Project Report

On

“FilmHive Movie App”

Under subject of

MAJOR PROJECT

B.Tech, Semester – VIII

(Department of Information Technology)

Submitted by:

1. Vishal Vansjariya 190004047

Prof. Piyush Kashiyani

(Faculty Guide)

Prof. Darshan Jani

(Head of the Department)

Academic Year

(2022-23)

CANDIDATE'S DECLARATION

We hereby declare that the work presented in this project entitled "**FilmHive Movie App"** submitted towards completion of project in **8th Semester** of B.Tech. (Information Technology) is an authentic record of our original work carried out under the guidance of **Prof. Piyush Kashiyani.**

We have not submitted the matter embodied in this project for the award of any other degree.

Semester: 8th
Place: Atmiya University, Rajkot

Signature:
Vishal Vansjariya (190004047)

CERTIFICATE

Date: **05/04/2023**

This is to certify that the "**FilmHive Movie App**" has been carried out by **Vishal Vansjariya** under my guidance in fulfillment of the subject Major Project in Information Technology (8th Semester) of Atmiya University, Rajkot during the academic year 2022-23.

Prof. Piyush Kashiyani
(Project Guide)

Prof. Darshan Jani
(Head of the Department)

ACKNOWLEDGEMENT

We have taken many efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to Prof. Piyush Kashiyani Sir for their guidance and constant supervision as well as for providing necessary information regarding the Major Project titled **"FilmHive Movie App".** We would like to express our gratitude towards staff members of Information Technology Department, Atmiya University for their kind co- operation and encouragement which helped us in completion of this project.

We even thank and appreciate to our colleague in developing the project and people who have willingly helped us out with their abilities.

Vishal Vansjariya(190004047)

ABSTRACT

This Report describes all the requirements for the project. The purpose of this research is to provide a virtual image for the combination of both structured and unstructured information of my project “FilmHive Movie App”. This is an online Movie and Series website for the check rating and information about trending movies and TV shows. The user can see different kind of movies, TV Series around world in the websites, though the website user can search movies and TV Shows from 1M large set of movies data. Also filtering the data by different type of genre be so easy for a find person choice of movies.

INDEX

1. Purpose :

The "FilmHive Movie App" project is a web-based application developed using the React framework and Material UI library. The application is designed to allow users to manage and display various types of entertainment content, including movies and TV series. The project uses the TMDB API to fetch the entertainment content and displays it in an organized manner. The application is designed to be a Progressive Web App (PWA) and has four main pages, including Trending, Movies, TV Series, and Search.

2. Scope :

In the fast-growing field of software engineering and development and even more rapidly growing sector of website development the future is hard to predict. In general software project is a project focusing on the creation of software. Consequently, Success can be measured by taking a look at the resulting software. In a website project, the product is a website. But and here comes the point: A website is much more than just its software. It has to provide content to become enjoyable.

3. Technology and Tools :

3.1 HTML

HTML (HyperText Markup Language) is a markup language used to create and structure content on the web. It is the backbone of any web page and serves as a building block for other web technologies such as CSS and JavaScript.

HTML is based on a system of tags and attributes that define the structure and appearance of web content. These tags are placed within the HTML document and are used to define elements such as headings, paragraphs, lists, images, and links. Attributes provide additional information about an element, such as its size, color, or location on the page.

HTML is a flexible and powerful language that allows developers to create complex web pages with ease. It has evolved over the years, with new versions and features being introduced regularly. HTML5, the latest version of HTML, includes several new elements and attributes that allow for more dynamic and interactive web content.

One of the key advantages of HTML is its simplicity and ease of use. The language is relatively easy to learn and can be mastered with just a basic understanding of its syntax and structure. HTML also provides a high degree of flexibility, allowing developers to create web pages that are tailored to the needs of their users.

HTML is supported by all modern web browsers, making it a universal language for the web. This ensures that web pages created using HTML can be viewed by anyone, regardless of their device or operating system.

In addition to its basic syntax and structure, HTML also supports a wide range of features and techniques that allow for more advanced web development. These include CSS (Cascading Style Sheets), which is used to control the appearance and layout of web content, and JavaScript, which is used to add interactivity and dynamic functionality to web pages. Overall, HTML is an essential tool for web developers and designers. Its flexibility, ease of use, and universal support make it a critical building block for creating modern, dynamic, and interactive web content.

3.2 CSS

CSS (Cascading Style Sheets) is a styling language used to define the appearance and layout of web content. It is used in conjunction with HTML and JavaScript to create visually appealing and interactive web pages.

CSS is a powerful language that allows developers to control every aspect of a web page's visual design, including color, font, layout, and animation. With CSS, developers can create responsive, mobile-friendly designs that adapt to different screen sizes and device types.

CSS works by using selectors to target specific HTML elements and applying style rules to them. For example, a developer might use a selector to target all the paragraph elements on a page and then apply a rule that sets their font size to 16 pixels. CSS also supports cascading, which means that if conflicting style rules are applied to the same element, the most specific rule will be used.

