IPMC TRAINING INSTITUTE, GHANA

TRAILERS HOME – A WEB APPLICATION FOR MOVIE RECOMMENDATIONS AND BOOKMARKING

Submitted in partial satisfaction of the requirements for A Professional diploma in

SOFTWARE ENGINEERING

BY

STEPHEN OKYERE

PROJECT SUPERVISOR – Mr. Abdul Basit

December

2023

ABSTRACT

Trailers Home is a web application developed by me, Stephen Okyere for the IPMC College of Technology as part of the university program requirements. The application provides movie enthusiasts with movie recommendations, bookmarking abilities, the watching of movie trailers, and allows them to leave reviews on movies they love.

The primary objective of Trailers Home is to provide a user-friendly platform for users to discover new movies, save their favorite movies, and watch trailers before deciding to watch the full movie. The application utilizes modern web development technologies, including Next.js for the front-end development, Tailwind CSS for styling, and Firebase for the backend (Authentication and Database) . The integration of the TMDB API provides access to a comprehensive movie database for movie recommendations and trailers. The use of Next.js enables server-side rendering, which improves the performance and user experience of the application.

This documentation provides a detailed overview of the Trailers Home application's features, including its user flow, technologies used, and database schema. The future improvements to the application are also outlined to enhance its functionality and user experience.

Overall, Trailers Home demonstrates my ability to develop a fully functional web application that meets the needs of its users and showcases the skills and knowledge acquired during the IPMC College of Technology university program.

PREFACE

The project presented in this documentation is an individual project developed by Stephen Okyere for the IPMC College of Technology. The project is a movie application that provides movie recommendations, bookmarking movies, and watching trailers. It was developed as part of the requirements for the university program.

The main objective of this project was to provide a user-friendly platform for movie enthusiasts to explore new movies, save their favorites, and watch trailers before deciding to watch the full movie. In addition, the project aimed to showcase the skills and knowledge acquired during the university program, including front-end and back-end development, database management, and API integration.

The development of this project involved the use of modern web development technologies, including Next.js for the front-end development, which is based on HTML, CSS, and JavaScript. Tailwind CSS was also utilized for the styling of the application. The back-end of the application was developed using Firebase, which provided a scalable and reliable infrastructure for the project.

The project also involved the integration of the TMDB API, which provided access to a comprehensive movie database for the movie recommendations and trailers. The use of Next.js enabled server-side rendering, which improved the performance and user experience of the application.

This documentation provides a detailed overview of the project, including its features, technologies used, user flow, and database schema. In addition, it outlines the future improvements that could be made to the project to enhance its functionality and user experience.

I believe that this project meets the requirements of the IPMC College of Technology university program, as it demonstrates my ability to develop a fully functional web application that meets the needs of its users. I hope that this project serves as a testament to the skills and knowledge acquired during the university program and showcases my potential as a future developer in the field of web development.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to several individuals who contributed to the success of this project. First and foremost, I am deeply thankful to my mother for her unwavering support and encouragement throughout my academic journey.

I am also immensely grateful to my mentor, Sir Langford Quarshie, whose invaluable guidance and support during the development of this project were instrumental in overcoming various challenges.

I extend my heartfelt appreciation to my brother, Maxwell Okyere, for always being there to listen to my tech rants throughout this project, even though he isn't a tech person. His attentive listening and occasional insights were invaluable.

I would also like to acknowledge the contributions of my fellow students who provided me with valuable feedback and suggestions that significantly improved the quality of this project. Their constructive feedback and insightful perspectives played a crucial role in shaping the final product.

Lastly, I wish to extend my appreciation to the IPMC College of Technology for their support and resources throughout this project. Their dedication to providing high quality education and valuable resources has been a guiding force that enabled the successful completion of this project I am grateful for the privilege of being a part of their esteemed learning community.

TABLE OF CONTENTS

ABSTRACT

PREFACE

ACKNOWLEDGMENTS

1. INTRODUCTION

1.1 Background

1.2 Approach

1.3 Research Questions

1.4 Proposed System

1.5 Features and Functionalities

2. LEGAL, SOCIAL, ETHICAL AND PROFESSIONAL ISSUES AND CONSIDERATIONS

3. REQUIREMENTS

3.1 Comparison of Systems

3.2 Functional Requirements

3.3 Non - Functional Requirements

4. DESIGN

4.1 Database Design

4.1.1 Database Collections

4.1.2 Database Flow

4.2 UI Design

5. PROTOTYPE

LIST OF FIGURES

Figure 1 Users Collection View *(Firebase)*

Figure 2 User Bookmarks Collection *(Firebase)*

Figure 3 Saved Bookmarks Collection *(Firebase)*

Figure 4 User Review Collection View and Data Structure (Firebase)

Figure 5 Authenticated User Data Example *(Firebase)*

Figure 6 Example Firebase Query to Fetch user bookmarks *(Using Query Builder)*

Figure 7 Screenshot Of The Homepage Layout

Figure 8 Screenshot Of The Movies List Page Layout

Figure 9 Screenshot Of The Movie Details Page Layout

Figure 10 Visual Representation Of Color Scheme

Figure 11 Wireframe Of Homepage Layout

Figure 12 Wireframe Of Movie Details Page Layout

Figure13 Wireframe Of Movie Lists Page Layout

Figure 14 Wireframe Of Login Modal

Figure 15 Wireframe Of Signup Modal

Figure 16 Wireframe of Reset Password Modal

INTRODUCTION

BACKGROUND:

In today's digital age, the way people consume entertainment has undergone a significant transformation. With the advent of online streaming platforms, movie enthusiasts are now seeking convenient and user-friendly ways to access information about their favorite movies. This shift in consumer behavior has led to an increased demand for comprehensive movie databases that provide detailed insights into movie titles, trailers, reviews, and ratings.

Trailers Home was conceptualized as an individual project with the vision of creating a web application that caters to the needs of movie enthusiasts who seek a one-stop platform for movie information. The inspiration for this project arose from the realization that existing movie streaming applications often focus solely on providing access to movies, without prioritizing the need for detailed movie data and a seamless user experience.

Being a big fan of movies and a web developer, I recognized the opportunity to create an application that would cater to the specific requirements of movie enthusiasts. Trailers Home was designed to be more than just another movie information platform. It aims to set itself apart by offering a comprehensive and user-friendly experience that goes beyond simple movie viewing.

The primary goal of Trailers Home is to provide users with an extensive movie database that not only includes a vast collection of movies from various genres but also offers in-depth information about each movie. By leveraging the power of the TMDB API, the application can access an up-to-date repository of movie data, including detailed movie synopses, release dates, ratings, and trailers.

Moreover, Trailers Home aims to be a platform where users can not only browse and watch movie trailers but also engage with the content and share their thoughts and opinions. The integration of user reviews allows movie enthusiasts to express their views, share recommendations, and contribute to a vibrant community of film lovers.

With an intuitive user interface built using NextJS and Tailwind CSS, Trailers Home seeks to provide a seamless and visually appealing experience for users. File-based routing and search engine optimization were chosen to ensure easy navigation and improved discoverability of movies.

Overall, Trailers Home is a project driven by passion, aiming to offer users a comprehensive and user-friendly platform to explore, engage, and enjoy detailed movie information, including trailers and more.

APPROACH:

In developing Trailers Home, my approach was centered around simplicity, practicality, and constant iteration. Rather than being confined to rigid methodologies, I focused on adopting a flexible and agile approach. The initial phase involved extensive research to understand user needs and preferences in the movie-streaming domain.

With a clear vision in mind, I embarked on designing the user interface with NextJS and Tailwind CSS to ensure a minimalistic yet visually appealing design. The aim was to create an intuitive experience that allows users to easily navigate and discover new movies.

For functionality, I leveraged JavaScript and the context API for state management, ensuring a seamless and efficient interaction with the TMDB API for movie data retrieval. Firebase was chosen for user authentication and database management, providing a secure and reliable solution.

Throughout the development process, I regularly sought feedback from users and incorporated their insights to refine the application. This iterative approach allowed me to address potential issues early on and deliver a more user-focused product.

Overall, the emphasis was on creating a platform that is user-centric, functional, and engaging, offering movie enthusiasts a delightful experience in exploring and enjoying their favorite movies to the fullest.

LITERATURE REVIEW

RESEARCH QUESTIONS

We also examined frequently asked questions related to building movie apps, such as "What features should a movie app have?" and "How can a movie app improve user engagement?" These questions guided our approach to developing Trailers Home and informed our research questions, which include:

- How can Trailers Home optimize its platform to deliver an extensive and user-intuitive experience for movie enthusiasts to explore and discover new movies?
- How can Trailers Home differentiate itself from other online movie service platforms and expand its user base?
- What are the most secure and user-friendly authentication methods to implement for user registration and login?
- How can we ensure user account data is securely stored and protected?
- What is the most efficient way to implement a bookmarking system for users to save their favorite movies?
- What features and functionalities can be added to Trailers Home to improve the user experience and increase user engagement?
- How can we provide a seamless experience for users to manage and organize their bookmarks?
- What APIs or services can be used to fetch and display movie trailers within the app?
- What criteria and algorithms can be used to determine related movies for a specific movie selection?
- How can we balance performance and user experience when displaying large datasets?
- What features and tools can be provided to users to write, submit and manage movie reviews easily?
- How can we retrieve and display movie details efficiently to reduce loading times?

