



Developing a Front-end web app using React

Abhishek khati Chhetri

Haaga-Helia University of Applied Sciences

Bachelor of Business Information Technology

Bachelor's thesis

2024

Author
Abhishek khati chhetri
Degree
Bachelor of Business Information Technology
Report/Thesis Title
Developing a Front-end web app using React
Number of pages and appendix pages
33
<p>The thesis titled "Developing a Front-end web app using React" aims to develop a comprehensive web application tailored to movie enthusiasts. By integrating React for interactive UI, API for data communication, and the application targets enhancing user experience and privacy.</p> <p>The research explores user-centric design principles to create a platform where users can explore and engage with a wide range of media content. The project follows a practical approach, focusing on continuous development, ensure usability and effectiveness.</p> <p>The objectives include developing a functional web application and deploying the product while considering sustainability and scalability. The key to successful completion of the project is the selection of the appropriate tools and frameworks. Visual Studio Code was one of the best selections for the project. Data fetching from The Movie Database(TMDb) API was the main task for the web app development, so several fetch functions were defined in the project. The crafted fetch functions retrieved movie data, genres, trending movies, upcoming movies and cast information. Firebase authentication has been used for the google authentication allowing marking features.</p> <p>Keeping the user interface as the main focal point, navbar, search bar and the homepage were developed ensuring seamless browsing experience. The main goal was meet as a generic web application for a movie was developed where different media contents were presented along the favourites marking options for the signed in user. The application is deployed using Netlify.</p> <p>During the project initiation, many new frameworks were new and learned by the author. The experience and skills gained during the project were one of the positive aspect of the project to the author.</p>

Author

Abhishek khati chhetri

Key words

React, JavaScript, web development, API, TMDb, HTML, CSS, Firebase, Frontend.

Table of contents

ABBREVIATION	3
1 Introduction.....	1
1.1 Objectives	1
1.2 Scope.....	1
1.3 Out of Scope	2
1.4 Thesis structure	2
2 Theoretical framework.....	3
2.1 HTML	3
2.2 CSS.....	4
2.3 React.....	5
2.4 Vite	7
2.5 React-Helmet.....	7
2.6 Node Package Manager(npm).....	8
2.7 Git	9
2.8 GitHub.....	9
2.9 Netlify	10
2.10 Firebase	10
2.11 The Movie Database(TMDb).....	11
3 Implementation of the Web App	13
3.1 Development Tools	13
3.1.1 Operating System, Terminal, and Web Browser	13
3.1.2 Code Editor	13
3.1.3 Code Hosting Platform	14
3.2 Data fetching	15
3.2.1 Fetching Data from TMDb API	15
3.2.2 Securing API key.....	16
3.2.3 Challenges and solutions	16
3.3 Project file structure	17
3.4 User Interface	20
3.4.1 Navigation bar and Search bar.....	20
3.4.2 Search suggestions and Search result page.....	22
3.4.3 Homepage.....	23
3.4.4 Trending Page	24
3.4.5 Upcoming Page.....	25
3.4.6 Favourites Page	27

3.4.7	Login.....	28
4	RESULTS.....	30
5	CONCLUSION	31
6	REFERENCES.....	32

ABBREVIATION

API	Application Programming Interface
CLI	Command Line Interface
CSS	Cascading Style Sheet
ES	ECMAScript
HTML	Hypertext Markup Language
JSON	JavaScript Object Notation
JSX	JavaScript XML
NPM	Node Package Manager
SEO	Search Engine Optimization
TMDb	The Movie Database
UI	User Interface
UX	User Experience
VS Code	Visual Studio Code
XML	Extensible Markup Language
URL	Uniform Resource Locator
.env	Environment Variable

1 Introduction

The introduction consists of the background, objectives, scope, out of scope, and structure of the thesis. In the modern world, the consumption of the internet has grown massively towards the online platforms for educational, entertainment, research, exploration and other many more purposes. As there are always various options available in the online platform these days, the main goal of this thesis is to provide users with an online video streaming web application using React, TMDb API, Firebase.

1.1 Objectives

The primary objective of this thesis is to address the growing need of the online platform for the users for entertainment, educational, research purposes. The result of the thesis will provide users with user-friendly web application for accessing different types of movies, series, documentaries and accessing the movie information, streaming options. The application also aims to provide rich information of the movie details, ratings, reviews, cast information and streaming availability.

The thesis seeks to explore modern web development framework, React. By utilizing these technologies, the aim is to build a scalable and user-friendly web application architecture which is capable of handling user interactions and data management.

1.2 Scope

The thesis will be focused to provide a user-friendly design and development of a responsive user interface using React. The application will use TMDb API for retrieving movie data. Bootstraps will be used to enhance the user interface.

React hook form will be used to handle the user input forms. React-Router-Dom will be used for the navigation between different pages. The development process will be well documentation according to the guidelines.

The application also provides the user to enjoy a stream community where reviews of the media will be addressed with the reviews from other users. The section to make a favourite will also be a scope of the application.

User authentication with a security to ensure secure login will be designed. Firebase authentication will be used in the application.

1.3 Out of Scope

Some other relevant information regarding the technologies used will be described in the theoretical framework. The design and the concept of the web application will not be included in the thesis.

1.4 Thesis structure

In this thesis, the theoretical framework will be tailored to provide with required elaboration about the terminologies and the technologies used to build the web application.

The structure of the report consists of the following chapters which is followed by a list of references and the appendices.

Chapter 1 consists of Introduction where the overall introduction about the report and the web application is presented. The introduction part is defined with some subtopics like Objectives, scope, out of scope and thesis structure.

Chapter 2 discusses the theoretical background of the terms which are used during the report and the technologies used during the development of the web application.

Chapter 3 provides information regarding the empirical part where the description of the starting point and presentation of the design is described, how is the outcome produced and finally the Final outcome.

Chapter 4 is the part where the discussion about the usability of the outcome is explained. More about the product itself, also about self-learning is explained in this part.

Finally, the references are listed in the reference part which includes the source material which is referenced in the report.

2 Theoretical framework

The modern world has evolved very fast in the case of internet. In a survey carried out by Worldwide digital population 2024, 5.3 billion were listed as internet users worldwide which is 66.2% of the global population. (Petersyan, 2024). Different people with different mindset go around the internet in the search of what they want and the people get along to those websites and platform where they find their actual interest and it can be anything like playing games, listening to music, watching videos, reading articles, and many more. Likewise, this thesis is all about creating a user centric web application where users can enjoy varieties of movies, series, documentaries. The application should be built in an order where every user can use all your functions and see the contents in your website so it should always be user centric. While building the application, few things should be taken into consideration in the case of common security threads. So, in this case, there are some best practices which can mitigate such risks like, input validation, encryption, authentication, session management.

The main focus of this thesis is to build a web app using React and TMDb API (Application Protocol interface). For the Front-end part, many different UI libraries are used from React and other libraries. Bootstrap was used in the application to make it look more interactive and responsive. React-router-Dom was used for the routing between the pages. React-hook-form was used to control the input field in the web application. To design the buttons, react-icons was used.

React integrates different types of frameworks to provide the best built ready-made components which can be used just by importing it to the project file by using “import” keyword. The development has been very easy and time saving as compared to the traditional way of developing the web applications.

Mostly the codes were hard coded and put into the same project folder with various files defining html, CSS and JavaScript files but currently the situation has changed allowing anyone to use different components. The application is designed in very simple way that user does not have to navigate to different page to find what they need. Infinite Scroll has been used to render the data continuously while scrolling down the page.

2.1 HTML

The HMTL (HyperText Markup Language) is known as the standard markup language for the web pages. Tags are used to enclose inside the angle brackets (<HTML>). There are some of the few tags which should be in a web page like <HTML>, <HEAD>, <TITLE>, <BODY>. Most of the tags

which are used needs to be closed like <h1>Tag</h1>. The extension used to save the HTML files are “.htm” or “.html”.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Figure 1. Example of a HTML document (W3Schools, n.d.)

The above figure represents a simple html document where the starting tag <!DOCTYPE html> defines the type of the document. The head element contains the <title> tag which defines the title of the web page. Each tag should be always closed to move on to the next html document, “</title>” this is the closing tag which closes the <title> element. The body of the html contains most of the data in the html like <h1>, <p>, <div>, etc. The document then is rendered in the visual form in the web site.

2.2 CSS

Cascading Style Sheets is a stylesheet language for styling the documents written in a markup language. Multiples styles can be defined using CSS along with the behaviour of the component. Font, background color, size, spacing, etc. can be defined using CSS. (Docs M. W., n.d.). It uses “.css” file extension.

CSS consists of certain set of rules which defines how does the HTML elements should be displayed. A set of declarations are declared inside the curly braces “{}”. The declaration types are the stylings for the elements like font-size, colour, padding, margin, etc. CSS provides different types of selectors for styling different elements in the HTML document. The selectors are elements selectors, class selectors, ID selectors, attribute selectors, pseudo-classes and pseudo-elements. Selectors allows to style some specific elements or groups of elements.

CSS follows the principle of cascading, which implies that multiple styles can be applied in one HTML document. It offers various features which allows developers to develop a responsive design

for their website. It has the features where you can define various screen sizes and devices. Media queries can be defined based on different styles based on viewports, dimension which enables to create mobile friendly layouts.

There are several frameworks of CSS which provide pre-defined CSS styles which can be easily integrated into your application. There has been transitional change in recent years introducing new features and specifications such as Flexbox, Grid Layout, Transitions, Animations, and filters.

2.3 React

React is a JavaScript library which was created by Facebook (Currently known as Meta) on 2013 for building UI. React is the most used frontend libraries these days. It is used for developing web applications with higher traffic. When compared to other web development technologies. React is one of the newest in the market. The time changed for the front-end developers to create user-friendly interfaces after the release of react.

The Facebook developers received some maintenance concerns in 2011 where the teams needed some extra workers as the Facebook Advertising app earned more capabilities. They began to experience lagging as they had updated the personal and app functionality. Slowly it was getting challenging for them to manage their software. After the Facebook programmers could not fix the cascade updates, their code needed to be updated to make it more effective. Finally, a prototype was developed, and it was the time when React.js was the outcome.

React uses the virtual DOM as only the components with changed state are updated rather than updating all the components in the DOM which makes it faster (Deshpande, 2023).

