React

Project 01 – Build a Movie Search Application

Course Value: 70%

Due: End of day – Jan 7, 2020 (11:59pm)

Instructions:

1.) Create a Movie App using React that meets the following specifications:
   a. Contains the following pages (detailed page specifications further down)
      i. Home Page
      ii. About Page
      iii. Discover Page
      iv. Search Results Page
      v. Individual Movie Page
      vi. My Favourites Page
      vii. My Rated Page

2.) The web application should be mobile responsive and work on devices with screen sizes between 320px and 1980px

3.) Be well styled, with a usable user interface that follows basic design best practices:
   a. Text content is readable
   b. Elements are in alignment
   c. Good spacing between elements
   d. Application navigation that is easy to use and allows the user to navigate to all the main pages of the application

4.) Use The Movie DB API to get your movie data for use in this application
   a. Details on how to setup an account and get your API key are shown further down

5.) The application should use the browser's local storage to store the user's favourited and rated movies which will be used on the "My Favourites" and the "My Rated" pages
   a. Click the link below for information on the localStorage JavaScript API:
      i. https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage

6.) Place your project on GitHub and provide your Repo URL in the submission of this project on the Learning Hub drobox for this assignment
   a. Make sure to add your instructor as a contributor if using a private repo (my GitHub username is: michaelwhyte)
   b. Your Repo should contain a build version that is ready for deployment and is simple to run in a browser with minimal setup
      i. Please provide detailed installation instructions for the application in your README.md file

7.) Zip up your project folder and upload it to the React – Movie App Project dropbox on the Learning Hub for this course. See your instructor if you are not sure where this is.

Detailed App Specifications

Note: These are minimum specifications. Feel free to add additional features

The Movie DB API

- This project should use The Movie DB API
- You will need to create a FREE account to obtain a FREE API Key
    - First you will need to create a FREE general account with The Movie DB API. Then…
    - …follow the instructions on how to get your API Key here (look under the section "How do I apply for an API key?"):
        - https://www.themoviedb.org/faq/api
    - When you apply for an API key, The Movie DB will ask for a few pieces of personal information (name, app name, app URL, phone number and address). If you do not feel comfortable giving your personal address and phone number, then feel free to use the BCIT Downtown Campus address and phone number
    - The entire application form must be filled in. Below is a list of boilerplate answers that you can give in order to get an API key.
        - **Application Name:** Movie App
        - **Appication URL:** N/A – Internal application for educational use
        - **Application Summary:**
            - An internal application for educational non-profit use, used mainly for learning how to use an API with React
        - **Address:**
            - Use your home address or use BCIT's DTC address:
                - 555 Seymour Street, Vancouver, BC, V6B 3H6
        - **Phone Number:**
            - Use your personal phone number or BCIT's main switch board:
                - 604.434.5734
        - **Email:**
            - Use an email address that you have access to. You will receive your API key in an email
    - Once you submit your application you will be approved instantly and will be given an API key. You will need this key to make API requests.
        - Do not share your API key with the general public.
        - You can share it with your teammate (if you choose to work in a team)

Application Page Specifications

All Pages

- The following should appear on all pages:
    - Application title and/or logo
    - Mobile friendly navigation that enables the user to go to any page in the application from any page in the application

Home Page

- All the requirements from the "All Pages" requirements plus…
- A Search form that allows a user to search for a Movie by Title
- Form select element or other UI element (select box, radio buttons, links or just buttons…the UI is up to you) that allows a user to change the current movies displayed between the following options:
    - Popular
    - Top Rated
    - Upcoming (yet to be released)
    - Now Playing
- On initial load of the application the page should display 12 of the most current popular movies (The Movie DB considers popular movies to be those that have the most ratings)
    - The user can switch this list by using the form select or other UI element to change these 12 movies to:
        - The 12 most highly rated movies currently out
        - The 12 most recently released movies (movies that are currently playing)
        - The next 12 movies that will be released soon (upcoming)
    - Each movie should display the following information:
        - The movie's poster
            - If a poster is not available, then you should load a generic placeholder image
        - The movie's title
        - The movie's release date
        - The movie's rating (review rating – eg: 67%)
        - A short summary of the movie's plot
        - A "More Info" button that the user can click on to get additional information on the individual movie page
            - Optionally the entire movie listing can be clickable to the individual movie page

About Page

- All the requirements from the “All Pages” requirements plus…
- A simple generic description about the application
- Meet the TMDb attribution requirements on this page
  - The TMDb attribution requirements include showing the TMDb logo and the following message:
    - "This product uses the TMDb API but is not endorsed or certified by TMDb."
    - The TMDb logo should be less prominent then the application logo and the logo should be placed in a way that does not give the appearance that your application is endorsed by TMDb
    - Read about attribution requirements on the FAQ page (look under the “What are the attribution requirements?” section)
      - https://www.themoviedb.org/faq/api
      - Click the link below for TMDb logo images:
        - https://www.themoviedb.org/about/logos-attribution