One of the key benefits of CSS is its ability to separate content and presentation. This means that developers can create well-structured, semantically meaningful HTML documents and then apply CSS styles to them to control their appearance. This makes it easier to maintain and update web pages, as changes can be made to the CSS without affecting the underlying HTML content.

CSS has evolved over the years, with new features and techniques being added to the language. For example, CSS3 introduced support for advanced layout techniques such as flexbox and grid, as well as support for animations and transitions.

In addition to its core syntax and features, CSS also supports a wide range of tools and frameworks that make it easier for developers to create complex designs and layouts. These include preprocessor languages such as Sass and Less, which provide additional functionality and features, as well as CSS frameworks such as Bootstrap and Foundation, which provide pre-built CSS styles and components that can be used to quickly create responsive web designs.

Overall, CSS is a powerful and essential tool for web developers and designers. Its ability to control the visual appearance and layout of web content makes it a critical component of modern web development, and its flexibility and versatility ensure that it will continue to be a key technology for years to come.

3.3 Javascript

JavaScript is a high-level, dynamic programming language used to create dynamic and interactive web content. It is used in conjunction with HTML and CSS to create modern, dynamic web pages that can respond to user input and provide a more engaging user experience.

JavaScript is a versatile and powerful language that can be used for a wide range of web development tasks, from simple form validation to complex web applications. It is a client-side language, meaning that it runs on the user's browser rather than on the server, allowing for faster and more responsive web applications.

One of the key features of JavaScript is its ability to manipulate the Document Object Model (DOM), which represents the structure of an HTML document. With JavaScript, developers can dynamically modify the content and structure of a web page in response to user input or other events, such as page load or mouse clicks.

JavaScript is also used to create interactive user interfaces, such as drop down menus, sliders, and forms. It can be used to create animations and visual effects, as well as to load and display data from external sources, such as APIs or databases.

In addition to its core features, JavaScript also supports a wide range of frameworks and libraries that make it easier to develop complex web applications. These include popular frameworks such as React, Angular, and Vue, as well as libraries such as jQuery, which simplifies DOM manipulation and event handling.

JavaScript has evolved over the years, with new features and capabilities being added regularly. ECMAScript, the standardized version of JavaScript, is updated regularly, with new versions introducing new syntax and features such as arrow functions, async/await, and modules.

Overall, JavaScript is a powerful and essential tool for web developers. Its ability to create dynamic and interactive web content, along with its versatility and ease of use, make it a critical component of modern web development.

3.4 React

React is an open-source JavaScript library used for building user interfaces for web applications. It was developed by Facebook and is now widely used by developers all over the world to create complex and dynamic applications.

React is known for its component-based architecture, which allows developers to break down complex user interfaces into smaller, reusable components. This makes it easier to build, maintain, and test large-scale applications.

React is designed to be declarative, meaning that developers can describe how a UI should look at any given point in time, and React takes care of updating the UI when necessary. This makes

it easier to create dynamic, responsive UIs that react to user input and changing data.
React uses a virtual DOM (Document Object Model), which is a lightweight representation of the actual DOM. When data changes, React updates the virtual DOM, and then uses a process called reconciliation to efficiently update only the parts of the actual DOM that have changed. This makes React much faster than traditional DOM manipulation, and allows for a smoother, more seamless user experience.

React is also highly customizable, with a wide range of third-party libraries and plugins available to enhance its functionality. For example, developers can use React Router to add routing to their applications, or Redux to manage application state.

One of the main advantages of React is its strong community and ecosystem. There are countless resources available online, including tutorials, documentation, and libraries, making it easier for developers to learn and use React effectively.

Overall, React is a powerful and popular tool for building modern web applications. Its component-based architecture, declarative syntax, and virtual DOM make it easier to build complex UIs, while its strong community and ecosystem make it easier to learn and use effectively.

3.5 Material UI

Material UI is a popular open-source library of React components for building user interfaces. It is based on Google's Material Design guidelines and provides a wide range of reusable UI components that can be easily customized to fit the needs of individual projects.

Material UI provides a consistent, modern look and feel for web applications, with a focus on clean, responsive design. It includes a wide range of components, from basic elements like

buttons and forms to more complex components like navigation menus and data tables.
One of the key benefits of Material UI is its flexibility and ease of use. The library includes pre-built components with a wide range of customization options, making it easy to build complex user interfaces without having to write a lot of custom CSS. It also includes a theme system that allows developers to easily customize the colors, typography, and other design elements of their application.

Material UI is also highly modular, with each component designed to be independent and reusable. This makes it easy to integrate Material UI components into existing projects, or to create new projects from scratch.

Another advantage of Material UI is its strong community support. The library is widely used and has a large and active community of developers who contribute to its development and provide support through forums, documentation, and other resources.

Overall, Material UI is a powerful and versatile library for building modern, responsive user interfaces for web applications. Its focus on clean, consistent design and ease of use make it a popular choice for developers looking to create high-quality, customizable user interfaces.

