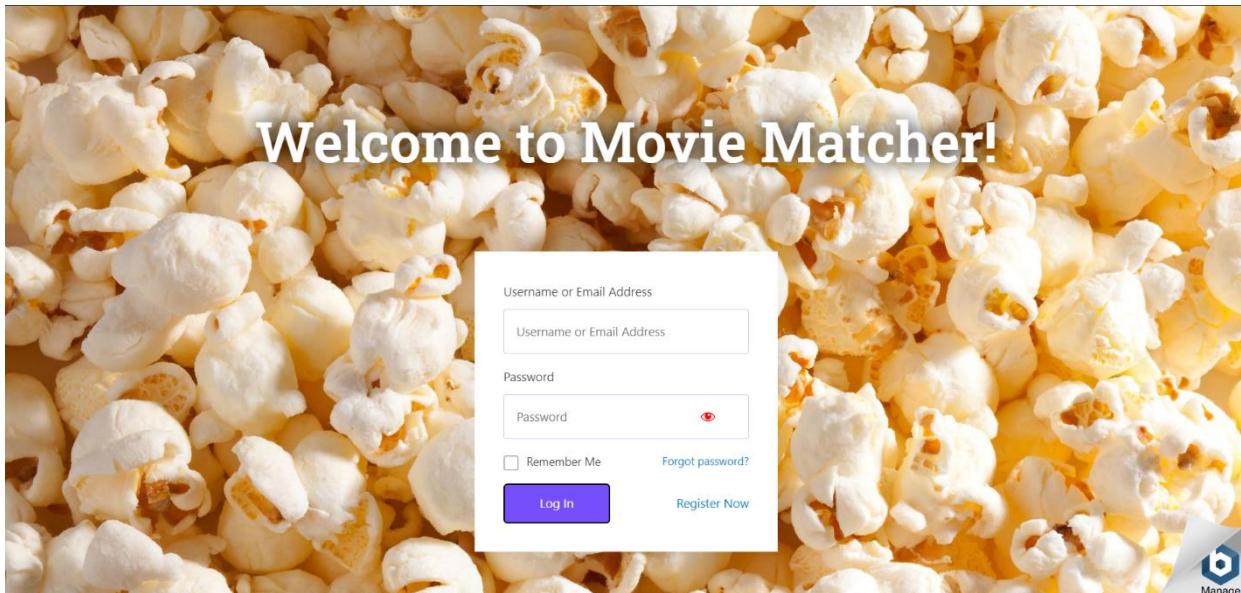
Producer

Actor

# The Green Muskoxen

Original Website Using Wordpress

**Login**



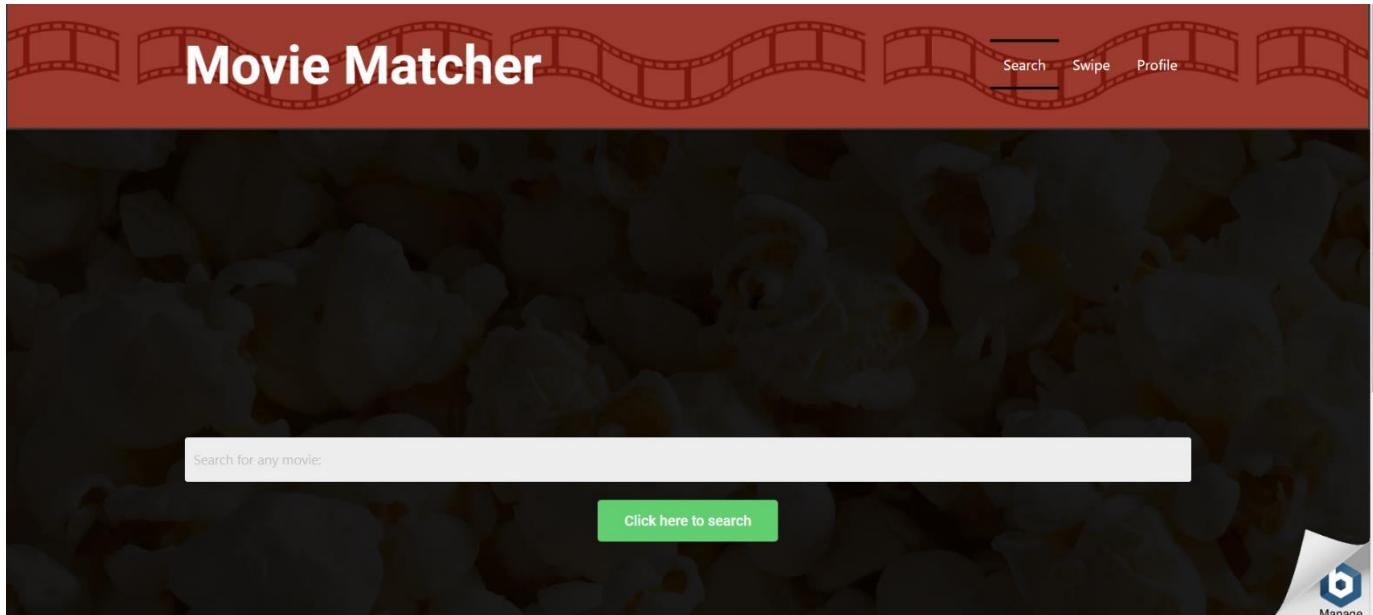
# The Green Muskoxen

## Original Swipe Page



# The Green Muskoxen

## Original Search Page



# The Green Muskoxen

## Original Search Result Page

The screenshot shows a mobile application interface for "Movie Matcher". At the top, there's a red decorative bar with film strip icons and the title "Movie Matcher". On the right side of this bar are three buttons: "Search", "Swipe", and "Profile". Below the bar, the main content area has a dark background with a faint image of a person's face on the right.

In the center, there's a movie card for "Tenet (2020)". The card includes the movie's title, release year, genre (Action, Sci-Fi), runtime (2h 30min), and release date (3 September 2020 (USA)). It also features a rating of 7.9 with a yellow star icon. The card displays two images: a poster showing two men in suits standing in a cityscape, and a scene from the movie showing a car in a dark, futuristic setting with a play button overlay.

Below the movie card, there's a green "Watch" button. Further down, there are two rating sections: "Your Rating: ★★★★☆" and "Movie Matcher Average Rating: ★★★★☆".

At the bottom right of the screen, there's a small "Manage" button with a gear icon.

# The Green Muskoxen

## Original Watchlist Page

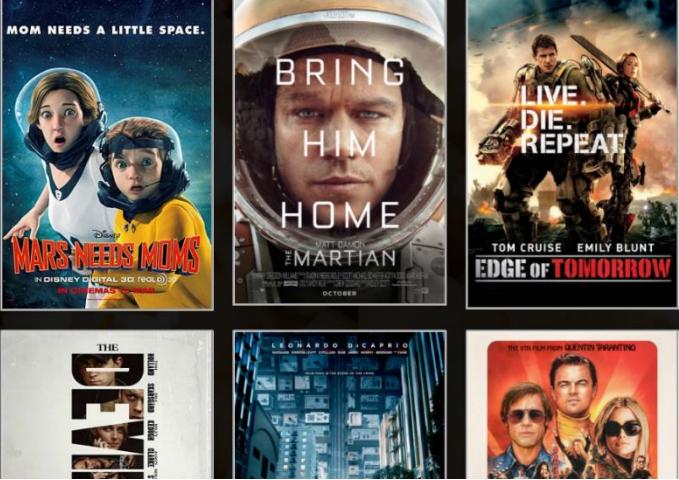


Elon Musk

**Watch List**

**Favorite Films**

[Update Favorites](#)

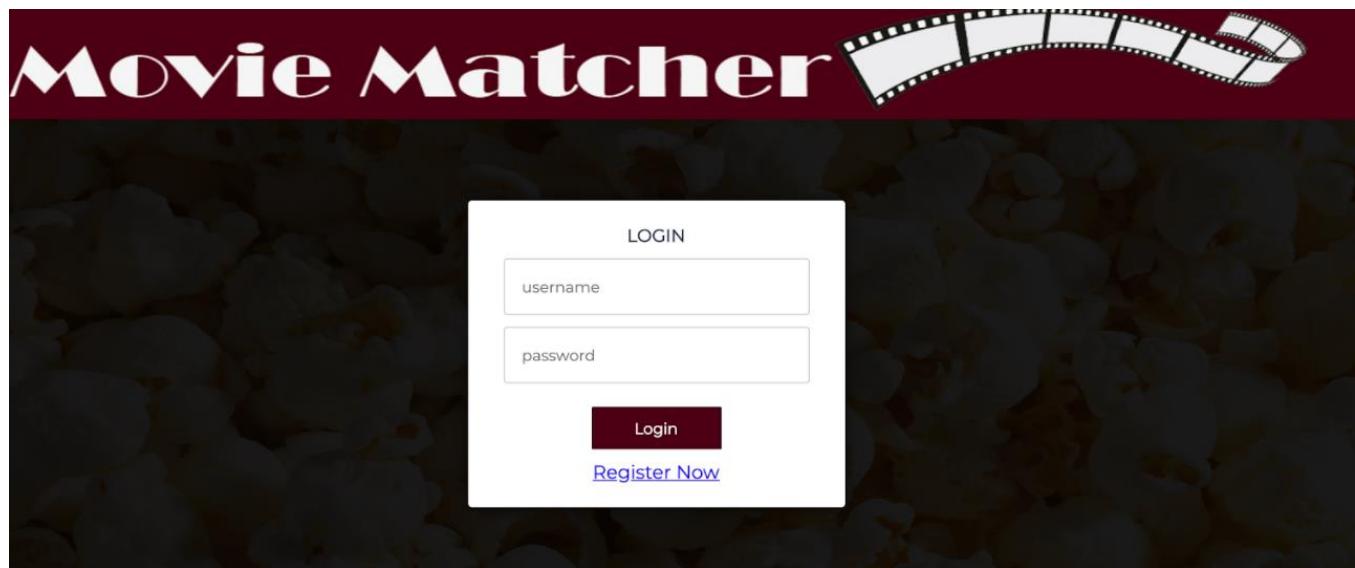


**Manage**

# The Green Muskoxen

Final Product

**Login Page**



# The Green Muskoxen

[Swipe Page](#)

## Movie Matcher

Search   Swipe   Profile   Contact Us

### The Avengers

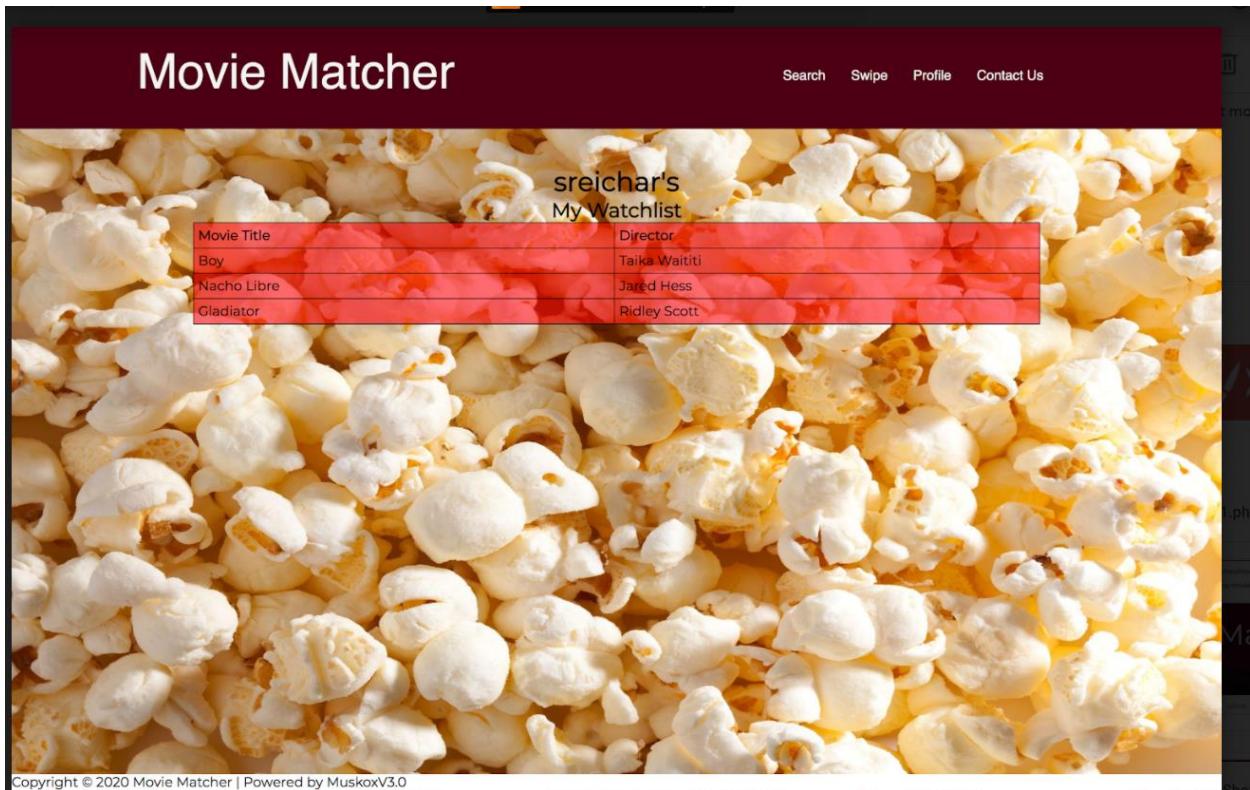


Pass      Watch

Director: Joss Whedon  
Release Year: 2012  
Actors: Robert Downey Jr., Chris Evans, Mark Ruffalo, Chris Hemsworth

# The Green Muskoxen

## Old Profile Page



A screenshot of the Movie Matcher mobile application's profile page. The background is a close-up photograph of popped popcorn. At the top, there is a dark header bar with the title "Movie Matcher" on the left and navigation links "Search", "Swipe", "Profile", and "Contact Us" on the right. Below the header, the user's name "sreichar's" and the section "My Watchlist" are displayed. A table lists three movies from the user's watchlist:

Movie Title	Director
Boy	Taika Waititi
Nacho Libre	Jared Hess
Gladiator	Ridley Scott

At the bottom of the screen, there is a copyright notice: "Copyright © 2020 Movie Matcher | Powered by MuskoxV3.0".