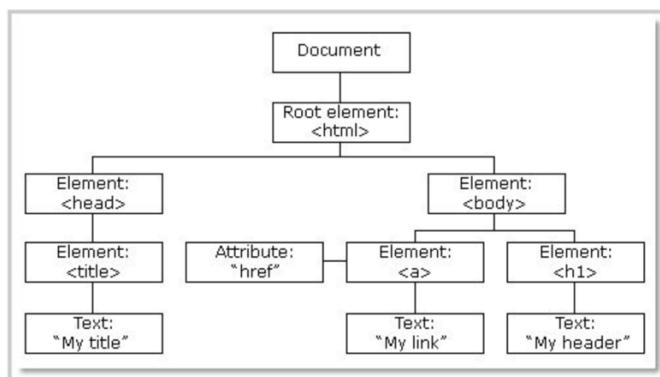


Figure 2. DOM of a webpage

The above figure illustrates the Document Object Model. The DOM is the data representation of the structure of the web document. DOM can be displayed in the browser window and the HTML source. The DOM allows it to be manipulated and it can be modified using any scripting language

like JavaScript. As the figure above has shown that the main element <html> which contains sub elements <head> and <body> tag which is a must in a web page. The <head> tag contains objects like <title> which can represent the title of the web page. The body element is then the part where most of the contents of the web page are located or are defined.

Facebook released React as open source in 2013 which is when developers started to learn and use it. The code then helped developers develop web applications with seaming less interface with fast and reusable components which are easy to maintain.

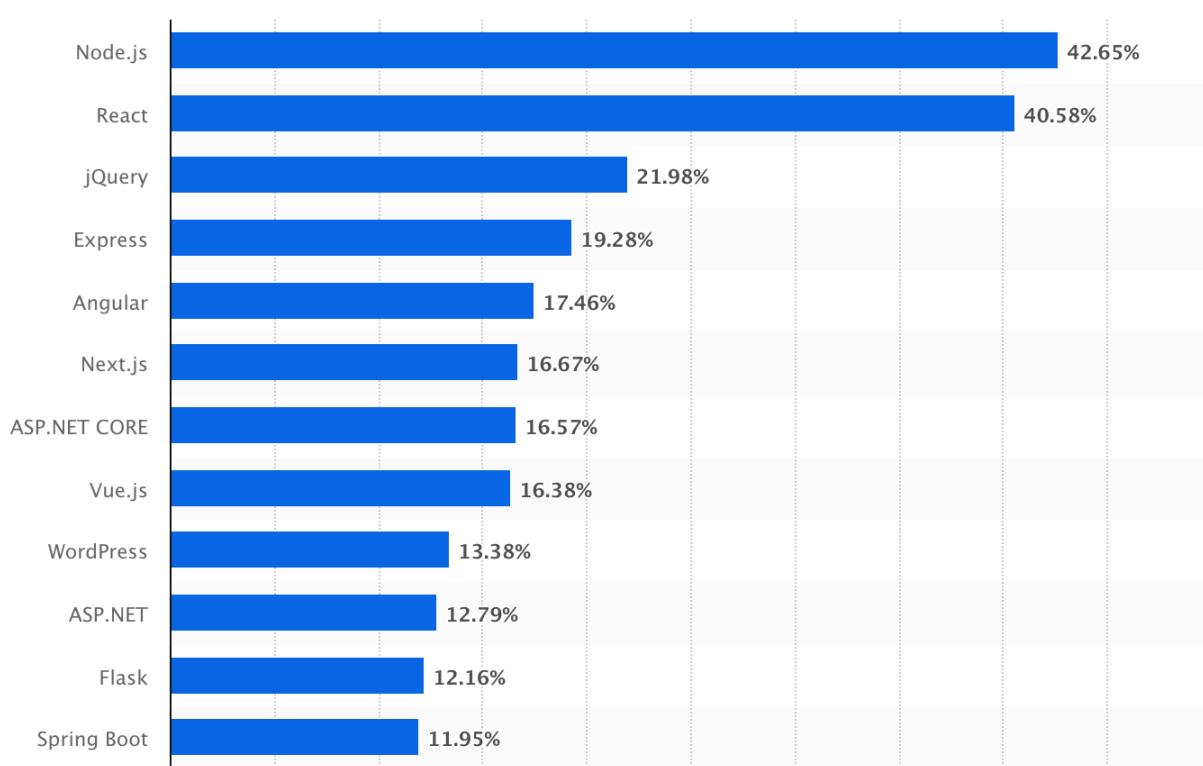


Figure 3. Most used web frameworks among developers worldwide (Statista, 2023).

React is easy to learn but it does not mean that you are ready to roll just after learning react as react has many frameworks which requires to use other tools that you might want while developing your app. It can be used to develop both mobile and web application. React Native is the JavaScript framework which lets you build both iOS and android. It provides the same experience as the native application.

A application basically contains multiple components so, React has a feature where each components can be defined in separate JSX file with their logic which is why the components can be re-used throughout the application which results in decrement in development time. React uses JSX (JavaScript syntactic extension) to define the user interface.

2.4 Vite

During the recent years, the frontend ecosystem has seen some very new tools and frameworks which has made developers easy and helped improve the developer experience. Vite has been used as a build tool for the web application. Vite provides a faster development experience for the modern web development. Vite was created by Evan Vue, developer of Vue.js. (Vite, n.d.)

Vite focuses on JavaScript and TypeScript frameworks like React, Vue and Svelte. Vite is basically used for bundling projects and create a web app easier and faster. The modern time has been very fast growing and developing the technology, there used to be a time where the web applications were developed by writing HTML, CSS, JavaScript files but the time has changed and now we have some tools like babel which has made development easier.

Babel is a JavaScript compiler while is used to convert ECMAScript 2015+ code into the backward compatible version of JavaScript in the current and older browsers. (Babel, n.d.) . In web development, Vite was released on 2020 with an alternative to Create React App (CRA). Vite uses native ES module which are supported in the browsers. ECMAScript is the official standard format to package the JavaScript code for reuse (Node.js, n.d.).

```
JavaScript

// Babel Input: ES2015 arrow function
[1, 2, 3].map(n => n + 1);

// Babel Output: ES5 equivalent
[1, 2, 3].map(function(n) {
  return n + 1;
});
```

Figure 4. Babel transforming arrow function

Vite runs a server which is used for the compilation of the project and serve any required dependencies via ES module. This process requires vite to only update the files that has been changed while running the development server making it faster.

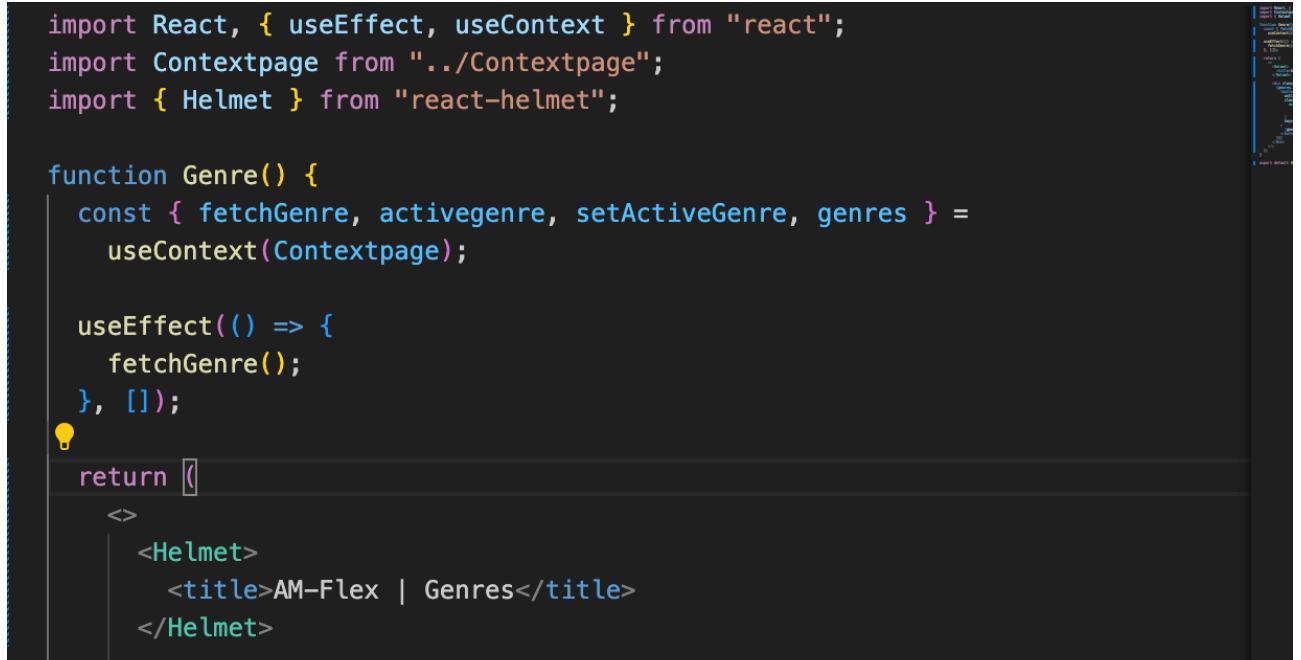
2.5 React-Helmet

React-Helmet is a document head manager which can manage all the changes made in the client side as well as server side. It takes plain HTML and output the plain HTML. It is a media friendly component as it is beneficial for the applications where SEO is important.

Every website developers hope to get a good SEO score and result in top listing in the browser's search result. The browser is unable to render JavaScript as the search engine crawler cannot be trusted with the data so the Single page applications using react is unable to favour the search engine crawlers. So, these kinds of sites score very low in the SEO.

React-Helmet was created to help solve this issue and focus on the SEO. React-Helmet can be used to set the document's title, description, and meta tags. This is very helpful when you need to update the SEO according to your project as it can be updated server-side.