3.6 Axios

Axios is a popular open-source library used by developers to make HTTP requests from JavaScript applications. It is a lightweight and efficient library that simplifies the process of making HTTP requests by providing a simple and consistent API.

Axios supports all major browsers and can be used both on the client-side and server-side of web applications. It can handle

complex requests, such as sending and receiving data in different formats like JSON, FormData, and binary data. It also supports common HTTP methods like GET, POST, PUT, and DELETE.

Axios is easy to use and requires minimal configuration. To make a request, developers simply need to import the Axios library, create an instance of the Axios client, and call the appropriate method, passing in the necessary data and configuration options.

Axios also provides several useful features, such as request and response interceptors, which can be used to modify requests and responses at runtime. It also supports canceling requests, handling errors, and automatic retries.

Axios can be used with different types of APIs, including REST APIs, GraphQL APIs, and others. It can also be integrated with other libraries and frameworks, such as React, Angular, and Vue.js.

In summary, Axios is a powerful and easy-to-use library that simplifies the process of making HTTP requests from JavaScript applications. Its simple API, cross-browser compatibility, and robust feature set make it a popular choice for developers working on a wide range of web applications.

3.7 API

An API (Application Programming Interface) is a set of protocols, tools, and standards that define how different software components can interact with each other. In general, an API is a set of methods and communication protocols that enable two different applications to communicate with each other and share data or functionality.

APIs are used for a variety of purposes, including:

1. Building applications: Developers use APIs to access data and functionality provided by other applications or services. This can include retrieving information from a database, interacting with a payment gateway, or integrating with a social media platform.

2. Integrating systems: APIs can be used to connect different systems or applications, allowing them to share data and functionality. For example, an API can be used to integrate an e-commerce website with a shipping provider, allowing customers to track their orders in real-time.

3.Building libraries and frameworks: APIs can be used to build libraries and frameworks that can be used by other developers to build their own applications. This can help to standardize development practices and improve code quality.

There are many types of APIs, including:
1. REST APIs: Representational State Transfer (REST) APIs are a popular type of API that uses HTTP requests to retrieve or manipulate data. REST APIs are often used for web applications and mobile applications.

2. SOAP APIs: Simple Object Access Protocol (SOAP) APIs are a type of API that uses XML to exchange data between different applications.

3. GraphQL APIs: GraphQL APIs are a newer type of API that provides a flexible and efficient way to retrieve data. Unlike REST APIs, which often require multiple requests to retrieve all the necessary data, GraphQL APIs allow developers to retrieve all the necessary data in a single request.

Overall, APIs are an essential tool for modern software development. They allow developers to build more complex and powerful applications by leveraging data and functionality from other applications and services.

3.8 TMDB API

The TMDB API (The Movie Database API) is a powerful API that provides developers with access to a vast collection of movie and TV show data. The API is maintained by The Movie Database (TMDB), which is a community-built movie and TV database that contains information about millions of movies, TV shows, and celebrities.

Developers can use the TMDB API to retrieve information about movies and TV shows, including titles, posters, trailers, release dates, ratings, and more. The API also allows developers to search for movies and TV shows based on a variety of criteria, such as genre, release year, and popularity.

The TMDB API is designed to be easy to use and includes extensive documentation and code examples. Developers can use a variety of programming languages and tools to interact with the API, including Python, Java, Node.js, and many others. The API also includes SDKs (Software Development Kits) for popular programming languages, which can help to simplify the process of integrating with the API.

In addition to providing movie and TV show data, the TMDB API also provides access to other types of data, such as movie credits, images, and reviews. This makes it a powerful tool for building applications that require detailed movie and TV show information.

Overall, the TMDB API is a valuable resource for developers looking to build applications that incorporate movie and TV show data. Its extensive collection of data, easy-to-use interface, and comprehensive documentation make it a popular choice for developers working on a wide range of projects, from mobile apps to web applications.

3.9 React Router DOM

React Router is a popular library for managing client-side routing in React applications. It provides a declarative way to define and navigate between different pages or views within a single-page application.

React Router DOM is a package that builds on top of React Router, providing additional tools and components for building web applications that run in a browser. It includes components like BrowserRouter, Switch, Route, and Link, which enable developers to create complex routing logic in a declarative and intuitive way.

The BrowserRouter component provides a way to define the root URL for your application and handle browser history. The Switch component is used to group Route components together and ensure that only one matches at a time. The Route component is used to define a route, which matches a specific path and renders a corresponding component. The Link component is used to create links between different routes within the application.

React Router DOM also provides features like nested routes, route parameters, and route guards, which enable developers to create complex routing scenarios. For example, developers can create a route with a dynamic parameter, such as /users/:id, which matches any URL that starts with /users/ and extracts the id parameter from the path.

React Router DOM also integrates well with other React libraries and tools, such as Redux and Material UI. It provides a flexible and extensible way to handle navigation and state management in modern web applications.

Overall, React Router DOM is a powerful and popular library for managing client-side routing in React applications. Its declarative syntax, intuitive components, and wide range of features make it easy for developers to create complex and dynamic routing scenarios, while also integrating seamlessly with other React libraries and tools.