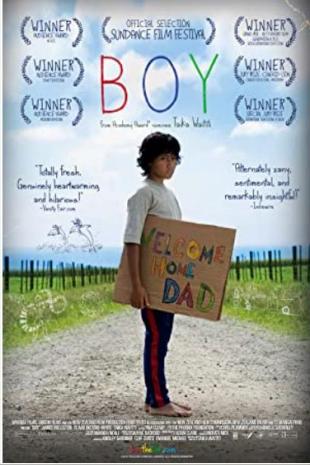
# The Green Muskoxen

## New Profile Page

**Movie Matcher** 

richl's Watchlist

  
The Avengers

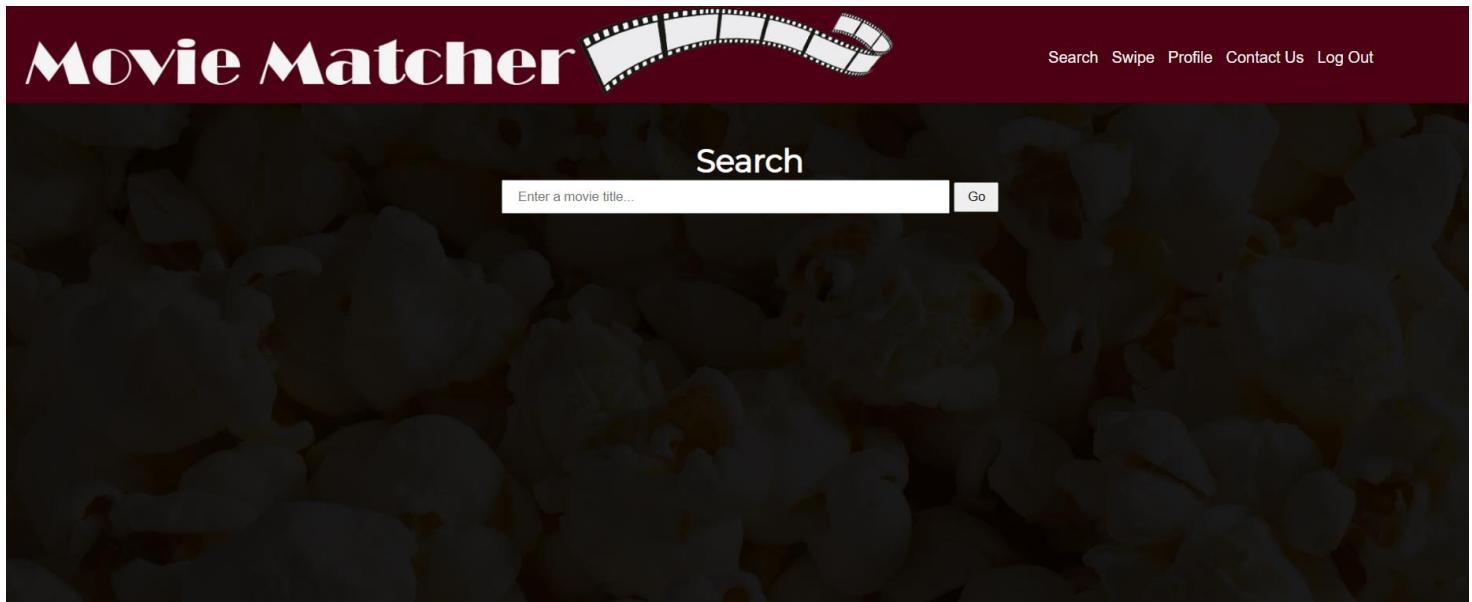
  
Boy

  
Nacho Libre

Search Swipe Profile Contact Us Log Out

# The Green Muskoxen

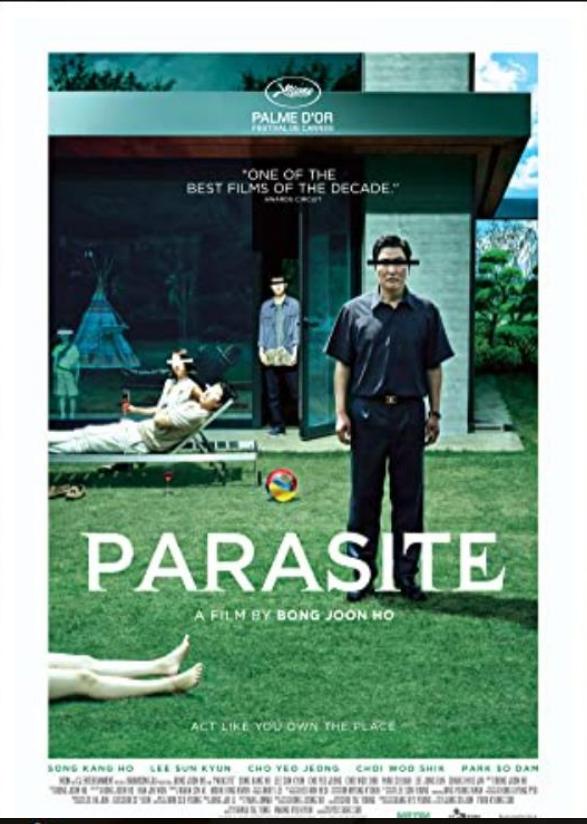
## Search Page



# The Green Muskoxen

## Search Result Page

# Movie Matcher



**Parasite**

Director: Bong Joon Ho

Year: 2019

Imdb Score: 8.6

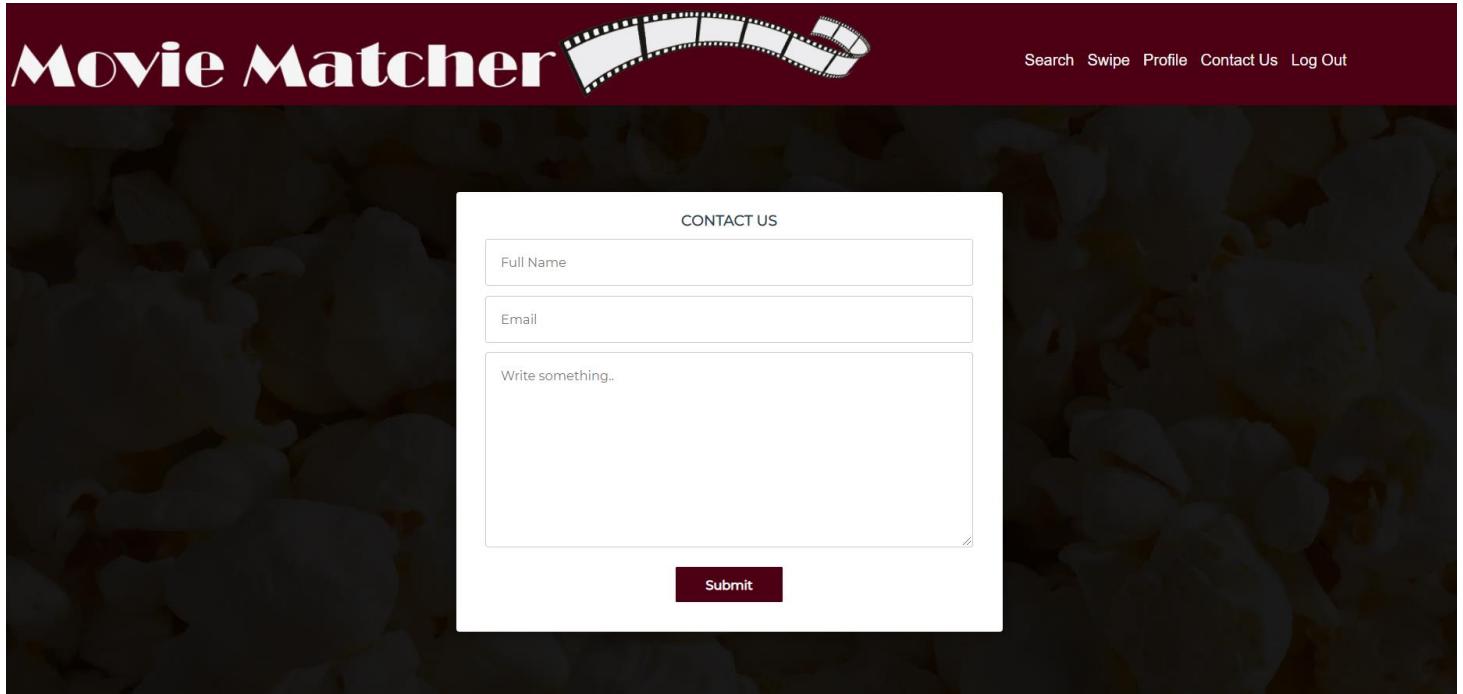
Actors: Kang-ho Song, Sun-kyun Lee, Yeo-jeong Jo, Woo-sik Choi

Genre: Comedy, Drama, Thriller

Rated: R

# The Green Muskoxen

## Contact Us Page



The screenshot shows the 'Contact Us' page of the Movie Matcher application. The header features the 'Movie Matcher' logo with a film strip icon on the right, and a navigation bar with links for Search, Swipe, Profile, Contact Us, and Log Out. The main content area is a white 'CONTACT US' form with three input fields: 'Full Name', 'Email', and a large text area labeled 'Write something..'. A 'Submit' button is located at the bottom of the form.

Search Swipe Profile Contact Us Log Out

CONTACT US

Full Name

Email

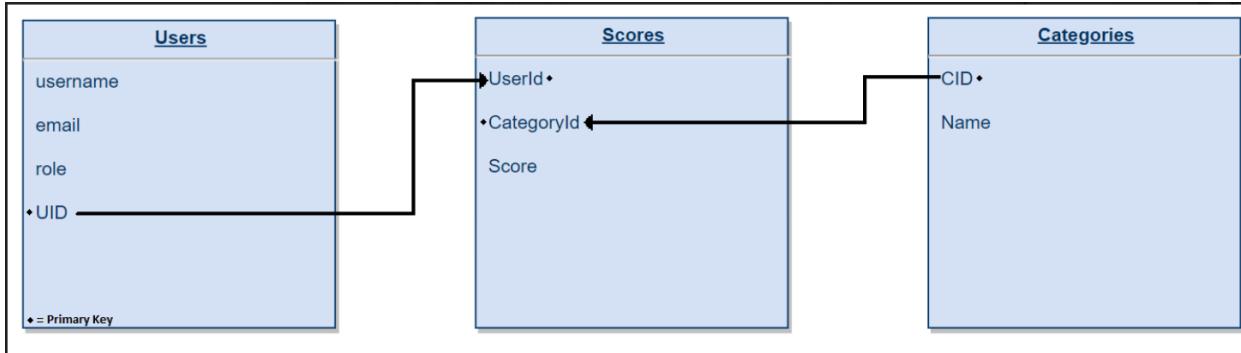
Write something..