Any duplicate changes to the head will be overridden by the nested components as the Helmet components can override values provided to other in the tree. HelmetProvider is used to stop the memory leaks. Helmet can be used in the pages where you need to define the meta tags (Dwivedi, 2023) .



```

import React, { useEffect, useContext } from "react";
import Contextpage from "../Contextpage";
import { Helmet } from "react-helmet";

function Genre() {
  const { fetchGenre, activegenre, setActiveGenre, genres } =
    useContext(Contextpage);

  useEffect(() => {
    fetchGenre();
  }, []);

  return (
    <>
      <Helmet>
        <title>AM-Flex | Genres</title>
      </Helmet>
    </>
  );
}

```

Figure 5. Using helmet to set the title of the page

2.6 Node Package Manager(npm)

npm is the world's largest software registry where 2.1 million packages were reported as listed in the npm registry. All developers use npm for using the packages. npm is commonly used for Command Line Interface (CLI). It is basically known as the library and registry for JavaScript. All the packages available can be installed using command line tools.

npm also provides version control for all the packages which allows developers to install the required version in your project. npm is the type of the package manager which also fixes the

dependency tree and install the required dependencies by the packages which simplifies the management of the project dependencies and reducing conflicts.

It provides wide range of commands for managing packages, initializing projects, running scripts, and publishing the packages to the registry. The CLI helps developers to seamlessly perform different development task by using CLI rather than wasting surfing for the desired dependency. It allows developers to define custom scripts in the “package.json” file such as building, testing, and deploying the application. The scripting helps in project maintainability and scalability by wrapping up complex workflows into simple and executable program.

2.7 Git

Git is an open-source version control system which was designed to keep record of each, and every change made in a project with efficiency. Git allows to create different branches according to the needs and make it independent of each other (Git, n.d.). Every change made by any user is kept in record and can be tracked by other user dealing in the same project.

Git is one of the topmost version control systems which is managed and maintained actively which was developed in 2005 by Linus Torvalds who is the creator of Linux Operating System kernel. Git is a distributed version control system with a working copy of the code which can also have the history of all the changes made in the project (Atlassian, n.d.).

All the data after a commit is timestamped and includes the metadata which provides smoothness in the development process giving it more transparency. It provides branching feature to work on some specific features without disrupting the main codebase. Unlike the centralized database, git follows distributed model where every developer has a copy of the main project allowing them to modify and commit the code when needed.

Git includes a staging area which is known as the “index” where changes can be added to the repository before committing them which allows developers to review the code and increase the productivity. Git uses advanced algorithms which compresses and stores the data efficiently minimizing the storage and optimizing performance.

2.8 GitHub

GitHub is a platform which allows developers can create, store, manage and share their code. GitHub uses software which provides version control, access control, task management. It is a website and a cloud-based service. It makes the development work faster, convenient and hassle

free. It has been designed in well order manner taking care of the data privacy and providing top level security to the repositories.

GitHub allows multiple users to work in a project which helps to reduce the risk of duplication which can reduce the production time. The changes made by any individual can be committed or saved in the repository only after the pull request by the certain authority holder of the project. All the saves are recorded.

GitHub also provides a project management tool through which the planning and tracking work can be done. A project can be created which contains spreadsheets, task-board and road map. This can be integrated into your issues and help you plan and keep track of the work. There are numerous ways of adding automation to your project which allows you to set fields after certain items are added or changed and you can also achieve the items after certain criteria has been reached.

2.9 Netlify

Netlify is a cloud computing platform which provides various development tools for build, deploy and serverless backend services for websites. Netlify has been providing very easy hosting services for developers so most of the developers prefer to use to build and deploy the website. It carries out all the works and let them deploy the site.

Netlify runs through your GitHub repository which runs a build process and pre-render it to the static HTML. It comes with different packages to offer but if you want to use it for your personal project then you can use it for free. The app can be deployed and hosted for free, and it is just for one member.

Netlify is one of the best web development platforms where you can build, test and deploy your application. It tends to provide better security, performance , speed and scalability. When it comes to deployment, it provides faster and easier service where users can easily deploy their site online. The deployment can be done through GitLab, GitHub, bitbucket. You do not need to worry about the further development of the application as you can normally push your code and the building and deploying is automated.

2.10 Firebase

Firebase is a Google's Mobile application development platform which provides a lot of set of tools through which you can build, grow and improve your app. It provides set of services which developer had to build, and which would take a lot of time to build like analytics, database, configuration,

authentication, file storage, push messages and many more services. The components are fully managed by Google.

Firebase works as a Backend-as-a-Service(BaaS) which means it provides readymade, scalable backend for user's application. It primarily focuses on providing a service to reduce the complex backend operation which every developer must build allowing developers to focus on their frontend and user experience.

Firebase also operates as a collection of APIs. The API is then fetched by the developers in their application. The API is linked to various cloud-based services.



Figure 6. List of products in Firebase (Firebase, 2023).

2.11 The Movie Database(TMDb)

The Movie Database is an online community-built database for movies and TV shows. It provides users with enormous data related to movies, tv shows. It also provides information related to cast, crew, summaries, ratings, posters, and more.

For the development of the web app, the author has used the API provided by TMDb. In order to use the TMDb API, a user has to login to the website and request for API which after a unique API is provided to the user. The key provided is unique and must be sent in every request. Developer API has been used in the app as the TMDb provides two API such as commercial API and developer API. Different URLs are used to fetch the data from the API.

<https://api.themoviedb.org/3/search/multi> (TMDB, 2024) is used for searching movies, TV shows and people in a single request.

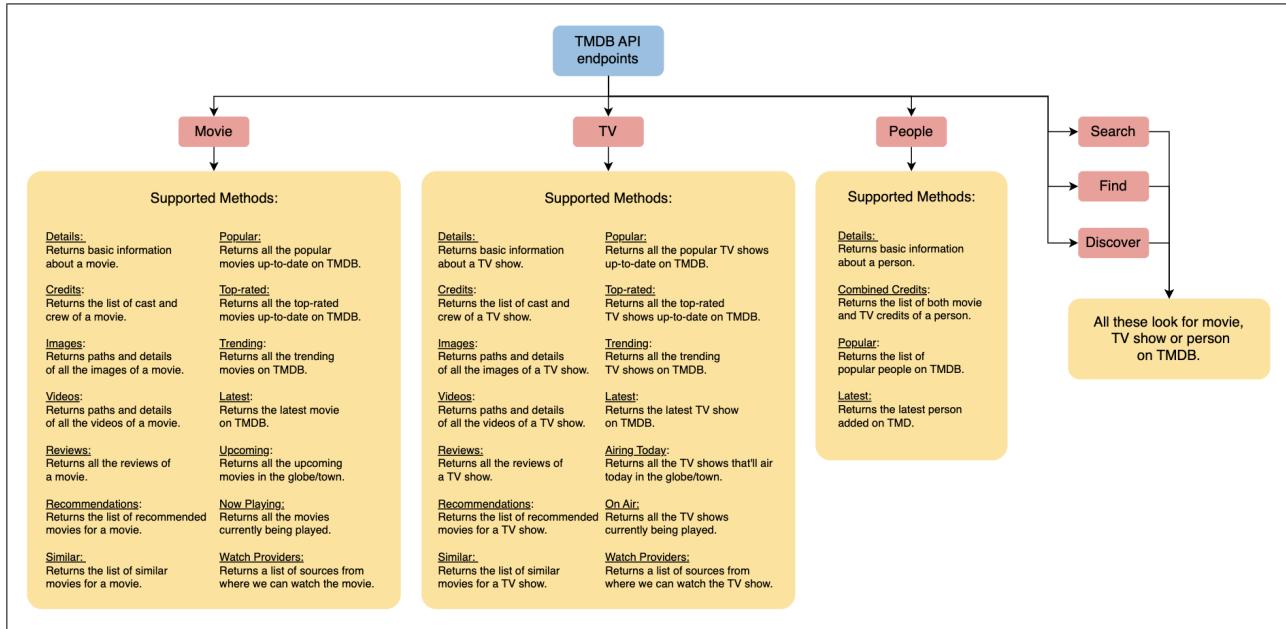


Figure 7. TMDB API endpoints (Educative.io, n.d.).

The above figure represents the API endpoints. Various parameters are used to fetch the data from the API. An API key is required for the authentication.

3 Implementation of the Web App

Designing a website can be very hectic sometimes as the development of the application takes a lot of time and resources so choosing the best development tools can play a crucial part in the development process. There are various tools and frameworks available which can help you in the development of the web application like editor, Frontend frameworks, Package Managers, Git clients or services, Local Development environments. This is the part where we discuss about the development process underlying the development tools and other frameworks which are used in the web applications.

3.1 Development Tools

Development Tools are software, frameworks, library, application which helps you to build, test and manage your website or application. These tools are used in the various steps of the web app development. Tools like code editor, web browser, terminal, platform for code hosting, are some of the important tools which are used to develop an application.

3.1.1 Operating System, Terminal, and Web Browser

The development of the web app was carried out using different operating systems such as Windows 11, Mac OS and iOS as the author was working from desktop and laptop respectively according to the location of work. Microsoft Windows and Mac OS was used to develop the web application and iOS was used to test the web app interface for mobile devices. Google chrome was used in previewing the web application in both environments(PC and Mobile). As two different operating systems were used during the development of the application, so the Command Prompt was used in Microsoft Windows and Terminal was used in Mac OS. Command prompt and Terminal were used to install the dependencies, framework using npm. The web app was initialized using vite and the keyword “**npm create vite@latest**”.

3.1.2 Code Editor

Selection of a good code editor can help in the development of the any project as it can affect the production time. Code editor allows developers to write the code error-free as they are provided with the features which facilitate the coding process. The color-coded syntax highlighting provides any coder to see the structure of the code. The editor automatically identifies the code element that the programmer is using in the code and the code highlighting provides different colour to the variables, commands, and functions.

Most of the editors have error-checking feature which detects errors reducing long and tiring debugging process. Also, the auto-complete feature helps you to code more proficiently as the editor suggests the next code that you want to write which saves time. There are some of the great code editors like VS Code, Notepad++, Atom, Sublime.

Among different code editors, VS Code was selected for the development of the project. VS Code is free and open-source text editor developed by Microsoft (Mustafeez, 2024). It is available for operating systems like Windows, macOS and Linux.

VS Code has been in the top spot letting developers with various addons to the project. VS Code is fast and robust as compared to other editors. Because of its IntelliSense which provides auto-complete feature. It supports many programming languages like Java, C++, C#, Python, TypeScript.

VS Code has in-built terminal which helps to run different commands. It also allows git features so that you can clone your project directly, make changes and finally push it. Because of its very big ecosystems for the extension, VS Code can be very useful for development within one application. Some add-ons like Prettier Code Formatter, Auto rename tag, Bracket Pair Colorizer that are provided by VS Code were used by the author.

