# **Client-Side React Application**

Graded

1 Day, 16 Hours Late

### Student

Krittaya Kruapat

**Total Points** 

78 / 100 pts

Autograder Score 0.0 / 0.0

### **Passed Tests**

Report

Build

package.json

package-lock.json

Source files

### Question 2

# **Functionality and performance**

**57** / 70 pts

**Functionality** 

→ + 30 pts Site has all functionality for a grade of 6 (SPA, login, reggo, refresh token, map, chart, table with mostly full utilisation of search endpoint) with some issues or limitations, and missing a feature

### Performance and Robustness

→ 14 pts Site is performant and robust. Only missing a 404 not found page.

# UI / Site design

- → + 13 pts Decent UI and site design, but one questionable choices in page layout or components behave strangely
- Couldn't search with empty query, couldn't view 12000 when entering the movie page
  - Year option is limited
  - Search only works in the initial 100 movies
  - Registration succeeded even though passwords did not match
  - Good layout and theme design
  - Refresh token doesn't work

### Question 3

### Development and code quality



- → + 5 pts The application shows limited evidence of a professional approach to development, with ad hoc or unclear organisation of the application and variable levels of code quality.
- Some steps are taken to introduce modular design.

Quite a lot of comments are used to guide the reader, but they tend to focus on low level implementation details, and fail to document the API of each component.

Hard-coded numbers and strings are widely used.

Functions tend to be too long

Components/functions have mixed responsibilities, e.g API calls should be moved to a dedicated location. Application logic, data and presentation are intertwined.

Better use of documentation comments, functions, styles and components would improve the project.

### Question 4

**Report** 16 / 20 pts

Report structure and professionalism

### **UI** Reflection

→ + 6 pts UI reflection demonstrates a near-professional critical analysis of design choices and overall usability, addressing either the points suggested or referencing an equivalent framework

# Accessibility reflection

✓ + 4 pts Accessibility reflection demonstrates a good understanding of the issues and their application in this environment, but may be superficial or unconvincing in parts, or may not reference an appropriate accessibility framework

# **Autograder Results**

### Report

Your report.pdf file was found.

### **Build**

Your static build was found in a directory named dist.

### package.json

Your "package.json" file was found.

### package-lock.json

Your "package-lock.json" file was found.

### Source files

Your source directory was found.

# **Submitted Files**

### ▼ CAP230\_Assignment2\_Krittaya/index.html **♣** Download 1 <!doctype html> 2 <html lang="en"> 3 <head> <meta charset="UTF-8"/> 4 <link rel="icon" type="image/svg+xml" href="/vite.svg" /> 5 6 <meta name="viewport" content="width=device-width, initial-scale=1.0" /> 7 <title>Vite + React</title> </head> 8 9 <body> <div id="root"></div> 10 <script type="module" src="/src/main.jsx"></script> 11 </body> 12 </html> 13 14

# **▼** CAP230\_Assignment2\_Krittaya/vite.config.js

**♣** Download

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'

// https://vite.dev/config/
export default defineConfig({
 plugins: [react()],
 })
```

# ▼ CAP230\_Assignment2\_Krittaya/README.md

**≛** Download

1

# React + Vite

2

This template provides a minimal setup to get React working in Vite with HMR and some ESLint rules.

4 5

Currently, two official plugins are available:

6 7

- [@vitejs/plugin-react](https://github.com/vitejs/vite-plugin-react/blob/main/packages/plugin-react) uses [Babel](https://babeljs.io/) for Fast Refresh

8

- [@vitejs/plugin-react-swc](https://github.com/vitejs/vite-plugin-react/blob/main/packages/plugin-react-swc) uses [SWC](https://swc.rs/) for Fast Refresh

9

## Expanding the ESLint configuration

10 11 12

If you are developing a production application, we recommend using TypeScript with type-aware lint rules enabled. Check out the [TS template](https://github.com/vitejs/vite/tree/main/packages/create-vite/template-react-ts) for information on how to integrate TypeScript and [`typescript-eslint`] (https://typescript-eslint.io) in your project.

13

# **▼** CAP230\_Assignment2\_Krittaya/package-lock.json

**♣** Download

1 Large file hidden. You can download it using the button above.

```
1
     {
2
      "name": "movie-client",
3
      "private": true,
      "version": "0.0.0",
4
5
      "type": "module",
      "scripts": {
6
7
       "dev": "vite",
8
       "build": "vite build",
       "lint": "eslint .",
9
       "preview": "vite preview"
10
11
      },
12
      "dependencies": {
       "ag-grid-community": "^33.2.4",
13
       "ag-grid-enterprise": "^33.2.4",
14
15
       "ag-grid-react": "^33.2.4",
16
       "chart.js": "^4.4.9",
       "react": "^19.1.0",
17
18
       "react-chartjs-2": "^5.3.0",
       "react-dom": "^19.1.0",
19
20
       "react-router-dom": "^7.5.3",
21
       "react-world-flags": "^1.6.0"
22
      },
      "devDependencies": {
23
       "@eslint/js": "^9.25.0",
24
       "@types/react": "^19.1.2",
25
26
       "@types/react-dom": "^19.1.2",
27
       "@vitejs/plugin-react": "^4.4.1",
28
       "eslint": "^9.25.0",
       "eslint-plugin-react-hooks": "^5.2.0",
29
       "eslint-plugin-react-refresh": "^0.4.19",
30
31
       "globals": "^16.0.0",
32
       "vite": "^6.3.5"
33
      }
34
     }
35
```

```
import js from '@eslint/js'
1
     import globals from 'globals'
2
     import reactHooks from 'eslint-plugin-react-hooks'
3
     import reactRefresh from 'eslint-plugin-react-refresh'
4
5
     export default [
6
7
      { ignores: ['dist'] },
8
9
       files: ['**/*.{js,jsx}'],
10
       languageOptions: {
        ecmaVersion: 2020,
11
        globals: globals.browser,
12
13
        parserOptions: {
         ecmaVersion: 'latest',
14
         ecmaFeatures: { jsx: true },
15
16
         sourceType: 'module',
17
        },
18
       },
19
       plugins: {
        'react-hooks': reactHooks,
20
        'react-refresh': reactRefresh,
21
22
       },
23
       rules: {
24
        ...js.configs.recommended.rules,
        ...reactHooks.configs.recommended.rules,
25
        'no-unused-vars': ['error', { varsIgnorePattern: '^[A-Z_]' }],
26
27
        'react-refresh/only-export-components': [
28
         'warn',
29
         { allowConstantExport: true },
30
        ],
31
       },
32
      },
33
     ]
34
```

▼ CAP230_Assignment2_Krittaya/report.pdf	<b>≛</b> Download
Your browser does not support PDF previews. You can <u>download the file instead.</u>	

# ▼ CAP230\_Assignment2\_Krittaya/dist/index.html

**♣** Download