Submit

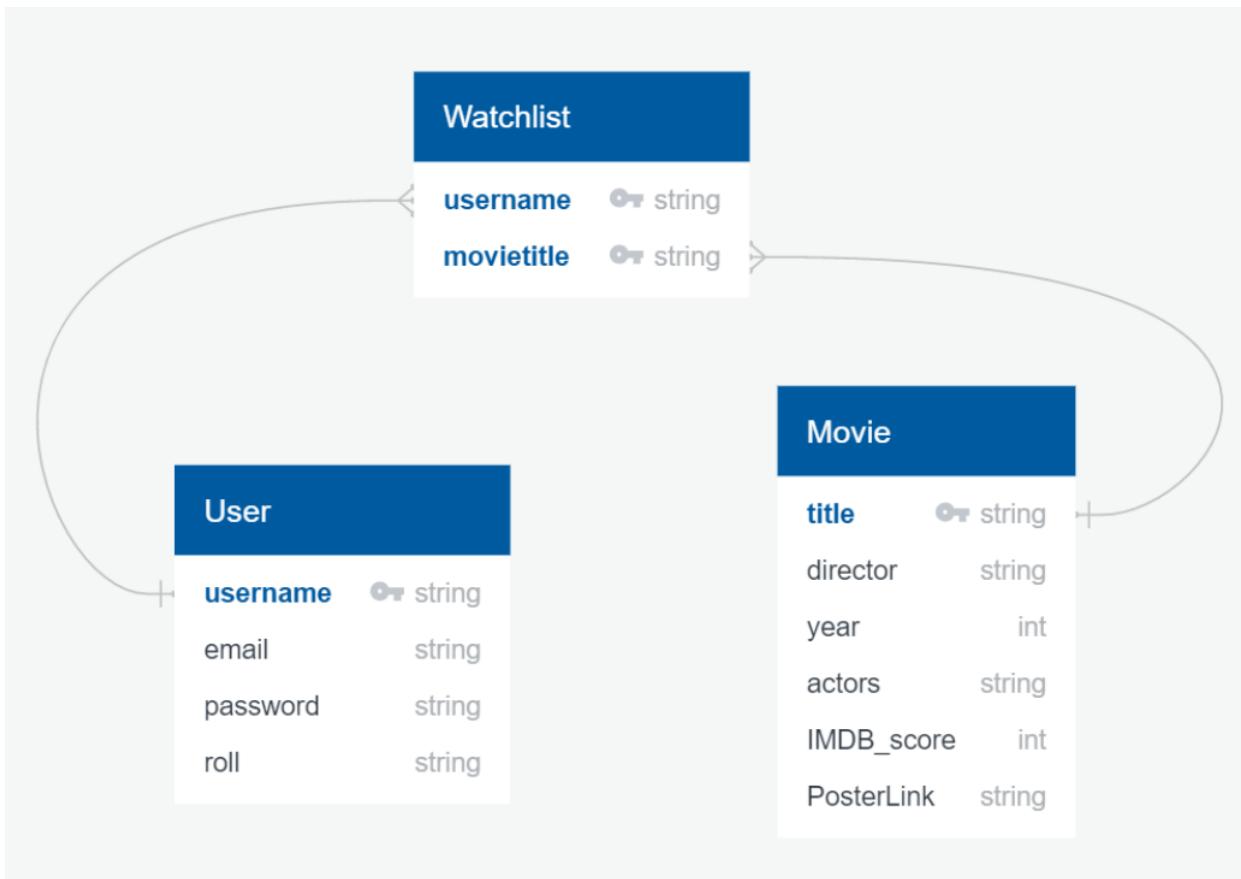
# The Green Muskoxen

## Schemas

### Scoring Schema

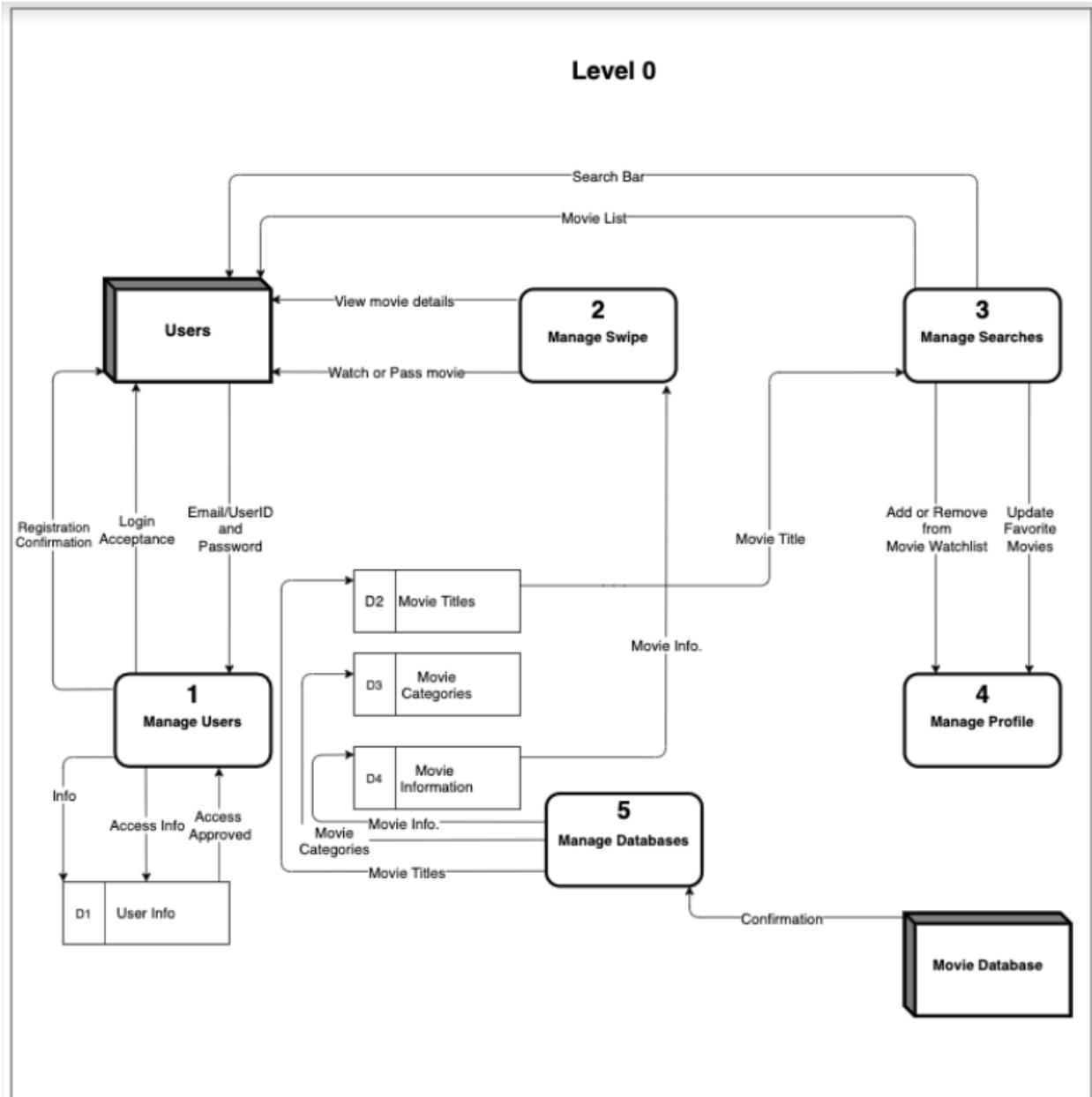


### Database Schema



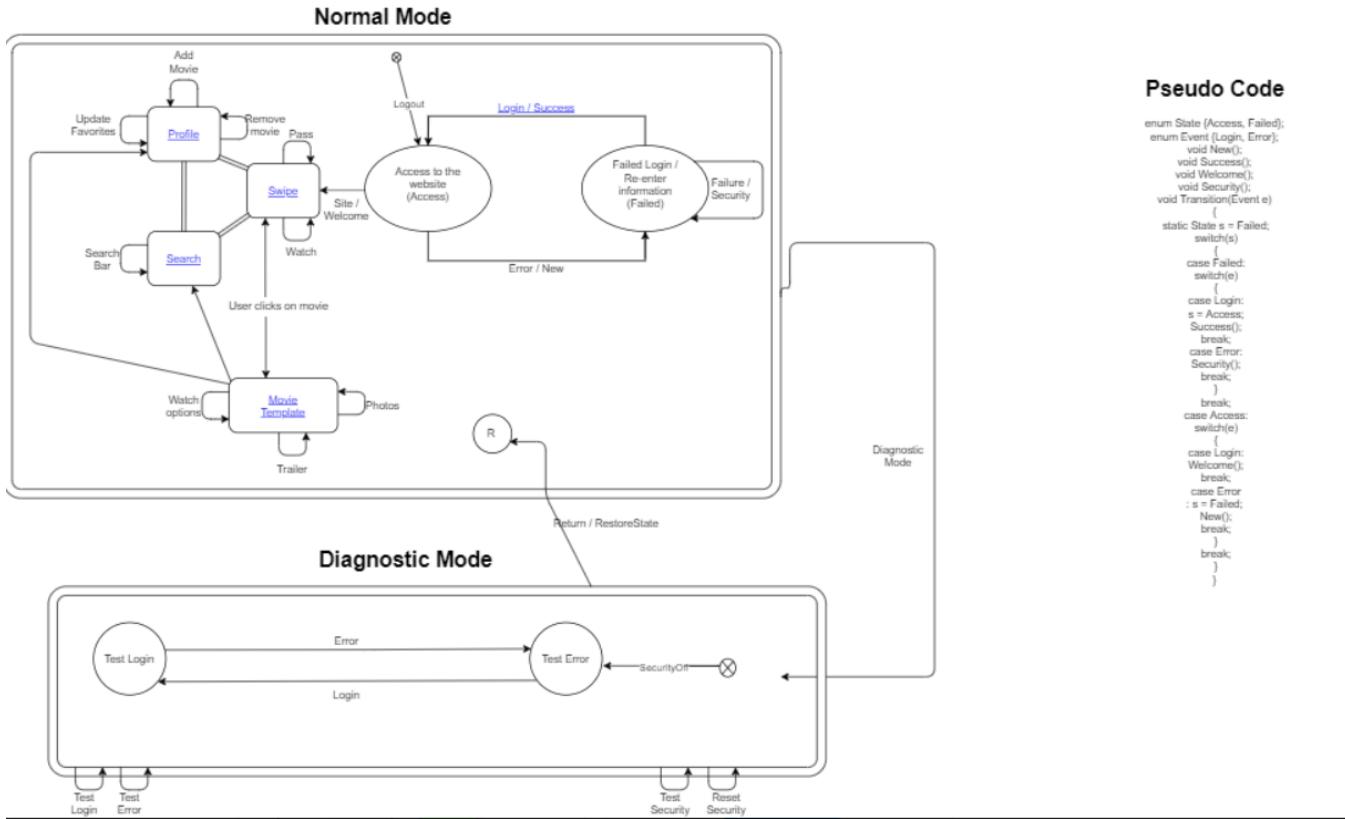
# The Green Muskoxen

## Level 0 Data Flow Schema



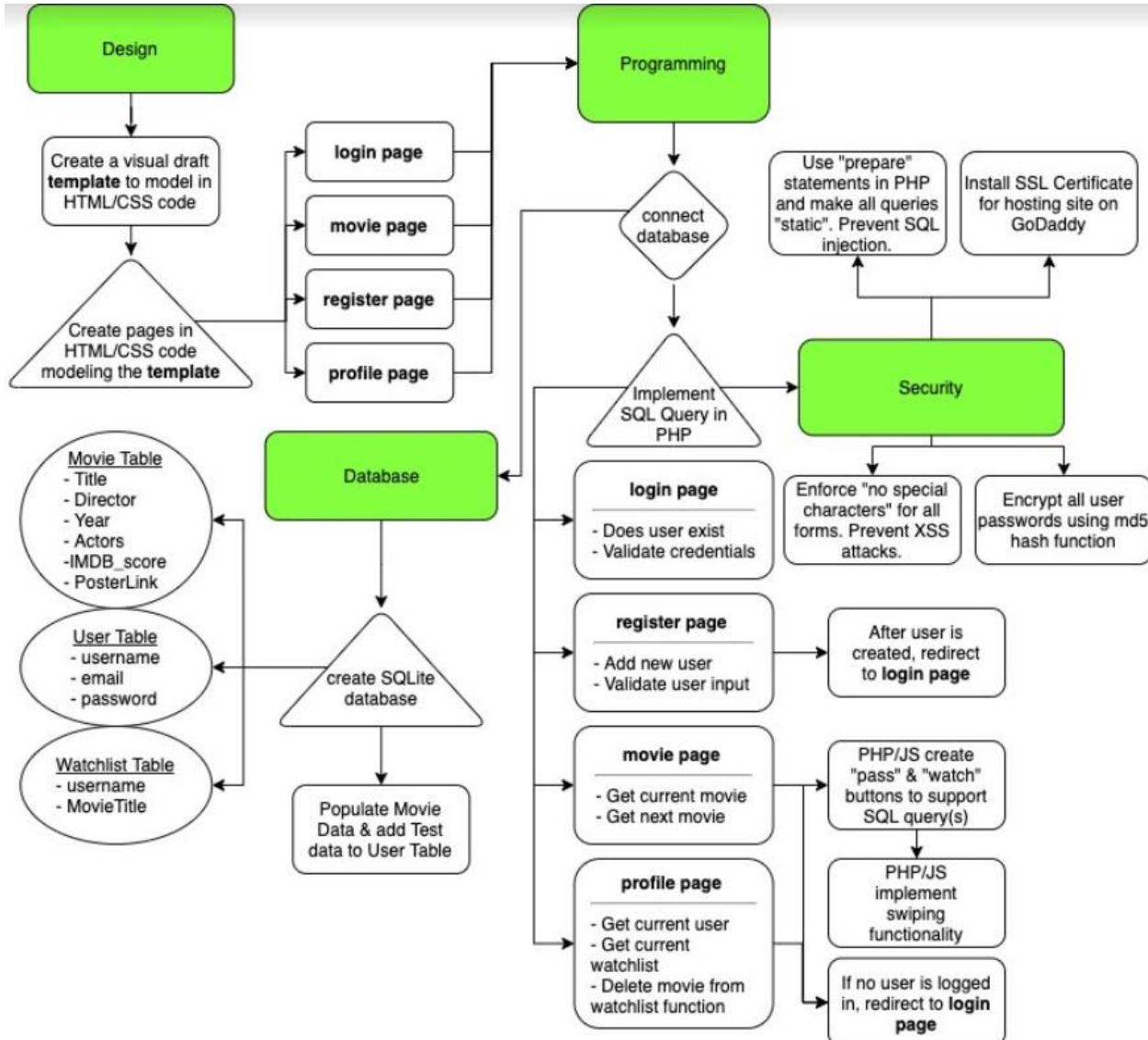
# The Green Muskoxen

## Finite State Machine Schema



# The Green Muskoxen

## Functional Relationship Between Modules Schema



# The Green Muskoxen

## BACKEND/FRONTEND CODE

### Login.php

```
login.php
1 <?php
2     function noSpecialChar($string)
3     {
4         if (preg_match('/[^\^f%&*()]{#~?><,|=_+-]/', $string)) {
5             // one or more of the 'special characters' found in $string
6             return false;
7         }
8         return true;
9     }
10
11    if(isset($_POST["username"]) && isset($_POST["password"]))
12    {
13        $username = trim($_POST["username"]);
14        $password = trim($_POST["password"]);
15        if (noSpecialChar($username) && noSpecialChar($password)) {
16            $pdo = new PDO("sqlite:MM DataBase.db");
17            $findUserQuery = "SELECT UID FROM Users WHERE username=?";
18            $stmt = $pdo->prepare($findUserQuery);
19            $stmt->execute([$username]);
20            // returns boolean "false" if user does not exist
21            $uid = $stmt->fetchColumn();
22            if ($uid != false) {
23                $hashPW = md5($password);
24                $findUserPasswordQuery = "SELECT Password FROM Passwords WHERE UserId=?";
25                $stmt2 = $pdo->prepare($findUserPasswordQuery);
26                $stmt2->execute([$uid]);
27                $existingHashedPW = $stmt2->fetchColumn();
28                if ($hashPW == $existingHashedPW) {
29                    session_start();
30                    $_SESSION["username"] = $username;
31                    $_SESSION["UID"] = $uid;
32                    // How are we getting the role? How should it be set for the session?
33                    // $_SESSION["role"] = $row["role"];
34                    header("Location: movie.php");
35    
```

# The Green Muskoxen

## Login.php Continued

```
35         }
36     } else {
37         echo "User account does not exist, please try again or register for an account";
38     }
39
40 }
41 ?
42
43
44
45 <html>
46
47     <title>Login</title>
48     <link rel="stylesheet" href="css/stylesheets.css">
49     <link rel="icon" href="assets/favicon/favicon.ico">
50 </head>
51
52 <body>
53     <div class = header style="font-size:4vw;">
54         
55     </div>
56     <div class= container>
57         <div class=overlay>
58             <br>
59             <form id = msform method="POST">
60                 <fieldset>
61                     <h2 class="fs-title">Login</h2>
62                     <input type="text" placeholder="username" name="username" required></input>
63                     <input type="password" placeholder="password" name="password" required></input>
64                     <input type="submit" value="Login" class = action-button></input><br>
65
66                     <a href="register.php">Register Now</a>
67                 </fieldset>
68             </form>
69         </div>
70     </div>
71     <?php include('components/footer.php'); ?>
72 </body>
73
74
75
76
77 </html>
```

# The Green Muskoxen

## SearchBar.php

```
1  <?php
2  session_start();
3  //If user is not logged in redirect to login page
4  if (!isset($_SESSION["UID"]))
5  {
6      header("Location: login.php");
7  }
8  //Store user ID and declare global variables
9  $userId = intval($_SESSION["UID"]);
10 $movieError = "";
11 $title = $director = $actors = $category = $poster = $rated = $plot = "";
12 $year = 0;
13 $imdbRating = 0.0;
14 $input_term = "";
15 // $pdo = new PDO("sqlite:MM DataBase.db"); //Establish Database connection
16
17
18 // Function to ensure no special characters are used in movie name
19 function noSpecialChar($string) {
20     if (preg_match('/[^\^%&*()]{#~?>>,|=_+-]/', $string)) {
21         // one or more of the 'special characters' found in $string
22         return false;
23     }
24     return true;
25 }
26
27 // This function uses the OMDB API to get the information of the movie
28 // the user has searched
29 function getOmdbRecord($movieName, $ApiKey)
30 {
31     //replace spaces " " with plus sign "+"
32     $movieName = str_replace(' ','+', $movieName);
33     //stitch url that gets JSON file
34     $path = "http://www.omdbapi.com/?apikey=$ApiKey&t=$movieName";
```

# The Green Muskoxen

## SearchBar.php Continued

```
35     //gets contents of json file
36     $json = file_get_contents($path);
37     return json_decode($json, TRUE);
38 }
39
40 function movieExistsinDB($movieTitle) {
41     $pdo = new PDO("sqlite:MMDataBase.db");
42     //Query to check if movie with specific name is already in Database
43     $sqlMovieExistsInDB = "SELECT MID FROM Movies WHERE Title LIKE :title ";
44     $stmtMovieExistsInDB = $pdo->prepare($sqlMovieExistsInDB);
45     $stmtMovieExistsInDB->bindParam(':title',$movieTitle);
46     $stmtMovieExistsInDB->execute();
47     $movieExistsInDB = $stmtMovieExistsInDB->fetchColumn();
48     // this will return the corresponding MID or false
49     return $movieExistsInDB;
50 }
51 //----- swipe methods -----//
52 function AddCategories($userId, $catArr) {
53     if (count($catArr) > 0) {
54         $pdo = new PDO("sqlite:MM DataBase.db");
55         // Get all records containing scores for categories specific to current user
56         $sqlCatExists = "SELECT CategoryName FROM Scores WHERE UserId=:uid AND CategoryName=:category";
57         // Query to add a category if a record does not already exist
58         $sqlInsertCat = "INSERT INTO Scores (UserId, CategoryName, Score) VALUES (:uid,:category,100)";
59         foreach ($catArr as $cat) {
60             $category = trim($cat); // remove spaces
61             $stmtCatExists = $pdo->prepare($sqlCatExists);
62             $stmtCatExists->bindParam(':uid', $userId);
63             $stmtCatExists->bindParam(':category', $category);
64             $stmtCatExists->execute();
65             $catExists = $stmtCatExists->fetchColumn();
66             //If recode does not already exist insert it into table with initial score of 100
67             if ($catExists == false) {
68                 $stmtInsertCat = $pdo->prepare($sqlInsertCat);
```

# The Green Muskoxen

## SearchBar.php Continued

```
69     $stmtInsertCat->bindParam(':uid', $userId);
70     $stmtInsertCat->bindParam(':category', $category);
71     $stmtInsertCat->execute();
72 }
73 // else do nothing, category exists
74 }
75 } else {
76     echo "Error, category array is empty";
77 }
78 }
79
80 //The following function adjusts the score
81 // of a category based on whether a user passes on
82 // movie of that category or adds to watchlist
83 function AdjustScore($amt, $userId, $catArr) {
84     if (count($catArr) > 0) {
85         $pdo = new PDO("sqlite:MMDataBase.db");
86         $sqlCatExists = "SELECT Score FROM Scores WHERE UserId=:uid AND CategoryName=:name";
87         $sqlUpdateScore = "UPDATE Scores SET Score=:score WHERE UserId=:uid AND CategoryName=:name";
88         foreach ($catArr as $cat) {
89             // assert the category exists
90             $category = trim($cat);
91             $stmtCatExists = $pdo->prepare($sqlCatExists);
92             $stmtCatExists->bindParam(':uid', $userId);
93             $stmtCatExists->bindParam(':name', $category);
94             $stmtCatExists->execute();
95             $catExists = $stmtCatExists->fetchColumn();
96             if ($catExists != false) {
```

# The Green Muskoxen

## SearchBar.php Continued

```
97         // category exists, update the score
98         $newScore = $catExists + $amt;
99         $stmtUpdateScore = $pdo->prepare($sqlUpdateScore);
100        $stmtUpdateScore->bindParam(':score', $newScore);
101        $stmtUpdateScore->bindParam(':uid', $userId);
102        $stmtUpdateScore->bindParam(':name', $category);
103        $stmtUpdateScore->execute();
104    }
105    // else do nothing, category exists
106}
107} else {
108    echo "Error, category array is empty";
109}
110}
111// ----- end swipe methods -----
112
113function populateMovie($movieTitle) {
114    $userId = intval($_SESSION["UID"]);
115    $mTitle = trim($movieTitle);
116    if ($mTitle == null) {
117        echo "error, movietitle is null";
118        return false;
119    }
120    // Using global variables to store information that will be used later
121    global $title,$director,$actors,$year,$imdbRating,$category,$poster,$rated,$plot;
122    $pdo = new PDO("sqlite:MMD DataBase.db");
123    if (movieExistsinDB($mTitle) != false) {
124        // movie exists in database, use database to populate
125        // Query to select movie with specific name
126        $sqlGetMovieDataByName = "SELECT * FROM Movies WHERE Title LIKE :title ";
127        // Prepare Query
128        $stmtGetMovieDataByName = $pdo->prepare($sqlGetMovieDataByName);
```

# The Green Muskoxen

## SearchBar.php Continued

```
129     // Different syntax to inject variable information into Query
130     $stmtGetMovieDataByName->bindParam(':title',$mTitle);
131     $stmtGetMovieDataByName->execute();
132     // Get all records from query execution
133     $movieData = $stmtGetMovieDataByName->fetch();
134
135     // store all relevant information into global variable
136     $title = $movieData["Title"];
137     $director = $movieData["Director"];
138     $actors = $movieData["Actors"];
139     //convert to int to add to database
140     $year = intval($movieData["ReleaseYear"]);
141     //convert to float (cannot do int because it rounds up/down)
142     $imdbRating =floatval($movieData["IMDB_score"]);
143     $category = $movieData["Category"];
144     $poster = $movieData["Poster"];
145     $rated = $movieData["Rated"];
146     $plot = $movieData["Plot"];
147
148 } else {
149     // movie does NOT exist, get data from omdbapi
150     //Use OMDB to get movie information
151     $movieData = getOmdBRecord($mTitle, "2f79417c");
152     //if movie was not found redirect to search page
153     if(count($movieData) < 3) {
154         // If the resulting JSON file has less than 3 keys
155         //then movie was not found, redirect to search page
156         header('Location: search.php');// replace with hosted URL
157         exit();
158     } else{
159         //Get relevant infomation from returned Jsonfile
160         $title = $movieData["Title"];
161         $director = $movieData["Director"];
162         $actors = $movieData["Actors"];
```

# The Green Muskoxen

## SearchBar.php Continued

```
163     //convert to int to add to database
164     $year = intval($movieData["Year"]);
165     //convert to float (cannot do int because it rounds up/down)
166     $imdbRating =floatval($movieData["imdbRating"]);
167     $category = $movieData["Genre"];
168     $poster = $movieData["Poster"];
169     $rated = $movieData["Rated"];
170     $plot = $movieData["Plot"];
171
172     // Insert movie information into Movie Table
173     $sqlInsertNewMovieIntoMovies =
174         "INSERT INTO Movies (Title, Director, Actors, ReleaseYear, Poster,
175             IMDB_score, Rated, Category, Plot) VALUES (?,?,?,?,?,?,?,?,?,?)";
176     $stmtInsertNewMovieIntoMovies = $pdo->prepare($sqlInsertNewMovieIntoMovies);
177     $stmtInsertNewMovieIntoMovies->execute([$title, $director,
178         $actors, $year, $poster, $imdbRating, $rated, $category, $plot]);
179     }
180   }
181   $categories = explode(',',$category);
182   AddCategories($userId, $categories);
183 }
184
185 // Add movie to watchlist called using Ajax
186 function addToWatchlist($movieTitle) {
187   $userId = intval($_SESSION["UID"]);
188   $pdo = new PDO("sqlite:MMDataBase.db");
189   $sqlGetMovieIdByTitle = "SELECT MID, Category FROM Movies WHERE Title LIKE :title";
190   $stmtGetMovieIdByTitle = $pdo->prepare($sqlGetMovieIdByTitle);
191   $stmtGetMovieIdByTitle->bindParam(':title',$movieTitle);
192   $stmtGetMovieIdByTitle->execute();
193   $movieExistsByTitle = $stmtGetMovieIdByTitle->fetch();
194   if ($movieExistsByTitle == false) {
195     $movieError = "This movie does not exist by associated title";
196     return;
```