3.1.3 Code Hosting Platform

GitHub was chosen during the development of the application as the code can be stored and updated in an efficient way. It allows for versioning of the code which is very useful during the development process.

Keeping track of all the records are very important as it helps for the smooth and precise development. As the code can be in public mode as well as private mode. The code can always be shared to anyone in case it is kept public where others do not have access to your code in private mode.

The deployment of the web application is also based on GitHub as Netlify integrates with the GitHub repository. After every commit, the code is rebuilt, and the changes are set to the deployed site. The project was deployed at: <https://amflex.netlify.app>.

3.2 Data fetching

Data fetching was one of the important steps of the project as the main element was the data which was fetched from API. A context page was created which passes the data down the tree component. Data fetching was the critical part of the development so here are the list of processes followed during the development.

3.2.1 Fetching Data from TMDb API

Several fetch functions were defined and utilized in the “ContextPage.jsx” to retrieve the data from the API key. Some of the functions are:

- FilteredGenres: This function fetches the movie data on a specific genre.
- fetchSearch: This function fetches the movie based on the search query.
- fetchGenre: This function fetches the list of movie genres.
- fetchTrending: This function fetches the trending movies.
- fetchUpcoming: This function fetches the upcoming movies.

```
src > ⚛ Contextpage.jsx > 🏃 MovieProvider
12  export function MovieProvider({ children }) {
48
49    const fetchSearch = async (query) => {
50      const data = await fetch(
51        `https://api.themoviedb.org/3/search/movie?api_key=${APIKEY}&query=${query}`
52      );
53      const searchmovies = await data.json();
54      setSearchedMovies(searchmovies.results);
55      setLoader(false);
56      setHeader(`Results for "${query}"`);
57    }
58
59    const fetchGenre = async () => {
60      const data = await fetch(
61        `https://api.themoviedb.org/3/genre/movie/list?api_key=${APIKEY}`
62      );
63      const gen = await data.json();
64      setGenres(gen.genres);
65    }
66
67    const fetchTrending = async () => {
68      const data = await fetch(
69        `https://api.themoviedb.org/3/trending/movie/day?api_key=${APIKEY}&page=${page}`
70      );
71      const trend = await data.json();
72      setTrending(trending.concat(trend.results));
73      setTotalPage(trend.total_pages);
74      setLoader(false);
75      setHeader("Trending Movies");
76    }
77
78    const fetchUpcoming = async () => {
79      const data = await fetch(
80        `https://api.themoviedb.org/3/movie/upcoming?api_key=${APIKEY}&page=${page}`
81      );
82      const upc = await data.json();
83      setUpcoming(upcoming.concat(upc.results));
84      setTotalPage(upc.total_pages);
85      setLoader(false);

```

Figure 8. Fetching data from API

The above figure illustrates several fetch functions which are defined in the contextpage.jsx to fetch the data from the TMDb API. The functions are 'filteredGenre', 'fetchSearch', 'fetchGenre', 'fetchTrending' 'fetchUpcoming'.

3.2.2 Securing API key

The API key is stored in .env(environment variable) to ensure the integrity of the data as the API key is sensitive and unique. The env file is not exposed in the version control system. While storing the API key in the environment variable , it is easier for updating the code without updating the whole codebase. The use of the environment variables ensures security and best practice for software development. The .env file should be also added in the .gitignore file which ignores the file while uploading the codebase in the GitHub.

The secrecy of the API should be always considered as it contains sensitive data. There is a greater risk of data leak and other problems related to huge bills due to the API usage limits if it is exposed to the wrong hands.

The API key was kept in an environment variable as:

```
"import.meta.env.VITE_API_KEY"
```

3.2.3 Challenges and solutions

While handling sensitive data coming from the API keys, as the unauthorized access should be always taken into consideration. Though, in this web application, the data may not be too sensitive as compared to other personal data but for the best practices, the author has also focused to handle the sensitive data by storing the API key in an environment variable.

The data handling is also one of the security concerns in a development process as the web application has heavily relied on the external data fetched from the API so, the author has also presented the best practice for the asynchronous operations by using modern JavaScript the `async/await` syntax.

The `await` syntax is used to follow the asynchronous operation where it passes the promise and waits for the result . The promise is always in three states(`pending`, `fulfilled` and `rejected`). It waits for the promise and suspends the execution until the promise has been fulfilled or rejected. (Docs M. W., n.d.) This process lets other operations to run independently making the code non-blocking.

3.3 Project file structure

React.js was used to build the project. Vite was used as the development tool to build as Vite provides the basic template and dependencies which are required for any frontend development environment. Also, it provides fast development server which allows developers to see the changes saved without reloading the whole page.

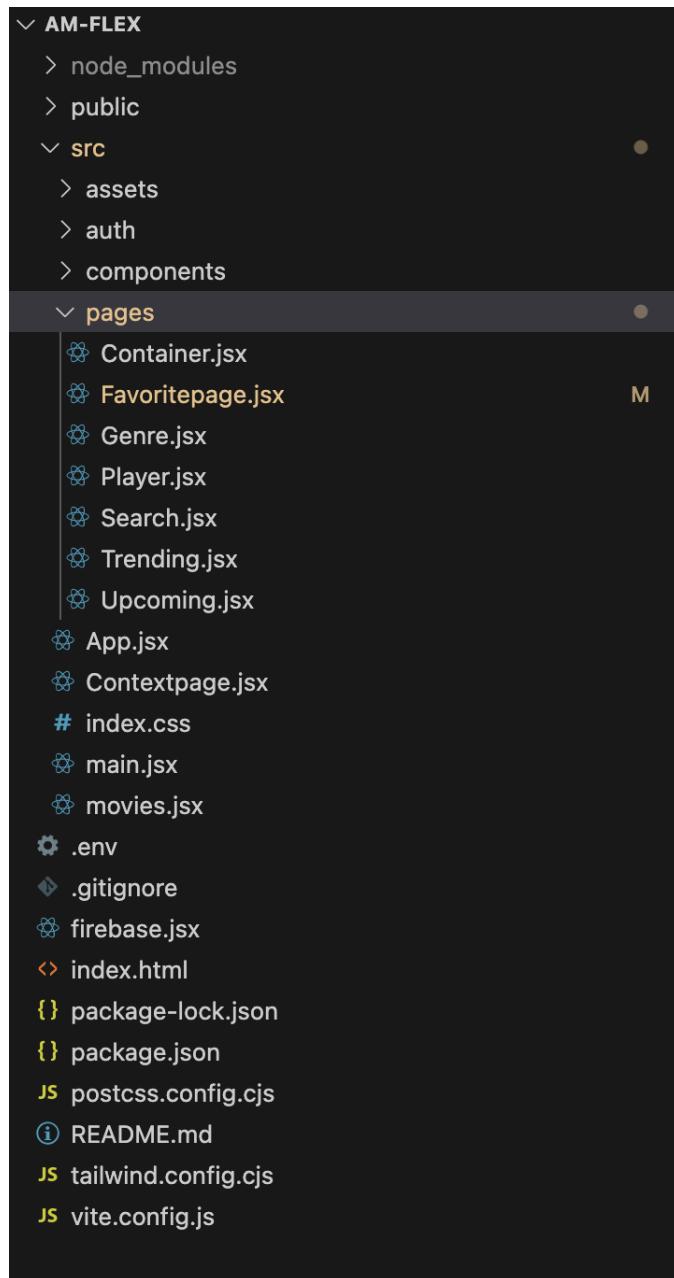


Figure 9. Structure of the project

The above figure illustrates the project file structure. The components such as navbar, movie details, cards, button, search bar are kept inside the components folder. Images and CSS are inside

the “images” and “styles” folder of assets folder. The auth folder contains the sign in page for the google authentication provided by Firebase. The pages folder contains the pages such as Genres, Trending, Favourite page.

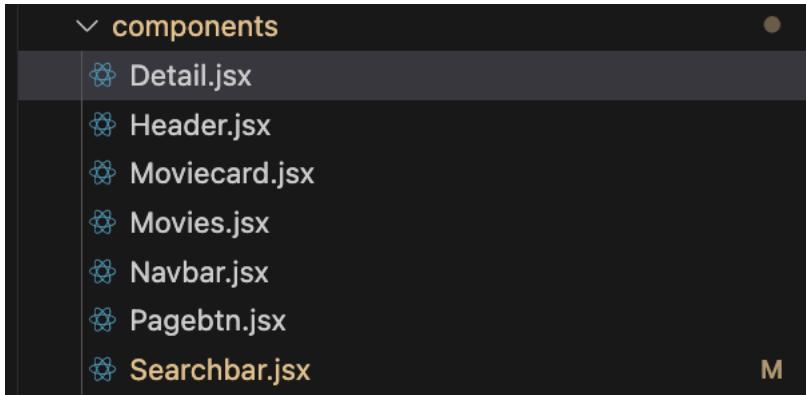


Figure 10. components folder

Main components like button, movie card, nav bar, and search bar were created and placed inside the component files.

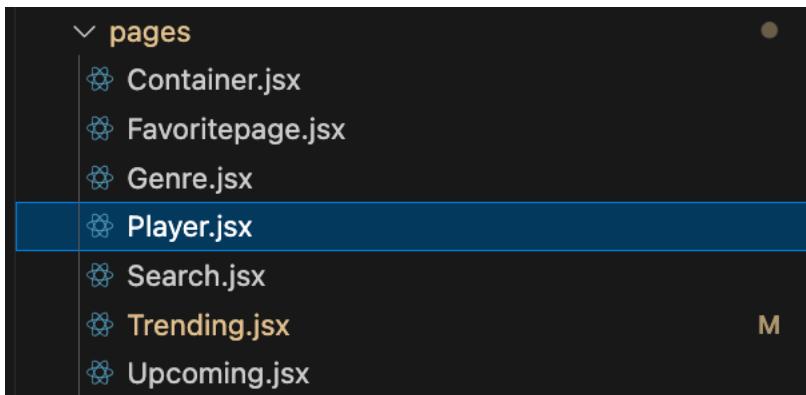


Figure 11. pages folder

The pages to navigate through in the web app were defined and placed in the pages folder.

