MATINEE (ADVANCED FULLSTACK PROJECT)

A PROJECT REPORT

Submitted by

RENU BOJJA

in partial fulfillment for the award of the degree

of

BTECH (HONS)

in

BRANCH OF COMPUTER SCIENCE

School of Computer Science and Engineering
RV University
RV Vidyaniketan, 8th Mile, Mysuru Road, Bengaluru, Karnataka, India - 562112

DECEMBER 2024

DECLARATION

I, **Renu Bojja (1RVU22CSE129),** student fifth semester B.Sc/B.Tech in **Computer Science & Engineering,** at School of Computer Science and Engineering, **RV University,** hereby declare that the project work titled "MATINEE (ADVANCED FULLSTACK PROJECT) " has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science & Engineering** during the academic year **2023-2024**. Further, the matter presented in the project has not been submitted previously by anybody for the award of any degree or any diploma to any other University, to the best of our knowledge and faith.

Name: Renu Bojja Signature
USN: 1RVU22CSE129

Place: RV University

Date: 20 December 2024

School of Computer Science and Engineering

RV University
RV Vidyaniketan,8th Mile, Mysuru Road, Bengaluru, Karnataka, India - 562112

CERTIFICATE

This is to certify that the project work titled **"MATINEE (ADVANCED FULLSTACK PROJECT) "** is performed by Renu Bojja **(1RVUEECSE129) ,** a bonafide students of Bachelor of Technology at the School of Computer Science and Engineering, RV university, Bengaluru in partial fulfillment for the award of degree Bachelor of Technology in Computer Science & Engineering , during the Academic year **2020-2021**.

Prof. Santhosh S Nair	**Dr. Sudhakar KN**	**Dr. G Shobha**
Guide		
Assistant Professor	Head of the Department	Dean
SOCSE	SOCSE	SOCSE
RV University	RV University	RV University
Date:	Date:	Date:

Name of the Examiner — Signature of Examiner

1.

2.

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to the School of Computer Science and Engineering, RV University, for providing us with a great opportunity to pursue our Bachelor's Degree in this institution.

In particular we would like to thank Dr. Sanjay R. Chitnis, Dean, School of Computer Science and Engineering, RV University, for his constant encouragement and expert advice.

It is a matter of immense pleasure to express our sincere thanks to Dr.Mydhili Nair, Head of the department, Computer Science & Engineering University, for providing right academic guidance that made our task possible.

We would like to thank our guide Santhosh S. Head of the Department Dept. of Computer Science & Engineering, RV University, for sparing his valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in completing the Project work successfully.

Date: 20 December 2024

Place: RV University

Renu Bojja

1RVU22CSE129

Semester 5C

TABLE OF CONTENTS

ABSTRACT

In today's digital era, personalized recommendations are essential for navigating the overwhelming abundance of content. The MATINEE project introduces an advanced entertainment recommender system designed to deliver tailored suggestions across movies, music, and books. By leveraging state-of-the-art technologies and methodologies, MATINEE simplifies content discovery and enhances user engagement.

MATINEE’s standout feature is its cross-category recommendation capability. Users can input a movie, book, or song and receive personalized suggestions from other categories, such as discovering music or movies inspired by a favorite book. This innovative integration ensures a seamless and holistic entertainment experience. Additionally, the system includes mood-based recommendations, allowing users to specify emotional states and genre preferences for highly tailored results.

The system’s foundation is built on hybrid recommendation methodologies. Content-based filtering employs TF-IDF vectorization and cosine similarity to analyze textual features, while hybrid approaches combine these techniques with collaborative filtering for greater accuracy and diversity. These methodologies are complemented by a scalable architecture, featuring React for a responsive frontend, Flask and Express.js for backend operations, and MongoDB for efficient data storage. Interactive 3D visualizations using Three.js further enhance the user experience.

Key innovations of the project include secure login features with JWT-based authentication, robust data preprocessing for clean datasets, and optimized algorithms for fast response times. Rigorous testing has validated MATINEE’s ability to deliver accurate and relevant recommendations, meeting diverse user needs.

MATINEE’s contributions to the field of recommendation systems are significant, showcasing the potential of integrating cross-domain recommendations with user-centric design. Future improvements include leveraging generative AI for enhanced mood-based filtering, expanding datasets for broader diversity, and adopting advanced machine learning models for improved accuracy. Incorporating user feedback mechanisms will further refine the system’s functionality.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol	Explanation
TF-IDF	Term Frequency-Inverse Document Frequency

1. INTRODUCTION

In the digital era, users are frequently overwhelmed by an enormous amount of online content, making it challenging to discover movies, music, or books that match their preferences. Personalized recommendations have become a crucial solution, providing tailored suggestions to enhance user experience and reduce the time spent searching for relevant content.

The MATINEE project aims to address this challenge by creating a unified entertainment recommender system that offers cross-category recommendations. The system allows users to input a favorite movie, book, or song and receive suggestions from other categories, such as discovering music based on a movie or books inspired by a song. This innovative approach ensures a seamless, comprehensive entertainment discovery experience.

The methodology employed in MATINEE involves a hybrid recommendation system that combines content-based filtering and collaborative techniques. By utilizing tools such as TF-IDF vectorization and cosine similarity, the system analyzes textual features to generate accurate and relevant recommendations. Collaborative filtering complements this by leveraging patterns from user interactions to further enhance suggestion quality.

To ensure scalability and performance, MATINEE is built on a robust architecture. The frontend is developed using React, providing a responsive and dynamic user interface, while Flask and Express.js power the backend operations. MongoDB serves as the database, offering flexible and efficient storage for user profiles and recommendation data. Interactive 3D visualizations using Three.js enhance the user interface, making the experience engaging and intuitive.

MATINEE also incorporates mood-based recommendations, enabling users to specify their mood and genre preferences for highly personalized results. With a focus on user satisfaction and efficient problem-solving, the project addresses key challenges, such as data redundancy, security, and response time, through rigorous preprocessing, JWT-based authentication, and optimized algorithms.

This system redefines entertainment discovery by combining technical innovation and user-centric design, providing a unique, personalized, and engaging experience.

2. RELATED WORK

Recommender systems have emerged as a critical solution for addressing content overload in digital platforms, enabling personalized suggestions based on user preferences and behaviors. Traditional systems are primarily classified into three categories: collaborative filtering, content-based filtering, and hybrid approaches. While these methods have significantly improved user engagement, they also have limitations, such as a dependency on data quality, lack of cross-category integration, and scalability issues.

