

SSD React

Project 01 – Build a Movie Application

Course Value: 40%

Due: Feb 24, 2024 (11:59pm)

Instructions:

- 1.) Create a Movie App as part of a team (max 4 students), or individually, using React such that it 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
 - v. My Watch List Page
- 2.) The app must be fully responsive and work on screens from 480px and up.
- 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 can either use the browser's local storage to store the user's favourites and watch list OR, for extra challenge/practice, you can also use the DB of your choice for persistent storage.

NOTE: Use of a database IS NOT REQUIRED and represents quite a bit of extra work since you'll need to set up a back end (Node or .Net) as well as implement an Auth layer. Ultimately that will be a more robust app, but given our time constraints using local storage will suffice. Grading remains the same regardless of which approach you take, so choose wisely!

Submission and Presentation

When you have completed your work

- 1.) Build your project using the Vite build script:
<https://vitejs.dev/guide/build>
- 2.) Deploy your built app to github pages or a host of your choice:
<https://vitejs.dev/guide/static-deploy.html>
- 3.) Also upload your project (including the build folder) 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. Please do NOT upload the node_modules folder. This folder is large and is not needed for marking. A 10% penalty will apply if node_modules are included in the final submission.
- 4.) Prepare a 5-10 minute presentation to be delivered to the class on Feb 28, 2024.
 - a. This is not intended to be a sales pitch style presentation, but rather a “technical” presentation.
 - b. Demonstrate the core functionality and discuss how you implemented it, as well as challenges that you surmounted along the way.
 - c. Think of it along the lines of the portion of an interview where you are asked to show samples of your previous work.
 - d. Barring illness or previously approved accommodations, ALL team members MUST participate in the presentation.
 - e. After each presentation there will be a Q&A period.

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](https://www.themoviedb.org/)
- 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:
 - o Application title and/or logo that is linkable back to the home page
 - o Navigation component 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:
 - o Popular
 - o Top Rated
 - o Now Playing
 - o 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)
 - o 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)
 - o 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
 - A visual indicator whether the movie is already in the Favourites and/or Watch List
 - The ability to add/remove the movie from Favourites and/or Watch List.

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
 - o 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 than 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:
 - o <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 favoured the movie
 - o When a user favourites a movie, the application should store information about the movie in localStorage using the localStorage web api
 - o When a user unfavourites a movie that movie should be removed from localStorage
- A button or similar user interface element that allows the user to add/remove the movie from their “Watch List”.

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

Watch List Page

- Analogous to the Favourites page.... See above.

FAQ

Q: Can I work in a team?

A: Yes, you can work in a team of up to 4 students, but this is optional. You can work solo if you wish.

Q: Can I use Bootstrap or another UI framework 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

- Application specifications fulfilled	30 marks
- Quality of the code	30 marks
- Design and Usability	20 marks
- Presentation	20 marks
- Total	100 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 without valid reason