3.10 React Alice Carousel

React Alice Carousel is a popular lightweight and responsive carousel component for React applications. It is built on top of the popular Alice Carousel library, and provides a simple and easy-to-use interface for creating carousels that can display images, videos, or any other content.

One of the key features of React Alice Carousel is its flexibility and customization options. Developers can customize the size, style, and behavior of the carousel by passing in props, including options for auto-playing, infinite looping, and responsive design. It also supports touch events, which allows users to swipe through the carousel on mobile devices.

React Alice Carousel also includes a range of callbacks and events that developers can use to handle user interactions, such as when a slide is changed, or when the carousel has reached the end of the slides. It also supports keyboard navigation, which allows users to navigate through the carousel using arrow keys. Another advantage of React Alice Carousel is its performance. It uses virtual rendering techniques to render only the visible slides, which makes it much faster and more efficient than other carousel libraries that render all slides at once. This is especially important for mobile devices, where performance is often a key consideration.

Overall, React Alice Carousel is a versatile and powerful carousel component for React applications. Its flexibility, customization options, and performance make it a popular

choice for developers looking to create dynamic and engaging carousels for their web applications.

3.11 PWA

Progressive Web Apps (PWAs) are web applications that are designed to look and feel like native mobile apps, while also providing the benefits of being accessible via a web browser. PWAs offer users a more seamless experience by leveraging features such as offline access, push notifications, and the ability to install on the home screen of a mobile device.

One of the key advantages of PWAs is that they are platform-agnostic, meaning they can be used on any device or operating system, regardless of the manufacturer or version. They are built using web technologies such as HTML, CSS, and JavaScript, and are designed to be responsive and adaptable to any screen size.

PWAs can also be easily updated and deployed without the need for an app store review process, which can be time-consuming and restricts the flexibility of native apps. This allows developers to quickly iterate and improve their apps based on user feedback.

In addition, PWAs can be accessed via a web browser, eliminating the need for users to download and install an app from an app store. This can lead to a more streamlined and frictionless user experience, which can improve engagement and retention.

Another key feature of PWAs is their ability to provide offline access to users, which is especially important in areas with poor network connectivity or limited data plans. PWAs can store data locally on the device, allowing users to continue using the app even when they are not connected to the internet.

Finally, PWAs can also provide push notifications to users, allowing them to receive timely updates and reminders without having to actively open the app. This can help increase engagement and keep users coming back to the app.

Overall, PWAs offer a compelling alternative to native mobile apps, providing many of the same benefits while also leveraging the advantages of web technologies. As more developers embrace PWAs, we can expect to see an increasing number of high-quality, user-friendly web applications that provide a seamless experience across all devices and platforms.

4. Project Management :

4.1 Project Planning

Project Planning is concerned with identifying and measuring the activities, milestones and deliverables produced by the project. Project planning is undertaken and completed sometimes even before any development activity starts. Project planning consists of following essential activities:

• Scheduling manpower and other resources needed to develop the system.
• Staff organization and staffing plans.
• Risk identification, analysis, and accurate planning.
•Estimating some of the basic attributes of the project like cost, duration and efforts. The effectiveness of the subsequent planning activities is based on the accuracy of these estimations.
• Miscellaneous plans like quality assurance plan, configuration management plan, etc.

Project management involves planning, monitoring and control of the people, process, and the events that occurs as the software evolves from a preliminary concept to an operational implementation. Cost estimation is a relative activity that is

concerned with the resources required to accomplish the project plan.

4.2 Project Scheduling

The scheduling is the peak of a planning activity, a primary component of software project management. When combined with estimation methods and risk analysis, scheduling establishes a roadmap for project management. The characteristics of the project are used to adapt an appropriate task set for doing work.

4.3 Risk Management

Risk management consists of a series of steps that help a software development team to understood and manage uncertain problems that may arise during the course of software development and can plague a software project.

Risks are the dangerous conditions or potential problems for the system which may damage the system functionalities to very high level which would not be acceptable at any cost. So in order to make our system stable and give its 100% performance we must have identify those risks, analyze their occurrences and effects on our system and must prevent them to occur.

4.4 Identification

Risk identification is a first systematic attempt to specify risks to project plan, Scheduling resources, project development. It may be carried out as a team process using brainstorming approach

Technology risk:
Technical risks concern implementation, potential design, Interfacing, testing, and maintenance problems
• Database Corruptness
• Garbage Collection

People Risks:

These risks are concerns with the team and its members who are taking part in developing the system.
• Leaking an important data
• Failure of the administration
• Lack of knowledge ,
• Lack of clear product vision.
• Technical staff conflict
• Poor communication between people

Tools Risks:
These are more concerned with tools used to develop the system
• Tools containing virus.

General Risks:
General Risks are the risks, which are concerned with the mentality and resources.
• Lack of resources can cause great harm to efficiency and timely productivity.
• Rapidly changing requirements.
• Changes in requirements can cause a great harm to implementation, designing and schedule of developing the system.
• Insufficient planning and task identification.