PROPOSED SYSTEM:

Trailers Home is a dynamic web application that caters to movie enthusiasts, offering real-time and comprehensive movie information, saving them valuable time and expenses. The application features

various browsing options such as Popular, Top-rated, Upcoming, and allows users to perform targeted searches for specific movies. Users can easily access movie posters, detailed synopses, trailers, and explore additional features like user reviews and ratings. Trailers Home is skillfully crafted using cutting-edge technologies, including NextJS for optimized performance and seamless user experiences, TailwindCSS for a modern and elegant interface, and Firebase for secure and efficient user authentication, database management, and real-time updates. With the integration of TMDB API, the app brings a vast and diverse collection of movie data to the user's fingertips. Continuously evolving, Trailers Home promises to deliver a delightful movie discovery journey.

FEATURES AND FUNCTIONALITY:

- User authentication and account management
- Save favorite movies to bookmarks
- Watch movie trailers
- Show related movies based on the selected movie
- Implement pagination or infinite scrolling for browsing movies
- Allow users to rate and review movies
- Enable users to add movie reviews
- Allow users to delete their movie reviews
- Share movies with others through social media platforms
- Search for movies by title
- Navigate between different tabs and view movies based on different categories like Popular, Top rated, and Upcoming
- Click on a movie poster to view the movie details, including Overview, Release Date, Rating, and Synopsis.

In conclusion, Trailers Home is an attempt to create a useful and user-friendly web application that provides movie enthusiasts with access to comprehensive movie-related information. The application is

developed using the latest technologies and is designed to meet the needs of users who prefer to watch movies at home.

LEGAL, SOCIAL, ETHICAL AND PROFESSIONAL ISSUES AND CONSIDERATIONS

In the development and deployment of the Trailers Home web application, several legal, social, ethical, and professional issues and considerations were taken into account. The following points highlight some of the key aspects related to these areas:

1. Data Privacy and Protection: The application adheres to relevant data privacy laws and regulations to ensure the protection of user data. Measures such as secure authentication, encryption of sensitive information, and data anonymization were implemented to maintain user privacy.
2. Copyright and Intellectual Property: The project acknowledges and respects copyright and intellectual property rights. The integration of the TMDB API ensures that movie data and trailers are used within the bounds of their respective licenses and terms of use.
3. User Consent and Permissions: The application obtains appropriate user consent for collecting and storing personal data. Users have control over their data and can choose to opt out or delete their accounts if desired.
4. Accessibility and Inclusivity: The application strives to be accessible to users with disabilities by following accessibility guidelines and standards. It ensures that the interface is perceivable, operable, and understandable to a wide range of users.
5. Ethical Use of Data: User data is handled responsibly and ethically, ensuring that it is not misused or shared without consent. Transparent privacy policies and terms of service are provided to users to maintain trust and transparency.
6. Cybersecurity and Data Breach Prevention: Appropriate security measures were implemented to safeguard user data against unauthorized access, data breaches, and cyber threats. Regular security audits and updates help ensure the application's resilience against potential vulnerabilities.
7. Ethical and Professional Standards: Throughout the development of this project, the utmost care was taken to uphold ethical and professional standards. It reflects a commitment to integrity, honesty, and accountability in every aspect, including code quality, documentation, and interactions with users.

It is important to note that this list is not exhaustive, and additional legal, social, ethical, and professional considerations may arise depending on the specific context and jurisdiction. Throughout the project, continuous monitoring and evaluation of these issues were undertaken to ensure compliance and responsible use of the application.

REQUIREMENTS

COMPARISON OF SYSTEMS

There are several applications available in the market that provide movie information, reviews, ratings, and personalized recommendations. For instance, GoodShow is an iOS app that helps users discover movies and TV shows based on their friend's reviews and enables them to add movies to their watchlist. However, the app's limitation is that it is only available on iOS.

Similarly, apps like Netflix and Hulu require users to subscribe and login to access their personalized content and recommendations. This subscription requirement limits users who are only interested in searching for movie titles and getting a brief overview of the movie. Furthermore, some applications provide movie reviews, while others provide release and rating information separately. For example, Guidebox provides a list of sources where movies are available.

In contrast, Trailers Home is a free web application that aims to provide a user-friendly platform for movie enthusiasts to discover new movies, save their favorite movies, and watch trailers before deciding to watch the full movie. The integration of the TMDB API provides access to a comprehensive movie database for movie recommendations and trailers. The user-friendly interface and bookmarking feature allow users to save their favorite movies and easily access them later. Additionally, the app provides a brief overview of the movie and its trailer, eliminating the need to subscribe to a streaming service or navigate multiple apps to find the necessary information.

Trailers Home distinguishes itself by offering a user-friendly platform, free access, comprehensive movie information, and a focus on trailers, providing movie enthusiasts with a unique and convenient experience.

FUNCTIONAL REQUIREMENTS:

Based on the Trailers Home web application's purpose and objectives, the following are the functional requirements:

1. User Registration and Authentication:

- ❿ Users should be able to create an account and provide necessary registration details.
- ❿ The application should authenticate users' credentials securely.
- ❿ Users should be able to reset their password when they forget

2. Movie Recommendations:
    - ❿ The application should provide personalized movie recommendations based on user preferences, ratings, and viewing history.
    - ❿ Users should be able to browse recommended movies by different categories such as popular, top-rated, and upcoming.

3. Movie Search and Filtering:
    - ❿ Users should be able to search for specific movies by title.
    - ❿ The application should allow users to apply filters to narrow down search results based on various criteria.

4. User Reviews:
    - ❿ Users should have the ability to write and submit reviews for movies they have watched.
    - ❿ The application should provide a form or interface where users can input their review text.
    - ❿ Users can rate movies using a star rating system or provide a numerical rating.
    - ❿ Reviews should include the reviewer's name or username for identification.
    - ❿ The application should display a list of user reviews for each movie, including the review text, rating, and reviewer information.
    - ❿ Users should have the option to edit or delete their own reviews if they wish to make changes or remove their feedback.
5. Movie Details:
    - ❿ Users should be able to view detailed information about each movie, including a synopsis, release date, rating, and duration.
    - ❿ Trailers Home should display the movie's poster, cast, crew, and additional relevant details.

6. Movie Bookmarking:

- Users should have the ability to bookmark movies they are interested in or want to watch later.
- The application should provide a bookmarking feature that allows users to create a list of their favorite movies for easy access.

7. Movie Trailers:
   - Users should be able to watch movie trailers directly within the application.
   - Trailers Home should seamlessly integrate with the TMDB API to fetch and display movie trailers for users to preview.

8. User Profile Management:
   - Users should be able to manage their profiles, including updating personal information and preferences.
   - The application should provide options for users to edit their profiles, change passwords, and manage their bookmarked movies.
   - Users should be delete their account entirely from the app

9. Responsive Design:
   - The application should be responsive and accessible across different devices, including desktops, tablets, and mobile phones.
   - The user interface should adapt to different screen sizes and resolutions for optimal user experience.

10. Error Handling and Feedback:
   - The application should provide meaningful error messages and notifications to users in case of errors or invalid inputs.
   - Clear feedback should be provided to users on successful actions, such as successful registration or bookmarking a movie.

11. Performance and Scalability:
   - The application should be optimized for performance, ensuring fast loading times and smooth navigation.

⑩ It should be able to handle a growing number of users and movie data without significant degradation in performance.

It's important to note that these are just initial functional requirements, and further analysis and discussions may refine and expand this list during the development process.

NON-FUNCTIONAL REQUIREMENTS:

In addition to the functional requirements, the Trailers Home web application also has non-functional requirements that define the quality attributes and constraints of the system. These non-functional requirements include:

1. Performance:

- The application should have fast response times, ensuring quick loading of movie information, recommendations, and trailers.
- The system should be able to handle multiple concurrent users without significant performance degradation.
- Response times for search queries and loading movie details should be within acceptable limits.

2. Security:

- User data, including personal information and login credentials, should be securely stored and transmitted.
- The application should implement proper authentication and authorization mechanisms to ensure data privacy and prevent unauthorized access.

3. Usability and User Experience:

- The user interface should be intuitive, visually appealing, and easy to navigate.
- The application should provide clear instructions and guidance to users on how to interact with its features.

4. Compatibility:

- The application should be compatible with popular web browsers, ensuring consistent functionality and appearance across different browsers and versions.

- It should be responsive and adaptable to various screen sizes, providing a seamless user experience on different devices.

5. Reliability and Availability:

- The system should be reliable, minimizing downtime and ensuring high availability to users.

- Proper error handling and graceful degradation should be implemented to handle exceptional situations and maintain system stability.

6. Scalability:

- The application should be designed to handle increasing numbers of users and growing movie data.

7. Compliance:

- The application should adhere to relevant laws and regulations regarding data privacy, intellectual property rights, and user consent.

- It should comply with ethical and professional standards, as outlined in the Legal, Social, Ethical, and Professional Issues section.

These non-functional requirements are crucial for ensuring that the Trailers Home web application not only meets the functional needs but also delivers a secure, performant, user-friendly, and compliant user experience.

DESIGN

DATABASE DESIGN:

Firebase is a cloud-based platform that offers a variety of backend services for web and mobile applications. It was chosen for this project due to its ease of use, real-time data synchronization, built-in authentication, and scalable database capabilities, providing a robust foundation for Trailers Home.