Collaborative filtering, widely used in systems like Netflix and Spotify, relies on identifying similarities among users or items. User-based approaches recommend content by finding users with similar preferences, whereas item-based approaches suggest items similar to those previously liked by a user. However, collaborative filtering suffers from challenges such as the "cold start problem," where limited data on new users or items affects recommendation quality.

Content-based filtering addresses some of these limitations by analyzing the attributes of items and matching them with user profiles. Techniques like TF-IDF (Term Frequency-Inverse Document Frequency) and cosine similarity are commonly used for this purpose. These methods are particularly effective in generating personalized recommendations within a single domain but often lack the ability to integrate across multiple categories, such as movies, music, and books.

Hybrid recommender systems combine the strengths of both collaborative and content-based filtering. By integrating diverse data sources and methodologies, hybrid systems provide more accurate and comprehensive recommendations. This approach underpins the MATINEE project, which uniquely incorporates cross-category recommendations alongside traditional methods.

MATINEE distinguishes itself from existing systems by offering a unified platform for entertainment recommendations across movies, music, and books. Features such as mood-based filtering and cross-category recommendations set it apart from conventional systems. Furthermore, the use of modern technologies like Flask, React, and MongoDB ensures scalability and flexibility, addressing the limitations of traditional implementations.

By overcoming challenges like data redundancy, cold start issues, and limited scalability, MATINEE advances the field of recommendation systems. It offers a robust solution that enhances user engagement and satisfaction while setting the stage for future innovations in personalized content discovery.

3. METHODOLOGY

3.1 Methodological Approach

The MATINEE project employs a hybrid recommendation methodology to deliver personalized and cross-category entertainment suggestions. This approach integrates content-based filtering, collaborative filtering, and mood-based recommendation techniques to provide accurate, diverse, and engaging user experiences.

3.2 Methods of Data Collection and Selection

Data for the project was collected from publicly available datasets, including metadata on movies, music, and books. Essential features such as genre, mood, user ratings, and keywords were extracted for analysis. Preprocessing steps included:

- **Data Cleaning:** Removal of duplicates and inconsistencies to improve dataset quality.
- **Feature Engineering:** Creating meaningful feature vectors using TF-IDF vectorization and cosine similarity for textual data.
- **Dataset Splitting:** Dividing the data into training, validation, and test sets to ensure robust evaluation.

3.3 Methods of Analysis

The analysis focused on building a hybrid recommendation system. Content-based filtering utilized TF-IDF to compute item similarity, while collaborative filtering analyzed user interaction patterns. Mood-based recommendations were implemented by mapping mood keywords to relevant content features. The system was rigorously tested to optimize accuracy and response time.

3.4 Evaluation and Justification of Methodological Choices

The hybrid approach was chosen to address limitations in traditional methods. Content-based filtering ensured personalized recommendations, while collaborative filtering added diversity. Mood-based filtering enhances user satisfaction by aligning results with emotional preferences. The choice of React, Flask, and MongoDB ensure scalability and flexibility, supporting real-time recommendations and efficient data management.

Table 3.1 : Feature Categories and Data Sources

Category	**Features**	**Data Source**
Movies	Genre, Ratings, Mood	Kaggle
Music	Mood, Genre, Keywords	Kaggle
Books	Keywords, Genre	Generated using Faker lib

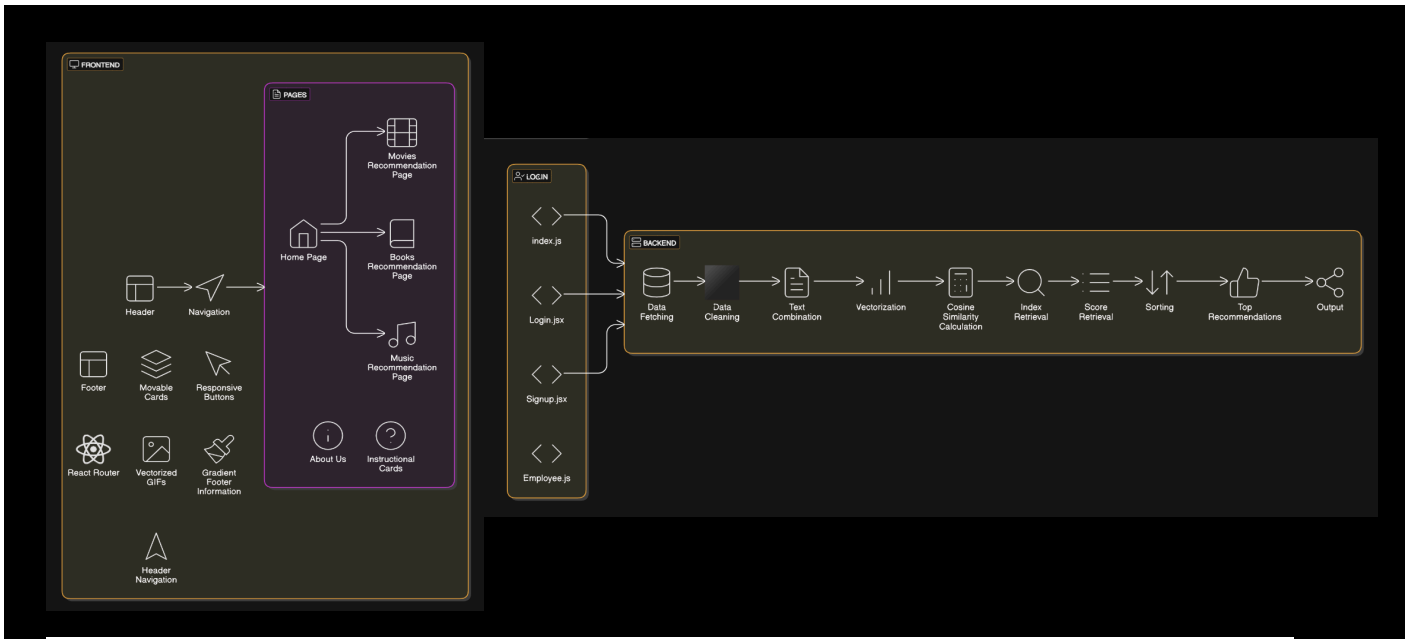

Figure 3.1: MATINEE Recommendation System Architecture

4. IMPLEMENTATION

4.1 Frontend, Backend and Login

Frontend :