4.5 Risk Analysis

“Risk analysis = risk assessment + risk management + risk communication.” Risk analysis is employed in its broadest sense to include:

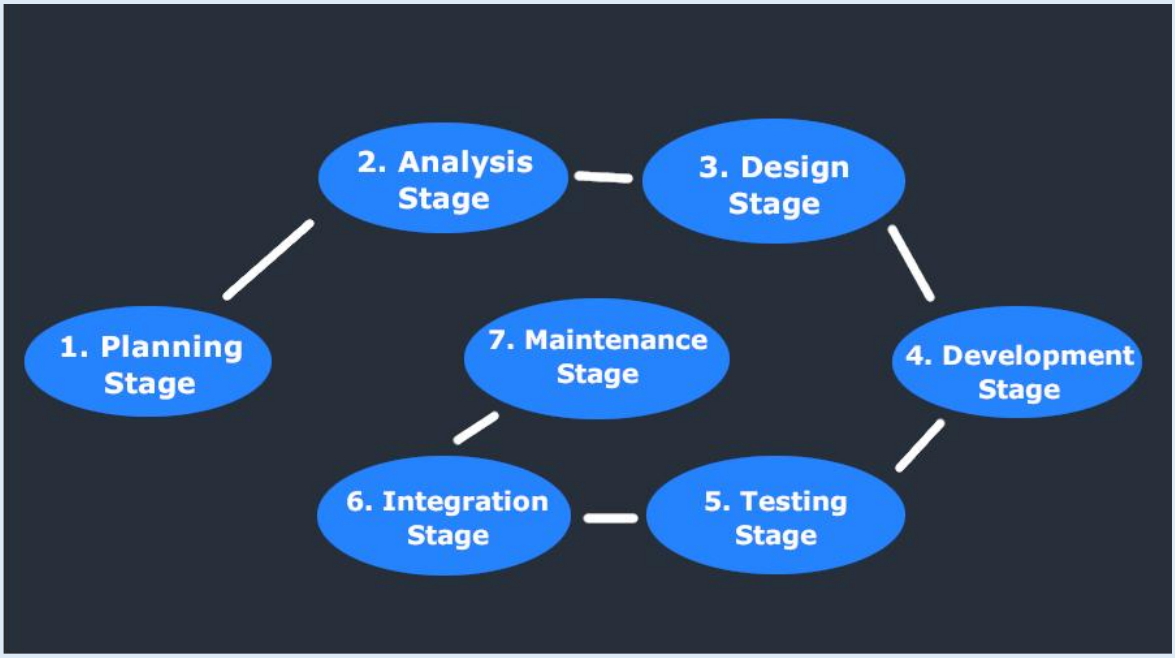
Risk assessment:
Involves identifying sources of potential harm, assessing the likelihood that harm will occur and the consequences if harm does occur.
For this project It might be :-

• System Crash.

Risk management:
Evaluates which risks identified in the risk assessment process require management and selects and implements the plans or actions that are required to ensure that those risks are controlled. Precautions taken to make risks minimal are as under:-
• Periodical backups are taken to avoid major loss in case of system crash.