DATABASE COLLECTIONS

1. Users Collection:

- The "users" collection will store user data, including user profiles and authentication information.
- Each user will have a unique identifier (UID) automatically generated by Firebase upon successful signup.
- User data can include fields such as name, email, username, and any additional profile information.
- The document structure will include the following fields:

  ➓ UID: [automatically generated by Firebase]

  ➓ name: [user's name]

  ➓ email: [user's email]

  ➓ imageUrl: [URL to user's profile picture]

  ➓ dateRegistered: [any additional profile information]

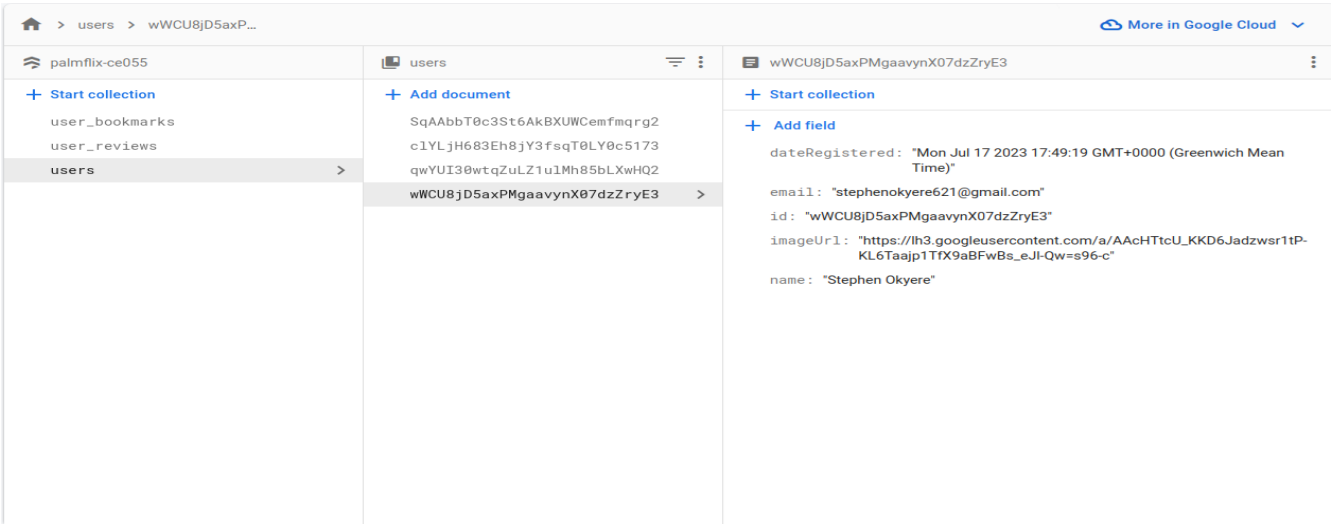


Fig 1 Users Collection View and Data structure (Firebase)

2. Users Bookmarks Collection:

- The "users_bookmarks" collection will store user bookmarks and their corresponding saved movies.
- Each document in the "users_bookmarks" collection will have the unique user ID (from the "users" collection) as its document ID.
- Under each document, a subcollection named "saved_bookmarks" will be created to store individual movie bookmarks.

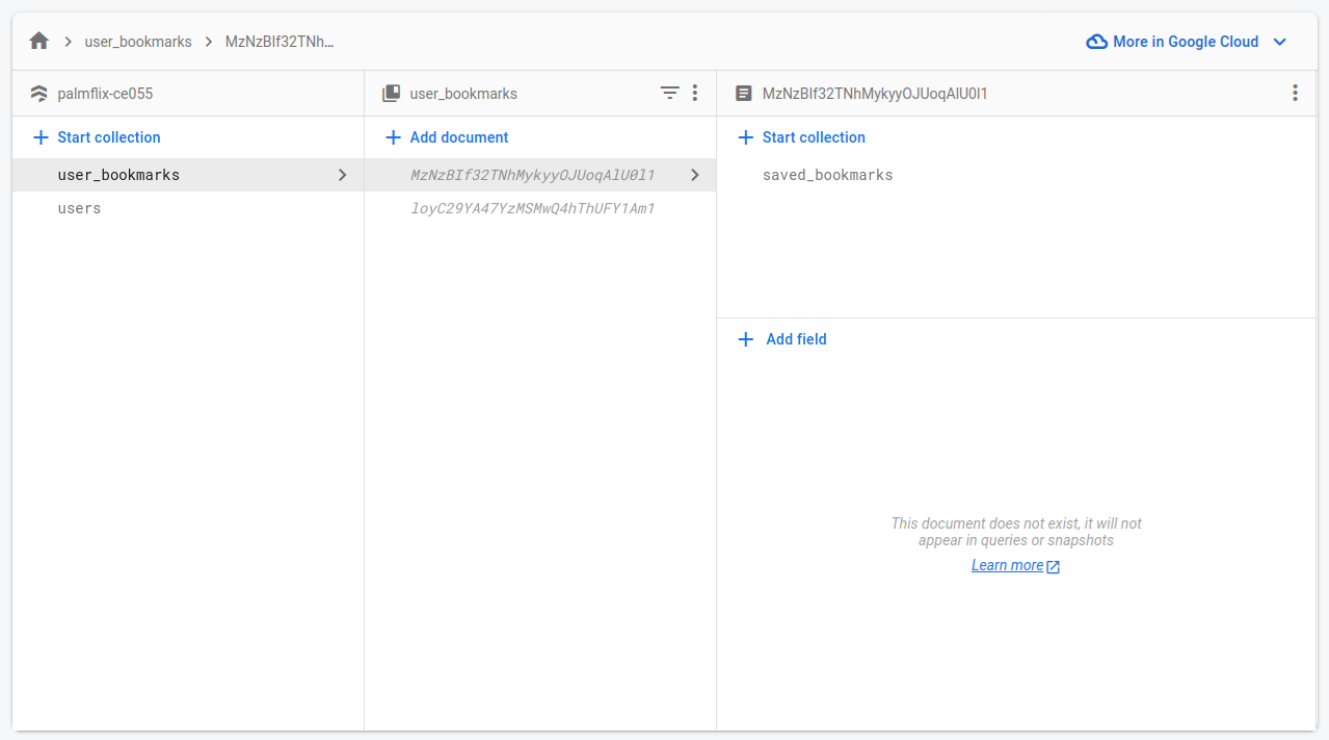


Fig 2 Users Bookmarks Collection (Firebase)

3.User Reviews Collection:

- The "user_reviews" collection will be responsible for storing user reviews and their corresponding movie details.
- Each document in the "user_reviews" collection will be identified by a unique combination of the movie ID and the Unix date at the particular time when the review was submitted.
- The document structure will include the following fields:
  - movieID: The unique identifier of the movie for which the review is submitted.
  - uid (UserID) : The unique identifier of the user who submitted the review (from the "users" collection).

- email: The verified email of the user who submitted the review
- username: The username of the user who submitted the review
- content: The actual text content of the user's review for the movie.
- rating: The numeric rating given by the user for the movie (e.g., out of 5 stars).
- Id (document ID): The unique identifier of the submitted review
- created_at: The Unix timestamp when the review was submitted, which allows for chronological sorting.

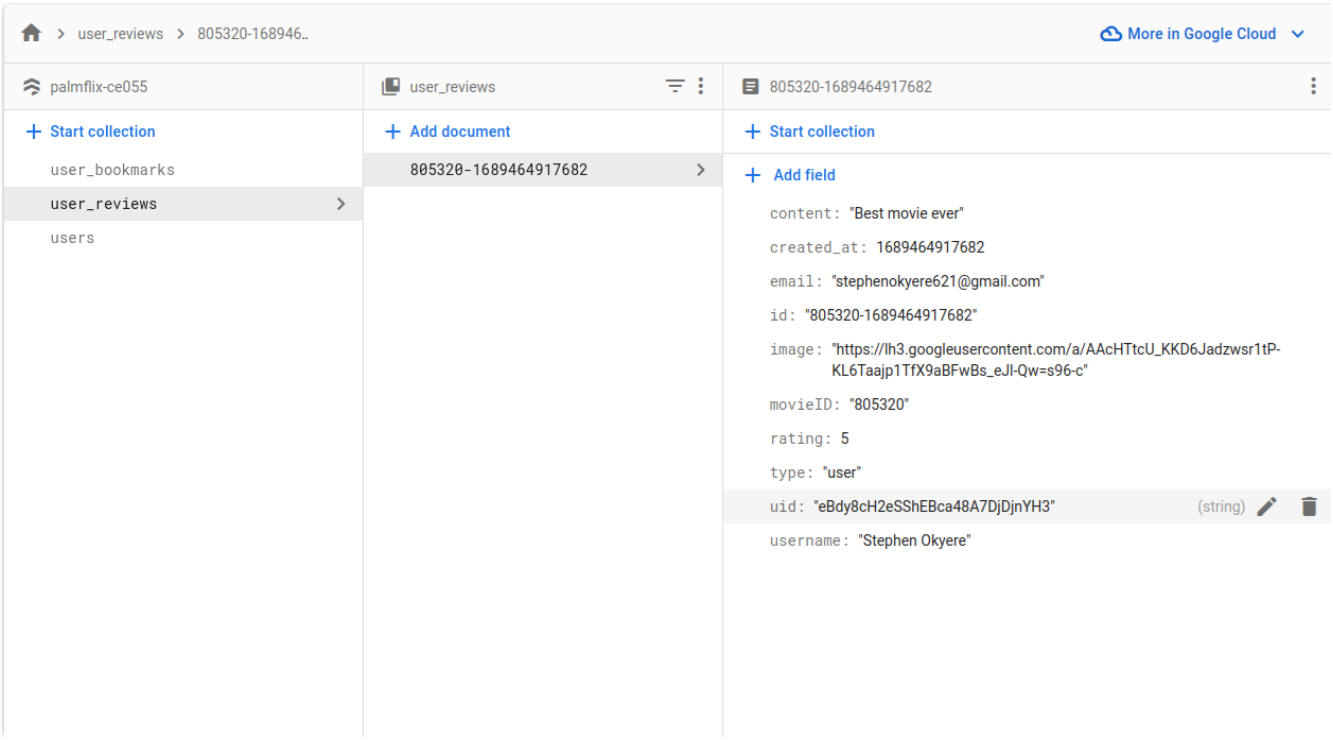


Fig 3 User Review Collection View and Data Structure (Firebase)

4. Saved_Bookmarks Subcollection:

- The "saved_bookmarks" subcollection will contain documents representing individual movie bookmarks for each user.