1. Header
    a. Movable Cards: Implemented movable cards to enhance interactivity.
    b. Responsive Buttons: Designed buttons that adapt to different screen sizes for a responsive user experience.
2. Footer
    a. Vectorized GIFs: Included vectorized GIFs for dynamic visual effects.
    b. Gradient Footer Information: Applied gradient color changes to footer information for a visually attractive design.
3. Navigation
    a. React Router: Utilized React Router for seamless navigation between pages.
    b. Header Navigation: Enabled navigation through the header to various sections of the website.
4. Pages
    a. Home Page: Contains interactive cards. Clicking on these cards redirects users to the Movies, Books and Music recommendation pages.
    b. Movies Recommendation Page: Provides personalized movie recommendations.
    c. Books Recommendation Page: Offers tailored book suggestions.
    d. Music Recommendation Page: Suggests music tracks based on user preferences.
5. About Us
    a. Created an "About Us" section to provide information about the project and the team.
6. Instructional Cards
    a. A series of instructional cards, guid the user through the website for the first time.
    b. Users can navigate through these cards using next and previous buttons.

Backend :

1. Data Fetching : Connect to MongoDB and fetch collections (MOVIES, MUSIC, BOOKS)
2. Data Cleaning : clean data; ensure no NaN columns or missing values
3. Text Combination : combine relevant text fields i.e genres+keywords+overview = movies
4. Vectorization : Fit on combined corpus to ensure consistent vocabulary
5. Transform individual datasets using the fitted TF-IDF vectorizer.
6. Calculate cosine similarity matrices : movie_music_sim, movie_book_sim, movie_movie_sim, music_music_sim, music_book_sim and book_book_sim
7. Input: Movie name (e.g., "2012").
8. Index Retrieval: Find the index of "2012" in the movie DataFrame.
9. Score Retrieval: Extract similarity scores from movie_music_sim, movie_book_sim, and movie_movie_sim using the retrieved index.
10. Sorting: Sort the similarity scores in descending order.
11. Exclude the queried movie
12. Top Recommendations: Selects the top n similar movies, music tracks and books.
13. Output: Return the records of the top n recommended movies, music, and books as JSON.

Login:

1. Login.jsx and Signup.jsx are responsible for user interaction and form handling on the client side.
    a. Handles user input for email and password.
    b. Uses axios to send a POST request to the server on form submission.
    c. Toggles password visibility.
    d. Navigates to the login component on successful registration. (Signup.jsx)
    e. Navigates to the header component on successful login. (Login.jsx)
2. index.js sets up the server, connects to the database, and handles API requests for login and registration. Implemented using Node.js and Express.js.
    a. Set up an Express server.
    b. Connects to a MongoDB database.

    c. Defines endpoints for user login and registration.
3. Employee.js defines the schema for the user data stored in MongoDB.

4.2 Algorithms and Data Structures

Algorithms :

1. TF-IDF Vectorization : A method that utilizes numerical representations to summarize the combined text fields. The TfidfVectorizer from sklearn.feature_extraction.text is used to transform text data into TF-IDF feature matrices with the corpus being composed of a mixture of text features from movies, music, and books.
2. Cosine Similarity : cosine_similarity from sklearn.metrics.pairwise is used to compute similarity scores between the vectorized items. This results in similarity matrices for movies-music, movies-books, movies-movies, music-music, music-books, and books-books. Quantifies the similarity between items across different datasets. Enables cross-domain recommendations by linking movies with music and books based on textual features.
3. Index Retrieval and Scoring : For each recommendation type (movie, music, book), the index of the queried item is found. Similarity scores are then retrieved from the respective similarity matrices. Ensures the correct items are compared based on their textual similarity.
4. Sorting and Selecting Top Recommendations : The scores are sorted in descending order to get the indices of the top n similar items. These indices are then used to fetch the corresponding records from the datasets. Prioritizes the most relevant recommendations.

Data Structures :

1. DataFrames : Pandas DataFrames are used to store the movie, music, and book datasets.
2. Matrices : TF-IDF feature matrices and cosine similarity matrices are used to store vectorized data and similarity scores, respectively.

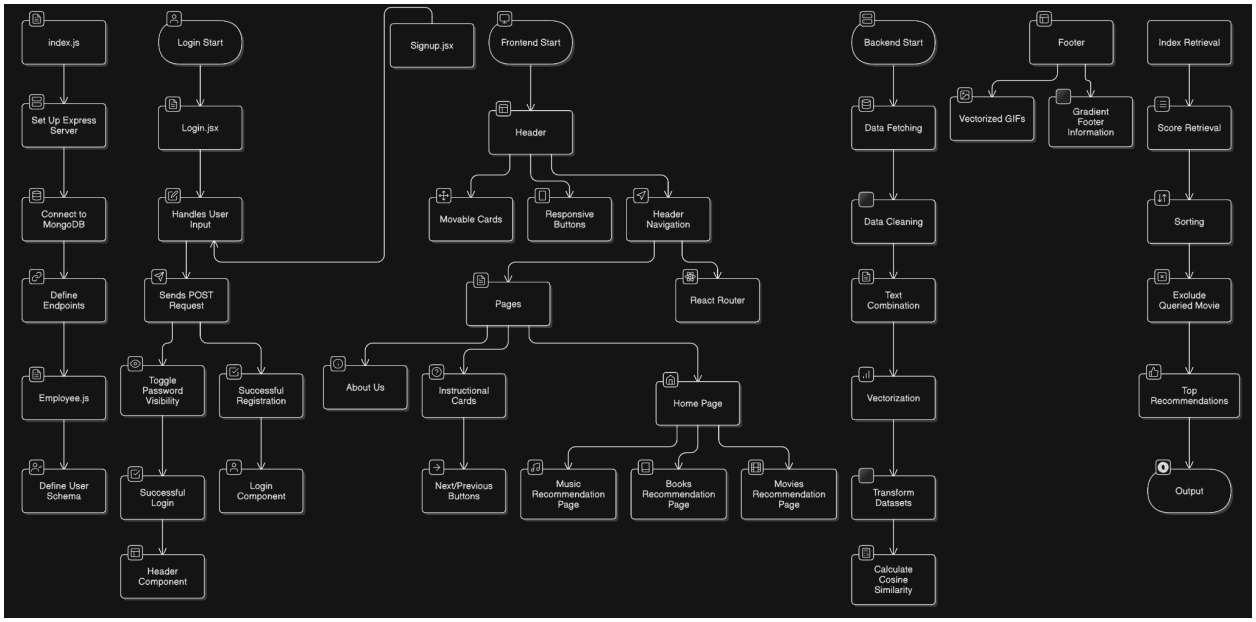

Figure 4.1 : FlowChart of System Architecture of the Recommendation Platform

5. RESULT AND DISCUSSION

The Matinee Recommender System successfully achieves its primary objective of delivering personalized and cross-category recommendations for movies, music, and books. The system integrates mood-based preferences and secure authentication to enhance user experience. Key results include:

- Accurate and relevant recommendations across entertainment categories based on user input.
- Seamless navigation and interactive design powered by React and Tailwind CSS.
- A robust backend system that efficiently processes and analyzes data using MongoDB and Flask.

The system demonstrates the effectiveness of using hybrid recommendation techniques by combining TF-IDF vectorization and cosine similarity. This approach ensures that recommendations are not only relevant but also diverse, catering to user-specific preferences. The ability to provide cross-category recommendations—such as suggesting music and books based on a movie input—enhances the entertainment discovery experience, aligning with findings in the literature review about the advantages of hybrid systems.

The research question aimed to determine whether an integrated recommender system could improve content discovery across movies, music, and books. The results affirm this, showing that the system successfully connects users with personalized content across different categories, reducing the time and effort spent on finding new entertainment.

The choice of using TF-IDF for text vectorization and cosine similarity for similarity computation was justified by their ability to process diverse textual features. These techniques are well-supported in the literature and are proven to deliver accurate recommendations. The system’s design, leveraging React for a dynamic frontend and Flask for lightweight API handling, ensures scalability and maintainability, which are critical for long-term performance.

Strengths:

- Integration of cross-category recommendations provides a novel and enriched user experience.
- Mood-based recommendations and secure login enhance personalization and usability.
- Use of modern tools and frameworks ensures scalability and efficiency.

Limitations:

- The accuracy of recommendations depends on the quality and comprehensiveness of the input data, as noted in the report.
- Challenges in dataset completeness, particularly for books, were addressed using synthetic data, which may not fully represent real-world scenarios.
- Overlapping data occasionally resulted in redundant recommendations, though preprocessing helped mitigate this.

6. CONCLUSION

Matinee is a very intelligent movie, music, and book recommender system. The system uses both personalized and cross-category recommendations. It takes into account user mood, genre preferences, and favorites to offer a more tailored experience. The user experience is made seamless and engaging by means of modern technology like React, Flask, and MongoDB. The project was successful in achieving its goals, which included the development of a secure login prototype with authentication and the provision of valuable information regarding user preferences and system performance.

A scalable and maintainable recommendation system was developed using React, Flask, MongoDB, and other modern technologies. The system features a unique cross-category recommendation capability, enhancing user engagement and satisfaction through personalized content discovery. It contributes to the field of recommender systems by integrating multiple entertainment domains. The website includes cards for understanding and navigating the platform, along with a prototype of mood-based recommendations that takes into account user mood and genre preferences. Additionally, the system stores user favorites to provide more user-specific recommendations.

7. FUTURE SCOPE

To improve the recommendation system, randomly switching genres for recommendations could be implemented to provide a broader variety of content. Enhancing mood-based recommendations using Generative AI will further improve the user experience. Collecting more information about users will help provide better user-specific recommendations. The system should continue to be optimized for better performance, particularly when handling larger datasets. Complementing the system with additional data sources could boost both recommendation accuracy and diversity. Refining recommendations using machine learning techniques is a potential next step in the exploration. Additionally, introducing functionalities that allow users to modify their recommendation preferences would further personalize the experience.

REFERENCES

1. Jain, S. (2024, June 25). *Installation of Node JS on Windows*. GeeksforGeeks. Retrieved August 4, 2024, from https://www.geeksforgeeks.org/installation-of-node-js-on-windows/
2. *Installing Express*. (n.d.). Express.js. Retrieved August 4, 2024, from https://expressjs.com/en/starter/installing.html
3. Emmanuel, O. (2024, February 5). *How to Install React – A Step-by-Step Guide*. freeCodeCamp. Retrieved August 4, 2024, from https://www.freecodecamp.org/news/how-to-install-react-a-step-by-step-guide/
4. *Install Tailwind CSS using PostCSS*. (n.d.). Tailwind CSS. Retrieved August 4, 2024, from https://tailwindcss.com/docs/installation/using-postcss
5. guide, s., & Jain, S. (n.d.). *How to Install MongoDB on Windows?* GeeksforGeeks. Retrieved August 4, 2024, from https://www.geeksforgeeks.org/how-to-install-mongodb-on-windows/
6. (n.d.). ngrok - Online in One Line. Retrieved August 16, 2024, from https://dashboard.ngrok.com/get-started/setup/windows

APPENDIX

Source code :

https://drive.google.com/drive/folders/1oNxBu1H8SUlrYH3CTQJUaoNmJohJYEoK?usp=drive_link

GitHub Link: https://github.com/b21renu/Matinee.git

Screen Shots:

