Web Scripting 3 – FWDP 3400

Project 01 – Build a Movie Application

Course Value: 60%

Due: End of day – May 24, 2022 (11:59pm)

Instructions:

1.) Create a Movie App as part of a team (max 3 students) or individually using React that meets the following specifications:
   a. Contains the following pages (detailed page specifications further down)
      i. Home Page
      ii. About Page
      iii. Individual Movie Page
      iv. My Favourites 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 present on all the application's pages

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 users favourited movies which will be used on the "My Favourites" page

6.) Build your project using the Create React App build script and upload it to your portfolio server and submit a link to your application on the Learning Hub
   a. Detailed instructions on how to properly upload your application to your portfolio server will be provided during the course

7.) Upload your raw development files to the Learning Hub.
   a. Name your project folder using this format:
      i. [student names…]-movie-app
      ii. For example: if your group was Larry, Curly and Moe, your folder name would be:
         1. larry-curly-moe-movie-app
   b. If you worked in a team of 2 or 3, have one member of your group message your instructor via Slack or email with all the members of your group
   c. Please do NOT upload the node_modules folder. This folder is large and is not needed for marking

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
        - **Application 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 that is linkable back to the home page
    - 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…
- 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
    - Now Playing
    - Upcoming (yet to be released)
- 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 – example: 67%)
        - A short summary about the movie
        - 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

Individual Movie Page

- This page is accessed when a user clicks on the “More Info” link on an individual movie
- All the requirements from the “All Pages” requirements plus…
- 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 – example: 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

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. Return to the home page to add a favourite movie”
- 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 – example: 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

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, but I would recommend trying to make your application a showcase portfolio piece by giving it a good dose of custom styling.

Marking Criteria

- All instructions followed	10 marks
- Application specifications fulfilled	10 marks
- Design and Usability of the web site	20 marks
- Quality of the code	20 marks
- **Total**	**60 marks**

See the marking rubric for this project on the following page for more details on how you will be marked for this project.

Late Project Policy:

- 5% per day up to a maximum of 35%
- No assignments accepted that are more than 14 days past due

Project 01 – Movie Application - Marking Rubric

Criteria	Grunt 40	Padawan 60	Ninja 80	Wizard 100	Score
All instructions followed in regard to project submission	- Failed to follow submission instructions -Project was unable to be run	- Failed to follow submission instructions -Project could be installed and run	- Submission instructions followed -Link to the live website does not work or no link was provided	- Submission instructions followed -Link to the live website was provided and the link to the website works	10
Application specifications fulfilled	- Several key application features missing -Application features not functioning	- 1 or 2 key application features missing - Some application features buggy	-All application features present -Some application features buggy	-All application features present -All application features work without bugs	10
Design & usability of the web site	- Application is unusable at desktop or mobile screen sizes - Unable to navigate around the application	- Application is usable at a basic level - UI elements not in alignment - Text content is difficult to read	- Application works well from mobile up to desktop screen sizes - Some UI elements do not match the overall design of the application - Text content is readable	- Application works well from mobile up to desktop screen sizes - UI is fully polished with care taken to make sure all elements of the application follow a consistent design theme - Text content is a pleasure to read	20
Quality of the code	- File structure of application is disorganized - Web safe filename conventions not followed - Code is difficult to read	- File structure is good -Some improvement needed in making the code more modular - Code is not properly indented	- File structure is well organized - Code is modular - Well defined functions present - Code is nicely commented and formatted	- File structure is well organized with good use of folders for better organization - Functions are well defined - Global variables are saved to a separate file - Code is nicely commented and formatted	20