- Each document representing an individual movie bookmark uses the movie ID fetched from the API as its unique identifier

- Within each document, relevant movie information can be stored, such as movie ID, title, genre, release date, and any additional details needed for bookmarked movies.

- The document structure will include the following fields:

  - movieID: [movie ID fetched from the API]

- uid: The unique identifier of the user whose bookmark is being stored
- title: [movie title]
- genre: [movie genre]
- description: [any additional details needed for bookmarked movies]

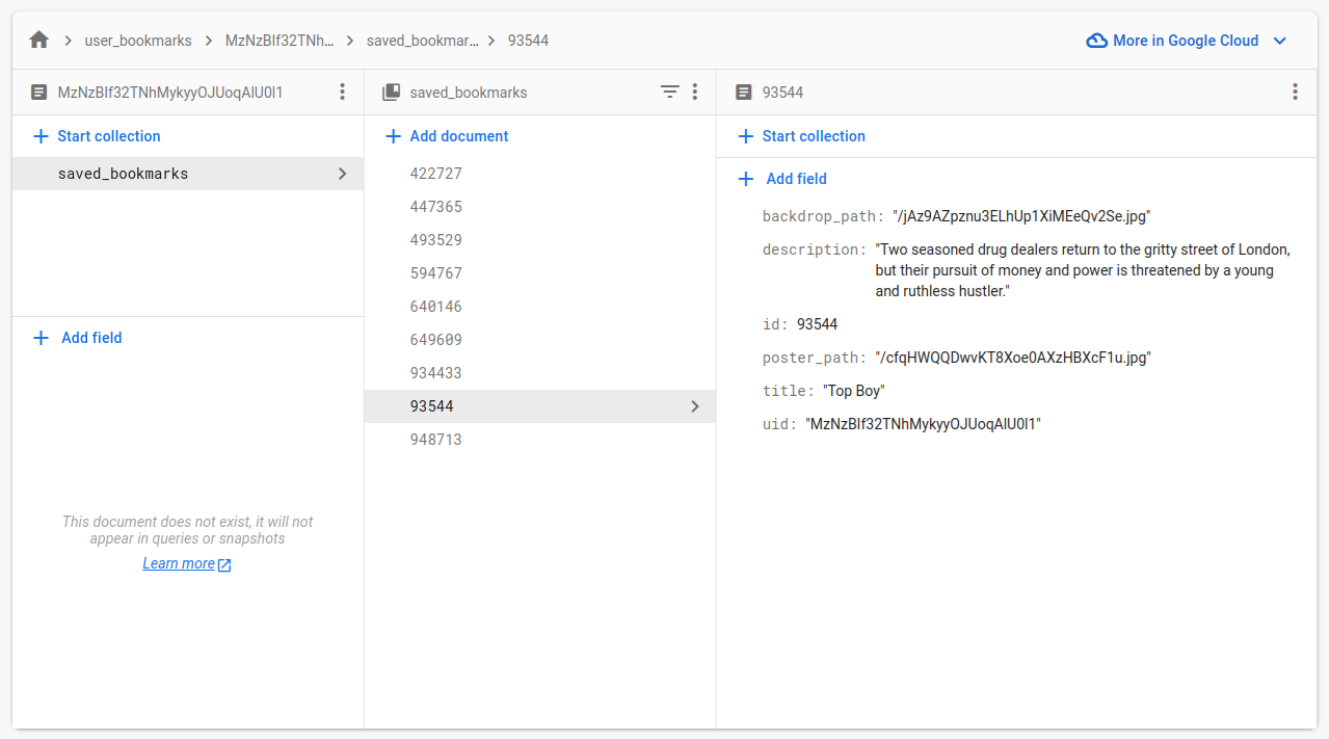


Fig 4 Saved Bookmarks Collection View (Firebase)

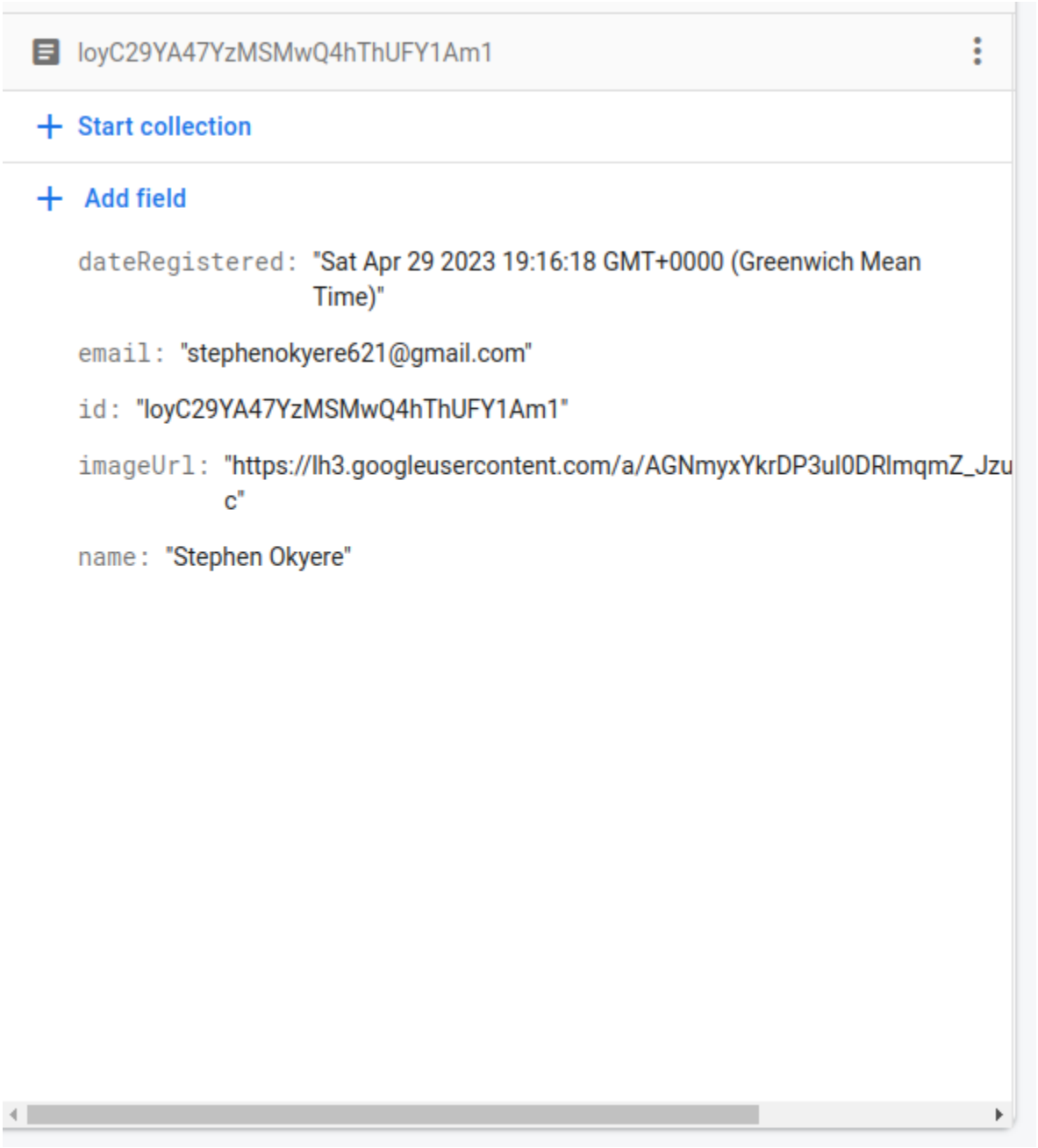
DATABASE FLOW

1. User Signup:

- When a user successfully signs up, their user profile information is stored as a new document in the "users" collection.

- The user is assigned a unique user ID (UID) generated by Firebase.

Fig *5*



Authenticated User Data Example (Firebase)

2. Bookmark Creation:

- When a user bookmarks a movie, a new document is created in the "saved_bookmarks" subcollection under their corresponding user document in the "users_bookmarks" collection.
- The document ID for the bookmarked movie can be automatically generated by Firestore.

3. Bookmark Retrieval:

- To retrieve a user's bookmarks, the application queries the "users_bookmarks" collection for the user's document using their unique user ID.

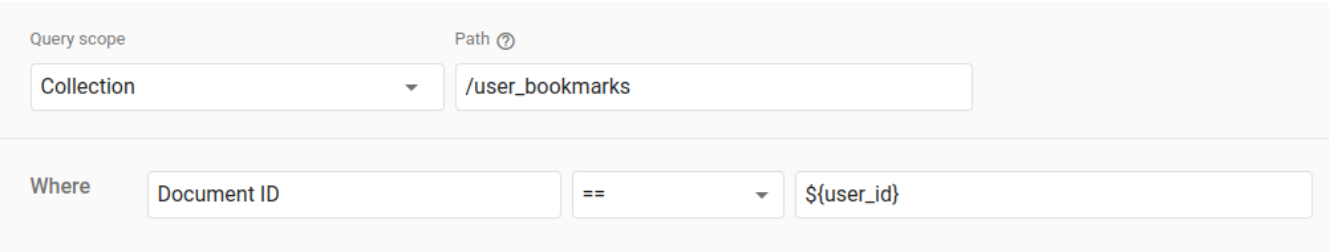


Fig 6 Example Firebase Query to Fetch user bookmarks(Using Query Builder)

- Within the user's document, the "saved_bookmarks" subcollection is accessed to retrieve the bookmarked movies.

