**Matthew Green**

**William Brooks**

**Alex Cragg**

**Website Concept: Movie Recommender**

**Project Goal**

Create a functional and useable website that recommends movies to users.

**Overview**

The Movie Recommendation website allows users to input the title of a movie they enjoyed, then receive a list of 5-10 similar movies they might also like.

**Development Team**

* Matthew Green: Responsible for the development of the front-end of the website and create any required documents pertaining to the front-end
* William Brooks: Responsible for being Scrum Master, AWS, Github, assisting with the general editing and revision of documents, and assist with front-end and backend implementation of the website
* Alex Cragg: Responsible for the development of the backend of the website and create any required documents pertaining to the backend.

**Key Features**

**1. Movie Search & Recommendations**

* **User Input**: Users enter the title of a movie they want recommendations based on.
* **Movie Confirmation**: A confirmation step ensures the user selects the correct movie, especially if multiple films share the same title.
* **Recommendations**: The website returns 5-10 movie recommendations, complete with key information like movie poster, release year, and genre.

**User Interface (UI)**

**1. Page Structure**

The website will consist of three main pages, designed for ease of use and clear navigation:

* **Page 1: Movie Search**
  + Users can search for a movie by typing its name into the search bar.
  + After typing, a drop-down or modal with potential matches appears, and users must confirm their selection to ensure they pick the correct movie.
  + **Key Elements**:
    - Movie search bar.
    - Confirmation step for selecting the correct movie.
* **Page 2: Recommendation Results**
  + After confirming their selected movie, users are presented with a list of 5-10 recommended movies.
  + Each recommendation includes:
    - **Movie Poster**: A visual thumbnail of the movie
    - **Movie Title and Release Year**
  + **Additional Features** (Future Considerations):
    - Users can rate or mark a movie as “watched.”
    - User Profiles where people search history could be tracked
    - Ability to integrate machine learning to better recommend movies
    - Collecting data from streaming services to let users know where they can watch or purchase movies

**Technical Details**

**1. Data Handling and Security**

* **Movie Information**: The website will pull key movie details, such as runtime, director, main cast, genre, and movie posters, through a movie database API like **The Movie Database (TMDb)** or **OMDb**.

**2**. Front End

* Front End Framework: HTML and Javascript will be used for the front-end side of the website. HTML is used to create the UI of the website, while Javascript will be used to interact with the backend to retrieve and send requests and for processing user interactions.

**3. Backend**

* **Backend Framework**: Python will be used for the backend using an algorithm that is given data from an API request to request other movies.

**4. Movie API**

* **Integration with TMDb**: To retrieve the detailed movie information, including title, poster, cast, and genre, the site will query an external movie database.
* **Recommendation Logic**: The recommendation engine could use factors such as genre similarity, director overlap, cast, and user ratings from the API to suggest similar films.

**Technology Used**

* **Front-End**
  + HTML
  + CSS
  + Javascript
* **Back-End**
  + Python
  + The Movie Database API
* **Web-Hosting**
  + AWS Lightsail
* **Other**
  + Github
  + JIRA Scrum Board