StreamSavvy: Your UltimateEntertainment Destination(ReactApplication)

The StreamSavvy web application underscores your proficiency in crafting dynamic and interactive user interfaces using React. This project ingeniously integrates external APIs to retrieve a diverse array of information on both moviesand TV shows, presenting this data in an engagingand user-friendly manner.Leveraging React.js as the core technology for the front end, it adeptly manages state and efficiently renders components. React Router streamlines application navigation, enhancingthe user experience.

The project ensures a visually appealing and responsive design by employing CSS/SCSS for styling. HTTP requests to external APIs are facilitated using Axios. You have the flexibility to choose between React's built-in state management or employ state management libraries like Redux to effectively handle data flow and application state. This comprehensive undertaking not only underscores your mastery of React.js but also serves as a testament to your ability to create a fully functional and user-centric web application that genuinely enrichesusers' experiences by presenting data in an engaging and informative manner.

➤ Scenario based Use-case:

Sarah is a 26-year-old professional who loves spending her evenings watching StreamSavvy. After a long day at work, she likes to unwind with a good film or catch up on the latest episodes of her favorite series. She's always on the lookout for new recommendations and enjoys exploring differentgenres.

- **Planning**: Sarah starts her day with a cup of coffee and a quick browse through her favorite entertainment websites and apps. Today,she noticed a new movie review showcase called StreamSavvy mentioned on a forum she frequents.

- **Exploring StreamSavvy:** Intrigued, Sarah opens her web browser and navigates to the StreamSavvy website. The clean and modern designimmediately catches her eye, and she's excitedto explore further.

- **During the Day:** Browsing at Lunch: During her lunch break, Sarah pulls up StreamSavvy again on her phone. She starts by searching for some of her favoritemovies and TV shows. The search featurequickly retrieves detailed information, including ratings,reviews, and trailers, all in one place.

- **Discovering New Content**: Sarah decidesto browse throughthe recommended sectionto discover new content. She's impressed by the diverserange of recommendations tailored to her preferences, thanks to StreamSavvy's smart recommendation algorithm.

- **Evening Entertainment**: Planningthe Evening: As Sarah wraps up her workday, she opens StreamSavvy on her laptop to plan her evening entertainment. She checks the schedule feature to see if any new episodes of her favoriteTV shows are airing tonight.

- **Movie Night:** Tonight, Sarah decides to watch a movie she found on StreamSavvy's top-rated list. She clicks on the movie'spage to read more about it, watchesthe trailer, and reads a few user reviews to get a sense of what othersthink.

- **Engaging Experience:** Throughout the movie, Sarah keeps StreamSavvy open on her phone to rate the film and leave her review once it's over. She appreciates the interactive nature of the platform and enjoys contributing to the community.

- **Wrap-Up**: Reflecting on the Experience: After the movie ends, Sarah spends a few minutes browsing through StreamSavvy's news section to catch up on the latest entertainment industry updates and gossip. She'simpressed by the depth of content available on the platformand makes a mental note to visit again tomorrow.

- **Recommendation Sharing**: Beforebed, Sarah sends a link to StreamSavvy to her friendswho share her love for movies and TV shows,recommending it as a must-visit for anyone lookingfor a comprehensive and user-friendly entertainment hub.

➤ **Target Audience:**

The primary audience for the OTT Platform project includes movie and TV show enthusiasts who seek a visually appealing and user-friendly interface for discovering and exploring the latest entertainment content. This platform caters to individuals who appreciate the convenience of accessing comprehensive details about movies and TV shows in one centralized location.

PRE-REQUISITES

PRE-REQUISITES:

Here are the key prerequisites for developing a frontend application using React.js:

- **Node.js and npm**:

  Node.js is a powerfulJavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications.

  Install Node.js and npm on your development machine, as they are required to run JavaScript onthe server-side.

- Download: https://nodejs.org/en/download/
- Installation instructions: https://nodejs.org/en/download/package-manager/

- **The API Source**: The API source involves LINK. These are the steps to get API key and put it into .env file.
1. Sign up for the account
2. Go to the settingsin you account
3. And move to api section
4. Copy the _API_TOKEN_KEY and paste it in .env folder

- **Vite:** Vite is a new frontendbuild tool that aims to improve the developer experience for development with the local machine, and for the build of optimized assets for production(go live). Vite (or ViteJS) includes: a development server with ES _native_ support and Hot Module Replacement; a build commandbased on rollup.

  ```
  npm create vite@latest
  ```

- **HTML, CSS, and JavaScript**: Basic knowledge of HTML for creating the structure of your app, CSS forstyling, and JavaScript for client-side interactivity is essential.