# The Green Muskoxen

## SearchBar.php Continued

```
197     }
198     $movieId = intval($movieExistsByTitle["MID"]);
199     $categories = explode(',', $movieExistsByTitle["Category"]);
200
201     // check if movie is in watchlist for user
202     $sqlMovieExistsInWatchlist = "SELECT MovieId FROM Watchlist
203         WHERE UserId=:userId AND MovieId=:movieId";
204     $stmtMovieExistsInWatchlist = $pdo->prepare($sqlMovieExistsInWatchlist);
205     $stmtMovieExistsInWatchlist->bindParam(':userId', $userId);
206     $stmtMovieExistsInWatchlist->bindParam(':movieId', $movieId);
207     $stmtMovieExistsInWatchlist->execute();
208     $movieExistsInWatchlist = $stmtMovieExistsInWatchlist->fetchColumn();
209     // fetchColumn() returns "false" if no records are found
210     if ($movieExistsInWatchlist == false) {
211         // movie does NOT exist in user's watchlist, ADD MOVIE, ADD CATEGORIES w/ Scores
212         // THIS IS ADDING 5 TO SCORES
213         AdjustScore(5,$userId, $categories);
214         // insert into watchlist
215         $sqlInsertMovieIdToWatchlist = "INSERT INTO Watchlist (UserId, MovieId, Watched) VALUES(?, ?, 0)";
216         $insertMovieIdToWatchlist = $pdo->prepare($sqlInsertMovieIdToWatchlist);
217         $insertMovieIdToWatchlist->execute([$userId, $movieId]);
218         header('Location: search.php');// replace with hosted URL
219     } else {
220         // movie already is in watchlist, error out
221         echo "Error, this movie is already watched";
222     }
223 }
```

```
226     /* ----- BELOW ALL FUNCTIONS ARE CALLED/INITIATED ----- */
227
228     // If Watch button is clicked, add to watchlist
229     if(isset($_POST['MovieTitle'])) {
230         $titleOfMovie = trim($_POST['MovieTitle']);
231         addToWatchlist($titleOfMovie);
232     }
233
234     if (isset($_POST['search'])) {
235         // Store user input from search page
236         $input_term = trim($_POST['search']);
237         // Make sure searched movie has no special characters
238         if(noSpecialChar($input_term) == true){
239             populateMovie($input_term);
240         } else {
241             header('Location: search.php');// replace with hosted URL
242             exit();
243         }
244     }
245 }
```

# The Green Muskoxen

## SearchBar.php Continued

```
247    ?>
248    <html>
249    <head>
250        <SCRIPT SRC="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></SCRIPT>
251        <link rel="stylesheet" href="css/stylesheet.css">
252        <link rel="stylesheet" href="css/searchbar.css">
253        <link rel="icon" href="assets/favicon/favicon.ico">
254        <title></title>
255    </head>
256
257    <body>
258        <?php include('components/header.php'); ?>
259        <div class= container>
260            <div class=overlay>
261                <br><br>
262                <div style="float: left; margin-left: 10%;">
263                    
264                </div>
265                <div style="float: left; padding-left:4%; color:#f5f5f5; width:60%; float:right:none;">
266                    <h1 id="title"><?= $display = ($title != null) ? $title : "Title: " ; ?></h1><br>
267                    <h3>Director: <?= $director ?></h3><br>
268                    <h3>Year: <?= $year ?> </h3><br>
269                    <h3>Imdb Score: <?= $imdbRating ?></h3><br>
270                    <h3>Actors: <?= $actors ?> </h3><br>
271                    <h3 id="category">Genre: <?= $category ?></h3><br>
272                    <h3>Rated: <?= $rated ?></h3><br>
273                    <h3>Plot:</h3>
274                    <p><?= $plot ?></p>
275                </div>
276                <h3><?php echo $movieError ?></h3>
277                <div class="addtowatchlist" style="width:100%; float:center; height:5%;">
278                    <button class="button" id="watch" onclick="watchmovie()">Watch</button>
279                </div>
280                <div class="searchMore" style="width:100%; float:center; height:5%;">
```

```
281                    <button class="button" id="searchMore" onclick="searchMore()">Back to Search</button>
282                </div>
283            </div>
284        </div>
285        <?php include('components/footer.php'); ?>
286    </body>
287    </html>
288    <script>
289        // This function is too overloaded for SQLite to handle, how can we support?
290        function watchmovie(){
291            var titleOfMovie = String(document.getElementById("title").textContent);
292            var category = String(document.getElementById("category").textContent);
293            const data = {
294                MovieTitle: titleOfMovie,
295                Category: category
296            };

```

# The Green Muskoxen

## SearchBar.php Continued

```
297     $.ajax({
298         type: 'POST',
299         url: 'searchbar.php',
300         data,
301         success: function(data)
302         {
303             alert(titleOfMovie + " added to watchlist");
304         }
305     });
306 }
307
308 function searchMore() {
309     alert("searchmore");
310 }
311 </script>
```

# The Green Muskoxen

## Profile.php

```
1  <?php
2
3      session_start();
4      if (!isset($_SESSION["UID"]))
5      {
6          header("Location: login.php");
7      }
8      $user = $_SESSION["username"];
9      $uid = intval($_SESSION["UID"]);
10
11     $pdo = new PDO("sqlite:MMDataBase.db");
12     $sqlUserWatchlistMID = "SELECT MovieId FROM Watchlist WHERE UserId=?"; //Query to get all movies
13     $stmt = $pdo->prepare($sqlUserWatchlistMID); //Prepare Statement
14     $stmt->execute([$uid]); // Inject variable information into query and Execute
15     $userWatchlistMID = $stmt->fetchAll(); // Get all records from executing query
16
17 ?>
18 <html>
19     <head>
20         <title>My Profile</title>
21         <link rel="stylesheet" href="css/stylesheet.css">
22         <link rel="icon" href="assets/favicon/favicon.ico">
23     </head>
24     <body>
25         <?php include('components/header.php'); ?>
26         <div class= container>
27             <div class = overlay style=" overflow:scroll; overflow-x:hidden; padding-right:10%;">
28                 <br>
29                 <center><h1 style="border-bottom:1px; border-bottom-style:solid;
30                     color:white;"><?php echo $user ?>'s Recommendations</h1></center>
31                 <div class= watchlist style="height:80%; width:90%;margin-left:10%; margin-right:5%; ">
32                     <?php
33                         $pdo = new PDO("sqlite:MMDataBase.db");
34                         $movieErrorLog = "";
35
36                         // Query to get top 5 categories with the highest scores
37                         $sqlTopCategoriesByScoreByUser = "SELECT CategoryName FROM SCORES WHERE UserId=:uid ORDER BY Score DESC";
38                         //prepare satement
39                         $stmtTopCategoriesByScoreByUser = $pdo->prepare($sqlTopCategoriesByScoreByUser);
40                         // inject parameter information into query
41                         $stmtTopCategoriesByScoreByUser->bindParam(":uid", $uid);
42                         // Execute the query
43                         $stmtTopCategoriesByScoreByUser->execute();
44                         $topCategoriesByScoreByUser = $stmtTopCategoriesByScoreByUser->fetchAll();
45
46                         // amount of rows returned by query MINUS ONE
47                         $lengthOfCategories = count($topCategoriesByScoreByUser) - 1;
48
49                         // shell of array to keep track of recs
50                         $existingRecommendationTitles = array("empty0", "empty1", "empty2", "empty3", "empty4");
51
52                         // count of categories that successfully added as recommendations on page.
53                         $countRecommendation = 0;
```

# The Green Muskoxen

## Profile.php Continued

```
54         foreach ($topCategoriesByScoreByUser as $cat) {
55             if ($countRecommendation > 4) {
56                 break;
57             }
58             $category = "%" . $cat["CategoryName"] . "%";
59             $sqlMoviesInWatchlist = "SELECT MovieId FROM Watchlist WHERE UserId=$uid";
60             $sqlRandomMovieLikeCategory = "SELECT Title, Poster FROM Movies WHERE
61                 Category LIKE :cat AND MID NOT IN ($sqlMoviesInWatchlist) ORDER BY random() LIMIT 1";
62             // Prepare Query
63             $stmtRandomMovieLikeCategory = $pdo->prepare($sqlRandomMovieLikeCategory);
64             //Bind variables
65             $stmtRandomMovieLikeCategory->bindParam(":cat", $category);
66             $stmtRandomMovieLikeCategory->execute();
67             $randomMovieLikeCategory = $stmtRandomMovieLikeCategory->fetch();
68             if ($randomMovieLikeCategory == false) {
69                 // Movie with category doesn't exist or is already contained in user's watchlistmovie, next query
70                 continue;
71             }
72             $movieTitle = $randomMovieLikeCategory["Title"];
73             $moviePoster = $randomMovieLikeCategory["Poster"];
74             if (!in_array($movieTitle, $existingRecommendationTitles)) {
75                 // echo $countRecommendation . " " . $movieTitle . " movie does not exist in recs. ";
76                 // A valid recommendation
77                 $existingRecommendationTitles[$countRecommendation] = $movieTitle;
78                 echo"<div class = watchlistmovie>
79                     <img src = '$moviePoster'>
80                     <br>
81                     <center>
82                         <h3 style = 'color:white;'>$movieTitle</h3>
83                     </center>
84                 </div>";
85                 $countRecommendation++;

86             if ($countRecommendation > 4) { break; }
87         } else {
88             $movieErrorLog = $movieErrorLog . " " . $movieTitle . "[" . $countRecommendation . "]: already recommended<br>";
89         }
90     }
91     // echo $movieErrorLog;
92
93     ?>
94     </div>
95
96
97     <center><h1 style="border-bottom:1px; border-bottom-style:solid;color:white;"><?php echo $user ?>'s Watchlist</h1></center>
98     <br>
99     <div class= watchlist style="height:80%; width:90%;margin-left:10%; margin-right:5%; ">
100    <?php
101        // iterate through all movieId's (MID) and find the according movie titles (or entire object... later) per each MID
102        foreach($userWatchlistMID as $row)
103        {
104            $mid = $row["MovieId"];
105            // Query to get all information of movies in watchlist by their Movie IDs from movie table
106            $sqlMovieTitle = "SELECT Title, Director, Poster FROM Movies WHERE MID=?";
```

# The Green Muskoxen

## Profile.php Continued

```
107         // Prepare Query
108         $stmt2 = $pdo->prepare($sqlMovieTitle);
109         // Inject variable information into query and execute
110         $stmt2->execute([$mid]);
111         // Get all records from query execution
112         $data = $stmt2->fetch();
113
114         //Storing Title and Poster information of movies
115         $title = $data["Title"];
116         $director = $data["Director"];
117         $poster = $data["Poster"];
118
119         //This code injects the HTML div that populates the movie information on the Profile Page
120         echo"<div class = watchlistmovie>
121             <img src = '$poster'>
122             <br>
123             <center>
124                 <h3 style = 'color:white;'>$title<h3>
125             </center>
126         </div>";
127     }
128     ?>
129     </div>
130     </div>
131     <div>
132         <?php include('components/footer.php'); ?>
133     </body>
134
135
136     </html>
```

# The Green Muskoxen

## Index.php

```
1 <?php
2     // if user is set in session data, then redirect to movie.php ?
3     //else redirect user to login.php to sign into their account (or create a new one)
4     header("Location: login.php");
5 ?>
```

## Movie.php

```
1 <?php
2 //If not logged in redirect to login page
3 session_start();
4 if (!isset($_SESSION["UID"]))
5 {
6     header("Location: login.php");
7 }
8 // Initialize PDO Object
9 $pdo = new PDO("sqlite:MMDataBase.db");
10
11 $userId = intval($_SESSION["UID"]);
12 // initialize user scores with categories
13 // Query top categories with the highest scores
14 $sqlTopCategoriesByScoreByUser = "SELECT CategoryName FROM SCORES WHERE UserId=:uid ORDER BY Score DESC";
15 //prepare satement
16 $stmtTopCategoriesByScoreByUser = $pdo->prepare($sqlTopCategoriesByScoreByUser);
17 // inject parameter information into query
18 $stmtTopCategoriesByScoreByUser->bindParam(":uid", $userId);
19 $stmtTopCategoriesByScoreByUser->execute();
20
21 // $topCategoriesByScoreByUser- All categories ordered from highest score to lowest
22 $topCategoriesByScoreByUser = $stmtTopCategoriesByScoreByUser->fetchAll();
23 if ($topCategoriesByScoreByUser != false) {
24     $moviesArrFiltered = array();
25     foreach ($topCategoriesByScoreByUser as $cat) {
26         $category = "%" . $cat["CategoryName"] . "%";
27         // $stmtRandomMoviesLikeCategory -
28         $sqlWatchedMovies = "SELECT MovieId FROM Watchlist WHERE UserId=$userId";
29         // SELECT * FROM Movies WHERE Category LIKE '%Action%' MID NOT IN
30         // (SELECT MovieId FROM Watchlist WHERE UserId=11) ORDER BY random();
31         $sqlRandomMoviesLikeCategory = "SELECT * FROM Movies WHERE Category LIKE ?
32             AND MID NOT IN ($sqlWatchedMovies) ORDER BY random()"; // ORDER BY random()
33         $stmtRandomMoviesLikeCategory = $pdo->prepare($sqlRandomMoviesLikeCategory);
```

# The Green Muskoxen

## Movie.php Continued

```
34     //Bind variables
35     $stmtRandomMoviesLikeCategory->execute([$category]);
36     // All movies from specified category
37     $randomMoviesLikeCategory = $stmtRandomMoviesLikeCategory->fetchAll();
38     $moviesArrFiltered = array_merge($moviesArrFiltered, $randomMoviesLikeCategory);
39 }
40 $moviesArrFiltered = array_unique($moviesArrFiltered, SORT_REGULAR);
41 $moviesArr = array_values($moviesArrFiltered);
42 } else {
43     $sqlNoCategoryOrderByRand = "SELECT * FROM Movies ORDER BY random()";
44     $stmtNoCategoryOrderByRand = $pdo->prepare($sqlNoCategoryOrderByRand);
45     $stmtNoCategoryOrderByRand->execute();
46     $moviesArr = $stmtNoCategoryOrderByRand->fetchAll();
47 }
48
49
50 // print_r($moviesArr);
51 // echo "Movies array has " . count($moviesArr) . " rows";
52 ?>
53 <?php
54
55 // This chuck of PHP code adds a movie to the users watchlist if the MovieId is set (if it is clicked)
56 // since "$userId" is defined in the first PHP code chunk, can we omit this variable declaration (below)?
57 function AddCategories($userId, $catArr) {
58     if (count($catArr) > 0) {
59         $pdo = new PDO("sqlite:MM DataBase.db");
60         // Get all records containing scores for categories specific to current user
61         $sqlCatExists = "SELECT CategoryName FROM Scores WHERE UserId=:uid AND CategoryName=:category";
62         // Query to add a category if a record does not already exist
63         $sqlInsertCat = "INSERT INTO Scores (UserId, CategoryName, Score) VALUES (:uid,:category,100)";
64         foreach ($catArr as $cat) {
65             $category = trim($cat); // remove spaces
66             $stmtCatExists = $pdo->prepare($sqlCatExists);
67             $stmtCatExists->bindParam(':uid', $userId);
68             $stmtCatExists->bindParam(':category', $category);
69             $stmtCatExists->execute();
70             $catExists = $stmtCatExists->fetchColumn();
71             //If recode does not already exist insert it into table with initial score of 100
72             if ($catExists == false) {
73                 $stmtInsertCat = $pdo->prepare($sqlInsertCat);
74                 $stmtInsertCat->bindParam(':uid', $userId);
75                 $stmtInsertCat->bindParam(':category', $category);
76                 $stmtInsertCat->execute();
77             }
78             // else do nothing, category exists
79         }
80     } else {
81         echo "Error, category array is empty";
82     }
83 }
```

