**Movie Discovery Documentation**

**Overview**

This document provides technical documentation and developer setup instructions for the Movie Discovery project included in this repository.

**Project Structure**

root/

├── .editorconfig

├── .gitignore

├── angular.json

├── package.json

├── proxy.config.json

├── README.md

├── tsconfig.app.json

├── tsconfig.json

├── tsconfig.spec.json

├── .angular/ # Angular cache/configuration

├── .vscode/ # VS Code workspace settings

├── public/ # Static assets

└── src/ # Source code of the Angular application

**Technologies Used**

* Angular (TypeScript)

**Feature Specifications**

**1. Landing Page Module**

* Main starting point is App Component.
* Implemented in MoodFindsComponent.
* File path: src/components/mood-finds/mood-finds.component.ts
* This component renders a homepage with a heading and introduction.
* Includes three buttons (e.g., feel-good, action-fix, mind-benders) bound to route-based or state-managed navigation, dynamically triggering API requests or filtering movie categories.

**2. Search Functionality**

* Implemented in SearchResultComponent.
* File path: src/components/search-result/search-result.component.ts
* This component captures user input using two-way data binding.
* Connected to a movie/actor search service that performs HTTP requests to an external API.
* Displays parsed response data with clear differentiation between movie and actor results using separate components.

**3. Movie Cards**

* A MovieCardComponent will receive input data through @Input() bindings.
* File path: src/components/movie-card/movie-card.component.ts
* Each card will include:
  + Lazy-loaded poster image (via Angular's image directive or third-party library)
  + Title and release year extracted from API
  + Average rating displayed with a rating visual (stars or numeric badge)

**4. Movie Detail Page**

* Routing to a MovieDetailsModalComponent via dynamic route parameters (e.g., /movie/:id).
* File path: src/components/movie-details-modal/movie-details-modal.component.ts
* Component will fetch movie details including metadata, trailer links, and cast list from an API.
* Related or similar movies can be shown using a horizontally scrollable SimilarMoviesComponent.

**API Integration**

**Using TMDB API**

This application fetches movie and actor data from [The Movie Database (TMDB)](https://www.themoviedb.org/).

**Getting a TMDB API Key**

1. Go to <https://www.themoviedb.org/settings/api>.
2. Sign in or create an account.
3. Click on the “Create” button under the API section.
4. Fill out the application form to request an API key.
5. Once approved, you’ll receive an API key for use in your application.

**Where to Use the API Key**

The TMDB API key is stored in the Angular project in the constants configuration file:

* Path: src/app/constants/constants.ts
* Edit the environment constant to include your TMDB key:  
  export const environment = {  
   production: false,  
   apiKey: 'your\_tmdb\_api\_key\_here',  
   apiUrl: 'https://api.themoviedb.org/3'  
  };
* This environment.apiKey value is imported and used by services making HTTP requests to TMDB endpoints.

**Note:** Do not commit your actual API key to version control.

**Setup Instructions**

**Prerequisites**

Ensure the following software is installed on your system:

* Node.js (v14 or higher recommended)(used v22.17.0)
* npm (v6 or higher)
* Angular CLI (npm install -g @angular/cli)

**Installation**

1. Clone the repository or extract the ZIP archive:  
   ----> git clone <https://github.com/shivamj19/assignment_jktech.git>  
   ----> unzip project.zip && cd project-folder
2. Navigate to the project directory:  
   ----> cd path-to-project
3. Install the project dependencies:  
   ----> npm install

**Running the Application**

To start the development server:

----> ng serve

Visit http://localhost:4200/ in your browser.

**Building the Application**

To build the project for production:

ng build --prod

The output will be located in the dist/ directory.

**Project Configuration Files**

* angular.json: Angular CLI configuration
* tsconfig.json: TypeScript configuration for the project
* package.json: Lists project dependencies and scripts
* proxy.config.json: Proxy configuration for development

Test coverage  
