**Chess Movies with React**

Graphical user interface, text, application

Description automatically generated

1. Make an HTML page. Add a title: John Smith’s Chess Movies (with John Smith replaced by your name).
2. Add an HTML comment with your name.
3. Your HTML elements with ids cannot have default/meaningless ids such as Button1. They also cannot refer to some previous subject – such as books, dogs, etc.
4. Make the page’s background color and (font) color anything other than the default values.
5. Put a header with John Smith’s Chess Movies (with John Smith replaced by your name) at the top of the page.
6. Add a div element that will serve as the "target" for your react component.
7. Link to the React and possibly Babel code.
8. Place a JavaScript comment with your name at the top of the script area (which might be of the type Babel).
9. Use fetch to read the file chess\_movies.json. And use this information to set the state of your React component. Remember when reading a file to use LiveServer and to account for the time when the file is being read (isLoading?)
10. Use React to create the following:
    1. A drop-down select (with film titles)
    2. An image element (height 315) to display the poster
    3. Divs, spans or labels to display
       1. Year, Director, and Runtime
       2. Cast
       3. Genre
       4. Description
       5. Hyperlink (anchor tag) linking to the IMDB page (use the imdb code to concatenate a link such as <https://www.imdb.com/title/tt0471768/>
11. The drop-down select should be loaded with film titles from the JSON file.
12. When the user changes the title in drop-down select, that movies’s info (year, director, runtime, cast and description) should be displayed.
13. In addition, the IMDB link (href) should change, and that film’s poster should be displayed in the above image element. (Use target="\_blank" so that the link opens in a new tab.)
14. A header that says "Search by Genre"
15. Have a genres array in the state

["Biography", "Crime", "Documentary", "Drama", "History", "Mystery", "Romance", "Sport", "Thriller"]

Use it to dynamically generate a set of radio buttons (horizontally or vertically arranged – doesn’t matter)

1. A Search Button
2. Clicking the button should search through the movies. Use indexOf to produce an array of matches. See if the movies’ genre string contains the genre the user has selected using the radio buttons. Push all matching movies onto the matches array. Display the matches array in an unordered list <ul></ul> My search looked like this

Text

Description automatically generated

1. Changing the selected radio button should set “matches” to an empty array.
2. Zip up your code and email it.