# Movie Website Development — Project Documentation

## Project Title: Movie Application using ReactJS and OMDb API

## Project Objective

To design and develop a responsive and interactive movie website using **ReactJS** and the **OMDb API**, allowing users to browse, search, filter, and view detailed movie information.

---

## Tech Stack Used

- **Frontend**: ReactJS (with Hooks)
- **API**: OMDb API (`https://www.omdbapi.com/`)
- **Styling**: ShadCN UI Components, Tailwind CSS
- **Version Control**: Git and GitHub
- **Deployment**: GitHub Pages

---

## Project Features

### 1. Movie List Display

- Fetches and displays popular/trending movies using API.
- Shows posters, titles, release years, and ratings in a responsive grid layout.

### 2. Search Functionality

- Allows users to search for movies by title.
- Real-time display of matching movies with loading state.
- Handles cases of no matches or API errors.

### 3. Movie Details Page / Modal

- On clicking a movie, shows detailed information including:
  - Title, Year, Genre
  - IMDb Rating
  - Plot Summary
  - Poster Image
- Utilizes modals or separate route/page for display.

### 4. Filter & Sort Options

- Genre and Year-based filtering.
- Sorting options include:
  - Popularity
  - Release Date
  - Rating

---

## Folder Structure

```
/src
  ├── components/
  │     ├── MovieCard.jsx
  │     ├── SearchBar.jsx
  │     ├── FilterBar.jsx
  │     └── MovieModal.jsx
  ├── pages/
  │     ├── Home.jsx
  │     └── MovieDetails.jsx
  ├── App.jsx
  └── index.js
```

---

## Best Practices Followed

- Functional components and React Hooks (`useState`, `useEffect`)
- Modular code with reusable components
- Environment variables for API keys
- Axios / fetch for API calls with error handling
- Responsive design using Tailwind's utility classes

---

## Setup Instructions

### 1. Clone the Repository

```
git clone https://github.com/your-username/movie-app.git
cd movie-app
```

### 2. Install Dependencies

```
npm install
```

### 3. Setup API Key

Create a `.env` file in the root:

```
VITE_OMDB_API_KEY=your_api_key_here
```

### 4. Run the Development Server

```
npm run dev
```

---

## Deployment on GitHub Pages

The project was deployed using **GitHub Pages** by following these steps:

### 1. Install `gh-pages` Package

```
npm install gh-pages --save-dev
```

### 2. Add Deployment Scripts in `package.json`

```
"homepage": "https://<your-github-username>.github.io/<repo-name>",
"scripts": {
  "predeploy": "vite build",
  "deploy": "gh-pages -d dist"
}
```

### 3. Initialize and Push to GitHub

```
git init
git remote add origin https://github.com/<your-username>/<repo-name>.git
git add .
git commit -m "Initial commit"
git push -u origin main
```

### 4. Run the Deployment Command

```
npm run deploy
```

This command does the following:

- Builds the production version using `vite build`
- Publishes the contents of the `dist` folder to the `gh-pages` branch of your repository

### 5. Access the Live App

Your project is now live at:

`https://<your-github-username>.github.io/<repo-name>/`

---

## Challenges Faced

- Handling CORS and API key security
- Managing conditional rendering while loading API data
- Designing mobile-friendly layout with Tailwind CSS

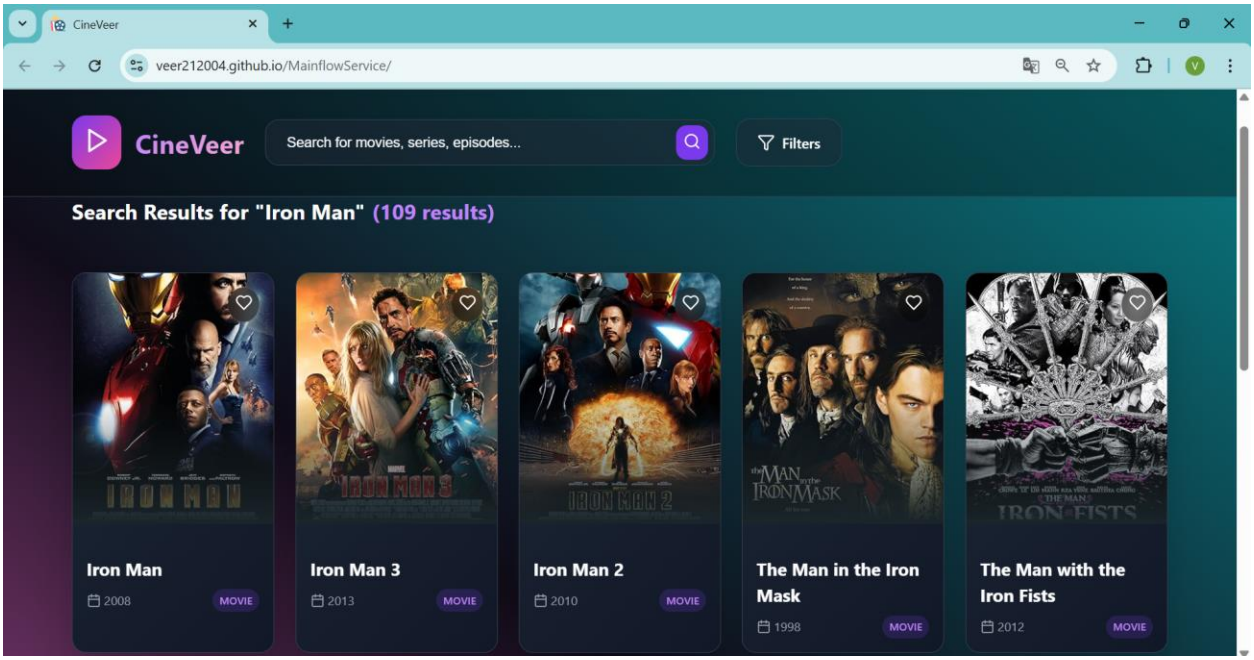
---

## Key Learnings

- Practical experience with React component-based structure
- Working with third-party APIs and managing real-time data
- Deployment workflows and handling environment variables
- UI/UX fundamentals and responsive design

---

## Screenshots




---

## Conclusion

This project provided real-world experience in building a functional frontend application. It demonstrates proficiency in React, API integration, responsive design, and modern deployment platforms.

### 🤳 Submitted by:

**Name:** Veeresh Hedderi

**Task**: MERN Stack Task 2 – Movie Website Development

**Submitted to:** Main Flow Services and Technologies Pvt. Ltd.

**Contact**: veereshhedderi18gmail.com