4. Bookmark Removal:

- When a user removes a bookmark, the corresponding document within the "saved_bookmarks" subcollection is deleted.

5. Review Creation:

- When a user submits a review for a movie, a new document is created in the "user_reviews" collection with a unique document ID.
- The document ID for the review can be generated based on the movieID and the Unix timestamp when the review was submitted.

6. Review Retrieval:

- To retrieve user reviews for a specific movie, the application queries the "user_reviews" collection using the movieID as a filter.

7. Review Removal:

When a user deletes their review for a movie, the corresponding document in the "user_reviews" collection is deleted based on the movieID and the Unix timestamp of the review.

UI DESIGN

For the Trailers Home web application, a simple and minimalistic design approach was adopted to create an intuitive and visually appealing user experience. The following considerations were made for the UI design:

1. Layout: The UI design maintains consistency throughout the application. It features a clean and organized layout to enhance usability. The homepage includes a slideshow of trailers after the header section, followed by subsequent sections that display movies from various categories in a grid format.

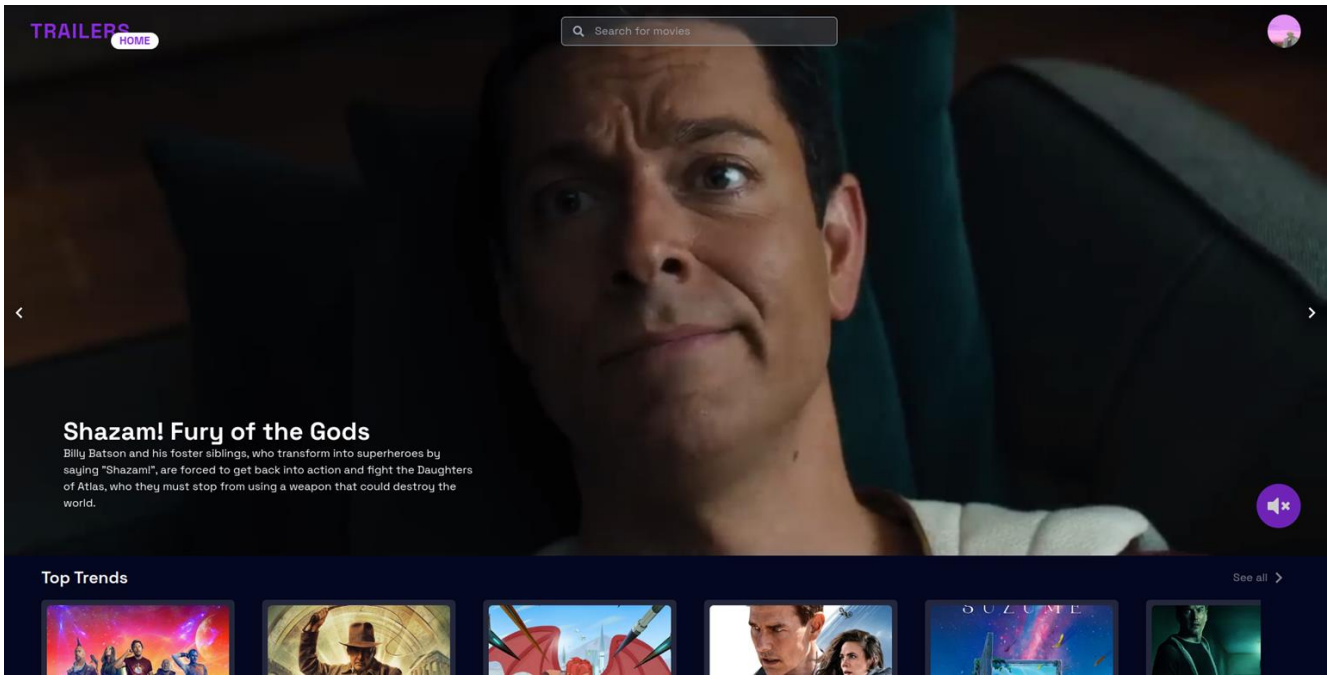


Fig 7 Screenshot Of The Homepage Layout

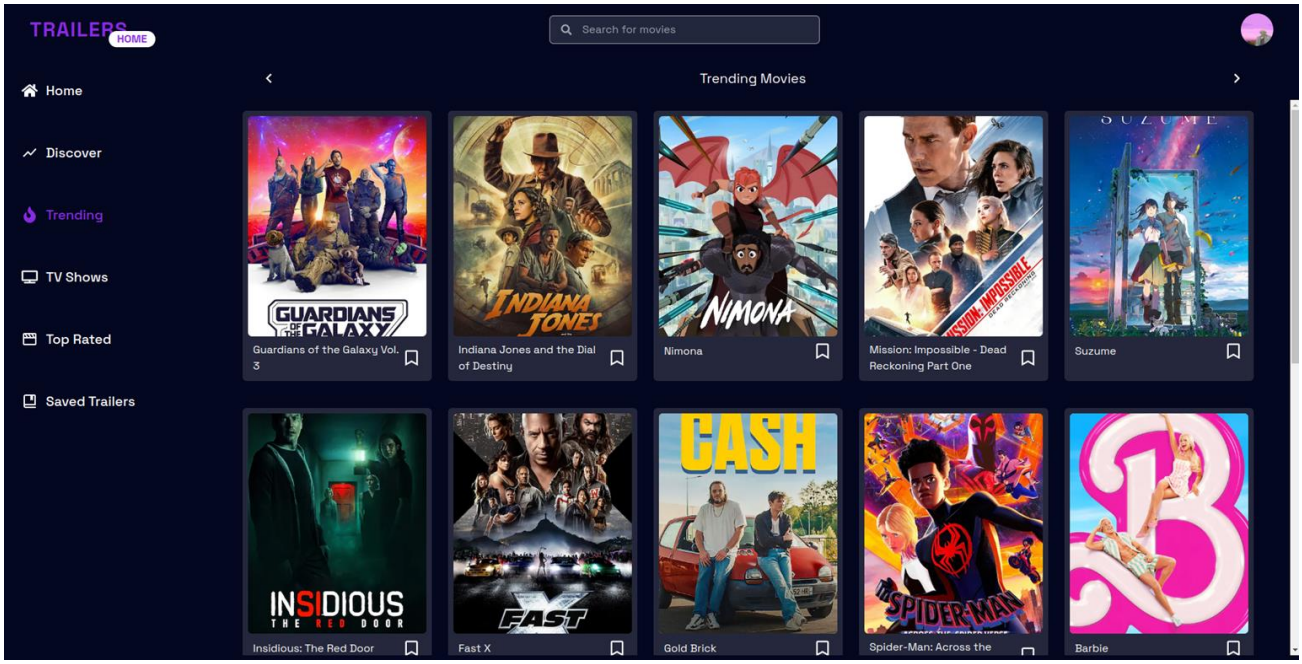


Fig 8 Screenshot Of The Movies List Page Layout

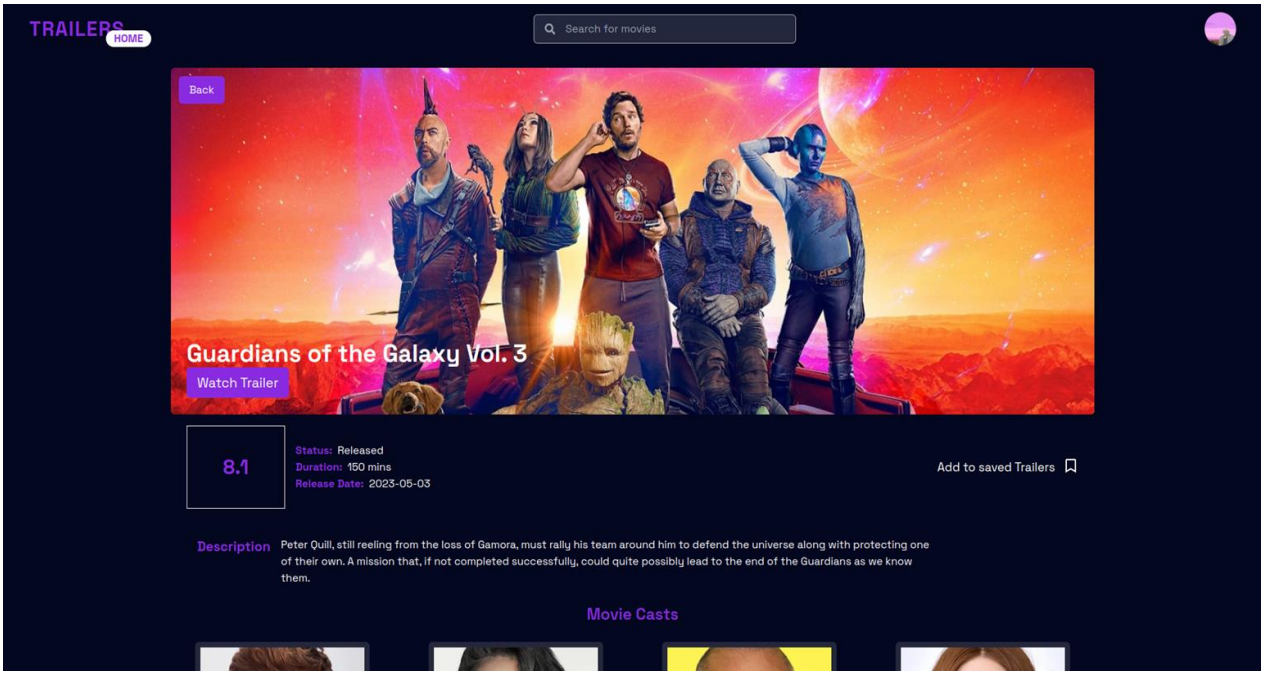


Fig 9 Screenshot Of The Movie Details Page Layout

2. Color Scheme: The chosen color scheme reflects a modern and visually appealing aesthetic. The primary color is "#8a2be2" (a shade of purple), which adds vibrancy and a touch of elegance to the UI. The background color is "#040720" (a deep blue shade), creating a visually immersive

experience. White text is used for optimal contrast against the dark background, ensuring readability.