- **Version Control**: Use Git for version control,enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.
  - Git: Download and installation instructions can be found at: https://git-scm.com/downloads

- **Development Environment**: Choose a code editor or Integrated Development Environment (IDE) thatsuits your preferences, such as VisualStudio Code, SublimeText, or WebStorm.

    - Visual Studio Code: Downloadfrom https://code.visualstudio.com/download
    - Sublime Text: Download from https://www.sublimetext.com/download
    - WebStorm: Download from https://www.jetbrains.com/webstorm/download

To clone and run the Application project from GitHub:

Follow below steps:

- **Get the code:**
    - Download the code from the drive link given below: [LINK](#)

        Install Dependencies:

        - Navigate into the cloned repository directory and install libraries:

            ```
            cd movies

            npm install
            ```

- **Start the Development Server**:
    - To start the development server, execute the following command:

        ```
        npm start
        ```

- **Access the App:**

    - Open your web browser and navigate to http://localhost:5173/.

    - You should see the recipe app's homepage, indicating that the installation and setup were successful.

You have successfully installed and set up the application on your local machine. You can now proceed with further customization, development, and testing as needed.

Project Structure

```
> node_modules
∨ public
    movix-logo.png
∨ src
  > assets
  ∨ components
    ∨ carousel
        Carousel.jsx
        carousel.scss
    ∨ circleRating
        CircleRating.jsx
        circleRating.scss
    ∨ contentWrapper
        ContentWrapper.jsx
        contentWrapper.scss
    ∨ footer
        Footer.jsx
        footer.scss
    ∨ genres
        Genres.jsx
        genres.scss
    ∨ header
        Header.jsx
        header.scss
    ∨ loadImage
        Img.jsx
    ∨ movieCard
        MovieCard.jsx
        movieCard.scss
```

```
∨ 🟢 src
  ∨ 🟠 components
    ∨ 🟠 movieCard
    ∨ 🟠 spinner
        ⚛ Spinner.jsx
        𝒮 spinner.scss
    ∨ 🟠 switch
        ⚛ Switch.jsx
        𝒮 switch.scss
    ∨ 🟠 videoPopUp
        ⚛ VideoPopUp.jsx
        𝒮 videoPopUp.scss
  ∨ 🟠 hooks
      ⚛ useFetch.jsx
  ∨ 🔴 pages
    ∨ 🟠 404
        𝒮 error.scss
        ⚛ PageNotFound.jsx
    ∨ 🟠 details
      ∨ 🟠 cast
          E:\web documents\react\r
          𝒮 cast.scss
      ∨ 🟠 detailsBanner
          ⚛ DetailsBanner.jsx
          𝒮 detailsBanner.scss
        ⚛ Details.jsx
        𝒮 details.scss
        ⚛ PlayBtn.jsx
    ∨ 🟠 explore
        ⚛ Explore.jsx
```

```
∨ 📦 src
  ∨ 📕 pages
    ∨ 📁 explore
    ∨ 📁 home
      ∨ 📁 heroBanner
          ⚛ HeroBanner.jsx
          🎨 herobanner.scss
      ∨ 📁 popular
          ⚛ Popular.jsx
      ∨ 📁 top rated
          ⚛ TopRated.jsx
      ∨ 📁 trending
          ⚛ Trending.jsx
        ⚛ Home.jsx
        🎨 home.scss
    ∨ 📁 searchResult
        🎨 search.scss
        ⚛ SearchResult.jsx
  ∨ 📁 store
      JS homeSlice.js
      JS store.js
  ∨ 🌟 utils
      JS api.js
      🎨 App.css
      ⚛ App.jsx
      🎨 index.scss
      ⚛ main.jsx
      🎨 mixins.scss
    ⚙ .env
    ◉ .eslintrc.cjs
```

These are the screenshots of all folder structures that are used in this project. This shows how components and other folderstructures are used for the smooth buildingof projects.

- Here we can see that inside the src folder we have components folder, store folder, utils folder. Along with App.jsx file, App.css, etc.