```
1
     <!doctype html>
2
     <html lang="en">
     <head>
3
       <meta charset="UTF-8" />
4
5
       <link rel="icon" type="image/svg+xml" href="/vite.svg" />
       <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6
7
       <title>Vite + React</title>
8
       <script type="module" crossorigin src="/assets/index-_GoSY5hR.js"></script>
       <link rel="stylesheet" crossorigin href="/assets/index-BQOiJvqv.css">
9
10
      </head>
11
      <body>
12
      <div id="root"></div>
13
     </body>
     </html>
14
15
```

# ▼ CAP230\_Assignment2\_Krittaya/dist/vite.svg

**♣** Download

<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" aria-1 hidden="true" role="img" class="iconify iconify--logos" width="31.88" height="32" preserveAspectRatio="xMidYMid meet" viewBox="0 0 256 257"><defs>linearGradient id="lconifyld1813088fe1fbc01fb466" x1="-.828%" x2="57.636%" y1="7.652%" y2="78.411%"><stop offset="0%" stop-color="#41D1FF"></stop><stop offset="100%" stop-color="#BD34FE"></stop> </linearGradient><linearGradient id="IconifyId1813088fe1fbc01fb467" x1="43.376%" x2="50.316%" y1="2.242%" y2="89.03%"><stop offset="0%" stop-color="#FFEA83"></stop><stop offset="8.333%" stop-color="#FFDD35"></stop><stop offset="100%" stop-color="#FFA800"></stop></linearGradient> </defs><path fill="url(#IconifyId1813088fe1fbc01fb466)" d="M255.153 37.938L134.897 252.976c-2.483 4.44-8.862 4.466-11.382.048L.875 37.958c-2.746-4.814 1.371-10.646 6.827-9.67l120.385 21.517a6.537 6.537 0 0 0 2.322-.004l117.867-21.483c5.438-.991 9.574 4.796 6.877 9.62Z"></path><path fill="url(#IconifyId1813088fe1fbc01fb467)" d="M185.432.063L96.44 17.501a3.268 3.268 0 0 0-2.634 3.014l-5.474 92.456a3.268 3.268 0 0 0 3.997 3.378l24.777-5.718c2.318-.535 4.413 1.507 3.936 3.838l-7.361 36.047c-.495 2.426 1.782 4.5 4.151 3.78l15.304-4.649c2.372-.72 4.652 1.36 4.15 3.788l-11.698 56.621c-.732 3.542 3.979 5.473 5.943 2.437l1.313-2.028l72.516-144.72c1.215-2.423-.88-5.186-3.54-4.672l-25.505 4.922c-2.396.462-4.435-1.77-3.759-4.114l16.646-57.705c.677-2.35-1.37-4.583-3.769-4.113Z"></path></svg>

<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" aria-1 hidden="true" role="img" class="iconify iconify--logos" width="31.88" height="32" preserveAspectRatio="xMidYMid meet" viewBox="0 0 256 257"><defs>linearGradient id="lconifyld1813088fe1fbc01fb466" x1="-.828%" x2="57.636%" y1="7.652%" y2="78.411%"><stop offset="0%" stop-color="#41D1FF"></stop><stop offset="100%" stop-color="#BD34FE"></stop> </linearGradient><linearGradient id="IconifyId1813088fe1fbc01fb467" x1="43.376%" x2="50.316%"</pre> y1="2.242%" y2="89.03%"><stop offset="0%" stop-color="#FFEA83"></stop><stop offset="8.333%" stop-color="#FFDD35"></stop><stop offset="100%" stop-color="#FFA800"></stop></linearGradient> </defs><path fill="url(#IconifyId1813088fe1fbc01fb466)" d="M255.153 37.938L134.897 252.976c-2.483 4.44-8.862 4.466-11.382.048L.875 37.958c-2.746-4.814 1.371-10.646 6.827-9.67l120.385 21.517a6.537 6.537 0 0 0 2.322-.004l117.867-21.483c5.438-.991 9.574 4.796 6.877 9.62Z"></path><path fill="url(#IconifyId1813088fe1fbc01fb467)" d="M185.432.063L96.44 17.501a3.268 3.268 0 0 0-2.634 3.014l-5.474 92.456a3.268 3.268 0 0 0 3.997 3.378l24.777-5.718c2.318-.535 4.413 1.507 3.936 3.838l-7.361 36.047c-.495 2.426 1.782 4.5 4.151 3.78l15.304-4.649c2.372-.72 4.652 1.36 4.15 3.788l-11.698 56.621c-.732 3.542 3.979 5.473 5.943 2.437l1.313-2.028l72.516-144.72c1.215-2.423-.88-5.186-3.54-4.672l-25.505 4.922c-2.396.462-4.435-1.77-3.759-4.114l16.646-57.705c.677-2.35-1.37-4.583-3.769-4.113Z"></path></svg>

```
1
     #root {
      /* max-width: 1280px; */
2
      /* margin: 0 auto; */
3
      /* padding: 2rem; */
4
      text-align: center;
5
6
7
     }
8
     * {
      font-family: 'Signika', sans-serif;
9
10
11
     }
12
     .logo {
13
      height: 10em;
14
      padding: 1.5em;
15
      will-change: filter;
16
      transition: filter 300ms;
17
     }
18
     .logo:hover {
      filter: drop-shadow(0 0 2em #64f5ffaa);
19
20
21
     .logo.react:hover {
      filter: drop-shadow(0 0 2em #7361fbaa);
22
23
     }
24
25
     @keyframes logo-spin {
26
      from {
27
       transform: rotate(0deg);
28
      }
29
      to {
30
       transform: rotate(360deg);
31
      }
32
     }
33
34
     @media (prefers-reduced-motion: no-preference) {
35
      a:nth-of-type(2) .logo {
36
       animation: logo-spin infinite 20s linear;
37
      }
38
     }
39
40
     .card {
41
      padding: 2em;
42
     }
43
44
     .read-the-docs {
45
      color: #888;
46
     }
     /* App.css */
47
     .navbar {
48
49
      background-color: #fcc5ff;
```

```
50
      /* padding-top: 18px;
51
      padding-bottom: 18px; */
      font-size: 20px;
52
      text-align: left;
53
54
      display: flex;
55
     }
56
57
     .navbar-email {
      font-size: 20px;
58
59
      color: #a846be;
      margin-left: 10px;
60
61
      margin-top: 18px;
62
      text-align: right;
      font-family: 'Vollkorn', serif;
63
64
65
     }
66
     .navbar-link {
67
      /* margin-right: 25px;
68
      margin-left: 25px; */
69
70
      color: #a846be;
71
      padding: 18px 30px;
      font-weight: 700;
72
73
      font-family: 'Vollkorn', serif;
74
75
     .navbar-link:hover {
76
      background-color: #a846be;
77
      color: #fcc5ff;
78
      text-decoration: underline;
79
     }
     .navbar-nav {
80
81
      display: flex;
      justify-content: space-between;
82
83
     }
84
85
86
     .navbar-brand {
87
      font-size: 1.5rem;
     }
88
89
90
     .navbar-light {
91
      background-color: #f8f9fa;
92
     }
93
94
95
```

```
1
2
     /* Vollkorn font */
     @import url('https://fonts.googleapis.com/css2?family=Vollkorn:wght@400;700&display=swap');
3
4
5
     /* Signika font */
     @import url('https://fonts.googleapis.com/css2?family=Signika:wght@400;600&display=swap');
6
7
8
     :root {
      font-family: system-ui, Avenir, Helvetica, Arial, sans-serif;
9
      line-height: 1.5;
10
      font-weight: 400;
11
12
13
      color-scheme: light dark;
      color: rgba(32, 13, 52, 0.87);
14
      background-color: #f3f3f3;
15
16
      font-synthesis: none;
17
      text-rendering: optimizeLegibility;
18
      -webkit-font-smoothing: antialiased;
19
      -moz-osx-font-smoothing: grayscale;
20
21
     /* Home and movies */
22
23
     a {
24
      font-weight: 500;
      color: #e364ff;
25
26
      text-decoration: inherit;
27
28
29
     a:hover {
30
      color: #e7b7ff;
31
     }
32
33
     body {
      margin: 0;
34
      /* display: flex;
35
36
      place-items: center; */
37
      min-width: 320px;
38
      min-height: 100vh;
39
40
     }
41
42
     h1 {
43
      font-size: 2.5em;
44
      line-height: 1.1;
45
      color: rgb(81, 40, 130);
46
      font-family: 'Vollkorn', serif;
47
48
     /* search button*/
49
     button {
```

```
border-radius: 8px;
50
51
      border: 1px solid transparent;
      padding: 0.6em 1.2em;
52
      font-size: 1em;
53
      font-weight: 500;
54
      font-family: inherit;
55
      background-color: #e3b3ff;
56
      cursor: pointer;
57
      transition: border-color 0.25s;
58
59
60
     }
61
62
     /* Button for movie */
     button:hover {
63
64
      border-color: #9e2ff3;
65
     button:focus,
66
     button:focus-visible {
67
      outline: 4px auto -webkit-focus-ring-color;
68
69
70
     }
71
72
73
```

# ▼ CAP230\_Assignment2\_Krittaya/src/main.jsx

**♣** Download

```
import { StrictMode } from 'react'
1
    import { createRoot } from 'react-dom/client'
2
    import { BrowserRouter } from 'react-router-dom'
3
4
    import './index.css'
5
    import App from './App.jsx'
6
7
    createRoot(document.getElementById('root')).render(
8
     <StrictMode>
9
       <BrowserRouter>
10
        <App />
       </BrowserRouter>
11
12
     </StrictMode>
13
14
```

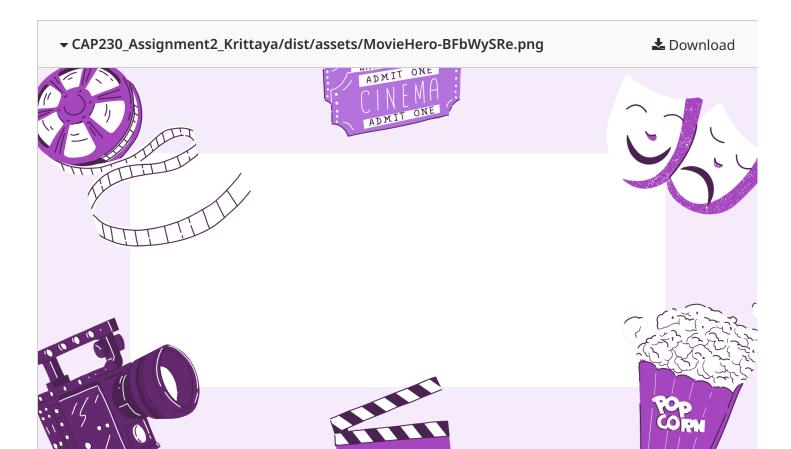
# ▼ CAP230\_Assignment2\_Krittaya/src/App.jsx