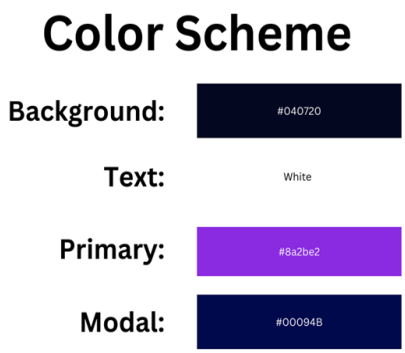


Fig 10 Visual Representation Of Color Scheme

3. Typography: Clear and legible typography is implemented across the application. Sans-serif fonts with appropriate font weights are used for headings, titles, and body text. The font choices contribute to a clean and modern look.
4. Pop-up Modals: Pop-up modals are utilized for the login and signup processes, providing a streamlined and focused user experience. They allow users to enter their credentials or create a new account without navigating to separate pages.

PROTOTYPE

Creating the prototype of the Trailers Home web application involved a systematic and iterative approach to design and development. The prototype aimed to provide a visual representation of the application's user interface and interactions. Here is an overview of the approach taken:

1. Design Ideation: The prototype process began with brainstorming to define the core features and functionalities of the application. This involved considering user requirements, industry trends, and competitor analysis to identify unique selling points and areas for innovation.

2. Sketches and Wireframes: Wireframes were created to visualize the layout, structure, and flow of the application. These wireframes served as a blueprint for the overall user interface design, helping to refine the information architecture and user experience.

3. Visual Design: Once the wireframes were finalized, attention was given to the visual design of the application. A minimalist and modern design approach was chosen, aligning with the desired user experience. The color scheme, typography, and visual elements were carefully selected to create a visually appealing and cohesive interface.

4. Feedback and Iteration: The prototype was shared with stakeholders, including potential users, for feedback and evaluation. Their input was crucial in identifying usability issues, areas for improvement, and potential enhancements. Iterative design cycles were undertaken to refine the prototype based on this feedback.

5. Refinement and Finalization: Based on the feedback and usability testing results, the prototype underwent further refinements and optimizations. Adjustments were made to enhance the user experience, streamline the navigation flow, and improve overall usability.

Throughout the prototype creation process, a user-centric approach was followed, prioritizing intuitive interactions, clear information presentation, and a visually appealing design. The iterative nature of the process allowed for continuous improvement and ensured that the final prototype reflected the envisioned user experience.

WIREFRAMES

HOMEPAGE:

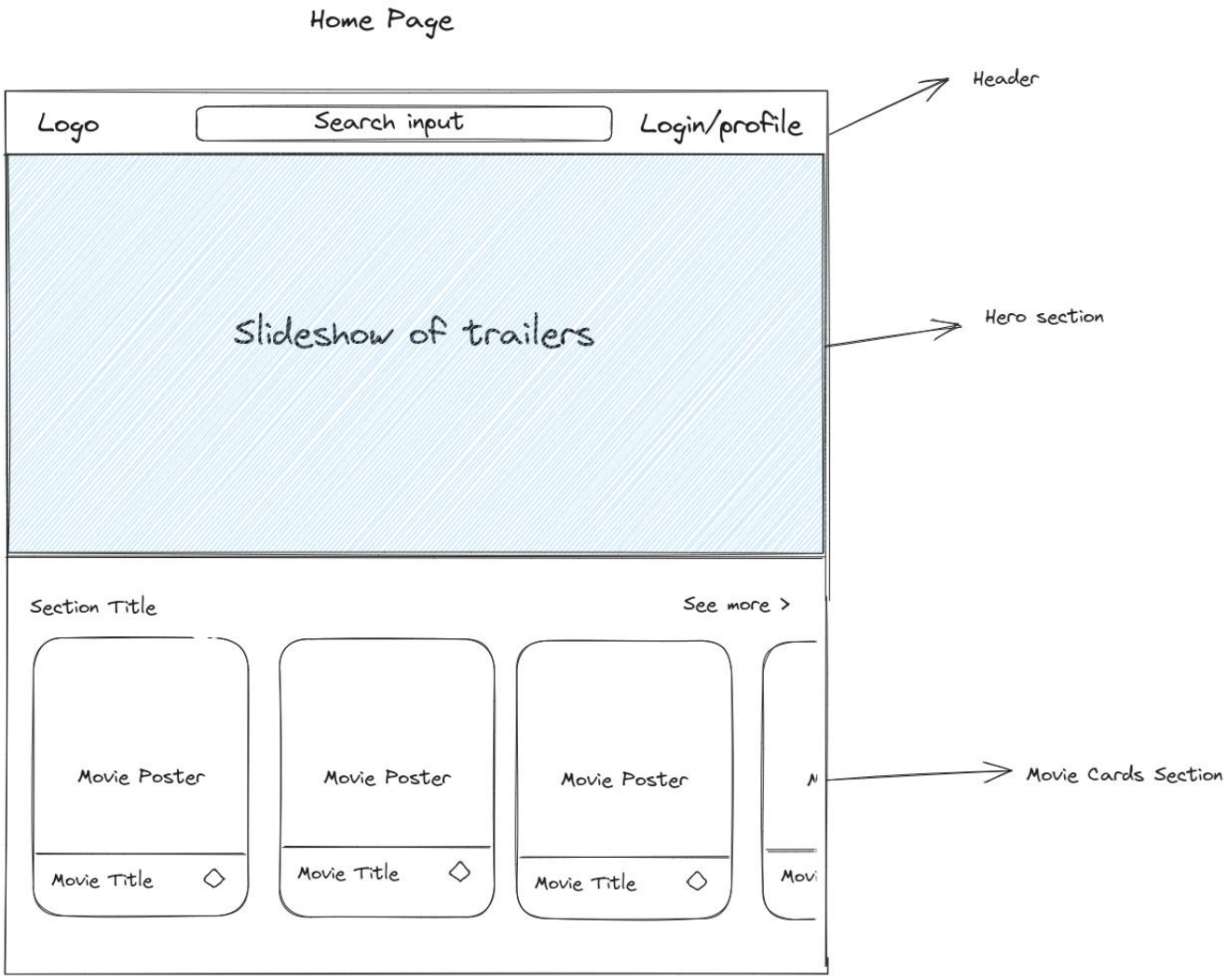


Fig 11 Wireframe Of Homepage Layout

MOVIE DETAILS PAGE:

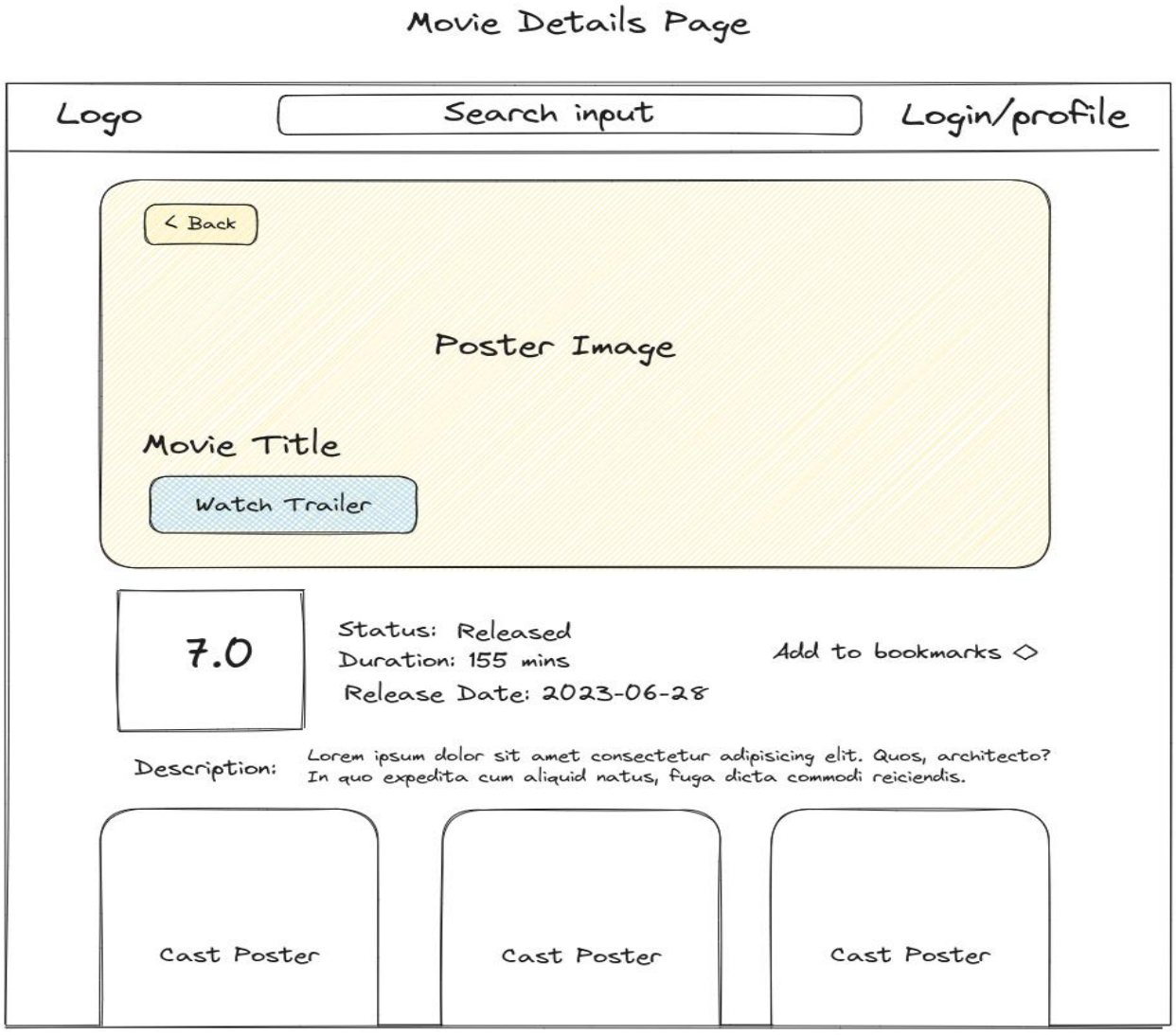


Fig 12 Wireframe Of Movie Details Page Layout

MOVIE LISTS PAGE:

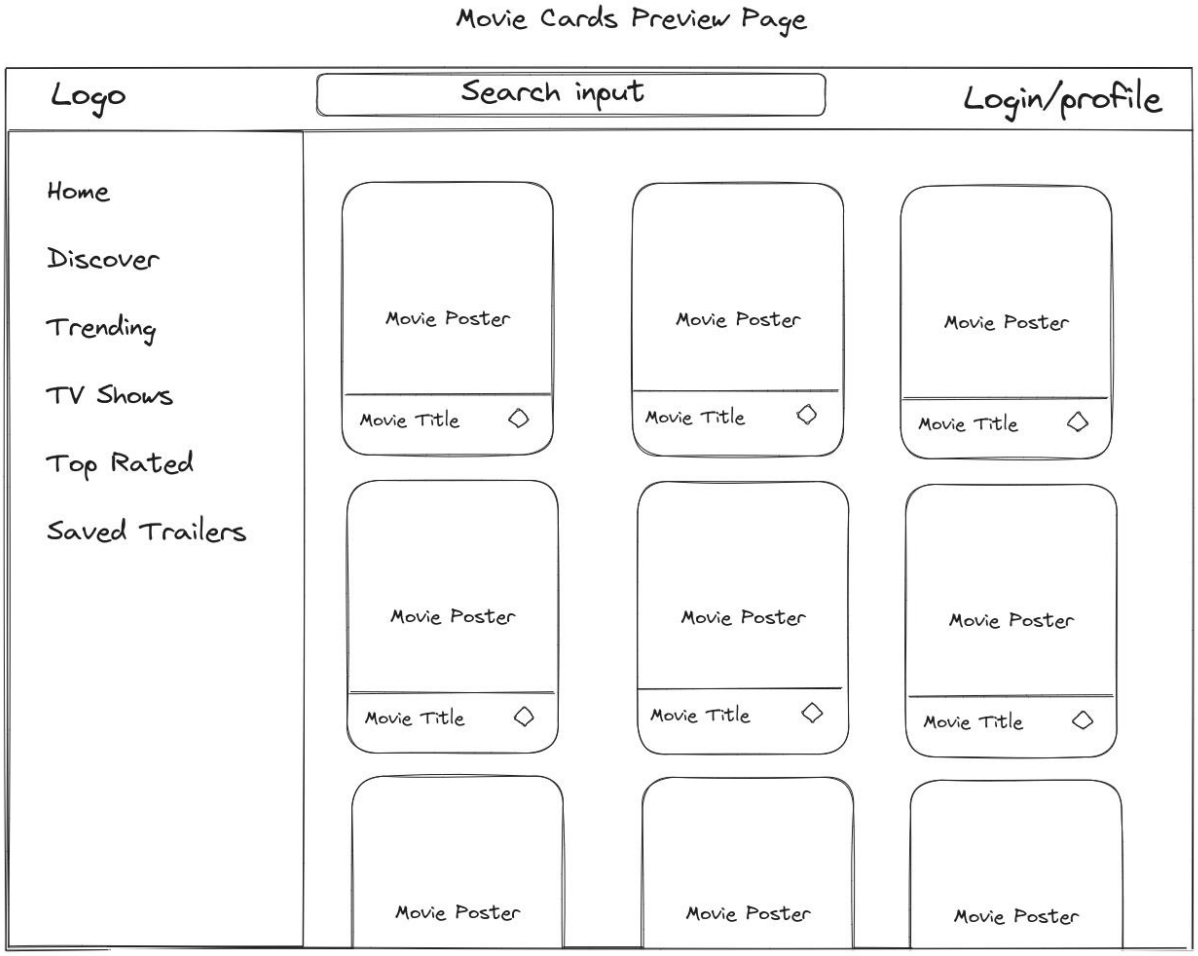


Fig 13 Wireframe Of Movie Lists Page Layout

LOGIN MODAL:

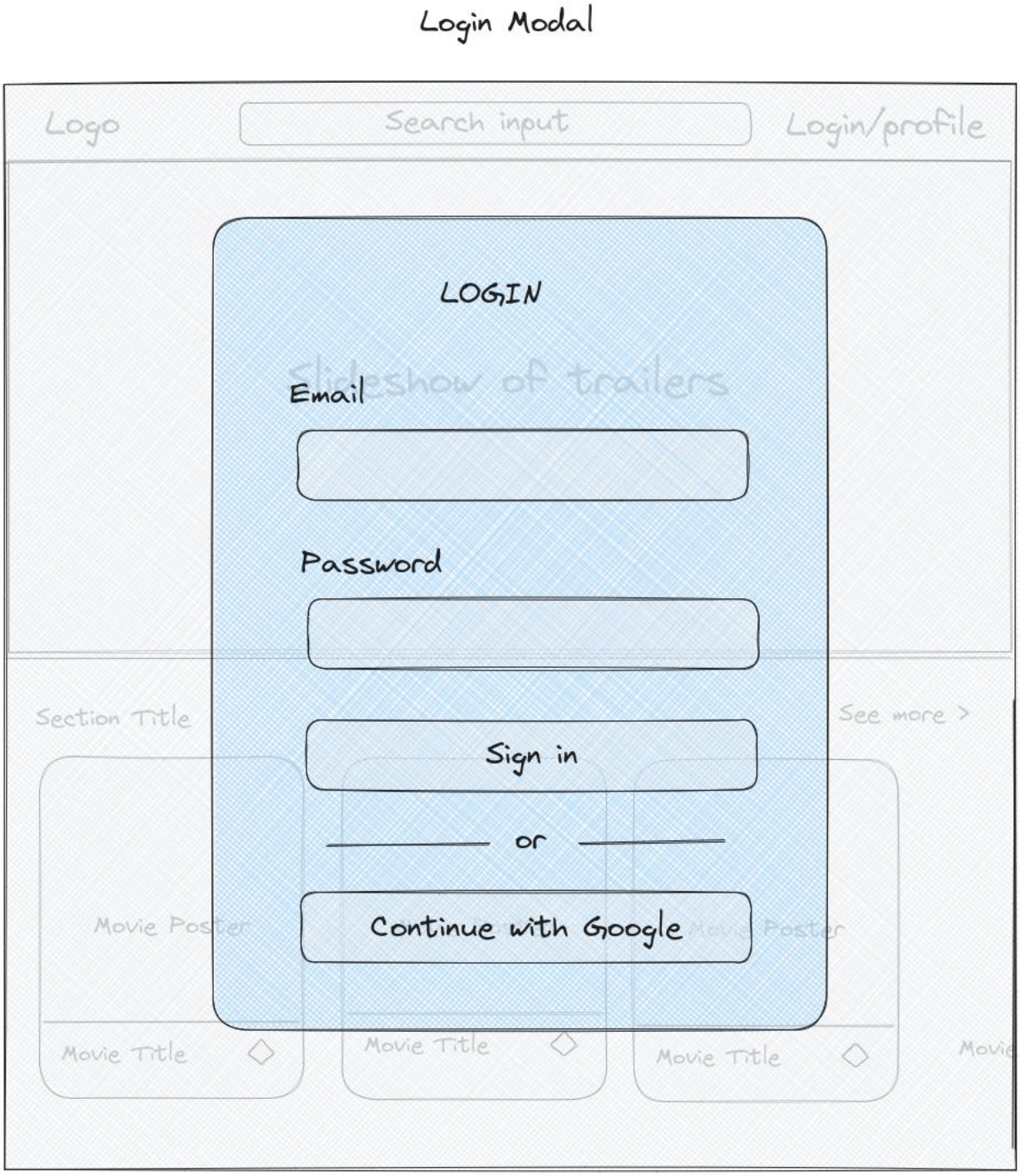


Fig 14 Wireframe Of Login Modal

SIGNUP MODAL:

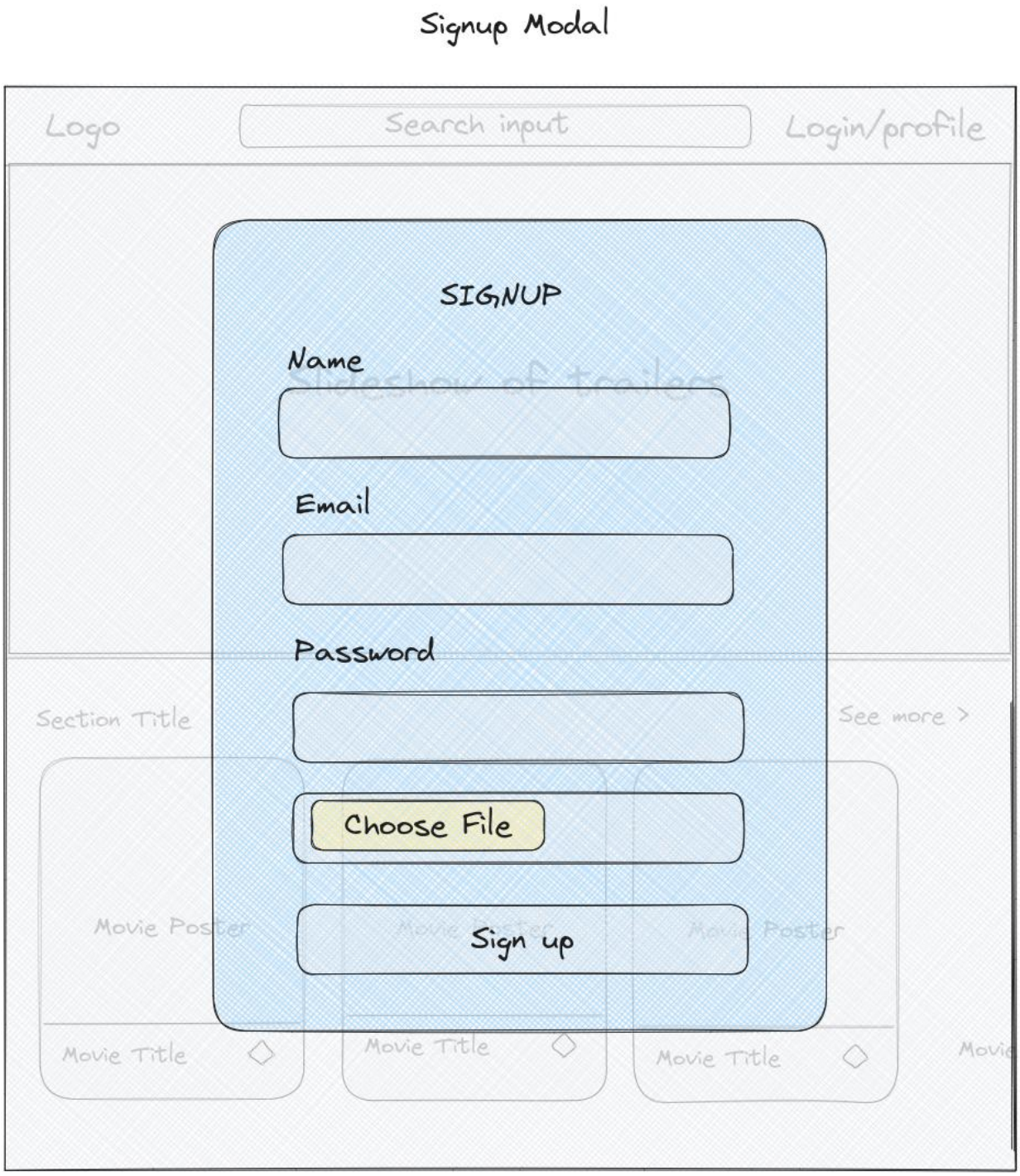


Fig 15 Wireframe Of Signup Modal

RESET PASSWORD MODAL:

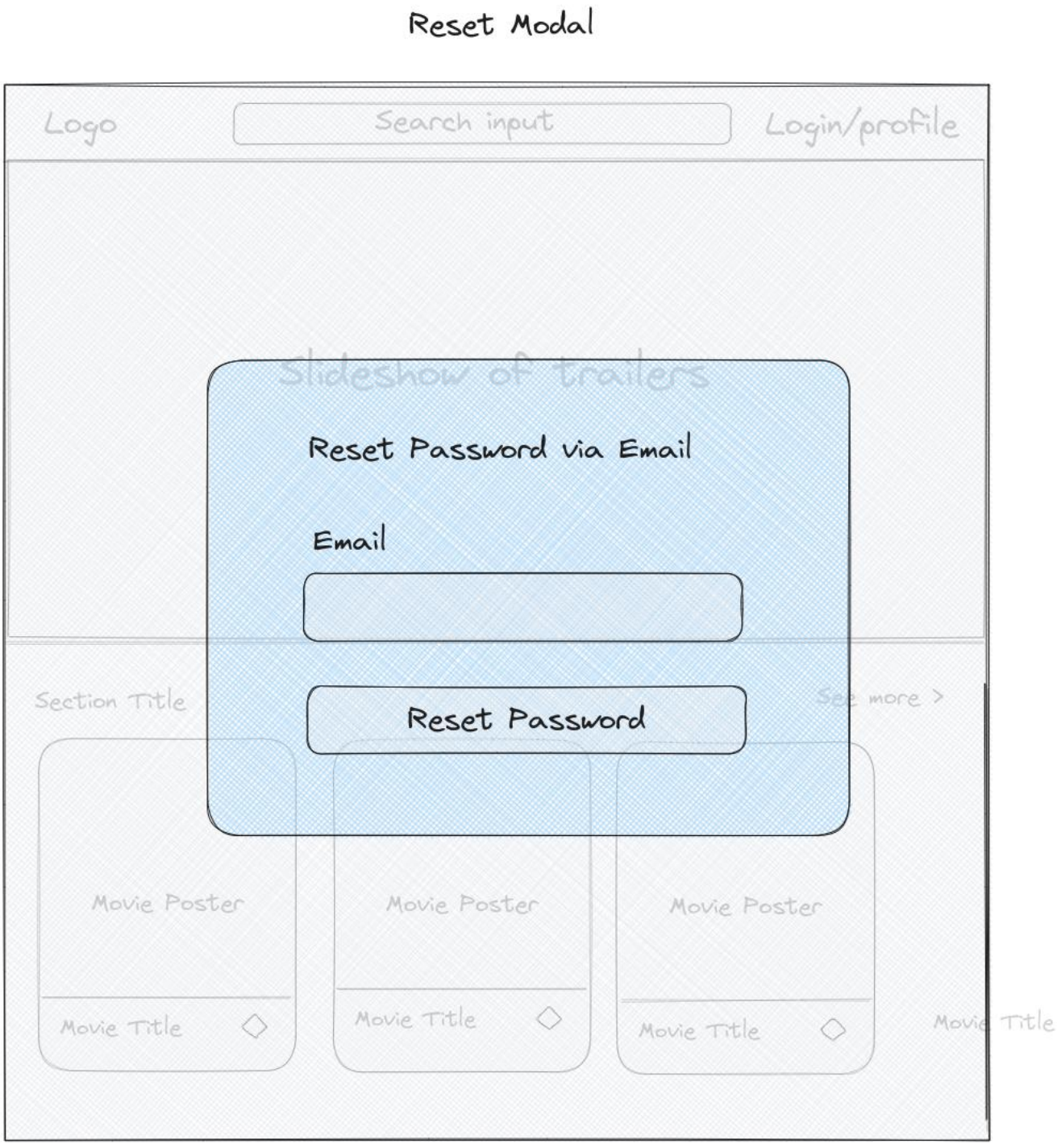


Fig 16 Wireframe of the Reset Password Modal

REFLECTION AND CONCLUSION:

In conclusion, the development of the Trailers Home web application has been an enriching experience that has showcased my skills and knowledge in software engineering. Throughout the project, several important aspects were addressed, including the application's purpose, functionality, design, and implementation. Reflecting on the project as a whole, I have gained valuable insights and perspectives.

From a technical standpoint, working with modern web development technologies such as Next.js, Tailwind CSS, and Firebase has enhanced my understanding of building robust and scalable web applications. The integration of the TMDB API has provided access to a vast movie database, allowing users to discover and bookmark their favorite movies effectively.

Moreover, the design considerations for a simple and minimalistic user interface have resulted in a visually appealing and intuitive application. The choice of color scheme, typography, and layout has contributed to a cohesive and engaging user experience.

During the development process, various challenges were encountered, such as ensuring seamless integration with external APIs and implementing secure authentication and data management. Overcoming these challenges has sharpened my problem-solving skills and expanded my knowledge of best practices in software engineering.

In retrospect, if given the opportunity to further improve the Trailers Home web application, I would consider incorporating additional features such as user reviews and ratings, personalized movie recommendations, and social sharing capabilities. These enhancements could further enrich the user experience and promote engagement within the application.

Overall, the completion of the Trailers Home web application has been a significant milestone in my journey as a software engineer. It has provided me with practical experience in developing a fully functional web application, while also encouraging critical thinking and personal growth. I am confident that the skills and knowledge gained through this project will serve as a strong foundation for future endeavors in the field of software engineering.

BIBLIOGRAPHY

[1] Firebase Documentation. Available: https://firebase.google.com/docs.

[2] Next.js Documentation. Available: https://nextjs.org/docs.

[3] Tailwind CSS Documentation. Available: https://tailwindcss.com/docs.

[4] TypeScript Documentation. Available: https://www.typescriptlang.org/docs.

[5] React.js Documentation. Available: https://react.dev/