- Inside the components folder we have carousel folder, circleRating folder, contentWrapper folder, footer folder, genres folder, header folder, loadImage folder, movieCard folder, spinner folder, switch folder,and videoPopUp folder.

- Inside the pages folder we have 404 folder, details folder, explore folder, home folder, and search Result folder.

Milestone 1 : ProjectFlow

Project demo:

Before startingto work on this project,let's see the

demo.Demo link: [LINK](#)

Usethe code: [LINK](#)

Let's Proceed with the project flow for the project
development phase Project setup and configuration:

- Setup React Application:

    - Create a React app in the client folder.

    - Install required libraries

    - Create required pages and components and add routes.

- Design UI components:

    - Create Components.

    - Implement layout and styling.

    - Add navigation.

- Implement frontend logic:

    - Integrationwith API endpoints.

    - Implement data binding.

Milestone 2 : Project Development

- Setup the Routing paths
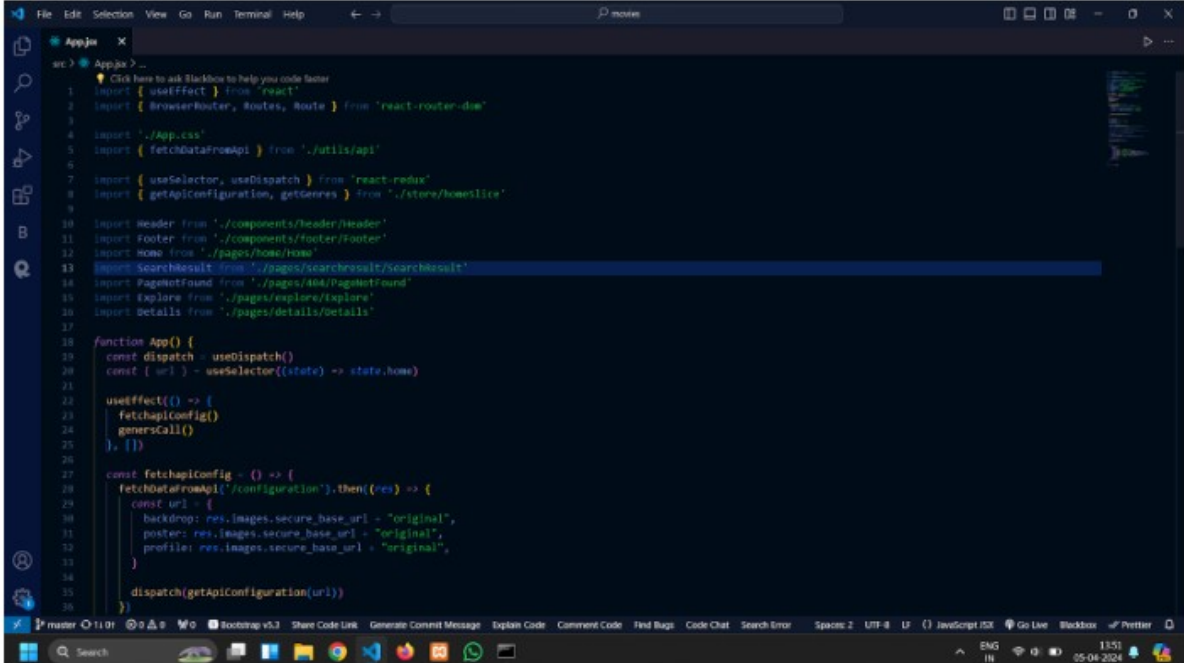
  Set the clear routingpaths to accessvarious files in the application.