Discover Page

- All the requirements from the “All Pages” requirements plus…
- This page will allow users to “Discover” new movies
- Users should be able to discover movies by:
  - Year (1920 – Present)
  - Genre (You can find a list of genre names on the Docs for the TMDb API)
    - Users should be able to choose either a single genre or multiple genres
  - A combination of year & genre (or multiple genres)
- The application should display the top 12 results from the user’s “discover” search
  - Each movie should display:
    - The movie’s poster (or generic placeholder if no poster is available)
    - The movie’s title
    - The movie’s release date
    - The movie’s rating (review rating – eg: 67%)
    - A short summary of the movie’s plot
    - A “More Info” button that the user can click on to get additional information on the individual movie page
      - Optionally the entire movie listing can be clickable to the individual movie page
- The user should be able to sort the results by the following:
  - Rating – highest first or lowest first
  - Release Date – latest first or oldest first
  - Title – A to Z or Z to A

Search Results Page

- This page is used when a user searches for a movie title from any general title search (not the discover search)
- All the requirements from the “All Pages” requirements plus…
- A Search form that allows a user to search for a Movie by Title
- A page subtitle that displays the user’s search query. See below for an example
    - Example (assuming the user searched for the movie “Star Wars”:
        - Search > Star Wars
- List of search results that match the user’s search query
    - Each result should display the following information:
        - The movie’s poster (or generic placeholder if no poster is available)
        - The movie’s title
        - The movie’s release date
        - The movie’s rating (review rating – eg: 67%)
        - A short summary of the movie’s plot
        - A “More Info” button that the user can click on to get additional information on the individual movie page
            - Optionally the entire movie listing can be clickable to the individual movie page

Individual Movie Page

- This page is accessed when a user clicks on the “More Info” link on an individual movie from a search or discovery result
- All the requirements from the “All Pages” requirements plus…
- A Search form that allows a user to search for a Movie by Title
- The movie’s poster (or generic placeholder if no poster is available)
- The movie’s title
- The movie’s release date
- The movie’s rating (review rating – eg: 67%)
- A short summary of the movie’s plot
- A button or similar user interface element that allows the user to “favourite” or “heart” or “like” a movie or “unfavourite” or “unheart” or “unlike” a movie if a user has already favourited the movie
    - When a user favourites a movie the application should store information about the movie in localstorage using the localstorage web api
    - When a user unfavourites a movie that movie should be removed from localstorage
- A user interface element that allows the user to rate the movie
    - You can use a star system (0 – 5 stars) or a simple percentage out of 100 (eg: 75%)
    - If the user rates the movie then you should store information about the rated movie in localstorage
    - If the user has rated the movie then the application should display the users rating

My Favourites Page

- All the requirements from the "All Pages" requirements plus…
- If the user has **NO** favourited movies, then:
    - Display a message similar to:
        - "Sorry you have no favourited movies. Search for a movie to add to your favourites."
    - A Search form that allows a user to search for a Movie by Title
- If the user DOES HAVE favourited movies, then display all the favourited movies
    - The movies should be retrieved from localstorage
    - Each movie should display the following information
        - The movie's poster (or generic placeholder if no poster is available)
        - The movie's title
        - The movie's release date
        - The movie's rating (review rating – eg: 67%)
        - A short summary of the movies plot
        - A "More Info" button that the user can click on to get additional information on the individual movie page
            - Optionally the entire movie listing can be clickable to the individual movie page

My Rated Page

- All the requirements from the "All Pages" requirements plus…
- If the user has **NO** rated movies, then:
    - Display a message similar to:
        - "Sorry you have no rated movies. Search for a movie to add to your favourites."
    - A Search form that allows a user to search for a Movie by Title
- If the user DOES HAVE rated movies, then display all the favourited movies
    - The movies should be retrieved from localstorage
    - Each movie should display the following information
        - The movie's poster (or generic placeholder if no poster is available)
        - The movie's title
        - The movie's release date
        - The movie's rating (review rating – eg: 67%)
        - A short summary of the movies plot
        - A "More Info" button that the user can click on to get additional information on the individual movie page
            - Optionally the entire movie listing can be clickable to the individual movie page

FAQ

Q: Can I work in a team?

A: Yes, you can work in a team of 2 or 3, but this is optional and you can work solo if you wish

Q: Can I use Bootstrap or Foundation for styling and layout?

A: Yes

Marking Criteria

All instructions followed	10 marks
Application specifications fulfilled (2 marks deducted for each missing application feature)	20 marks
Design and Usability of the web site	10 marks
Quality of the code	30 marks
Total	**70 marks**

Late Project Policy:

- 5% per day up to a maximum of 30%