The ContextPage.jsx manages the movies-related states and the authentication provided by Firebase. The data which are rendered through the web app is fetched from the ContextPage.jsx. TMDb API was used to fetch the movie data. App.jsx acts as the root component. React-router-dom has been used to manage the routing between the pages. The movie player component is inside the pages folder.

```

src > App.jsx > App
15   import { Helmet } from "react-helmet";
16   import logo from "./assets/images/logo.png";
17
18   function App() {
19     return (
20       <MovieProvider>
21         <Navbar />
22         <Helmet>
23           | <meta property="og:image" content={logo} />
24         </Helmet>
25         <ToastContainer
26           position="bottom-center"
27           autoClose={3000}
28           hideProgressBar={false}
29           newestOnTop={false}
30           closeOnClick
31           rtl={false}
32           pauseOnFocusLoss={false}
33           draggable
34           pauseOnHover
35           theme="dark"
36         />
37
38       <div className="md:ml-[ ]">
39         <Routes>
40           | <Route path="/" element={<Container />} />
41           | <Route path="/login" element={<Login />} />
42           | <Route path="/trending" element={<Trending />} />
43           | <Route path="/upcoming" element={<Upcoming />} />
44           | <Route path="/moviedetail/:id" element={<Detail />} />
45           | <Route path="/favorite" element={<Favorite />} />
46           | <Route path="/player/:id/:title" element={<Player />} />{" "}
47           | <Route path="/player/:id" element={<Player />} />{" "}
48           | <Route path="/search/:query" element={<Container />} />
49           | <Route path="/search/" element={<Container />} />
50         </Routes>
51       </div>
52     </MovieProvider>
53   ];
54 }
55
56   export default App;
57

```

Figure 12. App.jsx

The above image shows the “App.jsx” file where the “App” component serves as the entry point of the web application. The Helmet was used in the web app to provide the meta data. The toast provider is a container for managing the notifications to the user about the success and error actions. Routes has been defined for the navigation between different pages provided by react-router-dom. All the route link has been set to the corresponding pages.

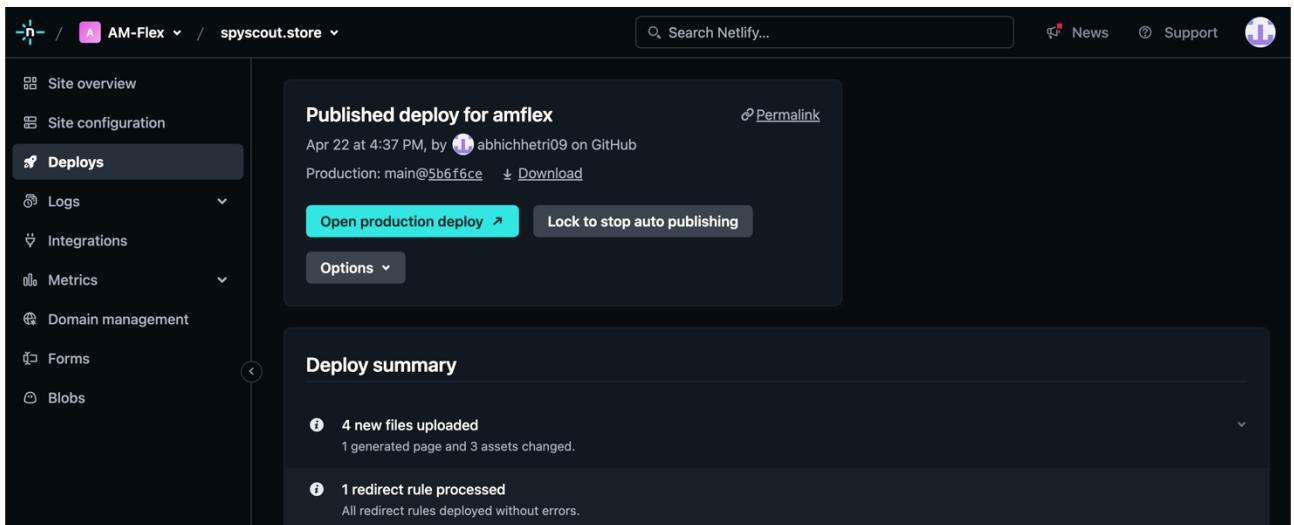


Figure 13. Deployment through Netlify

3.4 User Interface

User Interface design is done to provide an explicit view of any data to the user. It is a design which focuses on style and the experience. It usually represents the graphical user interface. The user interface in the web application designed by the author was very clear and descriptive to any user. It has been focused to the consistency as well as the usability of the application.

3.4.1 Navigation bar and Search bar

The Navigation bar is located in the top of the web page which allows user to navigate to different path. The logo on the left side of the nav bar also navigates to the main page. The nav bar has set the navigations to Genres, Trending, Upcoming, Favourites page. The login option is also on the right side of the nav bar.

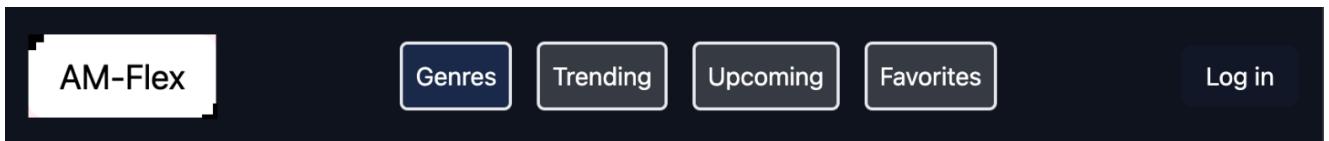


Figure 14. Desktop view



Figure 15. Mobile View

The above Figure 9 and Figure 10 illustrates the Desktop view and Mobile view of the web application. The left side has an icon for “AM-Flex” which navigates to the main page. The Genres, Trending, Upcoming and Favourites pages are only rendered in Desktop view. The login button changes to the profile picture after logging in to Google.



Figure 16. Search Bar

The Figure 11 illustrates the search bar in all the devices.

3.4.2 Search suggestions and Search result page

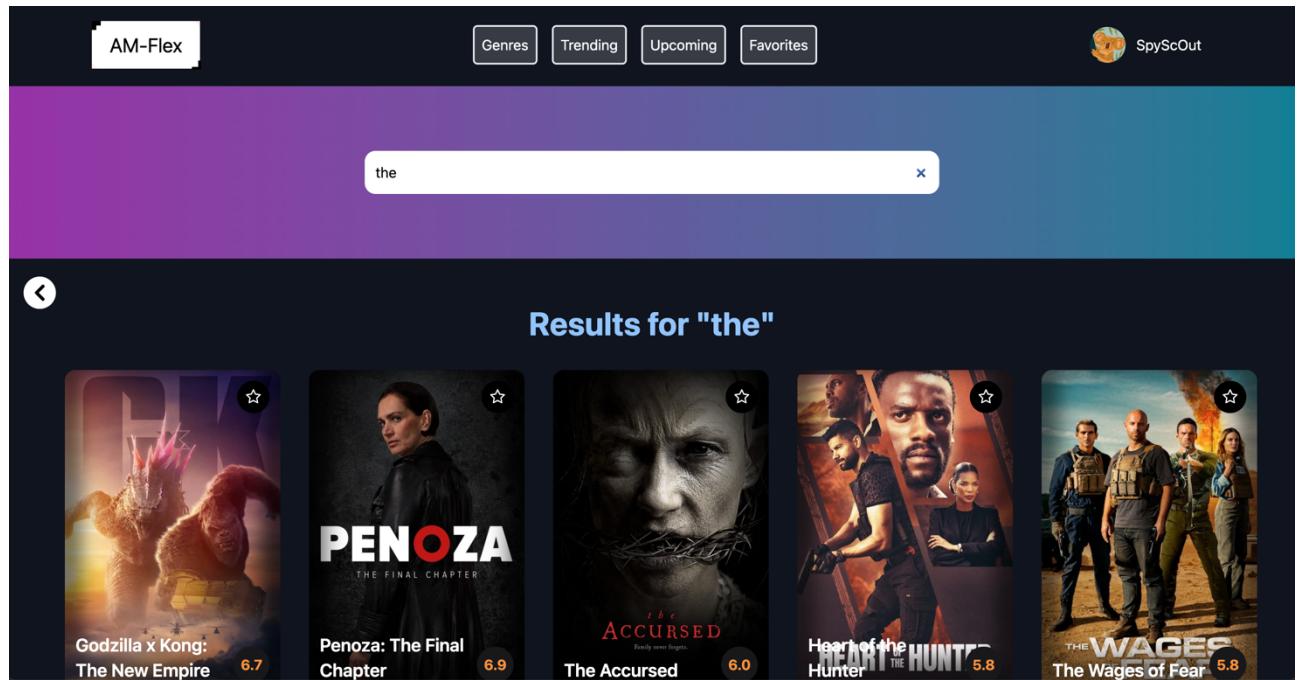


Figure 17. Search Suggestion

The above figure 12 illustrates the Search Suggestion according to the keyword provided in the search bar. The suggestions are suggested right after the user starts to type on the search bar.