# The Green Muskoxen

## Movie.php Continued

```
85      //The following function adjusts the score of a category based on whether
86      //a user passes on movie of that category or adds to watchlist
87      function AdjustScore($amt, $userId, $catArr) {
88          if (count($catArr) > 0) {
89              $pdo = new PDO("sqlite:MMDataBase.db");
90              $sqlCatExists = "SELECT Score FROM Scores WHERE UserId=:uid AND CategoryName=:name";
91              $sqlUpdateScore = "UPDATE Scores SET Score=:score WHERE UserId=:uid AND CategoryName=:name";
92              foreach ($catArr as $cat) {
93                  // assert the category exists
94                  $category = trim($cat);
95                  $stmtCatExists = $pdo->prepare($sqlCatExists);
96                  $stmtCatExists->bindParam(':uid', $userId);
97                  $stmtCatExists->bindParam(':name', $category);
98                  $stmtCatExists->execute();
99                  $catExists = $stmtCatExists->fetchColumn();
100                 if ($catExists != false) {
101                     // category exists, update the score
102                     $newScore = $catExists + $amt;
103                     $stmtUpdateScore = $pdo->prepare($sqlUpdateScore);
104                     $stmtUpdateScore->bindParam(':score', $newScore);
105                     $stmtUpdateScore->bindParam(':uid', $userId);
106                     $stmtUpdateScore->bindParam(':name', $category);
107                     $stmtUpdateScore->execute();
108                 }
109                 // else do nothing, category exists
110             }
111         } else {
112             echo "Error, category array is empty";
113         }
114     }
115 }
```

# The Green Muskoxen

## Movie.php Continued

```
116     $userId = $_SESSION["UID"];
117     if(isset($_POST['MovieId'])){
118         // implement scoring functionality
119         $pdo = new PDO("sqlite:MMDataBase.db");
120         $categories = $_POST['Category'];
121         $catArr = explode(',', $categories);
122         if (count($catArr) < 1) {
123             header("Location: login.php");
124         }
125         AddCategories($userId, $catArr);
126         AdjustScore(5,$userId, $catArr);
127
128         // add movie to user's watchlist
129         $movieId = $_POST['MovieId'];
130         $sqlInsertToWatchlist = "INSERT INTO Watchlist VALUES(?, ?, 0)";
131         $stmtInsertToWatchlist = $pdo->prepare($sqlInsertToWatchlist);
132         $stmtInsertToWatchlist->execute([$userId, $movieId]);
133
134         // $stmtFill = $pdo->query($sqlFilteredMovies);
135         // $all = $stmtFill->fetchall();
136     }
```

# The Green Muskoxen

## Movie.php Continued

```
138     if(isset($_POST['NotCategory'])) {
139         // decrement score for passed movie
140         $pdo = new PDO("sqlite:MMDataBase.db");
141         $categories = $_POST['NotCategory'];
142         $catArr = explode(',',$categories);
143         if (count($catArr) < 1) {
144             header("Location: login.php");
145         }
146         AddCategories($userId, $catArr);
147         AdjustScore(-5,$userId, $catArr);
148     }
149 ?>
150 <html>
151     <head>
152         <SCRIPT SRC="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></SCRIPT>
153         <title>Swipe</title>
154         <link rel="stylesheet" href="css/stylesheets.css">
155         <link rel="icon" href="assets/favicon/favicon.ico">
156     </head>
157     <body>
158         <?php include('components/header.php'); ?>
159         <div class= container>
160             <div class=overlay>
161                 <br><br>
162                 <div class="movieinfo">
163                     <h1 id="title">Movie Title</h1>
164                 </div>
165                 <div class="swipe">
```

# The Green Muskoxen

## Movie.php Continued

```
167         <button class="button" id="pass" onclick="pass()">Pass</button>
168     </div>
169     <div class="movieposter">
170         
171     </div>
172     <div class="swipe">
173         <button class="button" id="watch" onclick="watchmovie()">Watch</button>
174     </div>
175     <div class="movieinfo">
176         <h2 id="dir">Director</h2>
177         <h2 id="year">Release Year</h2>
178         <h2 id="act">Actors</h2>
179         <h2 id="cat">Categories</h2>
180         </div>
181     </div>
182 </div>
183 <?php include('components/footer.php'); ?>
184 </body>
185 </html>
```

# The Green Muskoxen

## Movie.php Continued

```
187 <script>
188 var movies = <?php echo json_encode($moviesArr)?>;
189 var moviecount = 0;
190 populateMovie();
191
192 function populateMovie(){
193     document.getElementById("poster").src=movies[moviecount]["Poster"];
194     document.getElementById("title").innerHTML = movies[moviecount]["Title"];
195     document.getElementById("dir").innerHTML = "Director: " + movies[moviecount]["Director"];
196     document.getElementById("year").innerHTML ="Release Year: " + movies[moviecount]["ReleaseYear"];
197     document.getElementById("act").innerHTML = "Actors: " +movies[moviecount]["Actors"];
198     document.getElementById("cat").innerHTML = "Category: " +movies[moviecount]["Category"];
199 }
200
201 function pass() {
202     var movietitle = movies[moviecount]["Title"];
203     var movieCategories = movies[moviecount]["Category"];
204     // console.log(movieCategories + " " + movieIdentity);
205     const data = {
206         NotCategory: movieCategories
207     };
208     $.ajax({
209         type: 'POST',
210         url: 'movie.php',
211         data,
212         success: function(data)
213     {
214         console.log(movietitle + " disliked");
215         nextMovie();
216     }
217 });
218 }
```

# The Green Muskoxen

## Movie.php Continued

```
219      }
220      function nextmovie(){
221          moviecount = moviecount + 1;
222          populatemovie();
223      }
224
225      function watchmovie(){
226          var movietitle = movies[moviecount]["Title"];
227          var movieIdentity = movies[moviecount]["MID"];
228          var movieCategories = movies[moviecount]["Category"];
229          // console.log(movieCategories + " " + movieIdentity);
230          const data = {
231              MovieId: movieIdentity,
232              Category: movieCategories
233          };
234          $.ajax({
235              type: 'POST',
236              url: 'movie.php',
237              data,
238              success: function(data)
239              {
240                  console.log(movietitle + " added to watchlist");
241                  nextmovie();
242              }
243          });
244
245      }
246
247      </script>
248
```

# The Green Muskoxen

## Search.php

```
1  <?php
2  //if not logged in redirect to login page
3  session_start();
4  if (!isset($_SESSION["UID"]))
5  {
6      header("Location: login.php");
7  }
8 ?>
9
10 <html>
11   <head>
12     <title>search</title>
13     <link rel="stylesheet" href="css/stylesheet.css">
14     <link rel="icon" href="assets/favicon/favicon.ico">
15   </head>
16   <body>
17     <?php include('components/header.php'); ?>
18     <div class= container>
19       <div class=overlay>
20         <br><br>
21         <center><h1 style="color:white;">search</h1>
22         <form method="post" action="searchbar.php">
23           <input type="text" placeholder=
24             |"Enter a movie title..." name ="search" style="width: 30%; height:3%; padding:1%;" required>
25             <input type="submit" Value="Go" name="submit" style="width:3%; height:4%;">
26           </form>
27         </center>
28       </div>
29     </div>
30     <?php include('components/footer.php'); ?>
31   </body>
32 </html>
```

# The Green Muskoxen

## Register.php

```
1 <?php
2 $usernameErr = $passwordErr = $emailErr = $globalErr = "";
3 $attUsername = $attPassword = $attEmail = "";
4 $username = $password = $email = "";
5
6 function noSpecialChar($string) {
7     if (preg_match('/[\^\E%&*()}{#~?><>,|=+-]/', $string)) {
8         // one or more of the 'special characters' found in $string
9         return false;
10    }
11    return true;
12 }
13
14 if(isset($_POST["username"]) && isset($_POST["password"]) && isset($_POST["email"]))
15 {
16     $attUsername = trim($_POST["username"]);
17     $attPassword = trim($_POST["password"]);
18     $attEmail = trim($_POST["email"]);
19     // assert inputs are not empty
20     if (empty($attUsername) ||
21         empty($attPassword) ||
22         empty($attEmail))
23     {
24         // fields were empty after trimming
25         $globalErr = "error, all fields must be filled";
26     } else if (!noSpecialChar($attUsername) || strlen($attUsername) < 3) {
27         // Username has special characters
28         $usernameErr = "Username must be 4 characters or longer and special
29             Characters are not allowed, please change your username and try again";
30     } else if (!noSpecialChar($attEmail) || strlen($attEmail) < 3) {
31         // email has special characters
32         $emailErr = "Special Characters are not allowed, please use a valid email and try again";
33     } else if (!noSpecialChar($attPassword) || strlen($attPassword) < 3) {
```

# The Green Muskoxen

## Register.php Continued

```
34     // Password has special Characters
35     $passwordErr = "Password must be 5 characters or longer and special characters are not
36         allowed, please change your password and try again";
37 } else {
38     // ALL INPUT IS SAFE (no risk of sql injection/xss attacks)
39     $username = $attUsername;
40     $email = $attEmail;
41     $password = md5($attPassword);
42
43     $pdo = new PDO("sqlite:MM DataBase.db");
44     $sqlUEPrep = $pdo->prepare("SELECT UID FROM Users WHERE username=?");
45     $sqlUEPrep->execute([$username]);
46     $sqlUsernameExists = $sqlUEPrep->fetchColumn();
47     if ($sqlUsernameExists == false) {
48         // insert username, email, role, and init - init is a
49         // value used to determine if a user goes to "personalize" page
50         $newUserAccountInsertSqlStmt = "INSERT INTO Users (username, email,
51             role, init) VALUES(?, ?, 'user', 0)";
52         $pdo->prepare($newUserAccountInsertSqlStmt)->execute([$username, $email]);
53
54         $stmt = $pdo->prepare("SELECT UID FROM Users WHERE username=?");
55         $stmt->execute([$username]);
56         $uid = $stmt->fetchColumn();
57
58         $newUserPasswordInsertSqlStmt = "INSERT INTO Passwords (Password, UserId) VALUES(?, ?)";
59         $pdo->prepare($newUserPasswordInsertSqlStmt)->execute([$password, $uid]);
60
61         echo "Account successfully created";
62         header("Location: login.php");
63     } else {
64         $usernameErr = "Username already exists";
65     }
66 }
```

# The Green Muskoxen

## Register.php Continued

```
68     ?>
69
70
71     <html>
72
73     <head>
74         <title>Register</title>
75         <link rel="stylesheet" href="css/stylesheets.css">
76         <link rel="stylesheet" href="css/register.css">
77         <link rel="icon" href="assets/favicon/favicon.ico">
78     </head>
79
80     <body>
81         <body>
82             <div class="header" style="font-size:4vw;">
83                 
84             </div>
85             <div class="container">
86                 <div class="overlay">
87                     <br>
88                     <form id="msform" method="POST">
89                         <fieldset>
90                             <h2 class="fs-title">Create your account</h2>
91                             <input type="text" placeholder="username" name="username"></input><br>
92                             <span class="help-block"><?php echo $usernameErr; ?></span><br>
93
94                             <input type="email" placeholder="email" name="email"></input><br>
95                             <span class="help-block"><?php echo $emailErr; ?></span><br>
96
97                             <input type="password" placeholder="password" name="password"></input><br>
98                             <span class="help-block"><?php echo $passwordErr; ?></span><br>
99                             <span class="help-block"><?php echo $globalErr; ?></span>
100                             <input type="submit" value="Create Account" class="action-button"></input><br>
101                             <a href="index.php">Log In</a>
102                         </fieldset>
103                     </form>
104                 </div>
105             </div>
106             <?php include('components/footer.php'); ?>
107         </body>
108     </html>
```

# The Green Muskoxen

## Contact.php

```
1 <?php
2 // If user is not logged in redirect to login page
3 session_start();
4 if (!isset($_SESSION["UID"]))
5 {
6     header("Location: login.php");
7 }
8 ?>
9 <html>
10 <head>
11     <title>Contact Us</title>
12     <link rel="stylesheet" href="css/stylesheets.css">
13     <link rel="icon" href="assets/favicon/favicon.ico">
14 </head>
15
16 <body>
17     <?php include('components/header.php'); ?>
18     <div class="container">
19         <div class="overlay">
20             <br><br>
21             <!-- Connect to custom Google form -->
22             <form id="msform" action="https://docs.google.com/forms/u/0/d/e/1FAIpQLSe59K68Mbo_D0oxof0kmTBfq2eCp8UuAk8-X10oMgbNds0pbq/formResponse"
23                 method="post" style="width: 700px;">
24                 <fieldset>
25                     <h2 class="fs-title">Contact Us</h2>
26                     <input type="text" name="entry.590676321" placeholder="Full Name" name="fullname" required>
27                     <input type="email" name="entry.730639748" placeholder="Email" name="email" required>
28                     <textarea type="text" name="entry.78704734" placeholder="Write something..." style="height: 200px" required></textarea>
29                     <input type="submit" value="Submit" class="action-button"><br>
30                 </fieldset>
31             </form>
32         </div>
33     </div>
34     <?php include('components/footer.php'); ?>
35 </body>
36 </html>
```

## Logout.php

```
1 <?php
2 session_start();
3 session_destroy();
4 header("Location: login.php");
5 //test
6 ?>
```

# The Green Muskoxen

## CSS

### Stylesheet.css

```
1  @import url(https://fonts.googleapis.com/css?family=Montserrat);
2
3  /*Fonts */
4  body {
5      font-family: montserrat, arial, verdana;
6      overflow-x: hidden;
7  }
8
9  /*Header with site title ad menu links*/
10 .header{
11     height: 15%;
12     background-color: #4d0013;
13     text-align: center;
14     color: #f5f5f5;
15 }
16
17 /* Site Title */
18 .titlepic{
19     padding-left: 0.5%;
20     float:left;
21     width: 60%;
22     height: 100%;
23 }
24 .titlepic img{
25     height: 100%;
26     width: 100%;
27 }
28
29 .header img{
30     max-height: 100%;
31     width: 100%;
32 }
33
```

# The Green Muskoxen

## Stylesheet.css Continued

```
34  /*Menu Links*/
35  .links{
36      float: left;
37      width: 30%;
38      height: 100%;
39      text-align: center;
40      font-family:sans-serif;
41      color: #f5f5f5;
42      position: relative;
43      top: 50%;
44      padding-left: 5%;

45
46
47  }
48  .links a{
49      margin-top:40%;
50      vertical-align: middle;
51      padding-left: 2%;
52      color: #f5f5f5;
53      text-align: center;
54      text-decoration: none;
55  }
56  .links a:hover{
57      text-decoration:underline;
58
59  }
```

# The Green Muskoxen

## Stylesheet.css Continued

```
61  /*Transparent blacked out background*/
62  .container{
63      background-image: url(..../assets/popcorn.jpg);
64      height:auto;
65  }
66  .overlay{
67      height: 100%;
68      width: 100%;
69      background: rgba(0, 0, 0, 0.9);/* set opacity*/
70      overflow-x: hidden;
71
72  }
73
74  * {margin: 0; padding: 0;}
75
76
77  /*form styles*/
78  #msform {
79      width: 400px;
80      margin: 50px auto;
81      text-align: center;
82      position: relative;
83  }
84
85  #msform fieldset {
86      background: white;
87      border: 0 none;
88      border-radius: 3px;
89      box-shadow: 0 0 15px 1px rgba(0, 0, 0, 0.4);
90      padding: 20px 30px;
91      box-sizing: border-box;
92      width: 80%;
93      margin: 0 10%;
94  }
```

# The Green Muskoxen

## Stylesheet.css Continued

```
95      /*stacking fieldsets above each other*/
96      position: relative;
97  }
98  /*Hide all except first fieldset*/
99  #msform fieldset:not(:first-of-type) {
100     display: none;
101  }
102
103 #message{
104     padding: 25%;
105  }
106 /*inputs*/
107 #msform input, #msform textarea {
108     padding: 15px;
109     border: 1px solid #ccc;
110     border-radius: 3px;
111     margin-bottom: 10px;
112     width: 100%;
113     box-sizing: border-box;
114     font-family: montserrat;
115     color: #2C3E50;
116     font-size: 13px;
117  }
118 /*buttons*/
119 #msform .action-button {
120     width: 110px;
121     background: #4d0013;;
122     font-weight: bold;
123     color: white;
124     border: 0 none;
125     border-radius: 1px;
126     cursor: pointer;
127     padding: 10px 5px;
128     margin: 10px 5px;
129 }
```

# The Green Muskoxen

## Stylesheet.css Continued

```
130     #msform .action-button:hover, #msform .action-button:focus {  
131         box-shadow: 0 0 0 2px white, 0 0 0 3px #4d0013;;  
132     }  
133  
134     /*headings*/  
135     .fs-title {  
136         font-size: 15px;  
137         text-transform: uppercase;  
138         color: #2C3E50;  
139         margin-bottom: 10px;  
140     }  
141     .fs-subtitle {  
142         font-weight: normal;  
143         font-size: 13px;  
144         color: #666;  
145         margin-bottom: 20px;  
146     }  
147  
148     /* Movie Title and information Divs. */  
149     .movieinfo{  
150         color: f5f5f5;  
151         text-align: center;  
152     }  
153  
154  
155     /*Container Divs for swipe and pass buttons */  
156     .swipe{  
157         float: left;  
158         height: 60%;  
159         width: 37.5%;  
160         margin-top: 2%;  
161         margin-bottom: 2%;  
162     }
```