```
1
     import React, { useState, useEffect } from 'react';
2
     import { Routes, Route, Link, useNavigate } from 'react-router-dom';
     import HomePage from './pages/HomePage.jsx';
3
4
     import MoviesPage from './pages/MoviesPage.jsx';
5
     import MovieDetailPage from './pages/MovieDetailPage.jsx';
6
     import PersonDetailPage from './pages/PersonDetailPage.jsx';
7
     import LoginPage from './pages/LoginPage.jsx';
8
     import RegisterPage from './pages/RegisterPage.jsx';
     import { getToken, getRefreshToken, clearToken, setToken } from './utils/auth'; // Import functions
9
10
     import './App.css';
11
12
     function App() {
      const [isAuthenticated, setIsAuthenticated] = useState(false);
13
14
      const [email, setEmail] = useState(localStorage.getItem("email"));
15
      const [loading, setLoading] = useState(true);
16
      const navigate = useNavigate();
17
18
      // Function to check if the token is expired
      const isTokenExpired = (token) => {
19
20
       if (!token) return true;
21
22
       const parts = token.split('.');
23
       if (parts.length !== 3) return true; // Invalid JWT
24
25
       const payload = |SON.parse(atob(parts[1])); // Decode the token's payload
26
       const expirationTime = payload.exp * 1000; // Expiry time is in seconds, convert to milliseconds
27
28
       return Date.now() >= expirationTime; // Compare expiration time with current time
29
      };
30
31
      // Handle user logout
32
      const handleLogout = async () => {
33
       const refreshToken = localStorage.getItem("refreshToken");
34
35
       if (!refreshToken) {
36
        // If no refresh token, force the user to log in again
37
        console.log("No refresh token found, forcing re-login.");
38
        clearToken();
39
        setIsAuthenticated(false);
        setEmail(null);
40
41
        localStorage.removeItem("email");
42
        navigate("/login");
43
        return;
44
       }
45
46
       try {
47
        const response = await fetch("http://4.237.58.241:3000/user/logout", {
         method: "POST",
48
49
         headers: {
```

```
50
           "Content-Type": "application/json",
51
          },
          body: JSON.stringify({ refreshToken }),
52
53
54
         if (!response.ok) {
55
          // If the logout request fails due to an expired token, force the user to log in again
56
          console.error("Failed to logout due to expired token, forcing re-login.");
57
58
          clearToken(); // Clear tokens from localStorage
59
          setIsAuthenticated(false);
60
          setEmail(null);
61
62
          localStorage.removeItem("email");
63
64
          // Redirect to login page
          navigate("/login");
65
66
          return;
67
         }
68
         // If logout is successful (i.e., refresh token was valid), clear tokens
69
70
         clearToken();
71
         setIsAuthenticated(false);
72
         setEmail(null);
73
         localStorage.removeItem("email");
74
75
         navigate("/login"); // Redirect to login page
76
        } catch (error) {
77
         console.error("Logout error:", error);
78
         clearToken(); // Clear tokens in case of any error
79
         navigate("/login"); // Redirect to login page
80
        }
81
      };
82
       // Function to refresh the token using the refresh token
83
       const refreshToken = async () => {
84
85
        const refreshToken = getRefreshToken();
86
        if (!refreshToken) {
87
         clearToken();
88
         return;
89
        }
90
91
92
         const res = await fetch("http://4.237.58.241:3000/user/refresh", {
93
          method: "POST",
94
          headers: {
95
           "Content-Type": "application/json",
96
          },
97
          body: |SON.stringify({ refreshToken }),
98
         });
99
100
         const data = await res.json();
101
         if (res.ok && data.bearerToken && data.refreshToken) {
```

```
102
         setToken(data.bearerToken.token, data.refreshToken.token);
103
         setIsAuthenticated(true);
104
        } else {
         clearToken();
105
106
        }
107
       } catch (err) {
108
       console.error("Token refresh error:", err);
109
        clearToken();
110
       }
111
      };
112
      // Check if the user is authenticated (i.e., if a token exists)
113
114
      useEffect(() => {
115
      const token = getToken();
116
      if (token) {
117
       setIsAuthenticated(true); // Token exists, user is authenticated
118
       }
       setLoading(false); // Stop loading when the check is complete
119
120
121
122
      if (loading) return <div>Loading...</div>; // You can show a loading spinner here.
123
124
      return (
125
      <div>
126
        <nav className="navbar">
127
         <Link className="navbar-link" to="/">Home</Link>
128
         <Link className="navbar-link" to="/movies">Movies</Link>
129
         {isAuthenticated?(
130
          <>
131
           <Link className="navbar-link" to="/login" onClick={handleLogout}>
132
            Logout
           </Link>
133
134
           135
          </>
136
         ):(
137
           <Link className="navbar-link" to="/login">Login</Link>
138
139
           <Link className="navbar-link" to="/register">Register</Link>
140
          </>
141
         )}
142
        </nav>
143
144
        <Routes>
145
         <Route path="/" element={<HomePage />} />
         <Route path="/movies" element={<MoviesPage />} />
146
147
         <Route path="/movies/:imdbID" element={<MovieDetailPage />} />
148
         <Route path="/people/:id" element={<PersonDetailPage />} />
         <Route path="/login" element={<LoginPage />} />
149
150
         <Route path="/register" element={<RegisterPage />} />
151
        </Routes>
        {/* Footer Section */}
152
153
        <footer className="footer">
```

# ▼ CAP230\_Assignment2\_Krittaya/dist/assets/MoviepageBG-DZbmksjN.png Labor ONE ADXIT ONE ADXIT ONE ADXIT ONE ADXIT ONE



# ▼ CAP230\_Assignment2\_Krittaya/dist/assets/index-BQOiJvqv.css

**≛** Download

# ▼ CAP230\_Assignment2\_Krittaya/dist/assets/index-\_GoSY5hR.js

**▲** Download

1 Large file hidden. You can download it using the button above.

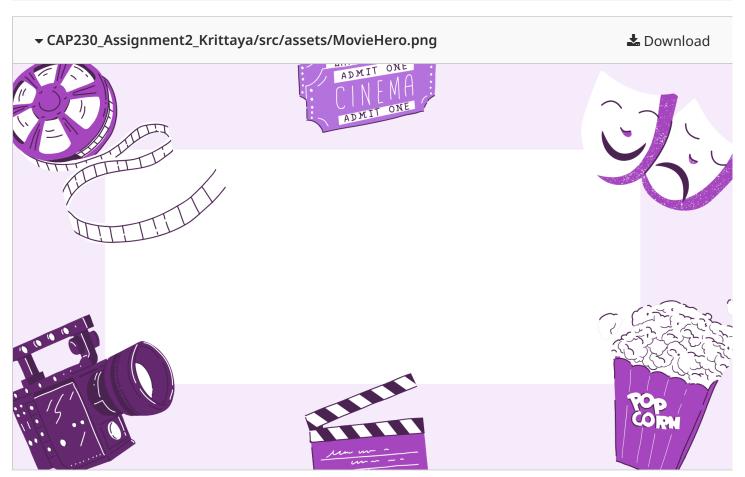
# ▼ CAP230\_Assignment2\_Krittaya/src/utils/auth.js

```
// Save the token and refresh token to localStorage
1
2
     export function setToken(token, refreshToken) {
       if (!token | | !refreshToken) {
3
        console.error("setToken error: Both token and refreshToken must be provided.");
4
5
        return;
6
       }
7
       localStorage.setItem("token", token);
8
       localStorage.setItem("refreshToken", refreshToken);
9
       console.log("Tokens saved:", token, refreshToken); // Log tokens for debugging
10
11
12
      }
13
      // Retrieve the bearer token from localStorage
14
15
      export function getToken() {
16
       const token = localStorage.getItem("token");
17
       if (!token) {
18
        console.log("getToken : No token found in localStorage.");
19
       }
20
       return token;
21
      }
22
23
      // Retrieve the refresh token from localStorage
24
      export function getRefreshToken() {
       const refreshToken = localStorage.getItem("refreshToken");
25
26
       if (!refreshToken) {
27
        console.error("getRefreshToken error: No refresh token found in localStorage.");
28
       }
29
       return refreshToken;
30
31
32
      // Clear both the token and refresh token from localStorage
33
      export function clearToken() {
34
       localStorage.removeItem("token");
35
       localStorage.removeItem("refreshToken");
36
      }
37
```

```
1
     // utils/fetchWithAuth.js
2
     export const fetchWithAuth = async (url, options = {}) => {
3
4
       const token = localStorage.getItem('token');
5
6
       if (!token) {
7
        throw new Error('No token found');
8
       }
9
10
       let response = await fetch(url, {
11
        ...options,
12
        headers: {
         ...options.headers,
13
14
         Authorization: `Bearer ${token}`,
15
        },
16
       });
17
18
       // If the token is expired (401 Unauthorized), refresh the token
       if (response.status === 401) {
19
20
        console.log('Token expired, refreshing...');
21
        await refreshToken(); // Call refresh function
22
23
        // Retry the request with the new token
24
        const newToken = localStorage.getItem('token');
25
        response = await fetch(url, {
26
         ...options,
          headers: {
27
28
           ...options.headers,
29
           Authorization: `Bearer ${newToken}`,
30
         },
31
        });
32
       }
33
       return response.json();
34
35
      };
36
     const refreshToken = async () => {
37
38
       const refreshToken = localStorage.getItem('refreshToken');
39
40
       if (!refreshToken) {
41
        console.error('No refresh token found');
42
        return;
43
       }
44
45
       try {
46
        const res = await fetch('http://4.237.58.241:3000/user/refresh', {
47
          method: 'POST',
48
          headers: {
49
           'Content-Type': 'application/json',
```