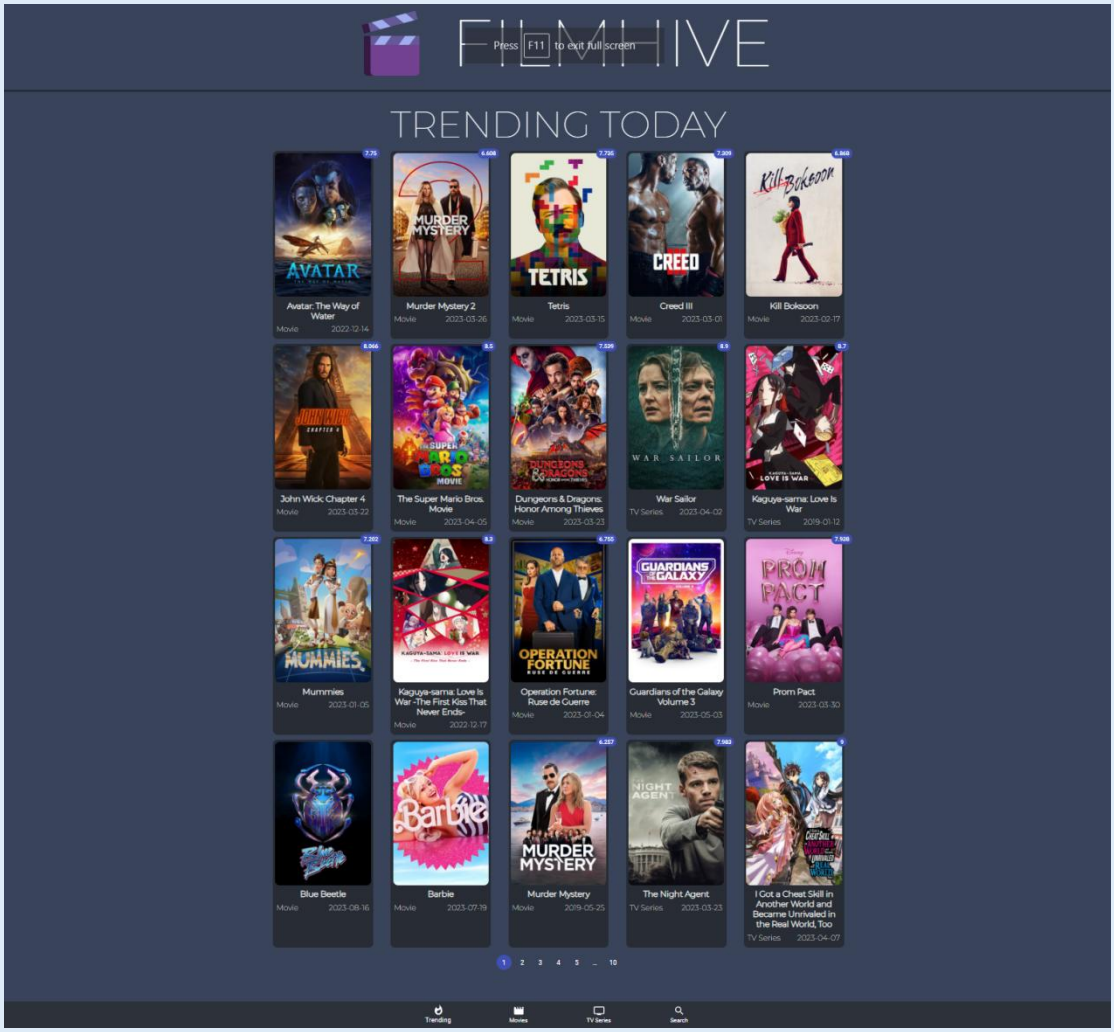
Risk communication:
Involves an interactive dialogue between stakeholders and risk assessors and risk managers which actively informs the other processes.
• Probability of certain risks is negotiated with client.
• All the possible risks are listed out during communication and project is developed taking care of that risks.