# The Green Muskoxen

## Stylesheet.css Continued

```
165     /*Buttons for Swiping Functionality*/
166     #watch{
167         width: 200px;
168         background: green;;
169         font-weight: bold;
170         color: white;
171         border: 0 none;
172         border-radius: 1px;
173         cursor: pointer;
174         padding: 20px 5px;
175         font-size: 20px;|
176         margin: 0;
177         position: relative;
178         top: 45%;
179         left: 20%;
180
181     }
182     #watch:hover, #watch:focus{
183         box-shadow: 0 0 0 4px white, 0 0 0 3px #4d0013;;
184     }
185
186     #pass{
187         width: 200px;
188         background: #4d0013;;
189         font-weight: bold;
190         color: white;
191         border: 0 none;
192         border-radius: 1px;
193         cursor: pointer;
194         padding: 20px 5px;
195         font-size: 20px;
196         margin: 0;
197         position: relative;
198         top: 45%;
```

# The Green Muskoxen

## Stylesheet.css Continued

```
199     left: 50%;  
200 }  
201  
202 #pass:hover, #pass:focus{  
203     box-shadow: 0 0 0 4px white, 0 0 0 3px #4d0013;;  
204 }  
205  
206 /*Container for image poster */  
207 .movieposter{  
208     margin-top: 2%;  
209     margin-bottom: 2%;  
210     float: left;  
211     height: 60%;  
212     width: 25%;  
213 }  
214  
215 .movieposter img{  
216     height: 100%;  
217     width: 100%;  
218 }  
219  
220  
221 /* Profile page movie CSS */  
222 .watchlistmovie{  
223     width: 300px;  
224     height: 500px;  
225     margin-top: 1%;  
226     margin-bottom: 3%;  
227     margin-left: 0.1%;  
228     float: left;  
229     background-color: white;
```

# The Green Muskoxen

## Stylesheet.css Continued

```
230     transition: transform .5s;
231 }
232
233 .watchlistmovie img{
234     width: 100%;
235     height: 100%;
236     padding: 2px;
237     background-color: white;
238 }
239
240 .watchlistmovie:hover{
241     -ms-transform: scale(1.1); /* IE 9 */
242     -webkit-transform: scale(1.1); /* Safari 3-8 */
243     transform: scale(1.1);
244 }
245 table, th, td {
246     border: 1px solid black;
247     border-collapse: collapse;
248 }
249 th, td {
250     padding: 5px;
251     text-align: left;
252 }
253 th {
254     font-style: bold;
255 }
256 table.centerTB {
257     margin-left: auto;
258     margin-right: auto;
259 }
260 .centerTB {
261     background-color: rgba(255, 0, 0, 0.6);
262 }
```

# The Green Muskoxen

## Stylesheet.css Continued

```
262     }
263     watchlist::-webkit-scrollbar{
264         display: none;
265     }
266 
```

## Register.css

```
1     .help-block {
2         color: red;
3         font-size: 14px;
4     }
5 
```

## Searchbar.css

```
1     .search-movie {
2         float: left;
3         margin-left: 10%;
4     }
5
6     .search-movie img {
7         height: 600px;
8         padding: 3%;
9         background-color: white;
10        width: 400px;
11    }
12
13    .movie-info{
14        float: left;
15        padding-left: 4%;
16        color: #f5f5f5;
17        width: 60%;
18        float-right: none;
19    }
20 
```

# The Green Muskoxen

## PLANNING AND WORKFLOW

### ClickUP

#### GANT View



#### Calendar View

This calendar view shows tasks scheduled throughout October 2020. Each day's tasks are listed in a grid format. Many tasks have descriptions and due times attached to them. For example, on October 2, there is a task 'Make movie images links' due at 11:30pm. On October 12, there are two tasks: '6.02 Manage all visual and graphic content for the website' and '2.08 Update GitHub Page' both due at 11:30pm. The tasks are color-coded and some have red dots indicating they are due or in progress.

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

# The Green Muskoxen

## Discord

### Session Planning Channel

The screenshot shows a Discord interface with the channel '# session-planning' selected. The left sidebar shows categories like INFORMATION, TEXT CHANNELS, and VOICE CHANNELS. The main pane displays messages:

- Robert** 10/20/2020: We're meeting wednesday again correct? Id like to go over the presentation then. Im envisioning about 12 min dedicated to rdp and the other 8 as a demo of our functioning webapp. Likely the same presenters for the rdp then someone new for the actual working app  
Reactions: 2
- reichard.stacey** 10/20/2020: And just a heads up ill be unavailable Friday and saturday so id like to clear up any questions with recording etc beforehand  
Reactions: 2
- reichard.stacey** 10/20/2020: @Robert Yes! Lets plan to meet on Wednesday. We may have to wait until 9pm to meet though because I know a few people have class.  
Reactions: 3
- nathanhester** 10/20/2020: @everyone Shooting to meet for 9:00 PM CST tomorrow night. We're going to discuss requirements for the Preview/Final Demo and any unfinished business with the algorithm. I'm hoping everyone will be able to join in.  
Reactions: 3
- nathanhester** 10/20/2020: Won't be able to make the 9:00pm meeting, but if I get off this other meeting early ill pop in.  
Reactions: 1
- Zaid Akel** 10/21/2020: @Rich @Ali (@everyone) Looking to meet tomorrow to go over algorithm implementation tomorrow!

At the bottom right of the message list, it says "October 21, 2020".

# The Green Muskoxen

## Meeting Notes

### Agenda – 10/11/20

Good job to everyone on the RDP Demo! And many thanks to Robert for being patient while we all worked to pull this one together.

- Has everyone had the chance to view all two (or three if done) demo videos? We all need to have a thumbs up so that Robert can officially post them to Panapto/D2L.
- Time Logs need to be updated ASAP. The professor will be looking at the time logs and it will affect our grade for the RDP Demo. We should also add a TOTAL row to the bottom of each of our time logs to show the total number of hours we've spent working on the project.
- Update on GoDaddy hosting (if available)  
Site will be available within the next 2 hours at [staceyreichard.com/CSC394](http://staceyreichard.com/CSC394)
- Zaid or Ali: Can one of you please get AWS/LightSail running so that we can host a live backup of the site? (Yes, by the 19<sup>th</sup>)
- Due dates for the last two demos have been pushed up:  
Preview Demo videos due to Robert no later than October 25<sup>th</sup>.  
Final Demo videos due to Robert no later than November 1<sup>st</sup>.
- Here are the details on the Preview Demo:  
<https://condor.depaul.edu/~elliott/394/demos/d-preview/>. Note that we must demonstrate the running program during this demo.
- Is there anything is particular that we need to accomplish during tonight's class meeting?
- This is due on the 16<sup>th</sup>:  
<https://condor.depaul.edu/~elliott/394/demos/Checklist-ReviewRDP.html>

Requirements document due on

### FTP credentials for GoDaddy [Staceyreichard.com/moviematcher](http://Staceyreichard.com/moviematcher)

Host: 107.180.21.70

Username: [elonslilbabies@staceyreichard.com](mailto:elonslilbabies@staceyreichard.com)

Password: XaeA-123

Port: 21

Implementation. What do we have left that needs to be worked on.

Brainstorming on the algorithm.

Search page needs to be done as well.

Start basic and let the alg. Grow based on what we are able to get done.

6:30 tomorrow for our mandatory meeting.

# The Green Muskoxen

## Agenda 10/19/20 Green Muskoxen

This week we need to make major headway on our final project. Demo videos are due to Robert no later than next Sunday night.

- The website is currently accessible at: staceyreichard.com/moviematcher. I also installed a free 90 day SSL certificate to it and set the site up to redirect to https automatically. This should keep us going through the end of the quarter.
- There are LOT of tasks that need to be marked as In Progress or even complete in ClickUp.
- The homepage of the site is showing all links. Did we want it to be solely a login page? If so, we need to remove the header.php directive from the index.php code and possibly leave the footer for version control.
- Where are we at with tasks and the algorithm (have Nathan give an update on this)
- Is everyone comfortable with the tasks that they need to complete this week? Do we need to move anything around?
- How does Rich feel about the project manual? Does he need any help to get this completed on time? (Going to look into this a bit more)
- What about Casey's requirements document?
- Preview demo:
  - o We need to include a including a short, updated section in our Preview Demo and/or Final Demo that shows off our rock-solid understanding of requirements development. This also needs to be a substantial part of our project manual. I would like to Casey to record a brief update on the requirements for the Preview/Final Demo. This does not need to be super long but would be a nice addition.  
.....  
o Would like for dan to either help with the demo (recording) and or with some tasks that still need to be done to complete everything on time.  
o Ali – test portion
- Reminder: Quiz is due on the 25<sup>th</sup> (Sunday) and the midterm is due on Friday.

### Notes:

- Push the Final Demo vids due to Robert on November 4<sup>th</sup>.
- Database is currently marked to "read only." We need to get this fixed/updated ASAP. Nathan is currently working on this as we speak.
- Nathan is going to spearhead the work on the algorithm.

# The Green Muskoxen

- Need someone to develop the questionnaire (Zaid)
- Need OMdB setup/Search page (Rich, Zaid)

## Class Zoom Call with Professor 10/19/20

**Presentation Managers** – Needs to be proactive with all presenters for the Preview/Final Demos. Need to make sure that the audio is top notch! We can go in any order that we want. Possibly put the stronger components more towards the beginning. We have the opportunity to take this on in any way we wish.

**Video Manager** – Need to be proactive along with the presentation manager. If we have everything by Sunday I think that we will be in great shape!

**Documentation Manager** – Need to make sure that all time logs are up to date! This needs to be included in the full manual. Need to be specific with our time logs. Need to figure out how to best translate the entire project plan into a 2D document. Make sure that the code has GOOD comments in it to be able to include in the project manual. Figure out how to represent the link information in a single document since those reviewing it won't be able to click on hyperlinks. Have the Documentation Manager reserve that last week after the final presentation to finish pulling the project manual together.

**Implementation Manager** – Manager needs to be in close communication with the entire group. Need to be prepared to tell everyone that the code has bugs in it. We can list code as Beta code and we are able to discuss this at the end of our presentation.

**Planners** – Need to be able to trim back and or plan to change things up if need be. Very loosely defined CRUD system. Just need to hold to the specs for the demos.

Do the work that you are interested in!

If we submit the Preview Demo early, let the Professor know and he will grade the demo earlier so we have adequate feedback for the Final Demo.

# The Green Muskoxen

10/21/20 Meeting Agenda

The Green Muskoxen

- The backup site is live!! <https://elonslilbackup.com>
- “Movie Matchers” need to be updated to “Movie Matcher” on the homepage of the site.
- Possibly try changing header “Movie Matcher” font to Broadway to match the favicon.  
<https://www.onlinewebfonts.com/download/19007e0c85468fd509414342e0ca9c68>
- Remove the “Register Now” link from the bottom of the Contact Us form.
- Do we want the Swipe page to always start with the Avengers? Can we make it start with a random movie every time using a rand() PHP function?
- Make the search bar on the Search and search results page more prominent. Add an “Add to Watchlist” button to the page as well.
- Where is the contact form posting to? Should we set it up as a google form? If someone wants to volunteer to do this that would be great, otherwise I’d be happy to take it on.  
<https://support.google.com/a/users/answer/9308623?hl=en>
- Add white border (move from inline code to css) to the search page. This might be nice on the Swipe page as well. Do we want to be consistent with this?
- Update the login.php page to require more secure passwords and to not take an empty form.
- It would be nice to have some design continuity between all of the pages on the site. Would someone be willing to take this on? I am happy to do it but wanted to give someone else the opportunity to work on this as well.

Popcorn background is a bit light on the Profile page. Do we want it to be consistent throughout the entire site (div background and overlay)? Also, to keep with the color scheme, do we want to change the table background color to the gray that we used in the footer (#4F4F4F)?

- Do I have any volunteers to work with @NathanHester to get the algorithm working properly?
- Does anyone else think I am missing something that needs to be completed by the end of the week?

# The Green Muskoxen

- Let @Robert discuss the requirements for the Preview Demo. Preview Demo video clips are due to Robert no later than Sunday at midnight.

## Notes:

**AIMING FOR SUNDAY AT MIDNIGHT.**

**VIDEOS MUST BE IN NO LATER THAN MONDAY AT MIDNIGHT.**

Stacey: Introduce the team and Movie Matcher application. Include what the app is called and who is on our team. 30 seconds or so.

Preview Demo: Same people from the RDP Demo (Casey, Nathan, Stacey) no more than 4 minutes recorded.

- Application Demo: Run through the application as a user. 1-2 minutes (Dan)
- Design: Discuss the security of the application. 4 minutes or less (Nathan)
- Business Model: 1-2 minutes (Rich will do this)
- Planning: Be sure to include time logs. 4 minutes or less (Stacey)
- Requirements: 4 minutes or less (Casey)

Hard deadline site has to look good by the end of the day on Saturday so we can fully have recordings ready to go.

Add vector image to header.

Ali will add a dummy image to the Swipe page for movies that don't populate a poster

Nathan wants to implement a "Forgot Password" field.

Required to the end of the input tags in the forms. They are non-existent right now.

Meeting regarding the algorithm Thursday @  
Need to work on a

Download GitHub desktop application

# The Green Muskoxen

Meeting Agenda 10/26/20

Green Muskoxen

- What's left that needs to be done in regard to the site? I would like to lock it down after X time (should agree upon a time so that those recording can get it done before Wednesday night at midnight). We cannot go any later on that deadline!
  - o Adding Watchlist Button on the searchbar.php (@Robert)
  - o Add searchbar to the header (@Robert)
  - o Algorithm (@NathanHester, @Ali)
  - o Movie posters are just posters. Can someone make them links to the info pages for them (@Zaid)
  - o Movie Matchers graphic turn that into a link (swipe.php) (@Stacey)
  - o Search button look a little prettier
  - o Test the site daily let others know about any bugs or problems (@Rich, @Stacey)
  - o Create brief questionnaire and work on connecting it to the da
  - o Record videos
- Would it still be helpful to have a questionnaire, or should we scrap that altogether? I'm happy to scrap it seeing that we are pretty much out of time right now. As long as the algorithm can handle a majority of what we want it to we shouldn't need it anyway.
- Discuss the scoring system. Have Nathan talk about this.
- Major styling issues. I need a few people to take some of this on and we need to get it done quickly. We can work on final styling touches after the Preview Demo but will then have to make sure there is time to rerecord certain things.

Nathan is going to help Dan get a repo going.

- How does the group feel about the contact for being through Google Forms?
- Let everyone know about the professor's e-mail.  
Hi Stacey.  
..
- 1. The demo should focus on your implementation, which we have not yet seen. But once you've shown that it does run and meets the simple specs you are free to show any aspect of your project that you feel best represents your efforts: planning, testing, group management, design, requirements, presentation management, user study, etc.
- 2. Yes, because requirements was a little weaker for this class (not all groups) then in others I'd like to see your requirements document, and we'll use part of the final demo points for that. I'll post a notice about it, but it might be something like 30 points. Maybe less.

# The Green Muskoxen

I've created a submission link for it in D2L. The FINAL final requirements document usually also appears in your project booklet, but that is up to you.

Good questions!

Best,

-ce

- Did we want to revise the video recording schedule? Talk about recording the parts that don't need to be implemented first.
- Have we heard from Casey at all? I've tried reaching out to help several times in the past week and haven't heard back. If he's not at the meeting tonight we will need to work something out for recording an update on requirements.

## **Meeting Notes:**

We need to make sure that we are documenting our code well. Rich, as the Documentation Manager, are you confident in the code? Do you think additional comments need to be included in our code files? Do you need to make sure that things are documented better? Can Dan document that code for our group?

Need to make sure that the CRUD system is clearly presented in the Preview/Final demo.

Discuss the GitHub workflow. I'd be happy to do that during planning if that is a good spot for it.

Talk about how we used Jira at the beginning and then switched up the ClickUp because things weren't working for us.

9pm tomorrow

# The Green Muskoxen

## Meeting Agenda 11/2/20

I can't believe it's the end of the quarter! Did everyone get a chance to see the grade we got for our Final Demo? We did a GREAT job! 100%!

I don't have a whole lot to cover in our meeting tonight. Tonight, is our final mandatory meeting as a group for this class (although we may need to touch base with one another for the Project Manual).

Is our website (both main and backup) fully updated? The professor is going to take a look at our final sites at some point but not sure when exactly. We need to make sure it's fully functioning.

- **Requirements (we need to get ready to turn this in and also forward everything on to Rich to include in the Project Manual)**

The professor has asked that this be turned in through D2L prior to the due date for the Project Manual. We are not receiving any points for this so no need to worry. We should worry more about what is going in our Project Manual.

I'd like to get Rich's input on what we have right now and if anything needs to be added or subtracted before the Project Manual is due.