```
50
         body: JSON.stringify({
51
          refreshToken,
52
53
         }),
        });
54
55
56
        const data = await res.json();
57
        if (res.ok && data.token && data.refreshToken) {
58
         setToken(data.token, data.refreshToken); // Save the new token
59
        } else {
60
         console.error('Failed to refresh token:', data);
61
         clearToken(); // Clear localStorage and force re-login
62
63
        }
64
       } catch (err) {
        console.error('Error refreshing token:', err);
65
        clearToken(); // Clear localStorage and force re-login
66
       }
67
     };
68
69
     const setToken = (token, refreshToken) => {
70
       localStorage.setItem('token', token);
71
       localStorage.setItem('refreshToken', refreshToken);
72
       console.log('Tokens saved:', token, refreshToken); // Log tokens for debugging
73
74
     };
75
     const clearToken = () => {
76
       localStorage.removeItem('token');
77
       localStorage.removeItem('refreshToken');
78
       console.log("Tokens cleared");
79
80
     };
81
```





1

<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" ariahidden="true" role="img" class="iconify iconify--logos" width="35.93" height="32" preserveAspectRatio="xMidYMid meet" viewBox="0 0 256 228"><path fill="#00D8FF" d="M210.483" 73.824a171.49 171.49 0 0 0-8.24-2.597c.465-1.9.893-3.777 1.273-5.621c6.238-30.281 2.16-54.676-11.769-62.708c-13.355-7.7-35.196.329-57.254 19.526a171.23 171.23 0 0 0-6.375 5.848a155.866 155.866 0 0 0-4.241-3.917C100.759 3.829 77.587-4.822 63.673 3.233C50.33 10.957 46.379 33.89 51.995 62.588a170.974 170.974 0 0 0 1.892 8.48c-3.28.932-6.445 1.924-9.474 2.98C17.309 83.498 0 98.307 0 113.668c0 15.865 18.582 31.778 46.812 41.427a145.52 145.52 0 0 0 6.921 2.165a167.467 167.467 0 0 0-2.01 9.138c-5.354 28.2-1.173 50.591 12.134 58.266c13.744 7.926 36.812-.22 59.273-19.855a145.567 145.567 0 0 0 5.342-4.923a168.064 168.064 0 0 0 6.92 6.314c21.758 18.722 43.246 26.282 56.54 18.586c13.731-7.949 18.194-32.003 12.4-61.268a145.016 145.016 0 0 0-1.535-6.842c1.62-.48 3.21-.974 4.76-1.488c29.348-9.723 48.443-25.443 48.443-41.52c0-15.417-17.868-30.326-45.517-39.844Zm-6.365 70.984c-1.4.463-2.836.91-4.3 1.345c-3.24-10.257-7.612-21.163-12.963-32.432c5.106-11 9.31-21.767 12.459-31.957c2.619.758 5.16 1.557 7.61 2.4c23.69 8.156 38.14 20.213 38.14 29.504c0 9.896-15.606 22.743-40.946 31.14Zm-10.514 20.834c2.562 12.94 2.927 24.64 1.23 33.787c-1.524 8.219-4.59 13.698-8.382 15.893c-8.067 4.67-25.32-1.4-43.927-17.412a156.726 156.726 0 0 1-6.437-5.87c7.214-7.889 14.423-17.06 21.459-27.246c12.376-1.098 24.068-2.894 34.671-5.345a134.17 134.17 0 0 1 1.386 6.193ZM87.276 214.515c-7.882 2.783-14.16 2.863-17.955.675c-8.075-4.657-11.432-22.636-6.853-46.752a156.923 156.923 0 0 1 1.869-8.499c10.486 2.32 22.093 3.988 34.498 4.994c7.084 9.967 14.501 19.128 21.976 27.15a134.668 134.668 0 0 1-4.877 4.492c-9.933 8.682-19.886 14.842-28.658 17.94ZM50.35 144.747c-12.483-4.267-22.792-9.812-29.858-15.863c-6.35-5.437-9.555-10.836-9.555-15.216c0-9.322 13.897-21.212 37.076-29.293c2.813-.98 5.757-1.905 8.812-2.773c3.204 10.42 7.406 21.315 12.477 32.332c-5.137 11.18-9.399 22.249-12.634 32.792a134.718 134.718 0 0 1-6.318-1.979Zm12.378-84.26c-4.811-24.587-1.616-43.134 6.425-47.789c8.564-4.958 27.502 2.111 47.463 19.835a144.318 144.318 0 0 1 3.841 3.545c-7.438 7.987-14.787 17.08-21.808 26.988c-12.04 1.116-23.565 2.908-34.161 5.309a160.342 160.342 0 0 1-1.76-7.887Zm110.427 27.268a347.8 347.8 0 0 0-7.785-12.803c8.168 1.033 15.994 2.404 23.343 4.08c-2.206 7.072-4.956 14.465-8.193 22.045a381.151 381.151 0 0 0-7.365-13.322Zm-45.032-43.861c5.044 5.465 10.096 11.566 15.065 18.186a322.04 322.04 0 0 0-30.257-.006c4.974-6.559 10.069-12.652 15.192-18.18ZM82.802 87.83a323.167 323.167 0 0 0-7.227 13.238c-3.184-7.553-5.909-14.98-8.134-22.152c7.304-1.634 15.093-2.97 23.209-3.984a321.524 321.524 0 0 0-7.848 12.897Zm8.081 65.352c-8.385-.936-16.291-2.203-23.593-3.793c2.26-7.3 5.045-14.885 8.298-22.6a321.187 321.187 0 0 0 7.257 13.246c2.594 4.48 5.28 8.868 8.038 13.147Zm37.542 31.03c-5.184-5.592-10.354-11.779-15.403-18.433c4.902.192 9.899.29 14.978.29c5.218 0 10.376-.117 15.453-.343c-4.985 6.774-10.018 12.97-15.028 18.486Zm52.198-57.817c3.422 7.8 6.306 15.345 8.596 22.52c-7.422 1.694-15.436 3.058-23.88 4.071a382.417 382.417 0 0 0 7.859-13.026a347.403 347.403 0 0 0 7.425-13.565Zm-16.898 8.101a358.557 358.557 0 0 1-12.281 19.815a329.4 329.4 0 0 1-23.444.823c-7.967 0-15.716-.248-23.178-.732a310.202 310.202 0 0 1-12.513-19.846h.001a307.41 307.41 0 0 1-10.923-20.627a310.278 310.278 0 0 1 10.89-20.637l-.001.001a307.318 307.318 0 0 1 12.413-19.761c7.613-.576 15.42-.876 23.31-.876H128c7.926 0 15.743.303 23.354.883a329.357 329.357 0 0 1 12.335 19.695a358.489 358.489 0 0 1 11.036 20.54a329.472 329.472 0 0 1-11 20.722Zm22.56-122.124c8.572 4.944 11.906 24.881 6.52 51.026c-.344 1.668-.73 3.367-1.15 5.09c-10.622-2.452-22.155-4.275-34.23-5.408c-7.034-10.017-14.323-19.124-21.64-27.008a160.789 160.789 0 0 1 5.888-5.4c18.9-16.447 36.564-22.941 44.612-18.3ZM128 90.808c12.625 0 22.86 10.235 22.86 22.86s-10.235 22.86-22.86 22.86s-22.86 10.235-22.86-22.86s10.235-22.86 22.86-22.86Z"></path></svg>

# ▼ CAP230\_Assignment2\_Krittaya/src/pages/HomePage.jsx

**▲** Download

```
import heroImage from "../assets/MovieHero.png"; // Adjust the path to your image
1
    import './style.css';
2
3
4
    function HomePage() {
5
     return (
      <div className="hero-section">
6
      <div className="hero-heading">
7
      <h1>Welcome Movie Search Website!</h1>
8
      <h1>By Krittya Kruapat</h1>
9
      Thank you for visiting! Enjoy browsing and discovering something new!
10
      </div>
11
      </div>
12
13
     );
14
15
     export default HomePage;
16
```

