

# CS 3200 – Spring 2019

## Assignment #4 – Moviemania

### Introduction

Moviemania (pronounced ‘Moo-Vee-May-Knee-Uh’) is American for *Movie Mania*. This app will actually be useful. You will create a tool to search and view movie information.

The purpose of this assignment is to introduce:

- Networking
- Promises
- Navigation
- 3<sup>rd</sup> Party Components (Nativebase)

### Moviemania Usage

Your program will allow a user to search for movies/people and view details of a selected item. It will be a very intuitive program with simple view-only capabilities.

### Resources

- <https://www.themoviedb.org>
- <https://www.themoviedb.org/documentation/api>
- <https://developers.themoviedb.org/3/getting-started/introduction>

### Requirements

Create a program that does the following:

1. (15 pts) Navigation bar (or whatever you want) that gives the following options for pages
  - a. Browse
  - b. Search
2. (30 pts) Browse Page
  - a. Lists genres
  - b. When a user taps a genre, it changes to a screen showing a list of movies within that genre.
  - c. On the list page, there should be a button that allows a user to go back to the genre selection page.
3. (30 pts) Search Page
  - a. Allow a user to select a search for
    - i. Movies
    - ii. People
  - b. After the search returns, change to a screen showing a listing of the search results.
  - c. There should be a button that allows a user to go back to the genre selection page.
4. (30 pts) List Pages
  - a. Pages should scroll infinitely by retrieving more data when scrolling is near the bottom of the list
  - b. The list pages (search results) should show a summary of each item
  - c. Movie
    - i. Poster image
    - ii. Title
    - iii. Popularity
    - iv. Release Date
    - v. Overview (Not the entire thing, just the first bit)
  - d. People
    - i. Name
    - ii. Popularity
    - iii. Image
5. (30 pts) Detail Pages
  - a. Movie

- i. All items from list page
    - ii. Genres
    - iii. Full overview
    - iv. Budget
    - v. Revenue
    - vi. Status
    - vii. Cast
      - 1. Image
      - 2. Person name
      - 3. Character name
  - b. Person
    - i. All items from list page
    - ii. Birth date
    - iii. Death date (if applicable)
    - iv. Place of birth
    - v. Biography
    - vi. Credits
      - 1. Image
      - 2. Title
      - 3. Year
  - c. Navigation
    - i. Users should be able to click on any movie, TV show, or person name and navigate to the detail page for that item. The back button should take them to their last viewed page (where they navigated from).
    - ii. From any list page a user can tap a selection and be shown a page that shows all details for the selected item
- 6. (30 pts) Organization
  - a. You must have the following services
    - i. api.service.js – Contains the information about the api access
    - ii. movie.service.js – Contains methods for app to call to make data requests
  - b. You must have the following custom components
    - i. MovieDetail.js
    - ii. MovieSummary.js
    - iii. PersonDetail.js
    - iv. PersonSummary.js
    - v. Any others you see fit (These required ones are meant to help you to know what to do).
  - c. Styling should be in a different file than the component
- 7. (15 pts) Other Items
  - a. No adult movies should show
  - b. All transitions between screens should be handled by the Navigator
  - c. You can decide how each item shows
    - i. Example: Listing a cast might be better to show a few with a link to show all on a new page. Or you may choose to make the cast list a horizontal scrolling list.
  - d. Must use a 3<sup>rd</sup> party navigation library
  - e. You cannot use 3<sup>rd</sup> party libraries for the API access
  - f. Check with me before using any 3<sup>rd</sup> party library besides:
    - i. Nativebase
    - ii. Navigation package

## Submission

Submit your files on Canvas. If you followed the guidelines in the class conventions documentation you will be fine.