- **Project Manual – Due 11/16**

Anything that isn't currently in the Google Drive folder needs to be included. Screenshots, planning, design and requirements docs.

We are expected to have our Time Logs in the Project Manual. If anyone has a time log that's currently out of date, please update it to ensure that we will get maximum points on the Project Manual. It's in everyone's best interest to make sure their time log is up to date.

In addition, we need to choose a time this week to freeze time logs. This will allow us plenty of time to use said time logs as a basis to rank everyone's contributions to our group.

Rich, do you need anything else for the Project Manual that you don't already have? Do you currently have a working document? Do you need anyone's help?

- **Individual Assessment – Due 11/20**

There is a lecture on D2L to help with the completion of everyone's final assessment.

- **Final Concepts Quiz – Due 11/17**

- **Capstone Quiz – Due 11/21**

# The Green Muskoxen

## Github

### Commit History Sample

The screenshot shows a GitHub commit history sample. At the top, there is a dropdown menu set to 'master'. Below it, a section titled 'Commits on Nov 1, 2020' contains four commits:

- Fixed bug in Movie page, initial users have no score so queries based... by nathanhester-depaul committed 15 days ago. (Commit hash: 54f37c4)
- Divided methods up by nathanhester-depaul committed 16 days ago. (Commit hash: fc47e78)
- Minor bug fixes by nathanhester-depaul committed 16 days ago. (Commit hash: 39386cc)
- refactored search page, added a button with no styling. Plot now will... by nathanhester-depaul committed 16 days ago. (Commit hash: 3a101e6)

Below this, another section titled 'Commits on Oct 31, 2020' contains six commits:

- Movies based on high scoring categories are shown first, no duplicate... by nathanhester-depaul committed 16 days ago. (Commit hash: 0071845)
- recommendations are now unique, slight styling fix by nathanhester-depaul committed 16 days ago. (Commit hash: 1d7db05)
- merge current master into this branch for merging into master compata... by nathanhester-depaul committed 16 days ago. (Commit hash: 8ed5e13)
- Recommendations no longer include movies in watchlist by alikumail99 committed 17 days ago. (Commit hash: 763eb4d)
- Added recommendations and added Stacey's additions by alikumail99 committed 17 days ago. (Commit hash: e4970da)
- This works including the added recommendations based on user preferen... by alikumail99 committed 17 days ago. (Commit hash: bcc3ea2)

# The Green Muskoxen

## TIMELOGS

### Robert Kaszubski

Name	Date	Hours	Work Done:
Robert Kaszubski	9/14/2020	2	Initial planning and role assignment, brainstorming ideas for our web app, setting up google drive
Robert Kaszubski	9/15/2020	1	Designed a sample diagram for the look of the website
Robert Kaszubski	9/15/2020	1	Met with group to plan (work on setting up website + discuss presentation)
Robert Kaszubski	9/16/2020	1.5	Created introduction for video
Robert Kaszubski	9/16/2020	1	Met with group to plan (finished setting up website and installed wordpress + plugins)
Robert Kaszubski	9/17/2020	0.5	Worked on scripting and organizing presentation
Robert Kaszubski	9/17/2020	1.5	Met with group to discuss next steps, answered questions about presentation, begun designing website
Robert Kaszubski	9/18/2020	2	Created the main 4 pages on the website, including polishing up the log in page, creating a sample swipe/movies page and a search page. Designed the overall look of the website
Robert Kaszubski	9/18/2020	2.5	Long meeting with the group, rebuilding the website due to a wordpress error and back up failure, teaching others to use elementor plugin
Robert Kaszubski	9/18/2020	0.5	Slightly tweaked the movie template page
Robert Kaszubski	9/19/2020	1	Added Profile page, reviewed video submissions and the presentation script
Robert Kaszubski	9/20/2020	4	Edited and uploaded final video
Robert Kaszubski	9/24/2020	0.5	Exploring plugins and potential databases
Robert Kaszubski	9/28/2020	1.5	Weekly group meeting, discussing future plans
Robert Kaszubski	9/30/2020	1	Group meeting
Robert Kaszubski	10/2/2020	1	Video planning and editing work
Robert Kaszubski	10/5/2020	3	group meeting followed by individual PHP/HTML/JS work
Robert Kaszubski	10/6/2020	2	Figuring out how to solve Swipe feature and utilizing both PHP and JS together
Robert Kaszubski	10/6/2020	2	Implementing a rough profile and watchlist page

# The Green Muskoxen

Robert Kaszubski	10/07/2020	1	Answer questions about the demo presentation, databases, and php code
Robert Kaszubski	10/08/2020	4	Planning Demo editing and uploaded
Robert Kaszubski	10/09/2020	1	Reviewing videos, and figuring out how to proceed with very delayed video submissions
Robert Kaszubski	10/09/2020	2	Assisting with video setup, reviewing videos
Robert Kaszubski	10/10/2020	5	Requirements demo editing and upload.
Robert Kaszubski	10/11/2020	5	Design demo editing and upload
Robert Kaszubski	10/11/2020	1	Group meeting and final review of RDP demo
Robert Kaszubski	10/11/2020	0.5	Final upload onto panopto and D2L post
Robert Kaszubski	10/12/2020	1	Group meeting/ reading project feedback/ planning for next demo
Robert Kaszubski	10/14/2020	1	Group meeting, brainstormed further implementation of algorithm
Robert Kaszubski	10/19/2020	1	Mandatory group meeting, went over algorithm, ideas on search functionality and database implementation
Robert Kaszubski	10/22/2020	3	Looking at editing, exploring a new intro and presentation design.
Robert Kaszubski	10/26/2020	3	Mandatory group meeting going over what still needs to be completed as well as individual work after working on figuring out how to use ajax + php to create add to watchlist button on searchbar page
Robert Kaszubski	10/26/2020	2	Finished implementing searchbar add to watchlist button - touched base with Nathan and Ali about it
Robert Kaszubski	10/27/2020	2	Working on implementing score system to the search page and answering questions regarding presentation
Robert Kaszubski	10/27/2020	0.5	Reviewing presentation material/ scripts/ powerpoints
Robert Kaszubski	10/28/2020	0.5	Reviewing video material sent in from Stacey and Rich
Robert Kaszubski	10/28/2020	2	Adding in scoring system to the search page, trying to solve a database bug
Robert Kaszubski	10/28/2020	1	Fixed database bug causing duplicate entries with Ali
Robert Kaszubski	10/29/2020	1	Reviewing video submissions, downloading material, starting editing, making suggestions
Robert Kaszubski	10/29/2020	1	Mini meeting with half the team to access what we have to improve and plan for the next three days
Robert Kaszubski	10/30/2020	3	Editing requirements section - deciding what to redo and rerecord it
Robert Kaszubski	10/31/2020	2	Editing the introduction, new requirements, and business plan section

# The Green Muskoxen

Robert Kaszubski	10/31/2020	1	Speaking with group members about what needs to be rerecorded, accepting and reviewing new recordings
Robert Kaszubski	11/1/2020	3	Editing all the material I had at the time while waiting for Nathan's clips
Robert Kaszubski	11/1/2020	1	Quick mini group meeting to talk about next steps, bug fixes, and presentation requirements
Robert Kaszubski	11/1/2020	3	Finished editing, render, upload to Youtube and Panopto - waited for processing and reviewed both versions for problems
Robert Kaszubski	11/2/2020	1	Weekly group meeting - went over changes to the presentation, changing the database
Robert Kaszubski	11/6/2020	2	Revisions to the requirement document, script, powerpoint
Robert Kaszubski	11/6/2020	1	Directing Casey in recording the requirements presentation - redoing any part that needs to be redone
Robert Kaszubski	11/6/2020	1	Reviewing footage for the final presentation
Robert Kaszubski	11/7/2020	2	Editing the brand new requirements portion of the presentation
Robert Kaszubski	11/8/2020	4	Finishing editing, getting new clips and rerecordings, final upload and reupload after issues with sound
	Total	92	

## Richard Watson

Name	Date	Hours	Work Done
Richard Watson	9/14/20	2	Initial planning and role assignment, brainstorming ideas for our web app, setting up google drive
	9/15/20	1	More detailed planning on workflow, nailed down wordpress/elementor, discussed server services
	9/16/20	1	Got AWS up and running for backend
	9/17/20	1	Setup jira
	9/18/20	2.5	Meeting, built out site after impromptu crash
	9/19/20	2	Worked on demo script, light styling to website, research on button binding for search bar
	9/20/20	1	Search bar plugin research

# The Green Muskoxen

	9/21/20	1	Monday Team Meeting
	9/24/20	1	Dataset research for local movie database
	9/28/20	1.5	Weekly Team Meeting
	9/29/20	1.5	Requirements Doc, Security within WordPress
	9/30/20		Team meeting. Discussed web service possibilities, communication with MySQL and Wordpress, Requirements Doc, Finite State Machine
	10/1/20	1.5	Half team meeting, troubleshoot SSH Lightsail
	10/3/20	1	Mandatory meeting, only participants were PM and I
	10/4/20	2	Booklet Started
	10/5/20	2.5	Team meeting, set agendas, RDP demo discussions
	10/7/20	1	Team meeting
	10/9/20	1.5	Populate design script, edit RDP docs
	10/11/20	1	Team Meeting, discussed RDP demo, future implementations
	10/12/20	1.5	Mandatory meeting, discussed algorithm brainstorming, future meeting delegations
	10/13/20	0.5	Team meeting
	10/14/20	1	Team meeting to check in and workshop algo ideas
	10/16/20		Team meeting, fixed database code, user code to populate database, 2 brainstormed algo ideas
	10/17/20	1.5	Soft time, studied cosine similarity algorithm to implement in our recommendation algorithm
	10/19/20	1	Mandatory meeting, algorithm refinement, search bar brainstorming and 1 delegation
	10/20/20	2	Got search bar to work with Zaid and Ali! Functions with OMDB API
	10/21/20	1	Team meeting to discuss finalized algorithm implementation
	10/22/20	3	Collab with Ali to get recommendations working, stop duplicates, discuss algo

# The Green Muskoxen

	10/22/20	3	Collab with Zaid to debug broken search bar branch
	10/26/20	1	Retweaked the business plan script to be more verbose, cleaner
	10/26/20	2	Team meeting to catch up on what needs to be done before recording, the future of the algorithm, trying to break the site.
	10/29/20	1	Team meeting to talk about Preview Demo
	10/31/20	2	Wrote out Requirements script with Stacey and Zaid
	11/2/20		Mandatory team meeting to discuss final recording requirements, feedback and additional movement forward, and trying to figure out credential situation with segueing to MySQL
	11/12/20	0.5	Team meeting to catch up on what is done and needs to be done.
	11/13/20	5	Project Manual
	11/14/20	4	Touching up Project Manual
		60	

## Stacey Reichard

Name	Date	Hours	Work Done:
Stacey Reichard	9/14/2020	2	Initial planning and role assignment, brainstorming ideas for our web app, setting up google drive.
Stacey Reichard	9/15/2020	1	Review e-mail from group and document prior to meeting with everyone in the evening.
Stacey Reichard		1.25	Met with The Green Muskoxen to discuss next steps and setting up the Apache server to host WordPress.
Stacey Reichard	9/16/2020	1.5	Met with team to discuss next steps and to finish setting up WordPress on AWS.
Stacey Reichard	9/17/20	1.5	Met with team to plan the Concept Demo Video and final changes to the WordPress site.
Stacey Reichard	9/18/20	1	Worked on setting up a backup WordPress site via GoDaddy hosting.

# The Green Muskoxen

		1	Recorded the video introduction for the Concept Demo.
		2.5	Met with the full team and redrafted the entire Wordpress site due to error. Setup backups to run more frequently.
Stacey Reichard	9/20/2020	1	Reviewed all documents and Comcept Demo Video requirements that are/will be fulfilled prior to class on 9/21.
Stacey Reichard	9/21/2020	1	Weekly required team meeting. Reassigned responsibilites for rest of quarter. Recorded weekly attendence and submitted documents to D2L.
Stacey Reichard	9/28/2020	1	Weekly required team meeting. Discussed the best way to manage our groups project and decided to use ClickUp.
Stacey Reichard	9/29/2020	2	Met with some team members to discuss the requirements document and testing for Movie Matcher.
Stacey Reichard	9/30/2020	1	Discussed the swipe algorithm and database connection of the applied with part of the group.
Stacey Reichard	10/1/2020	1	Assigned tasks to the group (part of the planning) and setup a meeting for 10/3/20 to touch base on said tasks.
Stacey Reichard	10/3/2020	3	Worked on the extensive planning documents for the rest of the quarter.
		1	Reviewed Requirements Document prior to meeting with Richard in the evening.
		1	Met with Richard regarding the requirements document and how things are looking for the RDP Demo.
Stacey Reichard	10/4/2020	2.5	Worked on tonight's meeting agenda, the overall tasklist, and ClickUp in preparation for class.
		1.5	Met with the group this evening to go over the agenda, assign tasks, and touch base about upcoming due dates for the RDP Demo.
Stacey Reichard	10/5/2020	3	Typed up tasks missing from the team's overall dependencies and re-reviewed existing dependencies.
		2	Mandatory weekly meeting and class Zoom meetings.
Stacey Reichard	10/6/2020	2	Entered all missing tasks for the rest of the quarter into ClickUp.
		4	Typed up Planning Document for group and worked on Planning RDP Demo video script.

# The Green Muskoxen

Stacey Reichard	10/7/2020	1	Worked on Planning RDP Demo video script.
		2	Finished getting all tasks into the Planning document and also ClickUp.
		1	Met with group to touch base on where we are at with the RDP Demo and video recordings for Video Manager.
Stacey Reichard	10/8/2020	1	Recorded Planning portion of the RDP Demo.
Stacey Reichard	10/9/2020	2	Worked on the video script for the Design portion of the RDP Demo.
		1	Touched base with several team members to make sure everyone was doing their part to get the RDP Demo finished on time.
Stacey Reichard	10/10/2020	0.5	Touched based with several team members individually to see where we were at with the Requirements and Design portions of the RDP Demo.
Stacey Reichard	10/11/2020	1	Met with team to discuss the final RDP Demo videos and to touch base on the rest of the project.
Stacey Reichard	10/12/2020	1.5	Mandatory team meeting: discussed the Preview Demo and rest of the overall Movie Matcher application as well as the main algorithm for it.
Stacey Reichard	10/13/20	0.5	Team meeting to finish discussing the main Movie Matcher algorithm.
Stacey Reichard	10/14/20	1	Team meeting to work on the algorithm design and implementation of the movie Swipe.php page.
Stacey Reichard	10/16/20	2	Team meeting to work on the main Movie Matcher algorithm implementation. Also worked on getting the main head and footer design of the site in place and coded them separate from all page. I used PHP to pull the header and footer code into the main pages of the site for easy updating later.
Stacey Reichard	10/19/20	1	Setup and installed a free 90-day SSL certificate to the Movie Matcher site for security. I also coded a redirect in .htaccess to for the site to redirect to https without the user having to do anything.
		2	Met with class on Zoom to discuss upcoming deadlines and expectations for different managers. Mandatory team meeting: discussed everything that's been done on the application up to this point and also what we have left to implement before the demo video clips are due to the Video Manager on Sunday evening.

# The Green Muskoxen

Stacey Reichard	10/20/20	1	Met with Zaid and Ali to work on getting the backup running flawlessly. Also, installed SSL certificate to backup. Backup is now live @ <a href="https://elonslilbackup.com">https://elonslilbackup.com</a> .
Stacey Reichard	10/21/20	2	Met with Ali to work on SQL code for the site. Worked on creating favicons for the site optimized for different kinds of devices.
		1	Met with the team to discuss final touches on the site before recording for the Preview Demo begins later this week. Discussed requirements for the Preview Demo with our Video Manager.
Stacey Reichard	10/22/20	3	Made several changes to the site's design including the header, and navigation. Embedded a Google form to the Contact Us page. Read through some of the HTML and PHP code and made slight improvements where needed.
Stacey Reichard	10/23/20	2	Touched base with several on the team regarding the algorithm and where we are at. Reached out to a few individuals within the group (Casey, Ali, Dan, and Zaid) regarding recording and CSS.
Stacey Reichard	10/26/20	3	Created meeting agenda for group meeting. Met with PMs and Implementation Managers on Zoom. Met with full group to discuss where we are at in our implementation, what's left to complete before recording the Preview Demo due next week.
Stacey Reichard	10/27/20	1	Worked on the header.php file and contact.php file for integrate the MovieMatcher site with Google forms. Wanted to use the site's existing CSS styling.
		1	Worked on proofreading and formatting the requirements document for the group. Touched base with team members on various aspects of the project (requirements, CSS, etc.)
Stacey Reichard	10/28/20	1	Worked on creating a script for the Planning portion of the Preview Demo along with a brief introduction.
		1	Video recorded the Introduction for the Preview Demo as well as the Planning portion.
Stacey Reichard	10/31/20	3	Worked with Zaid and Rich on a requirements script for the Preview Demo. This also included the creation of a PowerPoint.
		0.5	Touched base with a few team members on the Preview Demo. Recorded a 10 second outro for the demo and uploaded it to Google Drive.