3.4.3 Homepage

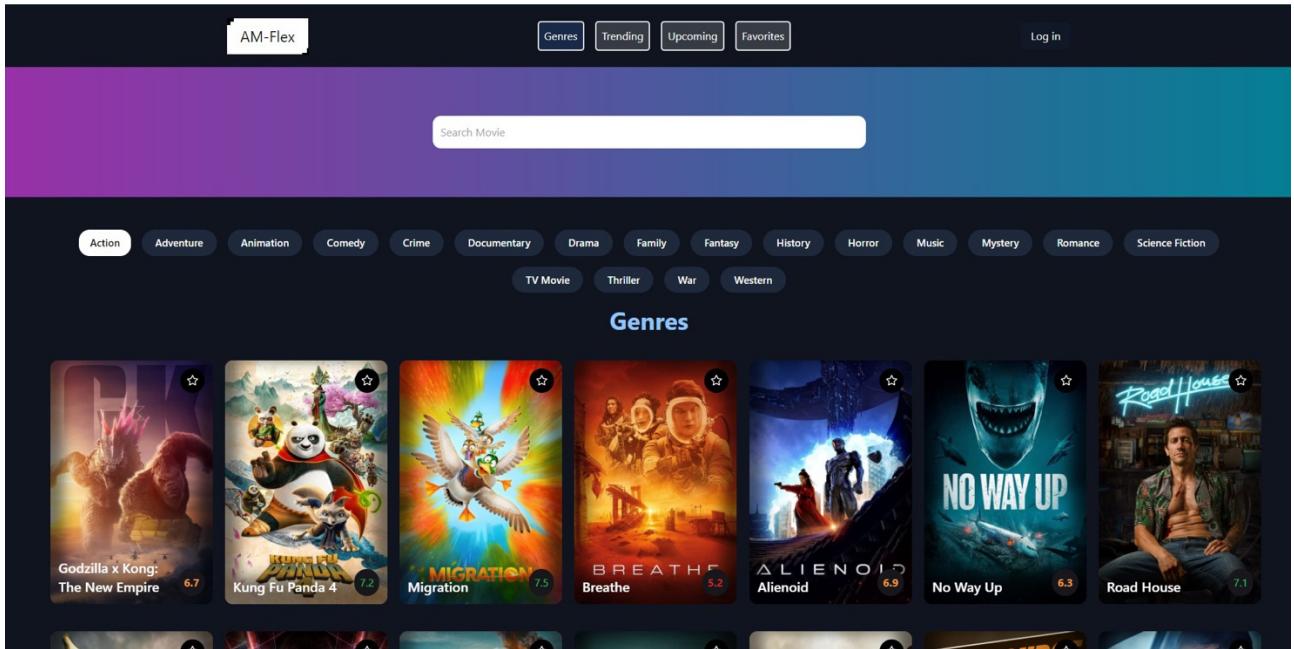


Figure 18. Homepage

The above figure shows the main page of the app. The main page contains nav bar, search bar, Genres and list of infinite movies which are rendered after scrolling to the bottom of the page.

The web app is built simple with no more components and navigating pages where user has to navigate through different pages but as the data here can be delivered in a style where the author has used infinite scroll which allows the data to be displayed in the same page.

The genre options can be found after the search bar where the suggestions of genres are listed allowing users to rather look for different kinds of movies according to the genres. When clicked in the suggestions, users get the list of the movies according to the picked genre.

3.4.4 Trending Page

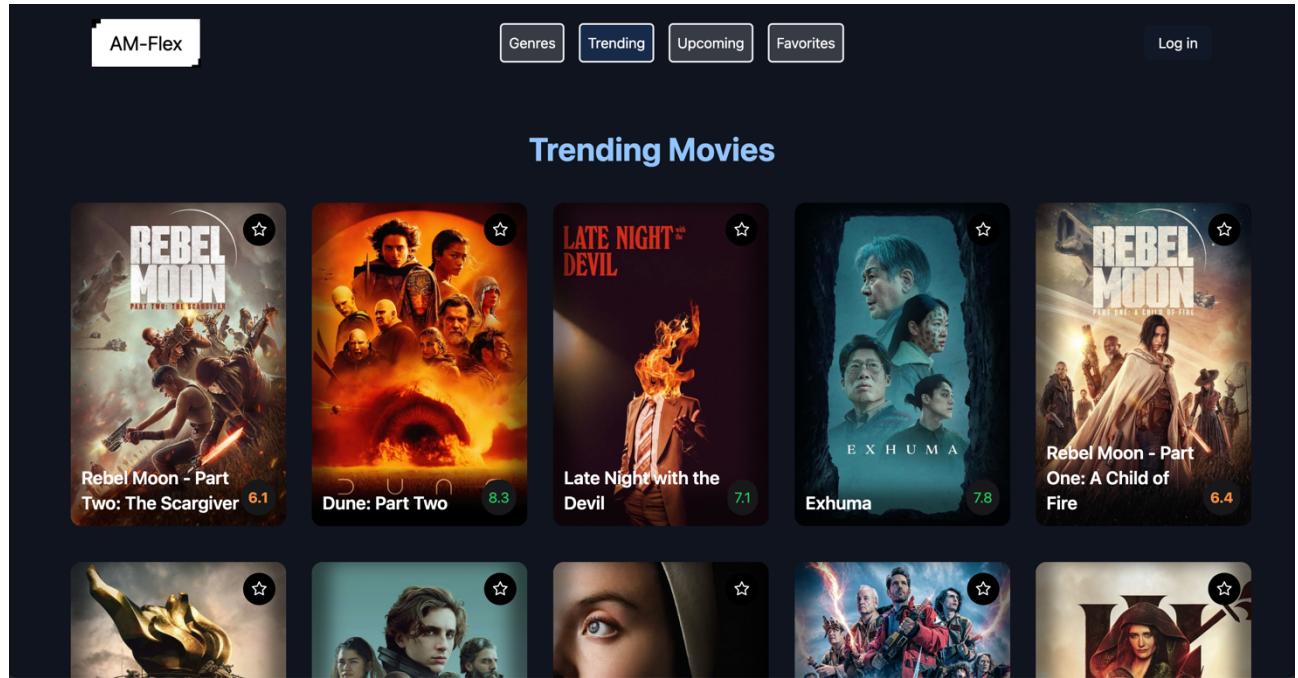


Figure 19. Trending page

The above figure shown is the trending page. This page provides the trending movies fetched from the API. It renders all the trending items. The favourite marking is also provided which helps you to bookmark your favourite ones. The marked movies are then stored locally and displayed in the Favourite page. User has to login to mark favourites.

The set of trending data is fetched from the TMDb API.

3.4.5 Upcoming Page

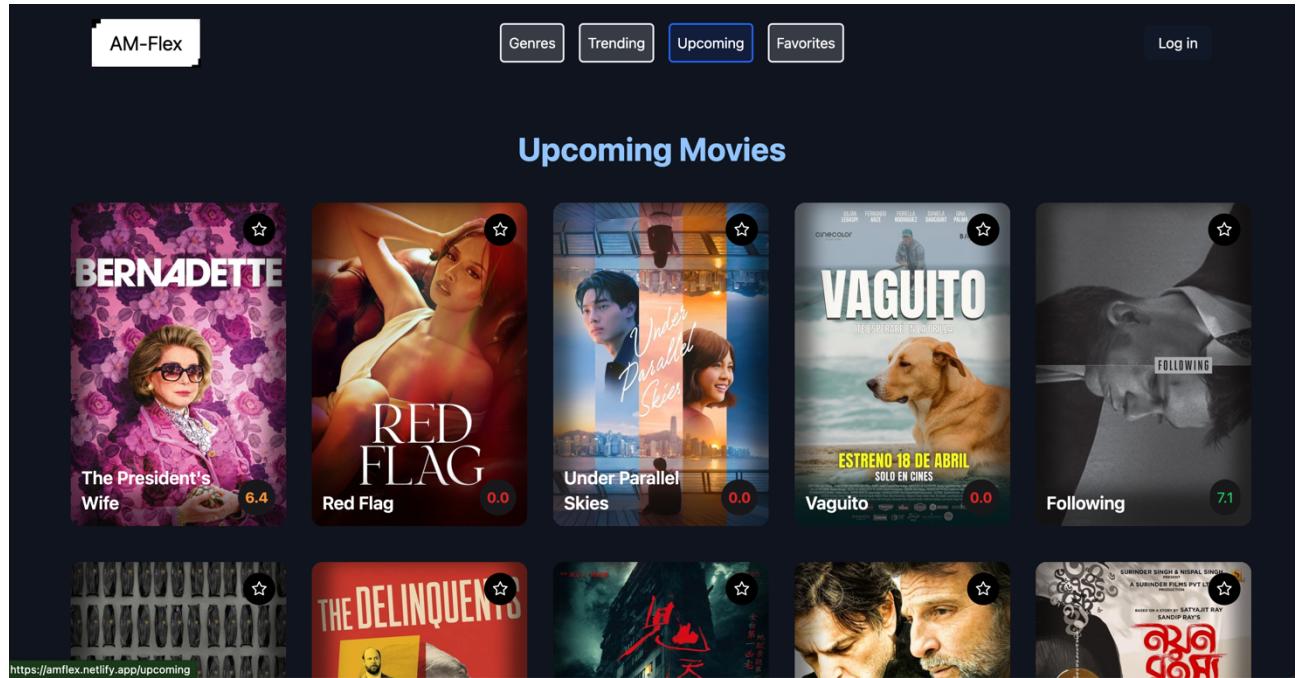


Figure 20. Upcoming Movies pages

The above figure shown is the page for upcoming movies. The upcoming movies are rendered and when clicked into the movie, the introduction, release date, cast information, and trailer are displayed on the movie page.