- some important code snips

1. App.jsx file: this is the counter file where all routes and all redux componentsare declared and used, to share data among all components and files. All Importing statements like import
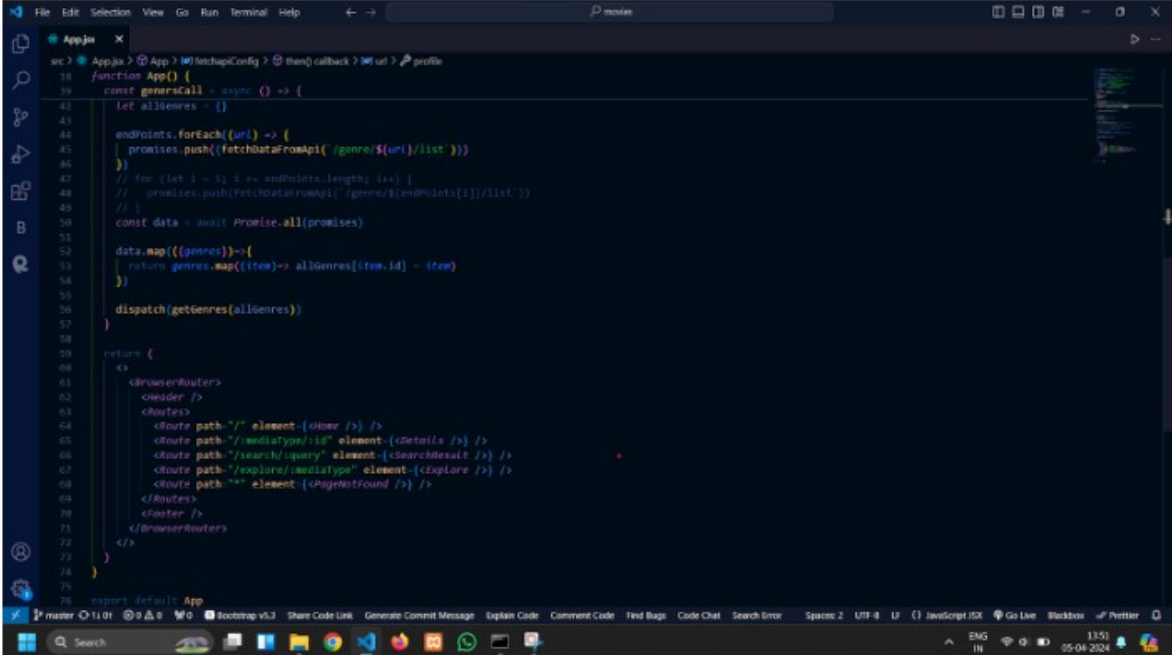
   {useEffect} from 'react', import './App.css', etc are present in this file only.

2. It contains function like **fetchApiConfig** to configure to the API endpoints which can be used to connect throughthe redux presentin the redux folder.

This file returns all the components and routes,params routing, etc. It containsBrowserRoutes, Router, and Route which are imported from react-router-dom above in code. Header and Footer components are present which are importedfrom their respective folders.