5. Software Development Life Cycle (SDLC)

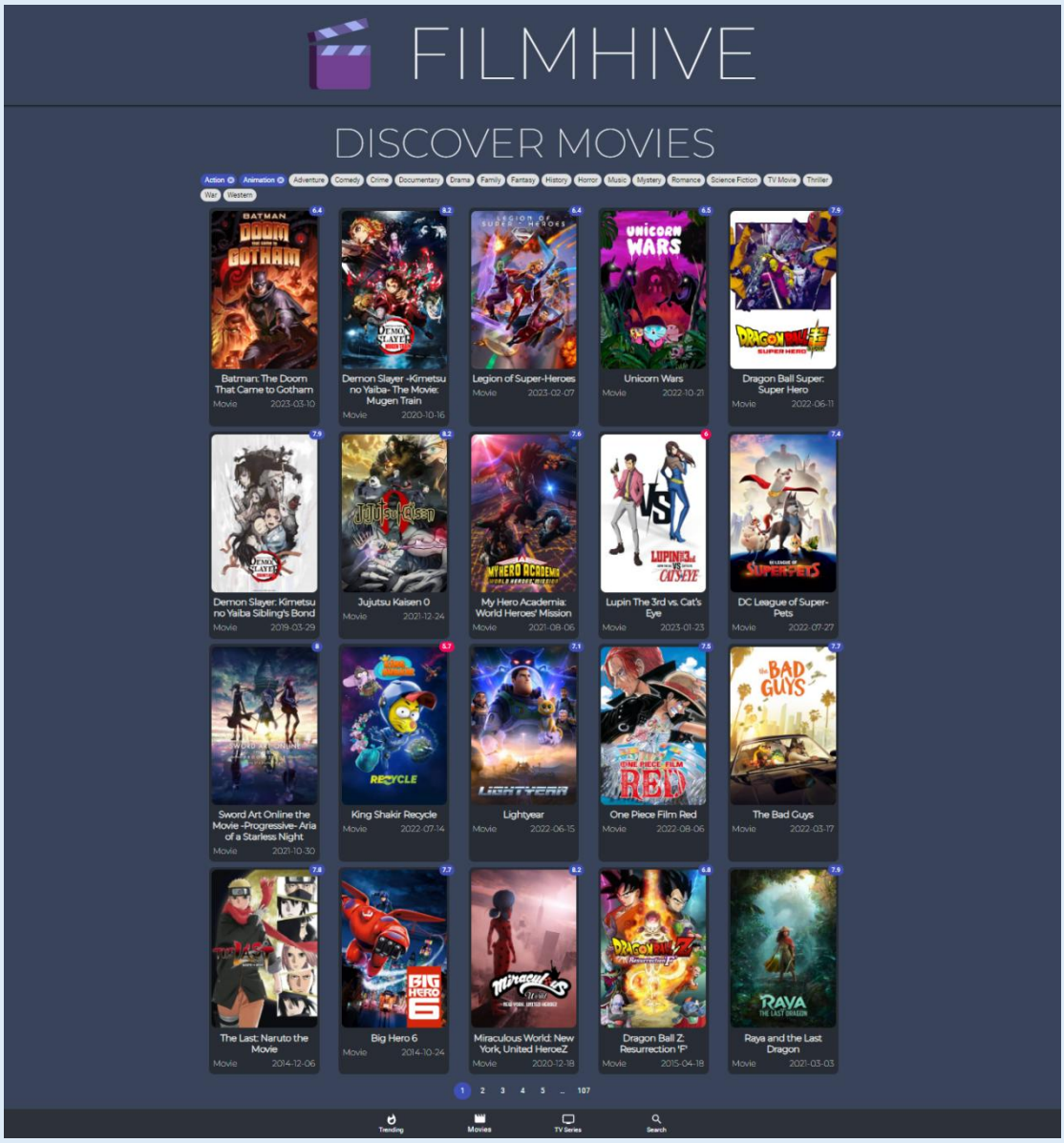


6. System Design

Page 1 : (Trending Page)

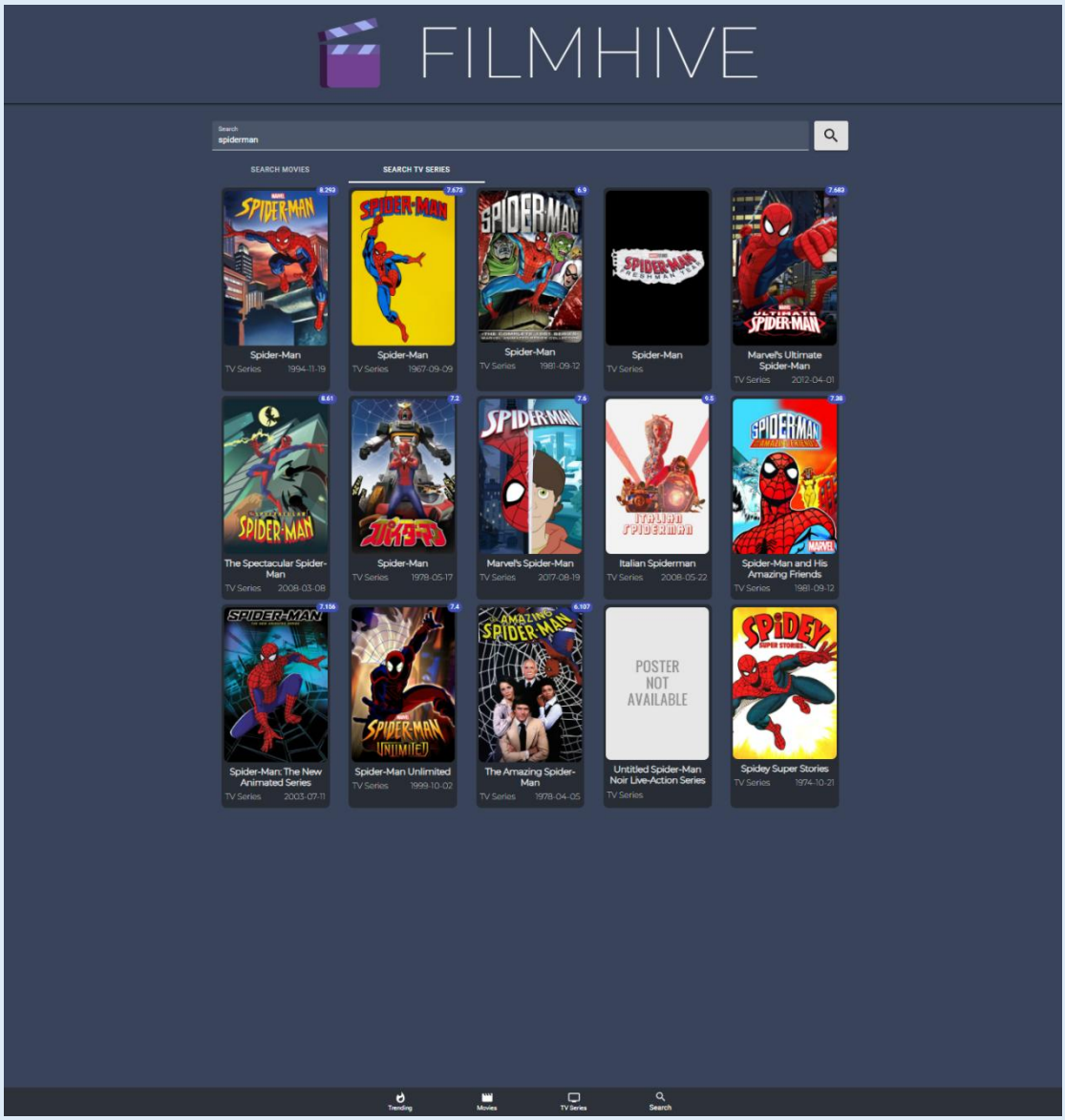


Page 2 : (Movies Page)