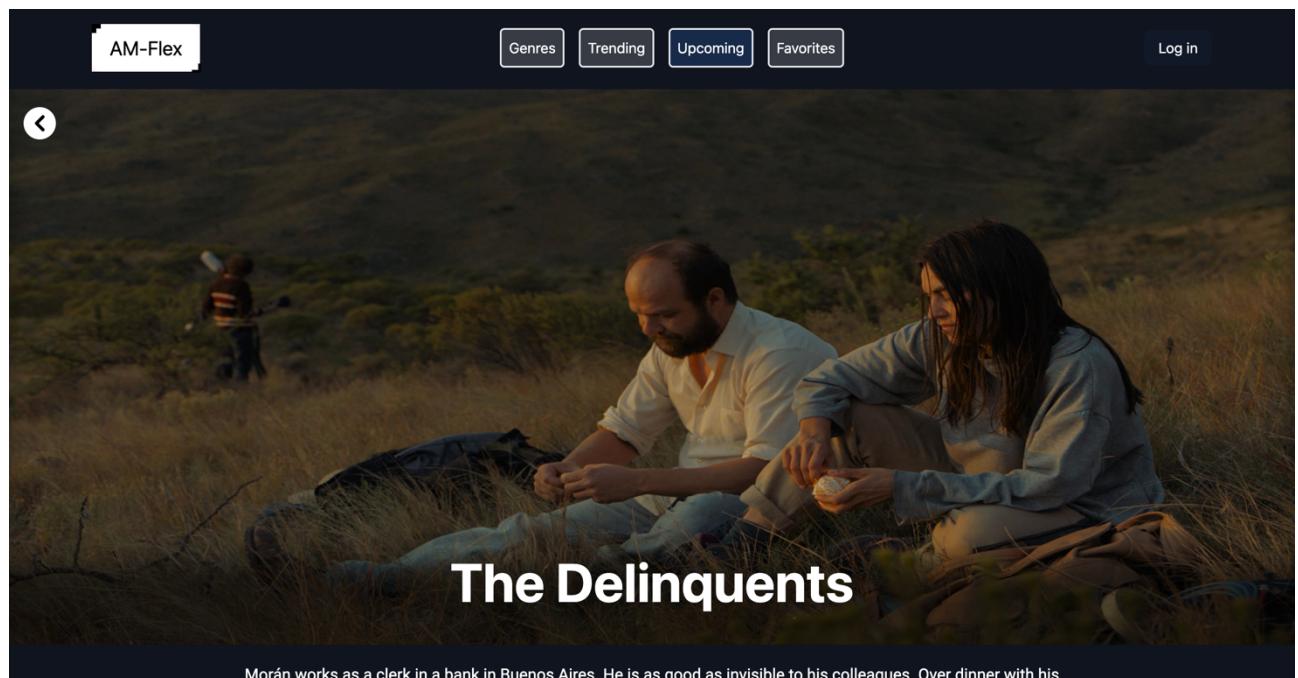


Figure 21. Movie Details page

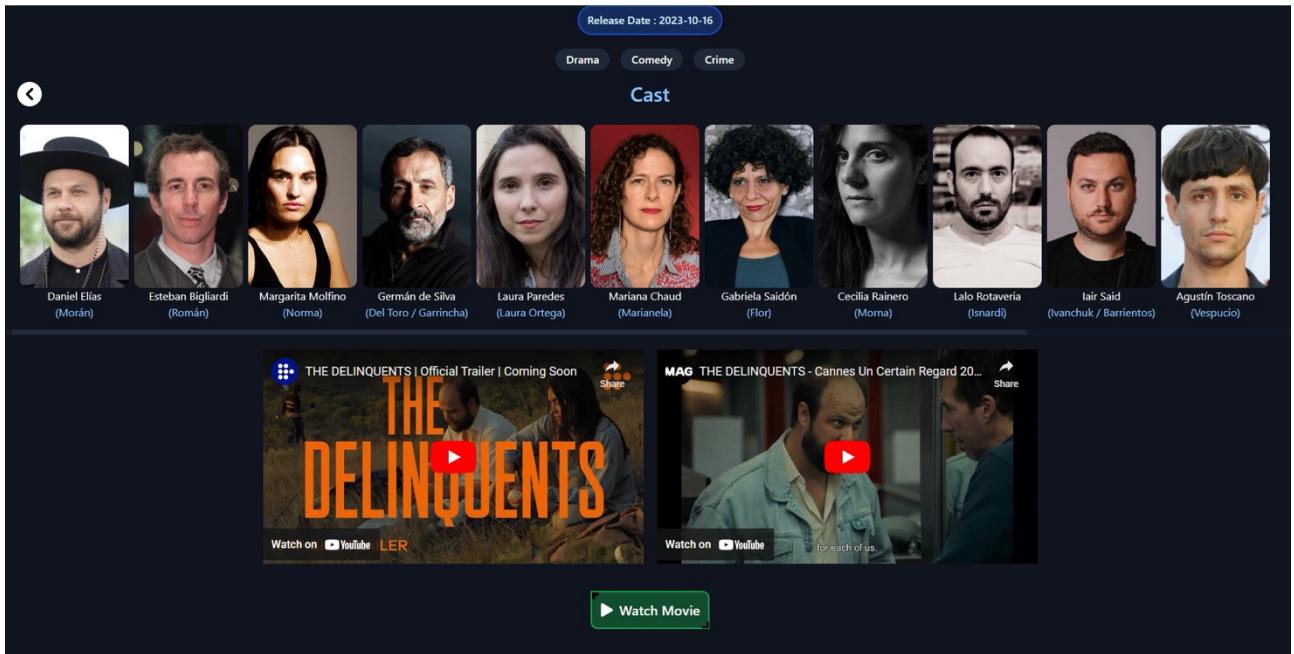


Figure 22. Movie Details page

The above figure 20 and 21 shows the movie detail page. This page provides detailed information about the movie along with the name of the movie in a large font. The details about the release date are also highlighted.

The page also contains the division to display the genre of the specific movie, along with the cast information in a carousel. The carousel is scrollable and renders all the cast information which are available.

The possible number official trailers are displayed in the player where the user can play the trailer on the same page. The trailer is fetched from YouTube. Finally, the button is placed which can okay the movie.

3.4.6 Favourites Page

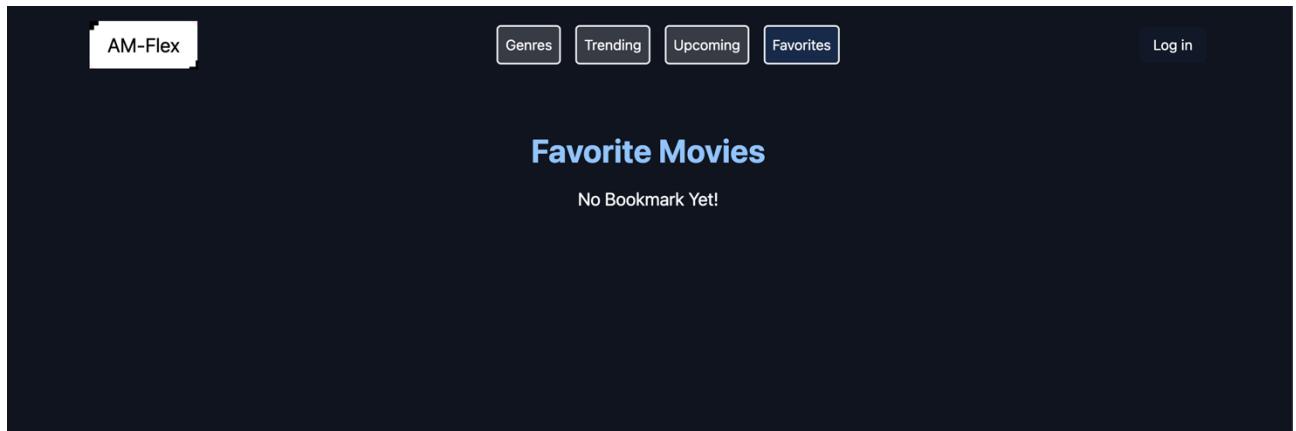


Figure 23 . Favourites page

The above figure shows the Favourite page which displays the favourite marked movies. The bookmarks can be marked only if the user is logged in. The user can mark the favourites by pressing the start button from the movie.

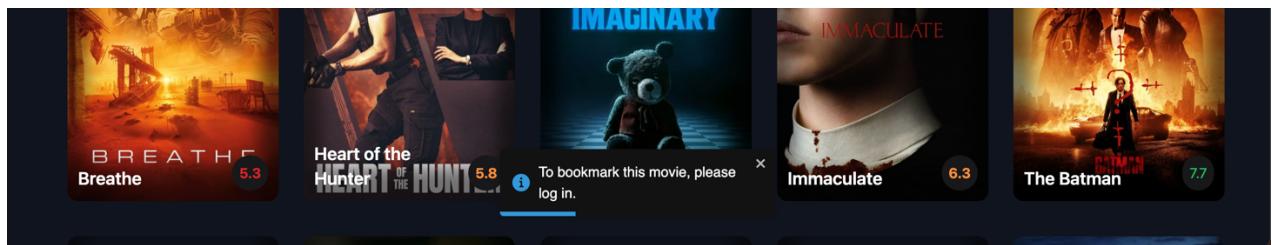


Figure 24. Login to mark favourite

The above figure shows that the movie cannot be marked favourite if the user is not logged in.

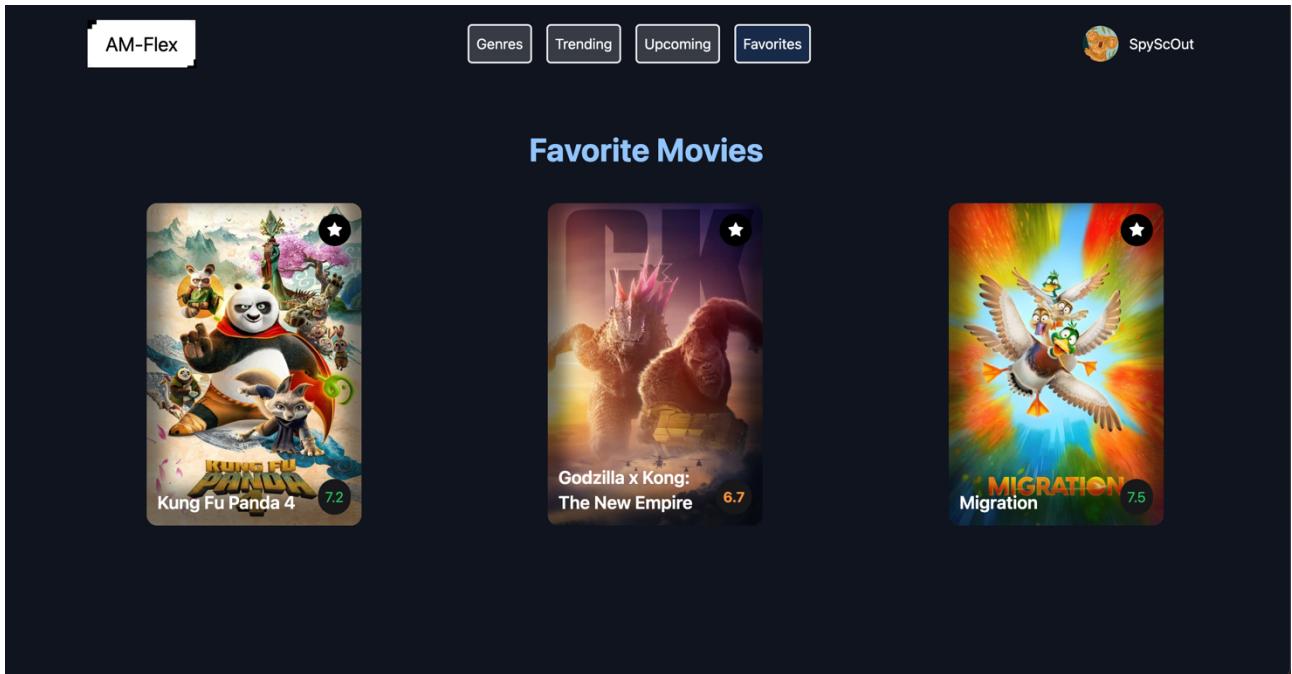


Figure 25. Favourites Movie page

The above figure shows the favourites page. All the marked movies are rendered in this page. The favourites are saved in the local storage temporarily.