API file where a custom hook is defined for the project to fetch data that is coming from the API and other useful variables like BASE_URL = "API", TMBD_TOKEN = "token", and returning data that coming from the API fetching using **fetchDataFromApi** function.
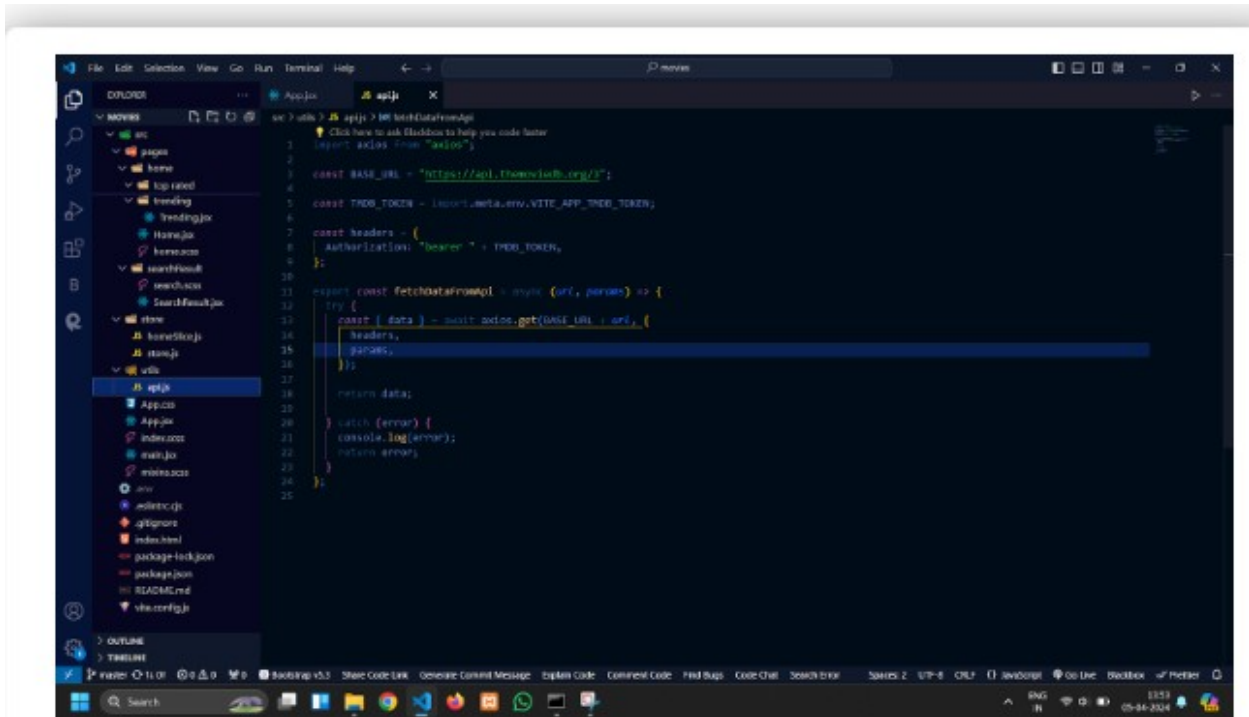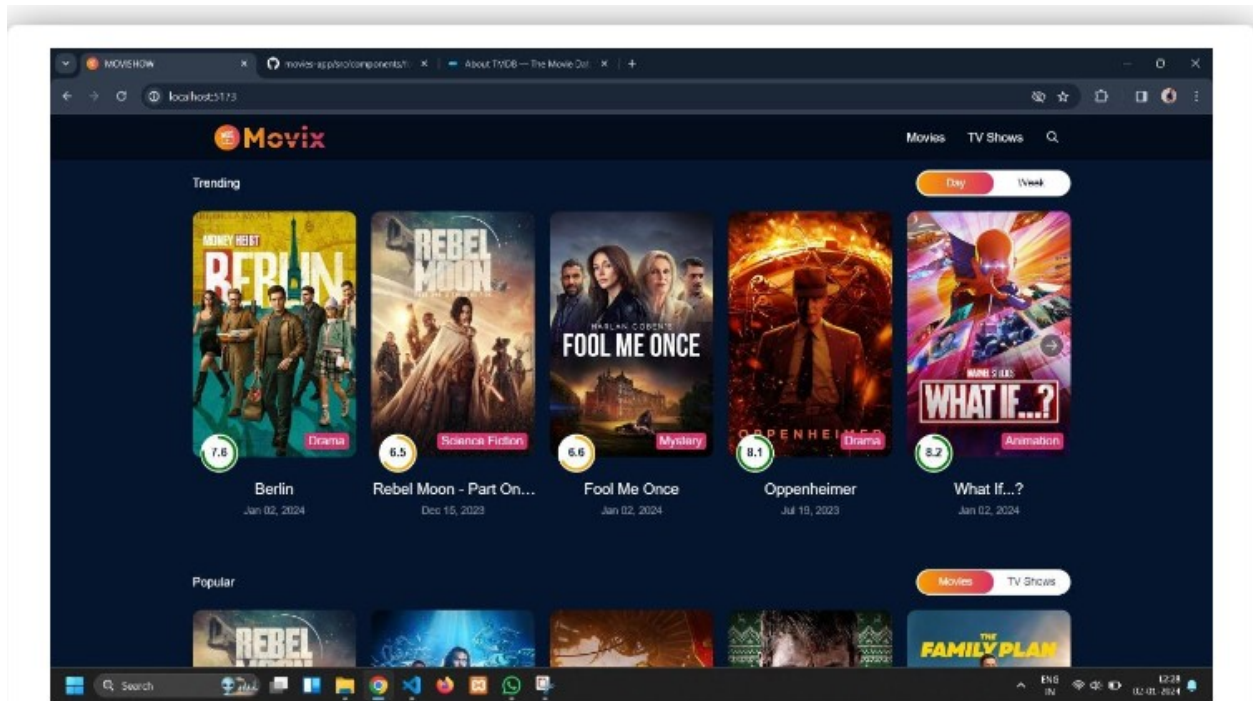
# Milestone 3 : ProjectImplementation & Execution