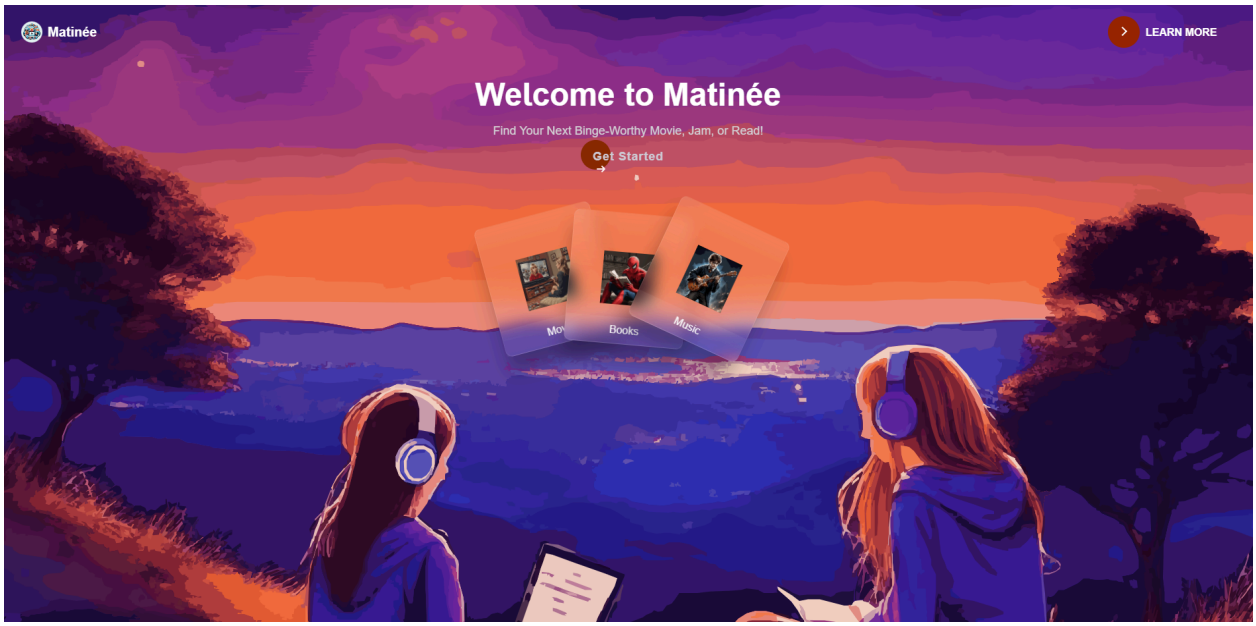

Figure 1 : Webpage Interface

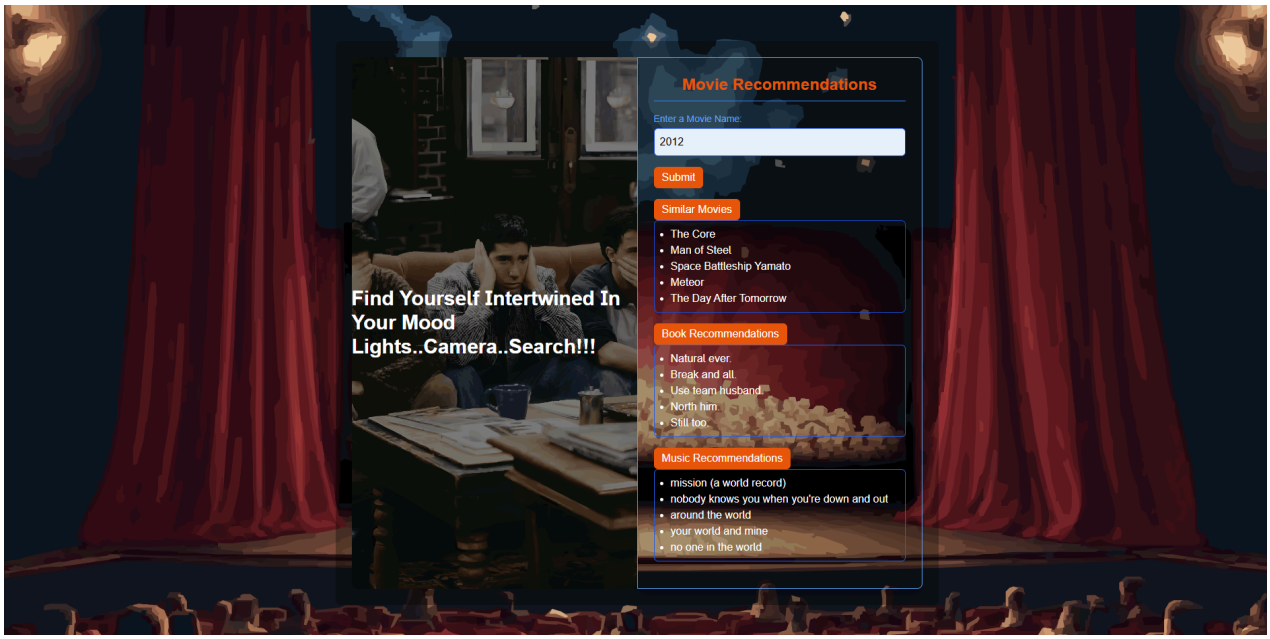

Figure 2 : Movie Recommendation

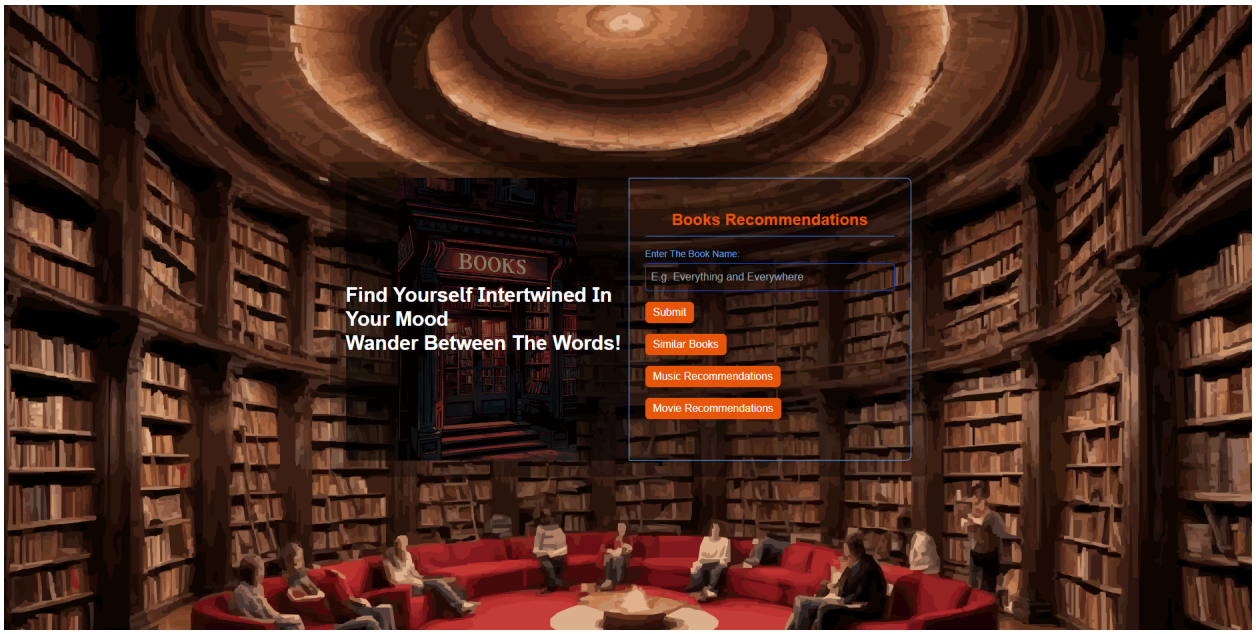