3.4.7 Login

The user needs to login in order to mark the favourite movies. The firebase google authentication has been used as an authentication. When the user clicks login, user is prompted to google for the authentication.

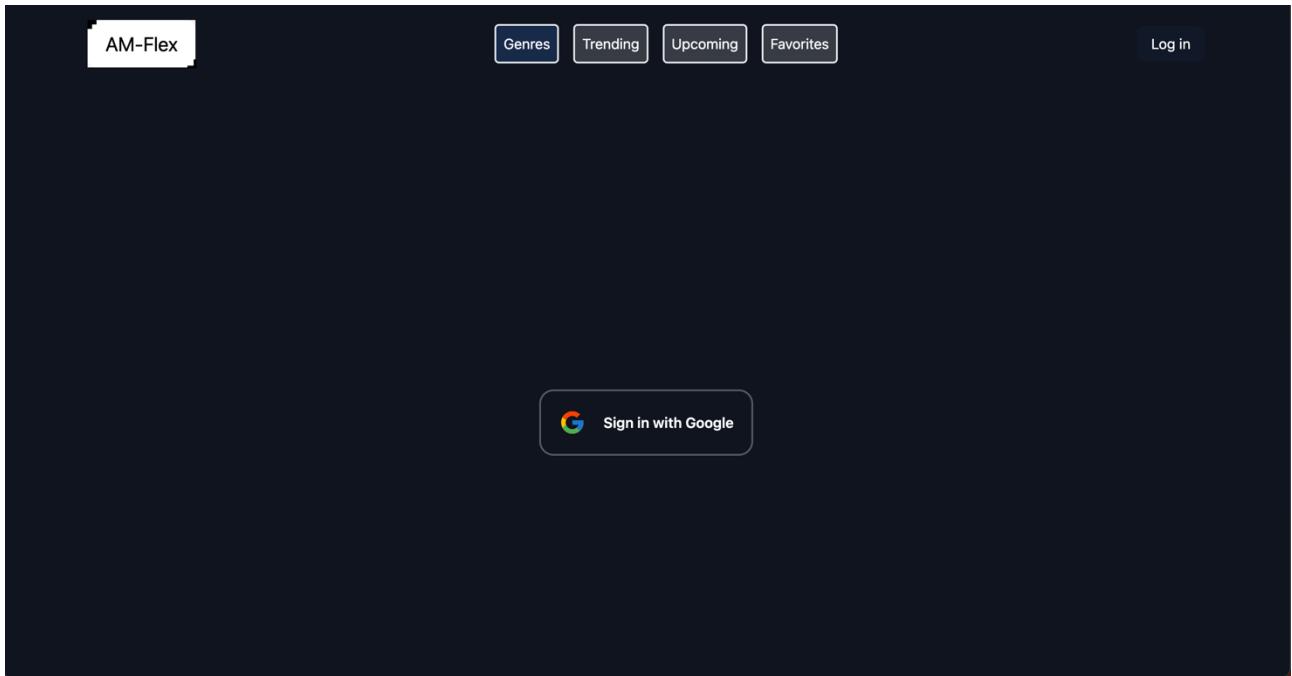


Figure 26. Google Authentication

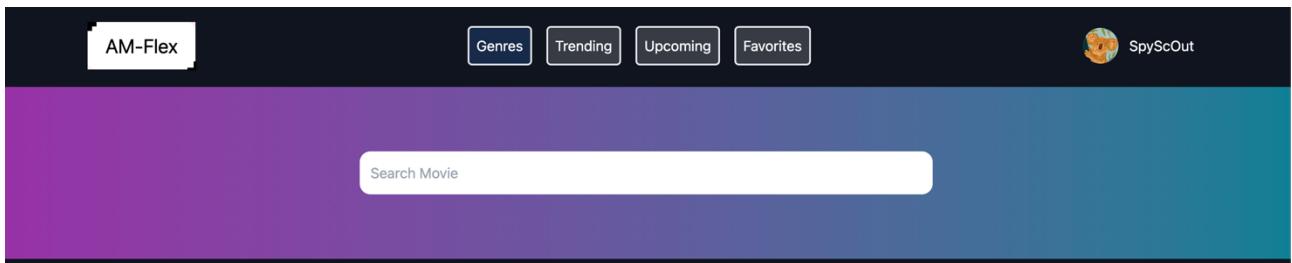


Figure 27. Logged in status with profile picture and username

The above figure illustrates the profile image of the user when logged in. The user has been logged in through google and the profile picture is also displayed from the same google account. The name of the google account is also displayed along the profile picture. The login button will be rendered after the user logs out.

4 RESULTS

The final product was developed after some theoretical studies and development. The product was a web application for movies. The main goal of the project was achieved. The user-friendly application was tested successfully both on mobile phone and desktop.

The outcome was a very user-friendly, fast, and simple movie web application. The platform was designed for desktop and mobile which was built using React and its frameworks and utilizing the TMDb API for the data retrieval.

User were able to perform the tasks which were meant to be developed in the web application. Some weeks of development along with some theoretical knowledge were gained using some online platform websites. Some of the react frameworks were used during the development process. Firebase was used for Google authentication.

The web app was then finally deployed using Netlify and is available at: <https://amflex.netlify.app/>. The deployment of the web application in Netlify has provided with accessibility throughout different devices remotely. Since the web application was designed for mobile and desktop, users can access the platform anywhere, anytime.

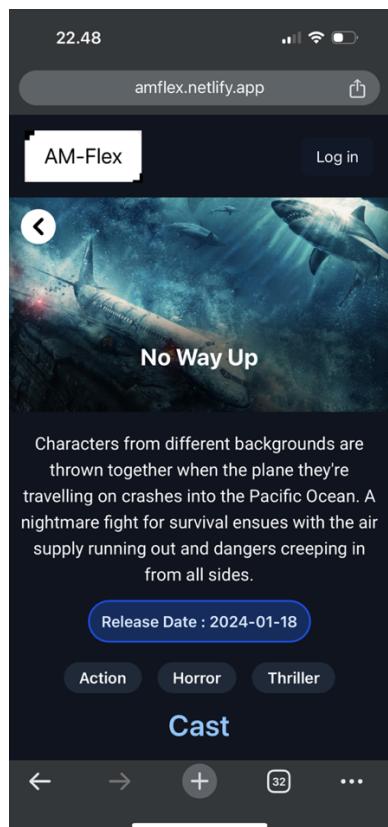


Figure 28. Mobile view

5 CONCLUSION

The main objective of this thesis was to implement and design a web application using react and TMDb API. The overall objective of the thesis was achieved although there occurred many problems in the development phase.

Expected feature was developed in the web application. For the development of the app, many documentations were researched and studied by the author as many of the react frameworks were new to the author. The primary focus was to be able to build a platform leading to a movie application where user can explore different kinds of movies and mark their favourites.

The application was built very simple with not many pages and routings to make it easier for the user to find the data they are searching for and according to the suggestions and recommendations. In case any user wants to explore any movies according to genres and trending, there are pages where you can find lists of movies according to the genres, trending, and upcoming movies. The app was designed for both mobile device and desktop making it accessible at any time.

The development was not as expected in the beginning as the author had to go through a lot of frameworks which were completely new to author and had to undergo research using the official documentations. Many other problems occurred during the development like errors in code, outdated frameworks, limited time and some self-employed project errors like lack of time management, but these problems were considered, and solutions were found after reading some well-prepared official documentations and YouTube videos which were also used for the information purposes.

After the completion of the thesis, author had gained some valuable information and skills to develop a frontend app using React and its libraries. React provides many different frameworks and libraries where development of a web app can be easier than the previous development methods. TMDb API was used to fetch the data freely as it was just used for the personal project.

The project also helped the author to gain some life long skills which can be used during the professional work life. The best practices for software development was very new to author but later with the help of online resources and materials, author was able to follow the procedures and tailor them in the project.

Further, the author had gained some comprehensive experience during the project which provided invaluable skills and enhanced the author's confidence. This skill can be applied further in other projects. The end product did not just provide the author with web application but a valuable skill and confidence was earned.

6 REFERENCES

Atlassian. (2024). *Atlassian*. Retrieved from www.atlassian.com:

<https://www.atlassian.com/git/tutorials/what-is-git>

Babel. (2024). *babeljs*. Retrieved from babeljs: <https://babeljs.io/docs>

Deshpande, C. (2023). <https://www.simplilearn.com/>. Retrieved from Simplilearn:

<https://www.simplilearn.com/>

Docs, M. W. (2024). *MDN Web Docs*. Retrieved from <https://developer.mozilla.org/>:

<https://developer.mozilla.org/>

Docs, M. W. (2024). *Mozilla*. Retrieved from <https://developer.mozilla.org/en-US/>:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function

Dwivedi, A. (2023). *Scaler Topics*. Retrieved from scaler.com:

<https://www.scaler.com/topics/react/react-helmet/>

Educative.io. (n.d.). *educative.io*. Retrieved from Educative:

<https://www.educative.io/courses/movie-database-api-python/introduction-to-the-movie-database-api>

Firebase. (2023). *twitter.com*. Retrieved from Twitter:

<https://twitter.com/Firebase/status/1699515319482515953/photo/1>

Git. (n.d.). *git*. Retrieved from git: <https://git-scm.com/>

Mustafeez, A. Z. (2024). *Educative.io*. Retrieved from <https://www.educative.io/answers/what-is-visual-studio-code>

Node.js. (n.d.). *Node.js*. Retrieved from nodejs: <https://nodejs.org/api/esm.html#introduction>

Petersyan, A. (2024). *Statista*. Retrieved from Statista:

<https://www.statista.com/statistics/617136/digital-population-worldwide/>

Statista. (2023). *Statista*. Retrieved from <https://www.statista.com/>:

<https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>

Stevenson, D. (2018). *Medium*. Retrieved from medium.com: <https://medium.com.firebaseio-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>

TMDb. (2024). Retrieved from <https://api.themoviedb.org/>:
<https://api.themoviedb.org/3/search/multi>

Vite. (n.d.). *Vite*. Retrieved from Vite: <https://vitejs.dev/guide/>

W3Schools. (n.d.). *w3schools*. Retrieved from w3schools.com:
https://www.w3schools.com/whatis/whatis_html.asp