```
1
     import React, { useState, useEffect } from "react";
2
     import { useNavigate } from "react-router-dom";
3
     import { AgGridReact } from "ag-grid-react";
4
     import { ModuleRegistry } from "ag-grid-community";
5
     import {
6
      ClientSideRowModelModule,
7
      PaginationModule,
8
      ValidationModule
9
     } from "ag-grid-community";
10
11
     import "ag-grid-community/styles/ag-theme-alpine.css";
12
     import './style.css';
13
14
15
     // Register modules
16
     ModuleRegistry.registerModules([
17
      ClientSideRowModelModule,
18
      PaginationModule,
      ValidationModule
19
20
     ]);
21
22
     function MoviesPage() {
23
      const [query, setQuery] = useState(""); // For title search
24
      const [selectedYear, setSelectedYear] = useState(""); // For year dropdown
25
      const [filteredMovies, setFilteredMovies] = useState([]);
26
      const [allMovies, setAllMovies] = useState([]); // Store all movies for later filtering
27
      const navigate = useNavigate(); // For navigation on row click
28
29
      // Fetch all movies initially to populate the year dropdown
30
      useEffect(() => {
31
       async function fetchMovies() {
32
        try {
33
         const res = await fetch("http://4.237.58.241:3000/movies/search", {
           headers: { Accept: "application/json" }
34
35
         });
36
         const result = await res.json();
37
38
         const data = Array.isArray(result) ? result : result.data | | [];
39
         setAllMovies(data); // Store all fetched movies
40
41
         setFilteredMovies(data); // Initially, display all movies
42
         console.log("First movie object:", data[0]);
43
44
        } catch (error) {
45
         console.error("Failed to fetch movies:", error);
46
        }
47
       }
48
49
       fetchMovies();
```

```
50
      }, []);
51
      // Get unique years for the dropdown, convert to numbers for consistency
52
      const years = [...new Set(allMovies.map((movie) => movie.year))].sort();
53
54
55
      // Ensure the year is a number for comparison
      const handleYearChange = (e) => {
56
57
       setSelectedYear(e.target.value ? Number(e.target.value) : "");
58
      };
59
      const searchMovies = () => {
60
       // Filter based on query and selected year
61
       const filtered = allMovies.filter((movie) => {
62
        const matchesTitle = movie.title?.toLowerCase().includes(query.toLowerCase());
63
64
        const matchesYear = selectedYear ? Number(movie.year) === selectedYear : true; // Convert
     movie.year to a number
        return matchesTitle && matchesYear:
65
66
       });
67
68
       setFilteredMovies(filtered);
69
      };
70
      const columnDefs = [
71
72
       { field: "title", headerName: "Title", sortable: true, flex: 2 },
       { field: "year", headerName: "Year", sortable: true, flex: 1 },
73
       { field: "imdbRating", headerName: "IMDB Rating", sortable: true, flex: 1 } // Change 'rank' to
74
     'imdbRating'
75
      1;
76
77
      const handleRowClick = (event) => {
78
       const imdbID = event.data.imdbID;
79
       if (imdbID) {
        navigate(`/movies/${imdbID}`);
80
81
       }
82
      };
83
84
      return (
       <div className="movies-search">
85
        <h1>Movie Search</h1>
86
87
        <div className="search-bar-wrapper">
88
         <input
          type="text"
89
90
          value={query}
91
          onChange={(e) => setQuery(e.target.value)}
          placeholder="Enter movie title"
92
93
         />
        </div>
94
95
        {/* Dropdown for year selection */}
96
97
        <div>
          <select
98
99
          value={selectedYear}
```

```
100
          onChange={handleYearChange}
101
         >
          <option value="">Select Year</option>
102
103
          {years.map((year) => (
104
           <option key={year} value={year}>
            {year}
105
106
           </option>
107
          ))}
108
         </select>
109
        </div>
110
111
        <button onClick={searchMovies}>Search</button>
112
113
        <div className="ag-theme-alpine ag-grid-container">
114
        {/* style={{ height: 400, marginTop: 20 }} */}
115
         <AgGridReact
          rowData={filteredMovies}
116
117
          columnDefs={columnDefs}
          pagination={true}
118
119
          paginationPageSize={10}
120
          paginationPageSizeSelector={[10, 20, 50, 100]} // Add 10 to the selector options
          onRowClicked={handleRowClick}
121
122
         />
123
        </div>
124
       </div>
125
      );
126 }
127
     export default MoviesPage;
128
129
```

# ▼ CAP230\_Assignment2\_Krittaya/src/pages/style.css

```
/* MovieDetailPage */
1
2
     .movie-detail-container {
3
      color: rgb(81, 40, 130);
4
      padding: 20px;
5
      max-width: 1200px;
6
      margin: auto;
7
      text-align: center; /* Ensure everything inside is centered */
8
      display: flex;
      flex-direction: column;
9
      align-items: center; /* Ensures centering of h2 */
10
11
     }
12
13
14
     .movie-detail-1{
15
      font-size: 18px;
16
     }
     .key-people {
17
18
      text-align: center;
      font-size: 24px;
19
20
      margin-top: 20px;
21
      width: 100%; /* Make sure the element takes full width */
      display: block;
22
23
     }
24
      .rating {
25
       font-size: 15px;
26
       padding: 0px 18px 10px;
27
       border: #b673de;
28
       border: 2px solid #ccc; /* Add a solid 2px border with light gray color */
29
       border-radius: 8px; /* Optional: Add rounded corners */
       background-color: #f9f9f9; /* Optional: Add a background color */
30
31
      }
32
      .movie-poster {
33
       width: 200px;
34
       margin-bottom: 20px;
35
      }
36
     .movie-detail{
37
      max-width: 1000px;
38
      margin-left:60px;
39
      margin-right:60px;
40
41
     }
42
43
      .key-people-grid {
44
       margin-top: 10px;
45
       font-size: 16px;
46
       border-radius: 8px;
47
       overflow: hidden;
48
       box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
49
       width: 90%; /* Ensure it takes the full available width */
```

```
50
      }
51
52
      .plot {
      font-style: italic;
53
      font-size: 18px;
54
55
56
57
      .key-people-grid table {
       width: 100%;
58
       border-collapse: collapse;
59
        background-color: #ffc4fb;
60
       font-weight: 600;
61
62
63
      }
64
      .key-people-grid table thead{
       color: #4b2c6f;
65
       font-weight: 800;
66
       font-size: larger;
67
68
      }
69
70
      .key-people-grid th, .key-people-grid td {
71
        padding: 12px;
72
       text-align: left;
73
       border-bottom: 1px solid #ca9bff;
74
75
      .key-people-grid th {
76
       background-color: #b673de;
77
78
79
      }
80
81
       .key-people-grid tr:nth-child(even) {
       background-color: #ca9bff; /* Default color for even rows */
82
83
      }
84
85
86
       .key-people-grid tr:nth-child(odd) {
87
       background-color: #ffc4fb;
      }
88
89
90
91
       .key-people-grid tr:hover {
       background-color: rgba(255, 255, 255, 0.719);
92
93
94
95
       .key-people-grid a {
       color: #642c70;
96
       font-weight: 800;
97
98
      }
99
100
       .key-people-grid a:hover {
       text-decoration: underline;
101
```

```
102
       color: #7f2d91;
103
104
105
      .movie-detail-people a{
106
       color: #642c70;
107
       text-decoration: 'none';
108
109
      }
110
      .movies-search h1 {
111
       font-size: 2.2em;
112
       font-weight: 800;
113
      }
114
115
116
      @media (min-width: 750px) {
117
118
       .movie-poster {
119
       margin-right: 20px;
        float: left;
120
121
       }
122
123
       .movie-detail-1 {
        /* display: block; */
124
125
        text-align: left;
126
        margin-left: 20px;
        justify-content: left;
127
128
129
       }
130
       .rating{
131
       margin-right: 20px;
132
        float: right;
133
       }
134
       .rating p{
135
        text-align: right;
136
       }
137
      }
138
139
140
      /* MoviePage */
141
142
     .movies-search{
143
      height: 100vh; /* Full height of the page */
144
      width: 100%; /* Full width of the page */
145
      background-image: url('../assets/MoviepageBG.png'); /* Background image path */
      background-size: cover; /* Makes sure the background covers the entire container */
146
147
      background-position: center center; /* Centers the background */
      background-repeat: no-repeat; /* Ensures the image doesn't repeat */
148
149
      padding-top: 40px;
150
      padding-bottom: 40px;
151
     }
152
     .movies-search h1{
153
```

```
154
      margin-top: 0;
155 }
156 /* Custom styles for ag-grid container */
     .ag-grid-container {
157
158
      height: 500px;
159
      margin-top: 40px;
160
      margin-bottom: 60px;
161
      padding-left:10%;
162
      padding-right:10%;
163
164
     }
165
     /* Custom styles for ag-Grid pagination */
166
     .ag-theme-alpine .ag-paging-panel {
167
168
      background-color: #b673de; /* Set background color of the pagination bar */
      color: #330e3b; /* Set text color */
169
170
      border-radius: 8px;
      padding: 10px;
171
172
      display: flex;
      justify-content: center;
173
174
      align-items: center;
175
      font-size: 16px;
176
     }
177
178
     /* Pagination controls (next, previous, page numbers) */
179
     .ag-theme-alpine .ag-paging-panel .ag-paging-button,
180
     .ag-theme-alpine .ag-paging-panel .ag-page-size-panel select {
181
      background-color: #642c70; /* Button background */
      color: #fff; /* Button text color */
182
183
      border: 2px solid #ca9bff;
184
      border-radius: 8px;
185
      padding: 5px 10px;
186
      margin: 0 5px;
187
      font-weight: 800;
188
      cursor: pointer;
189
190
     }
191
192
     /* Pagination buttons on hover */
193
     .ag-theme-alpine .ag-paging-panel .ag-paging-button:hover,
     .ag-theme-alpine .ag-theme-alpine .ag-paging-panel .ag-page-size-panel select:hover {
194
195
      background-color: #7f2d91; /* Change color on hover */
      border-color: #fff;
196
      text-decoration: underline;
197
198
199
200
     }
201
202
     /* Active page button */
203
     .ag-theme-alpine .ag-paging-panel .ag-paging-button[aria-pressed="true"] {
204
      background-color: #ca9bff;
205
      color: #4b2c6f; /* Color for the active button */
```