Figure 3 : Books Recommendation

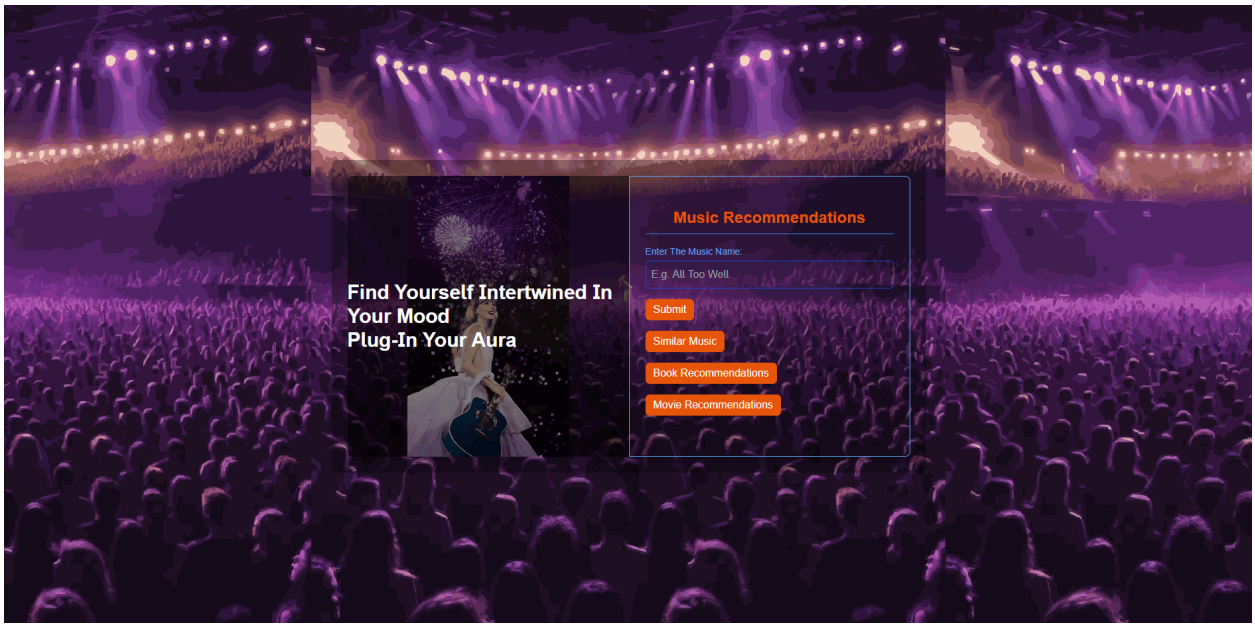

Figure 4 : Music Recommendation

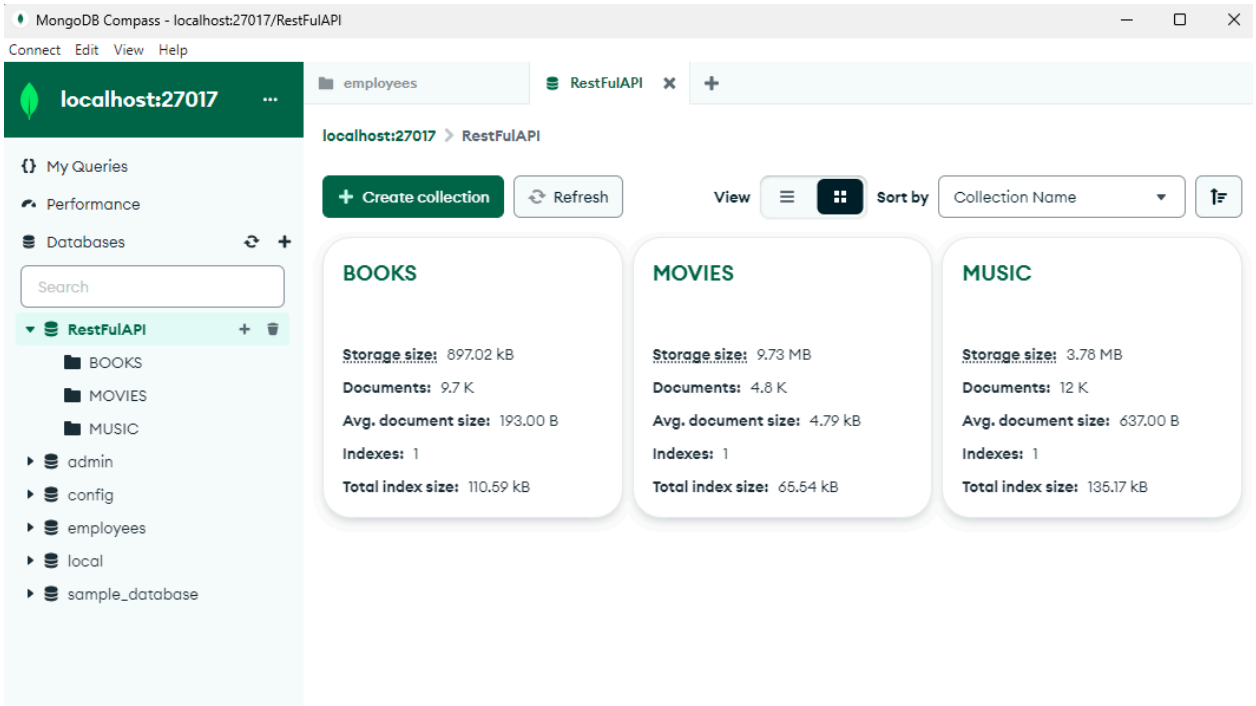