Home Page

1. Here we can see the Background Image.

2. We also see a search bar.

3. There is a Navbar which can navigate us to Movies and TV Shows sections.
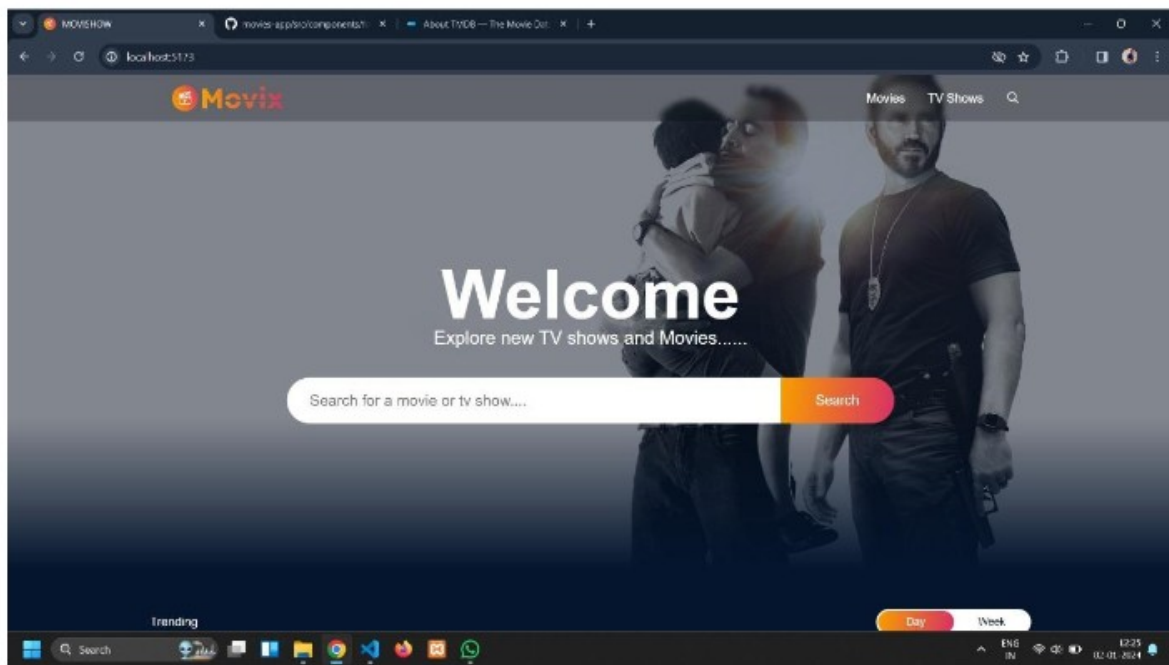
- We see the cards of the Movies along with their ratings, their posters as well as their titles.We can see that the navbar is still present in the page.