```
206
      border-color: #b673de;
207
208 }
209
210
     /* Custom page size dropdown */
     .ag-theme-alpine .ag-paging-panel .ag-page-size-panel select {
211
212
      background-color: #b673de;
213
      color: #fff:
214
      border-radius: 8px;
      /* border: 2px solid #ca9bff; */
215
216
      padding: 5px;
217
      font-weight: 600;
      text-decoration: underline;
218
219
220 }
221
222
     .ag-theme-alpine .ag-paging-panel .ag-page-size-panel select:hover {
223
      background-color: #7f2d91;
224
      border-color: #fff;
225
      text-decoration: underline;
226
227
     }
228
229
     /* Pagination dropdown on focus */
230
     .ag-theme-alpine .ag-paging-panel .ag-page-size-panel select:focus {
231
      outline: none;
232
      box-shadow: 0 0 5px rgba(75, 75, 75, 0.5);
233
      text-decoration: underline;
234
235
236 }
237
     /* Add custom styles for ag-Grid */
238
239
     .ag-theme-alpine {
240
      --ag-border-color: #ca9bff;
      --ag-header-background-color: #b673de;
241
242
      --ag-header-font-weight: 1000;
243
      --ag-header-font-size: 18px;
244
      --ag-header-text-align: center;
245
      --ag-font-family: 'Arial', sans-serif;
246
      --ag-font-size: 17px;
247
      --ag-font-color: #4b2c6f;
      --ag-row-hover-color: rgba(255, 255, 255, 0.719);
248
249
      --ag-row-border-radius: 8px;
250
      --ag-background-color: #ca9bff;
251
252 }
253
254
     .ag-theme-alpine .ag-header-cell {
255
      color: #4b2c6f;
256
257 }
```

```
258
259
     .ag-theme-alpine .ag-cell {
260
      color: rgb(81, 40, 130);
261
262
     }
263
264
     .ag-theme-alpine .ag-row-even {
265
      background-color: #ca9bff;
      padding-top: 2px;
266
      padding-bottom: 2px;
267
268
      font-weight: 600;
269
270
     }
271
272
     .ag-theme-alpine .ag-row-odd {
273
      background-color: #ffc4fb;
274
      padding-top: 2px;
275
      padding-bottom: 2px;
276
      font-weight: 600;
277
278
279
280
     .ag-theme-alpine .ag-row:hover {
      background-color: rgba(255, 255, 255, 0.719);
281
282
      color: #7f2d91;
283
284
285
     .ag-theme-alpine .ag-cell:hover {
286
      color: #7f2d91;
287
      text-decoration: underline;
288
289
     }
290
291
     .ag-theme-alpine .ag-cell a {
292
      color: #33133a;
293
      font-weight: 1000;
294
295
     .ag-theme-alpine .ag-cell a:hover {
296
297
      text-decoration: underline;
298
      color: #7f2d91;
299
     }
300
301
302
303
304
     /*login page and register page*/
305
     /* Styling for the movie title input field */
     input[type="text"], input[type="email"], input[type="password"] {
306
307
      background-color: #f9f9f9;
308
      color: #642c70;
309
      border: 2px solid #b673de;
```

```
310
      padding: 10px;
311
      font-size: 16px;
312
      border-radius: 8px;
313
      width: 250px;
314
      margin-bottom: 10px;
     }
315
316
317
     input[type="text"]:focus, input[type="email"]:focus, input[type="password"]:focus {
318
      outline: none;
319
      border-color: #7f2d91;
      background-color: #fff;
320
321
     }
322
323
324
     /* Styling for the year select dropdown */
325
     select {
326
      background-color: #f9f9f9;
327
      color: #642c70;
328
      border: 2px solid #b673de;
329
      padding: 10px;
330
      font-size: 16px;
331
      border-radius: 8px;
332
      width: 250px;
333
      margin-bottom: 10px;
334 }
335
     select:focus {
336
337
      outline: none;
338
      border-color: #7f2d91;
339
      background-color: #fff;
340
     }
341
     /* Styling for the Search button */
342
343
     button {
344
      background-color: #b673de;
345
      color: #fff;
346
      border: 2px solid #7f2d91;
347
      padding: 10px 20px;
348
      font-size: 16px;
349
      font-weight: bold;
350
      border-radius: 8px;
351
      cursor: pointer;
352
      width: 250px;
353
      margin-top: 10px;
354
     }
355
     button:hover {
356
357
      background-color: #7f2d91;
      border-color: #fff;
358
359
      color: #fff;
360 }
361
```

```
/* Wrapper for the search bar */
362
363
     .search-bar-wrapper {
364
     text-align: center;
365
     margin-top: 30px;
366
    }
367
368
     .movies-search {
369
     /* margin-top: 20px; */
370
     font-size: 24px;
371
      color: #512882;
372
     font-weight: 800;
373
    }
374
375
     .login-form {
376
      display: flex;
      flex-direction: column; /* Stack items vertically */
377
378
      align-items: center;
379
      gap: 5px;
380
      width: 250px;
381
      margin: 0 auto;
382
    }
383
384
     .login-form input, .login-form button {
385
      width: 100%;
386
      padding: 10px;
387
      font-size: 16px;
388
      border-radius: 8px;
389
    }
390
     /* ______*/
391
     /* Homepage */
392
393
     .hero-section {
394
      display: flex;
395
      justify-content: center;
396
      align-items: center; /* Aligns the text vertically in the center */
      height: 850px; /* 100% of the viewport height */
397
398
      width: 100%; /* Full width of the viewport */
399
      background-image: url('../assets/MovieHero.png'); /* Set the image as the background */
400
      background-size: cover; /* Makes the image cover the entire container */
401
      background-position: center center; /* Centers the background image */
402
      background-repeat: no-repeat; /* Prevents the image from repeating */
403
     }
404
405
     .hero-section h1{
406
      /* display: flex; */
407
      /* justify-content: center; */
408
      align-items: center;
409
      margin: 0;
410
      margin-block-start: 0em;
411
      margin-block-end: 0em;
412
    }
413
```

```
414
415
     .hero-heading h1{
416
      font-size: 3.5rem;
      font-weight: 800;
417
418
      color: #7f2d91;
419
      text-shadow: 2px 2px 5px rgba(0, 0, 0, 0.7); /* Adds shadow for better readability */
420
      margin: 0; /* Removes any default margin */
421
      padding: 20px; /* Adds space around the text */
422
      text-align: center; /* Center-aligns the text */
423
      width: 100%; /* Ensures the heading stretches across the full width */
424
425
426
     .hero-heading p{
427
      font-size: 1.2rem;
428
      color: #512882;
429
      margin-top: 30px;
430
     }
431
432
     /* Common color for both MovieDetailPage and PersonDetailPage */
433
434
     .common-text-color {
      color: #512882; /* Color used in MovieDetailPage */
435
436
     }
437
     /* PersonDetailPage styles */
438
439
     .person-detail-container {
440
      padding: 20px;
      max-width: 1200px;
441
442
      margin: auto;
443
      display: flex;
444
      flex-direction: column;
445
      align-items: center;
446 }
447
     .person-detail{
448
     display: flex;
449
     }
450
     .person-detail p{
451
     margin-left: 20px;
452
     margin-right: 20px;
453
454
     .person-detail-container h1,
455
     .person-detail-container h2,
456
     .person-detail-container h3 {
      /* Apply the common color to all header tags */
457
458
      color: rgb(81, 40, 130);
459
      margin-bottom: 10px;
460
     }
461
462
     .person-detail-container ul {
463
      margin-top: 10px;
464
      list-style-type: none;
465
      padding-left: 0;
```