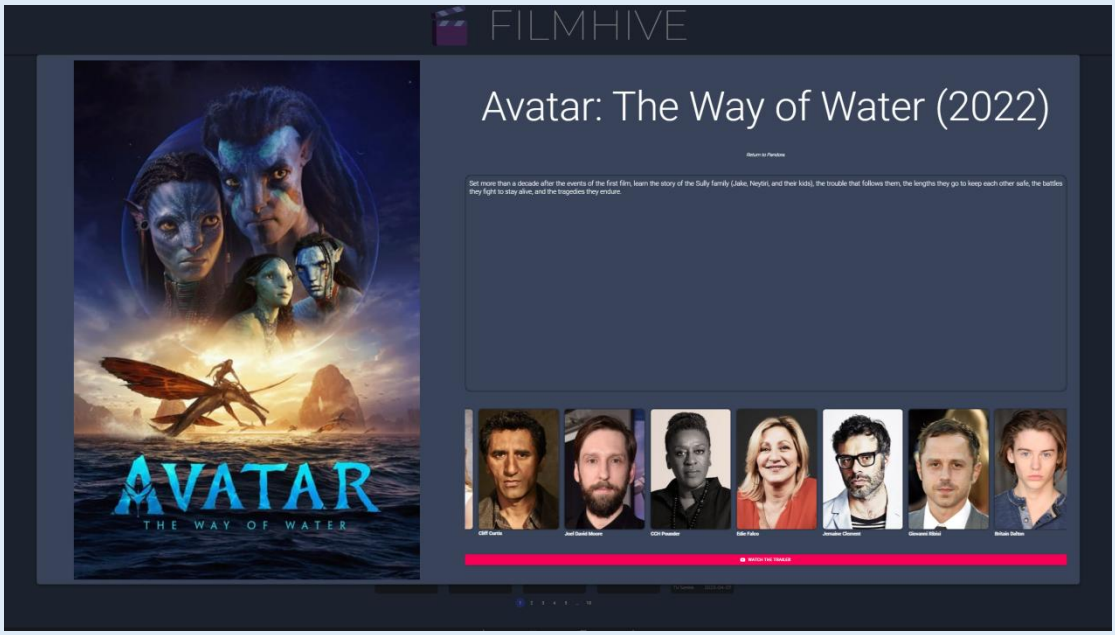


Page 3 : (TV Series Page)

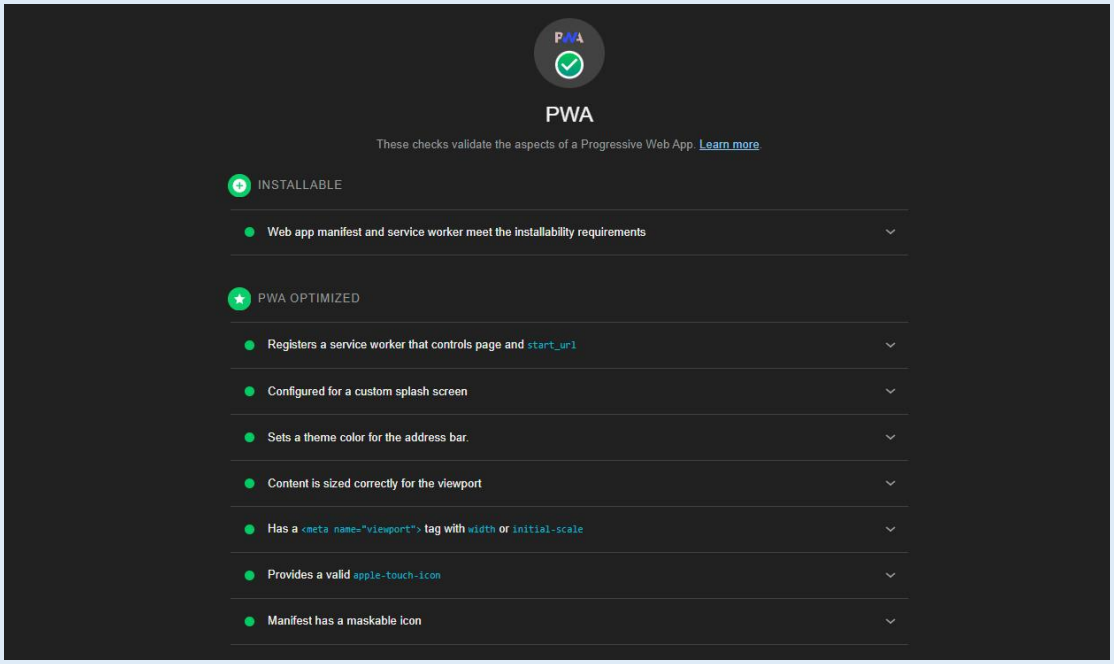
Page 4 : (Search Page)



Detail Page : (Popup Model)



Lighthouse PWA Result



7.Code Repo

Github : https://github.com/vishalk2512/FilmHIve
Live : https://vishal-filmhive.netlify.app/