Figure 5 : Books, Movies, Music Database

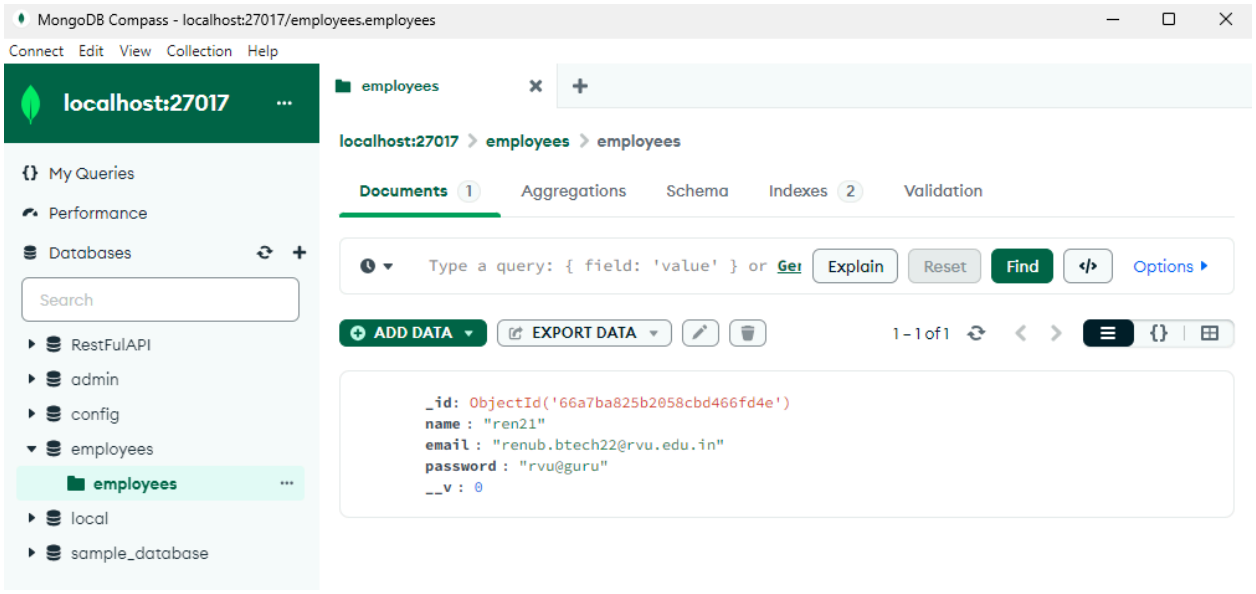

Figure 6 : Database for Login/SignUp

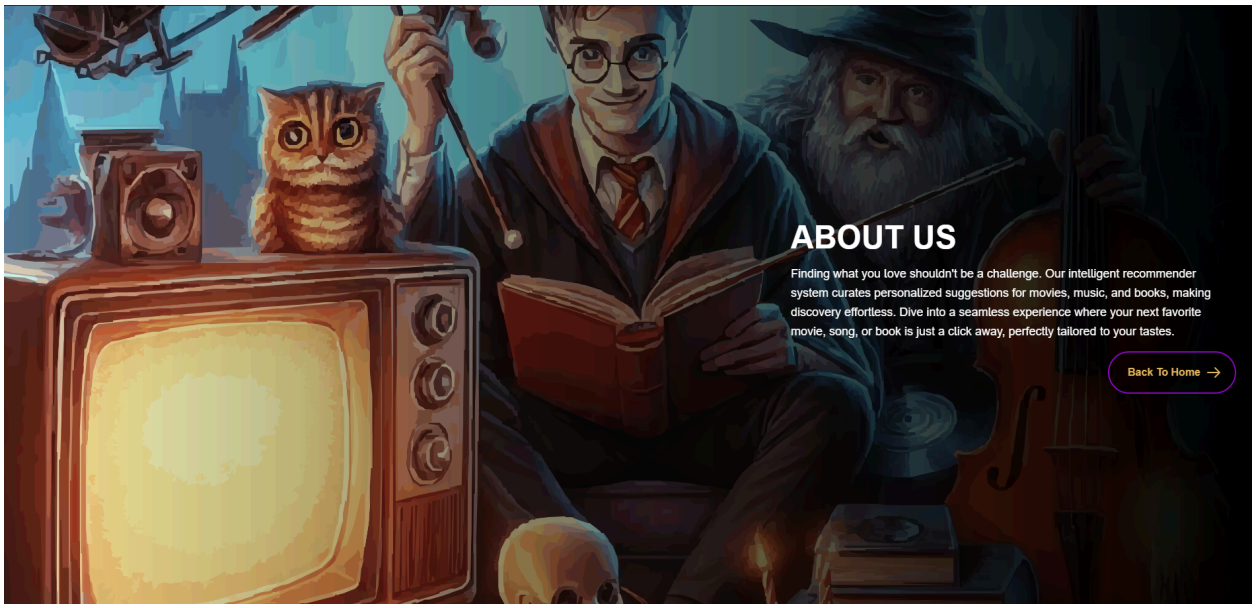

Figure 7 : About Us