```
466 }
467
468
     .person-detail-container li {
469
      margin-bottom: 5px;
470
     }
471
472
     /* Custom styles for Roles in Movies grid */
473
     .roles-grid {
474
      margin-top: 10px;
475
      font-size: 16px;
476
      border-radius: 8px;
477
      overflow: hidden;
478
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
479
      width: 90%;
480
      margin-bottom: 40px;
481
     }
482
483
     .roles-grid table {
484
      width: 100%;
485
      border-collapse: collapse;
486
      background-color: #ffc4fb;
487
      font-weight: 600;
488
489
     .roles-grid table thead{
490
      font-size: larger;
491
     }
492
493
     .roles-grid th,
494
     .roles-grid td {
495
      padding: 12px;
496
      text-align: left;
497
      border-bottom: 1px solid #ca9bff;
498
     }
499
500
     .roles-grid th {
501
      background-color: #b673de;
502
503
504
     }
505
506
     .roles-grid tr:nth-child(even) {
507
      background-color: #ca9bff;
508
     }
509
     /*
510
     .roles-grid tr:first-child {
511
512
      background-color: #d4a1ff;
     } */
513
514
515
     .kroles-gridtr:nth-child(odd) {
516
      background-color: #ffc4fb;
517 }
```

```
518
     .roles-grid tr:hover {
519
520
      background-color: rgba(255, 255, 255, 0.719);
521
     }
522
523
     .roles-grid a {
524
      color: #512882;
525
      text-decoration: none;
526
      font-weight: 800;
527 }
528
529
     .roles-grid a:hover {
530
      text-decoration: underline;
531
      color: #7f2d91;
532 }
533
534
     /* Custom styles for IMDB Rating Distribution chart */
535
     .chart-container {
536
      margin-top: 20px;
537
      margin-bottom: 60px;
538
      padding: 20px;
539
      background-color: #ffc4fb;
540
      border-radius: 10px;
541
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.5);
      width: 80%; /* Ensure it takes full width */
542
543
      height: 500px; /* Adjust the height as needed */
544
     }
545
     .chart-container h2 {
546
547
      margin-top: 40px;
548
      text-align: center;
549
      margin-bottom: 20px;
550
      color: #56ccf2;
551 }
552
553
     .chart-container canvas {
554
      max-width: 100%;
555
      height: 100%; /* Adjust the height to 100% of the container */
556
      height: auto;
557
      display: block;
558
      margin: 0;
559
     }
560
     /* MovieDetailPage styles */
561
562
563
     /* Apply the same color for MovieDetailPage headers */
564
565
```

## ▼ CAP230\_Assignment2\_Krittaya/src/pages/LoginPage.jsx

```
1
     import React, { useState, useEffect } from "react";
2
     import { useNavigate, useLocation } from "react-router-dom";
     import { setToken } from "../utils/auth"; // Import the setToken function
3
4
     import './style.css';
5
     function LoginPage() {
6
7
      const [email, setEmail] = useState("");
8
      const [password, setPassword] = useState("");
9
      const [error, setError] = useState("");
10
      const [expiredSession, setExpiredSession] = useState(false); // Track expired session
      const [isFromPersonDetail, setIsFromPersonDetail] = useState(false); // Track if the user came from
11
     PersonDetailPage
      const navigate = useNavigate();
12
13
      const location = useLocation();
14
15
      // Get the redirectTo URL from query params (this could be the previous page they were trying to
     access)
16
      const queryParams = new URLSearchParams(location.search);
      const redirectTo = queryParams.get("redirectTo") | | "/"; // Default to home page if not present
17
18
      const expired = queryParams.get("expired"); // Check if session expired
19
20
      // Determine if the user came from PersonDetailPage by checking for "redirectTo" in the query params
21
      useEffect(() => {
22
       if (localStorage.getItem("token")) {
23
        navigate(redirectTo); // If the user is already logged in, go to the previous page
24
       }
25
26
       // If expired=true, set the expired session state to true
27
       if (expired === "true") {
28
        setExpiredSession(true);
29
       }
30
31
       // Check if the user is coming from PersonDetailPage
32
       if (queryParams.has("redirectTo")) {
33
        setIsFromPersonDetail(true);
34
35
      }, [navigate, redirectTo, expired, location.search]);
36
37
      const handleLogin = async (e) => {
       e.preventDefault();
38
39
       setError("");
40
41
       try {
42
        const res = await fetch("http://4.237.58.241:3000/user/login", {
         method: "POST",
43
44
         headers: { "Content-Type": "application/json" },
45
         body: JSON.stringify({ email, password }),
46
        });
47
```

```
48
        const data = await res.json();
49
        if (res.ok && data.bearerToken?.token && data.refreshToken?.token) {
50
         setToken(data.bearerToken.token, data.refreshToken.token);
51
         localStorage.setItem("email", email);
52
         navigate(redirectTo); // Redirect to the previous page after successful login
53
54
         window.location.href = window.location.href; // Reload the current page
55
        } else {
         setError("Login failed. Please check your credentials.");
56
57
       } catch (err) {
58
        console.error("Login error:", err);
59
        setError("Something went wrong. Try again.");
60
61
       }
62
     };
63
64
      return (
65
       <div>
66
        <h1>Login</h1>
67
        {/* Display message based on whether the user came from PersonDetailPage or the session has
     expired */}
        {isFromPersonDetail &&!localStorage.getItem("email") && (
68
69
         Please login before viewing the details of the person.
70
71
        {expiredSession && localStorage.getItem("email") && (
72
         Please login again. Your session has expired.
73
        )}
74
75
        <form onSubmit={handleLogin} className="login-form">
76
         <input
77
          type="email"
          value={email}
78
          onChange={(e) => setEmail(e.target.value)}
79
80
          placeholder="Email"
          required
81
         />
82
83
         <input
84
          value={password}
85
          onChange={(e) => setPassword(e.target.value)}
86
          placeholder="Password"
          type="password"
87
          required
88
89
         />
90
         <button type="submit">Login</button>
        </form>
91
92
        {error && {error}}
93
       </div>
94
     );
95
    }
96
     export default LoginPage;
97
98
```

## ▼ CAP230\_Assignment2\_Krittaya/src/pages/RegisterPage.jsx

```
1
     import React, { useState } from "react";
2
     import './style.css';
3
4
     function RegisterPage() {
5
      const [email, setEmail] = useState("");
6
      const [password, setPassword] = useState("");
7
      const [confirmPassword, setConfirmPassword] = useState(""); // New state for confirm password
8
      const [error, setError] = useState("");
9
10
      const handleRegister = async (e) => {
11
       e.preventDefault();
12
       try {
        const res = await fetch("http://4.237.58.241:3000/user/register", {
13
14
         method: "POST",
15
         headers: { "Content-Type": "application/json" },
16
         body: JSON.stringify({ email, password }),
17
        });
18
19
        const data = await res.json();
20
21
        if (res.ok) {
22
         alert("Registration successful, please login.");
23
         window.location.href = "/login";
24
        } else {
25
         setError(data.message || "Registration failed");
26
        }
27
       } catch (err) {
28
        setError("Server error");
29
       }
30
     };
31
32
      return (
33
       <div>
34
        <h1>Register</h1>
35
        {error && {error}}
36
        <form onSubmit={handleRegister}>
37
         <input
38
          type="email"
39
          value={email}
          onChange={(e) => setEmail(e.target.value)}
40
41
          placeholder="Email"
42
          required
43
         />
44
         <br />
45
         <input
46
          type="password"
47
          value={password}
          onChange={(e) => setPassword(e.target.value)}
48
49
          placeholder="Password"
```

```
50
         required
51
        />
52
        <br />
53
        <input
         type="password"
54
         value={confirmPassword}
55
         onChange={(e) => setConfirmPassword(e.target.value)}
56
         placeholder="Confirm Password"
57
         required
58
59
        />
60
        <br />
        <button type="submit">Register</button>
61
62
       </form>
63
      </div>
64
     );
65
    }
66
67
    export default RegisterPage;
68
```

```
1
     import React, { useEffect, useState } from 'react';
2
     import { useParams, Link } from 'react-router-dom';
     import './style.css';
3
4
5
6
     function MovieDetailPage() {
7
      const { imdbID } = useParams();
8
      const [movie, setMovie] = useState(null);
9
10
      useEffect(() => {
       async function fetchMovieDetails() {
11
12
        try {
         const res = await fetch(`http://4.237.58.241:3000/movies/data/${imdbID}`, {
13
14
           headers: { Accept: "application/json" }
15
         });
16
17
         if (!res.ok) {
18
           console.error("X Failed to fetch movie:", res.statusText);
19
          return;
20
         }
21
22
         const data = await res.json();
23
         console.log(" Movie data:", data);
24
         setMovie(data);
25
        } catch (err) {
26
         console.error("  Error fetching movie details:", err.message);
27
        }
28
       }
29
30
       if (imdbID) fetchMovieDetails();
31
      }, [imdbID]);
32
33
      if (!movie) return Loading...;
34
35
      // Get people for all roles dynamically
36
      const getPeopleByRole = () =>
37
       movie.principals?.reduce((acc, person) => {
38
        if (!acc[person.category]) acc[person.category] = [];
39
        acc[person.category].push(person);
40
        return acc;
41
       }, {}) | | {};
42
43
      const peopleByRole = getPeopleByRole();
44
45
      // Extract ratings
46
      const imdbRating = movie.ratings?.find(r => r.source.includes('Internet'))?.value
      ? `${movie.ratings.find(r => r.source.includes('Internet')).value}/10`
47
      : 'Unknown';
48
49
```