- Then when we click on our favourite movie card, we can see that the poster is aligned on the left hand side and the info is displayed on the right hand side along with it's ratingsand watch trailer button.
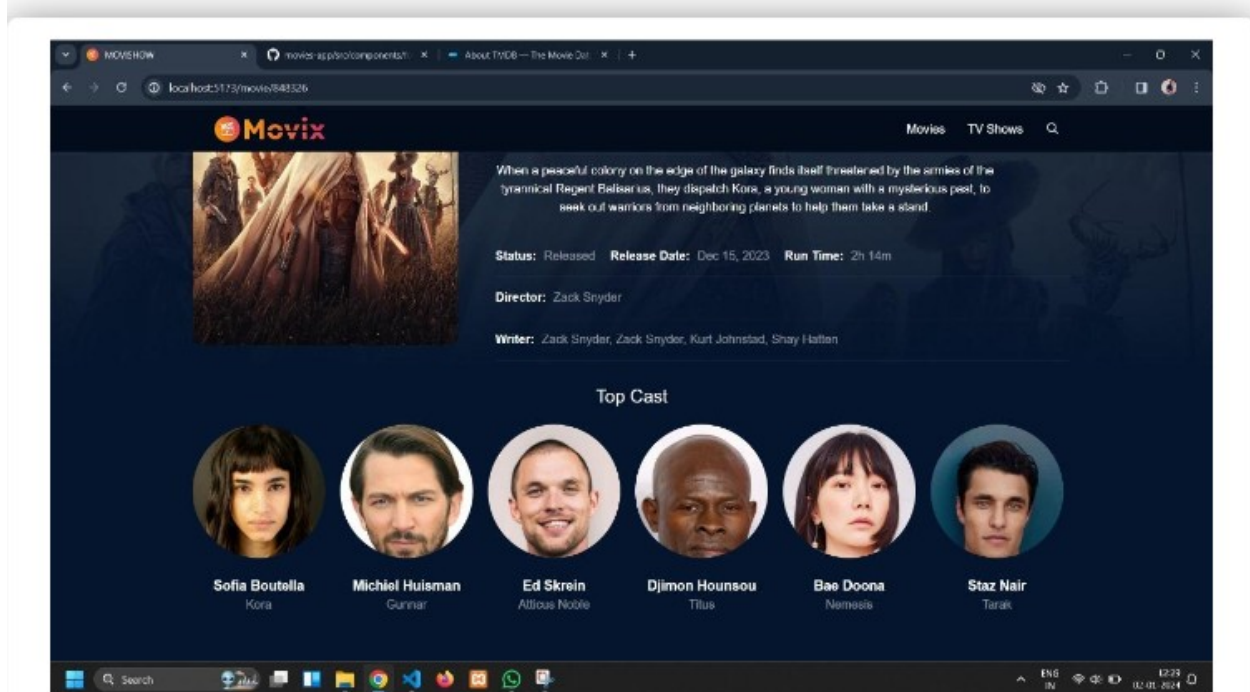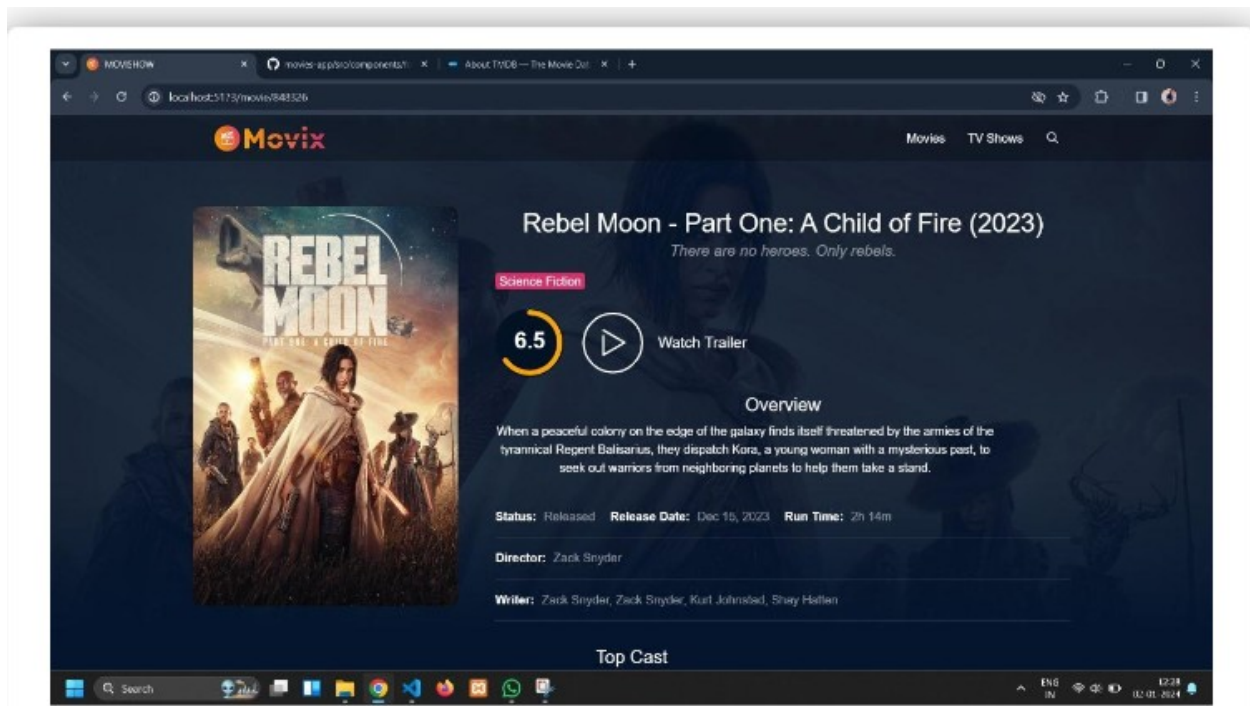
- We also see that clicking on the "watch trailer" button opens up the trailer of the selected movie.

**** Happy Coding !!****