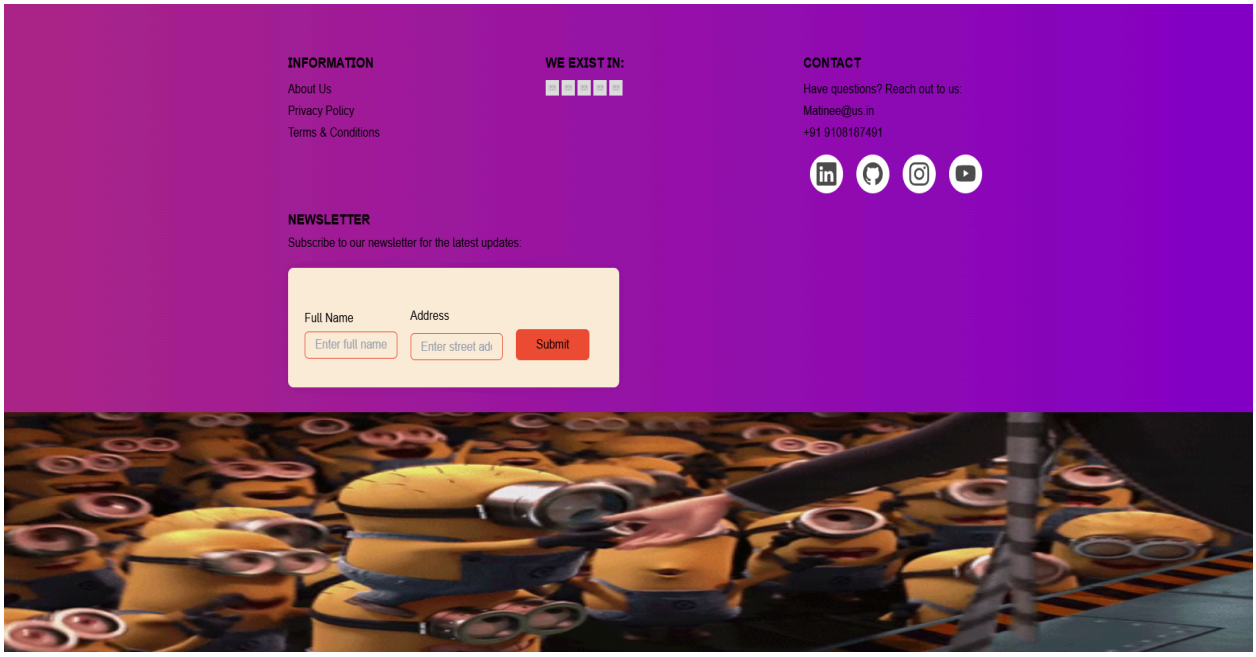

Figure 8 : Footer

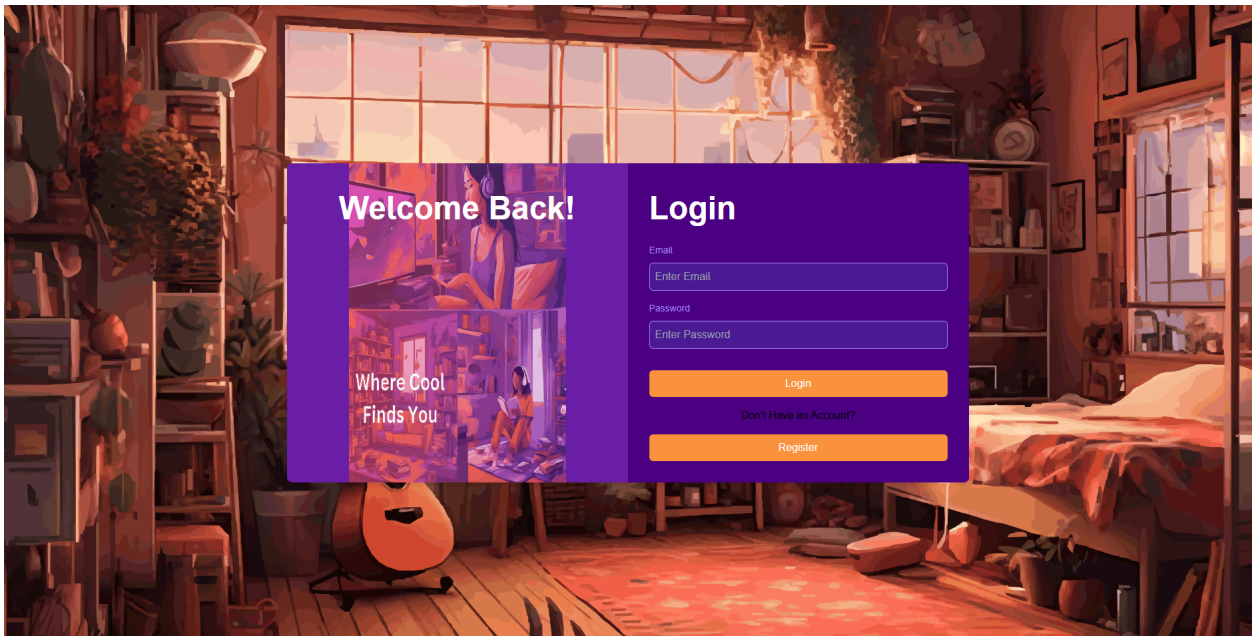

Figure 9 : Login Page

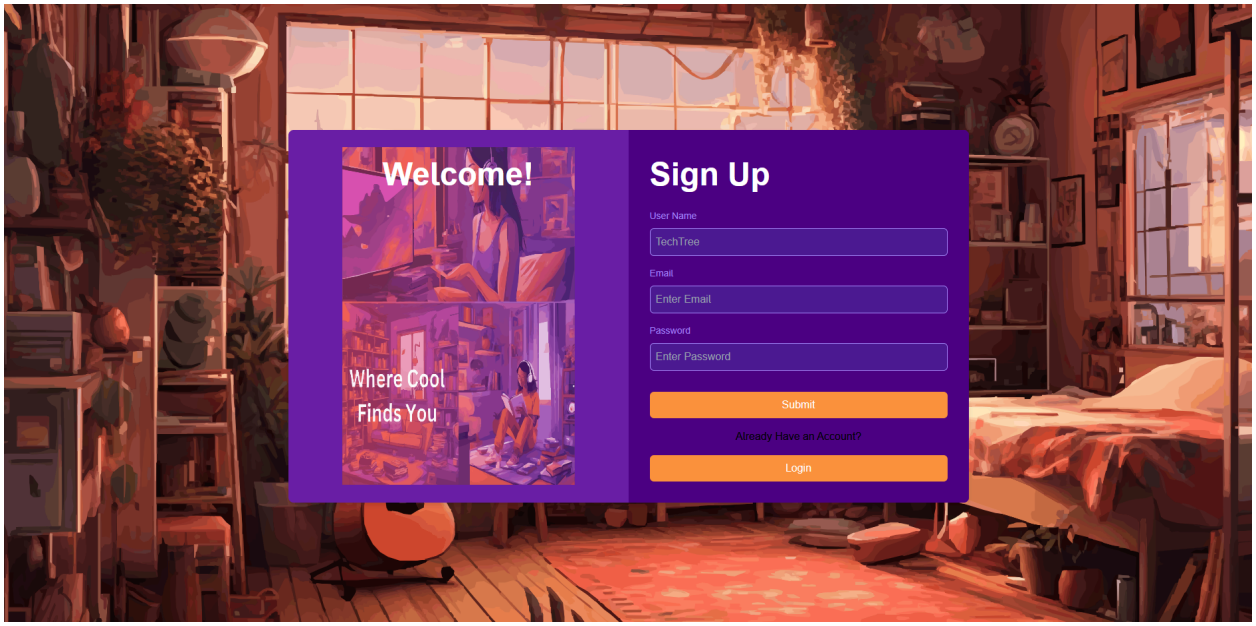

Figure 10 : Sign Up Page