# The Green Muskoxen

Stacey Reichard	11/1/20		Met with everyone on the team to review the final version of the Preview Demo and to talk about anything else needed prior to submitting it to the D2L.
Stacey Reichard	11/2/20	1	Met with the team to review the Preview Demo and to discuss what tasks are remaining in order to complete the Final Demo on time.
Stacey Reichard	11/5/20	3	Met with Zaid to discuss the Requirements Document and to revise what was previously documented for the Final Demo. These changes were made to be more in line with the feedback we received from the Preview Demo.
		2	Worked with Nathan and GoDaddy to get PHPMyAdmin running with the Movie Matcher application.
Stacey Reichard	11/6/20	3	Met with Zaid again to finish working on the Requirements Document prior to having Casey record the requirements portion of the Final Demo.
Stacey Reichard	11/8/20	1	Touched base with various group members and recorded some brief updates for the Final Demo.
Stacey Reichard	11/9/20	2	Met with everyone as a group for the final time. Discussed a time to freeze time logs, and any last minute details regarding requirements and the project manual.
Stacey Reichard	11/11/20	5	Worked on finishing up requirements.
	<b>Total</b>	<b>92.75</b>	

## Dan Kaminski

Name	Date	Hours	Work Done:
Daniel Kaminski	9/14	2	Team meeting where we started planning/role assignments and shooting ideas about our app. Set up drive.
Daniel Kaminski	9/15/20	1	Team meeting with more planning on the workflow, started wordpress/elementor and talked about server services.
Daniel Kaminski	9/17/20	1.5	Team meeting to plan for the concept demo and finish the wordpress site design.
Daniel Kaminski	9/18/20	2.5	Team meeting to go over the app, however ran into problems so had to quickly restart the app design.
Daniel Kaminski	9/19/20	1	Recorded video on description of website by implementation and programming team

# The Green Muskoxen

Daniel Kaminski	9/21/20	1	Monday meeting with team to reassign responsibilities.
Daniel Kaminski	9/28/20	1.5	Meeting with team to discuss the future on managing group projects and ended with ClickUp as our main.
Daniel Kaminski	9/30/20	1	Meeting with team to talk about our swipe algorithm and database connection.
Daniel Kaminski	10/1/20	1.5	Meeting with team to assign tasks throughout the group and setup a future meeting for progress reports on said tasks.
Daniel Kaminski	10/3/20	2	Worked on Finite State Machine diagram
Daniel Kaminski	10/4/20	1	Added finishing touches to the Finite State Machine diagram
Daniel Kaminski	10/5/20	1.5	Meeting with team where we set agendas and discussed the RDP demo
Daniel Kaminski	10/7/20	1	Meeting with team to discuss our progress on the RDP demo and on video recordings
Daniel Kaminski	10/8/20	2	Worked on diagrams(UML of website + Level 0 DFD) for RDP demo
Daniel Kaminski	10/9/20	2	Worked on diagrams (DataDictionary + FunctionalRelationship) for RDP demo
Daniel Kaminski	10/10/20	1	Answered some design questions for design document for RDP demo
Daniel Kaminski	10/12/20	1.5	Team meeting talked about algorithms, future meeting plans
Daniel Kaminski	10/14/20	1	Team meeting for workshop algorithm ideas
Daniel Kaminski	10/16/20	1	Team meeting to work on site algorithm implementation.
Daniel Kaminski	10/19/20	1	Team meeting, brainstorming search bar
Daniel Kaminski	10/21/20	1	Team meeting, discussing final touches on website.
Daniel Kaminski	10/23/20	2	Worked on CSS for website layout (mobile/pc)
Daniel Kaminski	10/26/20	2	Zoom meeting, as well as Team meeting to discuss where we are at in implementation and Preview Demo.
Daniel Kaminski	10/28/2020	1	Worked on script for my recording section for Preview Demo.
Daniel Kaminski	10/31/2020	2	Recorded videos for describing how our website works, etc for Preview Demo.
Daniel Kaminski	11/1/2020	1	Team meeting to discuss final version of Preview Demo, and any other thing to touch base on.
Daniel Kaminski	11/2/2020	1	Team meeting about how Preview Demo went, and what to do for the Final Demo.
Daniel Kaminski	11/5/20	2	Worked on the Requirements Document with Stacey, Zaid, and Casey to make necessary changes.
Daniel Kaminski	11/8/20	1	Rerecorded videos for describing our website for Final Demo.

# The Green Muskoxen

	total:	41	
--	--------	----	--

## Nathan Hester

Name	Date	Hours	Work Done:
Nathan Hester	09/15/2020	0.5	Team Meeting
Nathan Hester	09/16/2020	0.5	Team Meeting
Nathan Hester	09/17/2020	2.5	Team Meeting, Updated Jira and Made tasks
Nathan Hester	09/18/2020	1	Work on script for video pres
Nathan Hester	09/19/2020	2	Finished script, recorded video for presentation
Nathan Hester	09/21/2020	1	Team Meeting
Nathan Hester	09/28/2020	1	Team Meeting, click up integration
Nathan Hester	09/29/2020	1	Team Meeting
Nathan Hester	10/1/2020	1	Team Meeting, attempt to push to a repository
Nathan Hester	10/5/2020	2	Team Meeting, mild coding/fiddling with php locally.
Nathan Hester	10/6/2020	2	FrontEnd and BackEnd Diagrams
Nathan Hester	10/6/2020	2	Design Document Creation & outline
Nathan Hester	10/6/2020	2	Design Document edits, graph creation
Nathan Hester	10/7/2020	2	Design Document restructure
Nathan Hester	10/7/2020	1	Team Meeting
Nathan Hester	10/8/2020	3	Design document edits
Nathan Hester	10/9/2020	4	Design document edits
Nathan Hester	10/9/2020	2	Design Video Script creation

# The Green Muskoxen

Nathan Hester	10/9/2020	3	Edit Design script
Nathan Hester	10/10/2020	2	Final Design document edits
Nathan Hester	10/10/2020	4	Record Design Video
Nathan Hester	10/11/2020	1	Team meeting
Nathan Hester	10/15/2020	3	Refactor Database schema impl. & rewrite queries to use new database structure
Nathan Hester	10/16/2020	2	More PHP query coding & Team meeting
Nathan Hester	10/19	4	PHP modifications to support new database schema. Security additions were also added
Nathan Hester	10/19	2	Team meeting, addressing algorithm implementation, task management and redispursement
Nathan Hester	10/19	4	Fixed sql queries and how we interact with the database via PHP code
Nathan Hester	10/19	2	Styled website and restructured the file directory.
Nathan Hester	10/20/2020	1	Created GitHub branch creation Wiki for efficient collaboration
Nathan Hester	10/20/2020	0.5	Search Bar review and brainstorming.
Nathan Hester	10/21/2020	1	Team meeting to discuss finalized algorithm implementation
Nathan Hester	10/24/2020	1	Team meeting: we prioritized development of a function that will show movies that are not on the current user's watchlist. The team divided high priority tasks like creating video outlines and scripts while others were assigned to write code.
Nathan Hester	10/24/2020	3	Implemented functionality to query movies only based on the user's current watchlist
Nathan Hester	10/24/2020	2	Reviewed other team members branches and created pull requests. Reviewed all pull requests and merged in necessary additions from each member's branch.
Nathan Hester	10/24/2020	2	Brain stormed possible database additions (tables, columns) to assist the rating/scoring of each movie according to a specific user. Drawed a high level schema of how these tables would interact in our current application setting.
Nathan Hester	10/24/2020	2	Began refactoring jumbled CSS code due to problems with resizing the browser and maintaining application design. Created a new branch for this implementation and pumped out a new CSS skeleton for movie.php
Nathan Hester	10/24/2020	1	Merged the new profile page and cleaned up code
Nathan Hester	10/26/2020	2	Met with group and discussed remaining tasks before the preview demo deadline arrives.
Nathan Hester	10/28/2020	1	Merged code from other branches after changes were reviewed and approved by me

# The Green Muskoxen

Nathan Hester	10/30/2020	2	Began sorting movies by a user's score. Assisted Ali with implementing score queries by username and category.
Nathan Hester	10/31/2020	4	Complete code refactor. Redundant methods were removed and repacked into existing methods. Fixed recommendations for a user. A user will never get duplicate recommendations, and will always get random movies based on the highest scored category for that user. Also refactored the searchbar page.
Nathan Hester	10/31/2020	3	Movie page (swipe page) now randomly queries movies that are part of the highest scored categories for that user, once movies are completely expended, the next best category will be used and so on. Solved the problem of multiple lists of movies with the array_merge method, and solved the issue of duplicate movies with the array_values method.
Nathan Hester	11/01/2020	2	edit script design and security for recording
Nathan Hester	11/01/2020	4	Recorded preview demo: design portion
Nathan Hester	11/02/2020	1	Team meeting: discussed what we are all working on, how we approach the final push
Nathan Hester	11/03/2020	2	Worked on getting a phpmyadmin connection on our backup site, ran into a multitude of issues.
Nathan Hester	11/05/2020	3	Solved issue of database connectivity. Realized running a local server will not allow a connection to phpmyadmin db that is hosted online. Made separate config files for db, one for backup phpmyadmin db and the other for local mysql db
Nathan Hester	11/07/2020	4	Ran into query problems with original sqlite usage, original ajax calls are broken. Site is in critical condition
Nathan Hester	11/08/2020	3	Movie page populated with new queries, backend configuration is somewhat complete for original functionality with new phpmyadmin db
Nathan Hester	11/09/2020	7	Implemented mysql functionality with login, register, swipe, and profile page. Search page has bugs.
		109	

## Casey Berg

Name	Date	Hours	Work Done:
Casey Berg	9/16/2020	1.5	meeting
	9/17/2020	1.5	meeting
	9/18/2020	1.5	Meeting
	9/20/2020	2	Required document research start building it
	9/20/2020	0.5	Take professionally photo
	9/20/2020	1	Meeting

# The Green Muskoxen

	9/28/2020	0.5	requirment document finishing it
	9/28/2020	1	
	9/30	1	Requirements updates based on what they though
	10/5	1	meeting
	10/8	2	requirment document adding more to document
	10/09	1	meeting
	10/09	2	finish requirment document
	10/10	1	update requirment document
	10/10	2	work on practing presentation perform and geting perfect
	10/10	2	issues, correct format and then returning in the requirment video
	10/11	1	meeting
	10/12	1	meeting
	10/19	1	group meeting discussion what need with requirment document
	10/21		group meeting discussuion what need for presentation
	10/22	1	adding to requirment documment
	10/24	1	group meeting
	10/25	3	requirment document and pressentation
	10/26	2	
	10/28/2020	1.5	finish ing on the requirment document
	10/28/2020	4	group meeting
	10/30/2020	1	talking about presentation
	10/30/2020	1.5	practicing and presenttion
	11/2/2020	1	group meeting on project
	11/6/2020	3	finish requirment
	11/7/2020	3	finish requirment
	11/7/2020	1	finish requirment
	11/7/2020	1.5	group meeting finish project for final
		48.5	

# The Green Muskoxen

## Zaid Akel

Name	Date	Hours	Work Done:
	<b>Week 1</b>		
Zaid Akel	9/14/2020	2	Meeting: Initial planning and role assignment, brainstorming ideas for our web app, setting up discord
Zaid Akel	9/15/2020	1	Meeting: Committed to Movie Matcher. Started setting up the wordpress site, decided we'd try and work with Elementor
Zaid Akel	9/16/2020	1.5	Morning meeting: AWS initial set up for users and account registration with Rich and Ali
	9/16/2020	1.5	Night meeting: established lightsail & associated accounts; set up wordpress's backend
Zaid Akel	9/17/2020	1.5	Meeting: Began drafting home page in wordpress with the group. Updated wordpress & database
Zaid Akel	9/18/2020	2.5	Meeting: Redrafted the entire wordpress site due to error
Zaid Akel	9/19/2020	1	Solo: drafted scripts for presentation involving design of the site and implementation of AWS/lightsail
		1	Solo: recorded pieces to be included in the presentation of our concept due monday
	<b>Week 2</b>		
Zaid Akel	9/21/2020	1	Meeting: Mandatory monday meeting, pivoted towards implementation of our design
	<b>Week 3</b>		
Zaid Akel	9/28/2020	2	Meeting: Began planning for our next deadline, the RDP demo; I set up click up integration in Discord for ease of access
Zaid Akel	9/29/2020	1	Meeting: Group meeting; added SSH with Nathan after

# The Green Muskoxen

Zaid Akel	10/1/2020	1.5	Meeting with Nathan: Finished adding SSH
	<b>Week 4</b>		
Zaid Akel	10/4/2020	1.5	Meeting: Potential talks about Godaddy as a backend instead of lightsail
Zaid Akel	10/5/2020	2	Meeting: Began planning for our next deadline, the RDP demo; I set up ClickUp integration in Discord for ease of access
Zaid Akel	10/7/2020	2	Meeting: Set agendas with clickup for RDP demo
Zaid Akel	10/8/2020	1	Solo: Reviewed and edited requirements document
Zaid Akel	10/9/2020	1	Solo: Added details + pictures of code to design document
Zaid Akel	10/10/2020	0.5	Solo: Researched solutions for producing a running backup for MovieMatcher
	<b>Week 5</b>		
Zaid Akel	10/11/2020	1	Meeting: Touched bases regarding RDP demo; reviewed video as a team
		0.5	Solo: researched solutions for producing a running backup for MovieMatcher
Zaid Akel	10/12/2020	1.5	Meeting: Brainstormed swiping algorithms for MovieMatcher
Zaid Akel	10/13/2020	0.5	Meeting: continued brainstorming swiping algorithms for MovieMatcher; settled on a Weighted algorithm
Zaid Akel	10/14/2020	1	Meeting: Started building our schema which will then allow us to work on the weighted algorithm implementation when complete
Zaid Akel	10/15/2020	1	Meeting: Ironed out issues involving our schema implementation

# The Green Muskoxen

Zaid Akel	10/16/2020	1	Meeting: Completed implementation of schema
	<b>Week 6</b>		
Zaid Akel	10/19/2020	1	Meeting with Ali: Started implementing running backup
		1	Meeting: Brainstormed implementation of search page & how it will interact with OMDb
Zaid Akel	10/20/2020	2	Meeting with Ali and Rich: Implemented search page
		0.5	Meeting with Ali and Stacey: Solved our running backup issue; registered <a href="http://elonslilbackup.com">elonslilbackup.com</a> as our backup host url
Zaid Akel	10/21/2020	1	Meeting: Established final details that need implementing as we move onto the last couple of weeks of class
Zaid Akel	10/22/2020	3.5	Meeting with Nate, Rich, Stacey and Ali: Implemented search bar algorithm on search page and restructured some broken php code
Zaid Akel	10/24/2020	1	Meeting with Nate, Rich, Stacey, Ali and Dan: Established where we should all be pivoting towards for the preview
		2	Solo: Began writing the script for the presentation
	<b>Week 7</b>		
Zaid Akel	10/25/2020	3	Solo: Embellished upon current presentation script, edited requirements document; Tinkered with the 'Add to Watchlist' button on searchbar.php
Zaid Akel	10/26/2020	2	Solo: Tinkered with the 'Add to Watchlist' in searchbar.php some more
Zaid Akel	10/27/2020	1.5	Solo: Worked with searchbar.php, tried to get movies to display
		1.75	Meeting: Class meeting; as we enter the final weeks, we had to delegate final tasks to make sure we make Wednesday deadline
Zaid Akel	10/29/2020	1	Meeting with Rich, Rob and Stacey: Meeting regarding requirements documents and script

# The Green Muskoxen

Zaid Akel	10/31/2020	3.5	Meeting with Rich, Stacey and Casey: Completed the rewriting of requirements script, started redrafting requirements document
	<b>Week 8</b>		
Zaid Akel	11/1/2020	3	Meeting with Nathan, Stacey, Ali and Rob: Reviewed edited presentation video, continued redrafting requirements document
Zaid Akel	11/2/2020	2	Meeting: Reviewed comments on presentation continued redrafting requirements document
Zaid Akel	11/3/2020	2.75	Meeting with Casey and Stacey: continued redrafting requirements document, moved matrices to a google sheet for ease of reading
Zaid Akel	11/4/2020	5	Meeting with Casey, Stacey and Dan: started over on the matrices nad requirements document. Began the final draft
Zaid Akel	11/5/2020	2	Meeting with Stacey and Nathan: continued on requirements documentation, worked mostly on the entries in the matrices
Zaid Akel	11/6/2020	1	Meeting with Stacey: continued requirements documentation, worked mostly on formatting and grammar
	<b>Week 9</b>		
Zaid Akel	11/8/2020	2	Meeting with Stacey and Nathan: finalized everything in the requirements documentation
Zaid Akel	11/9/2020	0.5	Meeting: Went over final meeting agenda, determined time log freeze date of wednesday
TOTAL:	71.5		

## The Green Muskoxen