```
const rtRating = movie.ratings?.find(r => r.source.includes('Rotten'))?.value
50
51
     ? `${movie.ratings.find(r => r.source.includes('Rotten')).value}/100`
     : 'Unknown';
52
53
     const metaRating = movie.ratings?.find(r => r.source.includes('Metacritic'))?.value
54
     ? `${movie.ratings.find(r => r.source.includes('Metacritic')).value}/100`
55
     : 'Unknown';
56
57
     // Capitalize the first letter of the role
58
    // Capitalize the first letter of each word, remove underscores, and join them
59
     const capitalizeRole = (role) => {
60
61
     return role
       .split(' ') // Split the role by underscores
62
       .map(word => word.charAt(0).toUpperCase() + word.slice(1)) // Capitalize the first letter of each word
63
64
       .join(' '); // Join the words with a space
65
    };
66
67
     return (
       <div className="movie-detail-container">
68
69
        <h1>{movie.title} ({movie.year})</h1>
70
        <div className="movie-detail">
71
        {movie.poster && <img src={movie.poster} alt={movie.title} className="movie-poster" />}
72
        <div className="rating">
73
        <h3> \( \tau \) Ratings \( \( \tau < / h 3 > \)
         <strong>IMDb:</strong> {imdbRating}
74
75
         <strong>Rotten Tomatoes:</strong> {rtRating}
76
         <strong>Metacritic:</strong> {metaRating}
77
        </div>
78
        <div className= "movie-detail-1">
79
        <strong>Year:</strong> {movie.year | | 'N/A'}
80
        <strong>Runtime:</strong> {movie.runtime && movie.runtime !== 'N/A' ? `${movie.runtime}}
    minutes`: 'Unknown'}
81
        <strong>Genres:</strong> {movie.genres?.join(', ') || 'Unknown'}
        <strong>Country:</strong> {movie.country | | 'Unknown'}
82
83
        <strong>Box Office:</strong> {movie.boxoffice ? `$${new}
    Intl.NumberFormat().format(movie.boxoffice)}`: 'Unknown'}
84
85
        <strong></strong> {movie.plot | | 'Unknown'}
86
87
       </div>
88
89
90
        <h2 className="key-people">Key People</h2>
91
        <div className="key-people-grid">
92
         93
          <thead>
94
           95
            Role
            Name
96
97
            Character
98
           99
          </thead>
```

```
100
        101
         {/* Dynamically render all people by role */}
         {Object.entries(peopleByRole).map(([role, people]) => (
102
103
          people.map((person) => (
104
           {capitalizeRole(role)}{/* Capitalize the role */}
105
106
            <Link to={`/people/${person.id}`} >{person.name}
    </Link>
            {person.characters?.join(", ") | | "N/A"}
107
108
           109
          ))
110
         ))}
111
        112
       113
      </div>
114
      </div>
115
     );
116 }
117
    export default MovieDetailPage;
118
119
120
```

```
1
     import React, { useState, useEffect } from 'react';
2
     import { useParams, useNavigate, Link } from 'react-router-dom';
     import { clearToken } from '../utils/auth'; // Import clearToken function
3
4
     import { Bar } from 'react-chartjs-2'; // Import the Bar chart component
5
     import { Chart as ChartJS, CategoryScale, LinearScale, BarElement, Title, Tooltip, Legend } from 'chart.js';
     // Import necessary Chart.js components
     import './style.css';
6
7
8
     // Register the necessary Chart.js components
9
     ChartJS.register(CategoryScale, LinearScale, BarElement, Title, Tooltip, Legend);
10
11
     function PersonDetailPage() {
      const { id } = useParams(); // Get person ID from URL parameters
12
13
      const [person, setPerson] = useState(null); // State to store person details
14
      const [error, setError] = useState(null); // For error handling
15
      const [ratingData, setRatingData] = useState(null); // For chart data
16
      const navigate = useNavigate(); // Hook for navigation
17
18
      // Function to check if the token is expired
19
      const isTokenExpired = (token) => {
20
       const parts = token.split('.');
21
       const payload = |SON.parse(atob(parts[1]));
22
       return Date.now() >= payload.exp * 1000;
23
      };
24
25
      useEffect(() => {
26
       const fetchPersonDetails = async () => {
27
        const token = localStorage.getItem("token"); // Get the token from localStorage
28
29
        if (!token | | isTokenExpired(token)) {
30
          setError("Token expired or not found. Please log in.");
31
          clearToken(); // Clear the token from localStorage
32
          navigate(`/login?redirectTo=/people/${id}&expired=true`); // Redirect to login with a "redirectTo"
     param
33
         return;
34
        }
35
36
37
          const res = await fetch(`http://4.237.58.241:3000/people/${id}`, {
38
           method: 'GET',
39
           headers: {
            'Accept': 'application/json',
40
41
            'Authorization': `Bearer ${token}`, // Include the token in the header
42
          },
43
         });
44
45
          if (!res.ok) {
46
          if (res.status === 401) {
47
            clearToken(); // Remove the token from localStorage
```

```
setError("Unauthorized access. Please log in again.");
48
            navigate('/login'); // Redirect to the login page
49
50
            return;
51
           }
           throw new Error('Failed to fetch person data: ${res.statusText}');
52
53
          }
          const data = await res.json();
54
55
          setPerson(data); // Set the fetched data to state
56
57
          // Prepare the rating data for the chart
          const imdbRatings = data.roles?.map(role => role.imdbRating).filter(Boolean); // Get IMDB ratings
58
     and filter out falsy values
59
          // Count ratings in specific ranges
60
61
          const ratingRanges = [0, 2, 4, 6, 8, 10];
          const ratingCount = new Array(ratingRanges.length - 1).fill(0);
62
63
          imdbRatings.forEach(rating => {
64
           for (let i = 0; i < ratingRanges.length - 1; i++) {
65
            if (rating >= ratingRanges[i] && rating < ratingRanges[i + 1]) {
66
67
             ratingCount[i]++;
             break;
68
69
            }
70
           }
71
          });
72
73
          // Set the chart data
74
          setRatingData({
75
           labels: ratingRanges.slice(0, -1).map((r, i) => `${r}-${ratingRanges[i + 1]}`),
           datasets: [
76
77
             label: 'IMDB Ratings Count',
78
79
             data: ratingCount,
             backgroundColor: 'rgb(141, 75, 192)', // Bar color
80
81
             borderColor: 'rgb(141, 75, 192)', // Bar border color
82
             borderWidth: 1,
83
            },
84
           ],
85
          });
86
87
        } catch (err) {
88
          setError(`Error: ${err.message}`); // Handle other errors
89
        }
90
       };
91
92
       fetchPersonDetails();
93
      }, [id, navigate]); // Re-run effect when `id` or `navigate` changes
94
95
      if (error) {
96
       return <div>{error}</div>; // Show error message if there's any
97
      }
98
```

```
99
     if (!person) {
100
      return Loading...; // Show loading state until person data is fetched
101
     }
102
103
     return (
104
      <div className="person-detail-container">
105
       <h1>{person.name}</h1>
       <div className="person-detail">
106
        <strong>Born:</strong> {person.birthYear}
107
        <strong>Death:</strong> {person.deathYear || "Unknown"}
108
       </div>
109
110
       <h2>Roles in Movies:</h2>
111
       <div className="roles-grid">
112
113
        <thead>
114
115
          116
           Role
117
           Movie Title
118
           Character(s)
119
           IMDB Rating
120
          </thead>
121
122
         {person.roles && person.roles.length > 0?(
123
124
           person.roles.map((role, idx) => (
125
            <tr key={idx}>
             {role.category.charAt(0).toUpperCase() + role.category.slice(1)}{/* Capitalize the role
126
    */}
127
             128
              <Link to={`/movies/${role.movieId}`} className="movie-link">
               {role.movieName}
129
130
              </Link>
131
             {role.characters?.join(", ") | | "N/A"}
132
133
             {role.imdbRating | | "N/A"}
134
            135
           ))
136
          ):(
137
138
            No roles available.
139
           140
          )}
141
         142
        143
       </div>
144
145
       <h2>IMDB Rating Distribution:</h2>
146
       {ratingData?(
147
        <div className="chart-container">
148
         <Bar
149
          data={ratingData}
```

```
150
            options={{
151
             responsive: true,
152
             maintainAspectRatio: false, // Disable aspect ratio to make the chart fill the container
153
             plugins: {
154
              title: {
155
               display: true,
156
               text: 'Distribution of IMDB Ratings',
157
              },
              legend: {
158
159
               position: 'top',
160
              },
161
             },
162
             scales: {
163
              x: {
              beginAtZero: true,
164
165
              },
166
              y: {
              beginAtZero: true,
167
168
              },
169
             },
170
           }}
171
           />
172
173
         </div>
174
        ):(
175
          Loading chart...
176
        )}
177
       </div>
178
      );
179
     }
180
181
     export default PersonDetailPage;
182
```