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 / 10 pts

✓ **+ 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

✓ **+ 6 pts** Report is professional but lacks detail, or has errors significant enough to detract

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

```html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Vite + React</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>

```

▾ CAP230_Assignment2_Krittaya/vite.config.js

Download

```js
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

```
# React + Vite

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

Currently, two official plugins are available:

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

## Expanding the ESLint configuration

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.

```

CAP230_Assignment2_Krittaya/package-lock.json ⬇ Download

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

CAP230_Assignment2_Krittaya/package.json

Download

```json
{
  "name": "movie-client",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint .",
    "preview": "vite preview"
  },
  "dependencies": {
    "ag-grid-community": "^33.2.4",
    "ag-grid-enterprise": "^33.2.4",
    "ag-grid-react": "^33.2.4",
    "chart.js": "^4.4.9",
    "react": "^19.1.0",
    "react-chartjs-2": "^5.3.0",
    "react-dom": "^19.1.0",
    "react-router-dom": "^7.5.3",
    "react-world-flags": "^1.6.0"
  },
  "devDependencies": {
    "@eslint/js": "^9.25.0",
    "@types/react": "^19.1.2",
    "@types/react-dom": "^19.1.2",
    "@vitejs/plugin-react": "^4.4.1",
    "eslint": "^9.25.0",
    "eslint-plugin-react-hooks": "^5.2.0",
    "eslint-plugin-react-refresh": "^0.4.19",
    "globals": "^16.0.0",
    "vite": "^6.3.5"
  }
}

```

CAP230_Assignment2_Krittaya/eslint.config.js

Download

```js
import js from '@eslint/js'
import globals from 'globals'
import reactHooks from 'eslint-plugin-react-hooks'
import reactRefresh from 'eslint-plugin-react-refresh'

export default [
  { ignores: ['dist'] },
  {
    files: ['**/*.{js,jsx}'],
    languageOptions: {
      ecmaVersion: 2020,
      globals: globals.browser,
      parserOptions: {
        ecmaVersion: 'latest',
        ecmaFeatures: { jsx: true },
        sourceType: 'module',
      },
    },
    plugins: {
      'react-hooks': reactHooks,
      'react-refresh': reactRefresh,
    },
    rules: {
      ...js.configs.recommended.rules,
      ...reactHooks.configs.recommended.rules,
      'no-unused-vars': ['error', { varsIgnorePattern: '^[A-Z_]' }],
      'react-refresh/only-export-components': [
        'warn',
        { allowConstantExport: true },
      ],
    },
  },
]

```

▾ **CAP230_Assignment2_Krittaya/report.pdf**

Download

Your browser does not support PDF previews. You can download the file instead.

CAP230_Assignment2_Krittaya/dist/index.html Download

```html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Vite + React</title>
    <script type="module" crossorigin src="/assets/index-_GoSY5hR.js"></script>
    <link rel="stylesheet" crossorigin href="/assets/index-BQOiJvqv.css">
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>

```

CAP230_Assignment2_Krittaya/dist/vite.svg Download

```svg
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" aria-hidden="true" role="img" class="iconify iconify--logos" width="31.88" height="32" preserveAspectRatio="xMidYMid meet" viewBox="0 0 256 257"><defs><linearGradient id="IconifyId1813088fe1fbc01fb466" 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>
```

▾ **CAP230_Assignment2_Krittaya/public/vite.svg** Download

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" aria-hidden="true" role="img" class="iconify iconify--logos" width="31.88" height="32" preserveAspectRatio="xMidYMid meet" viewBox="0 0 256 257"><defs><linearGradient id="IconifyId1813088fe1fbc01fb466" 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>
```

▾ **CAP230_Assignment2_Krittaya/src/App.css** Download

```css
#root {
  /* max-width: 1280px; */
  /* margin: 0 auto; */
  /* padding: 2rem; */
  text-align: center;

}
* {
  font-family: 'Signika', sans-serif;

}
.logo {
  height: 10em;
  padding: 1.5em;
  will-change: filter;
  transition: filter 300ms;
}
.logo:hover {
  filter: drop-shadow(0 0 2em #64f5ffaa);
}
.logo.react:hover {
  filter: drop-shadow(0 0 2em #7361fbaa);
}

@keyframes logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}

@media (prefers-reduced-motion: no-preference) {
  a:nth-of-type(2) .logo {
    animation: logo-spin infinite 20s linear;
  }
}

.card {
  padding: 2em;
}

.read-the-docs {
  color: #888;
}
/* App.css */
.navbar {
  background-color: #fcc5ff;
  /* padding-top: 18px;
  padding-bottom: 18px;  */
  font-size: 20px;
  text-align: left;
  display: flex;
}

.navbar-email {
  font-size: 20px;
  color: #a846be;
  margin-left: 10px;
  margin-top: 18px;
  text-align: right;
  font-family: 'Vollkorn', serif;

}

.navbar-link {
  /* margin-right: 25px;
  margin-left: 25px;  */
  color: #a846be;
  padding: 18px 30px;
  font-weight: 700;
  font-family: 'Vollkorn', serif;
}
.navbar-link:hover {
  background-color: #a846be;
  color: #fcc5ff;
  text-decoration: underline;
}
.navbar-nav {
  display: flex;
  justify-content: space-between;
}


.navbar-brand {
  font-size: 1.5rem;
}

.navbar-light {
  background-color: #f8f9fa;
}



```

CAP230_Assignment2_Krittaya/src/index.css

Download

```css

/*  Vollkorn font */
@import url('https://fonts.googleapis.com/css2?family=Vollkorn:wght@400;700&display=swap');

/*  Signika font */
@import url('https://fonts.googleapis.com/css2?family=Signika:wght@400;600&display=swap');

:root {
  font-family: system-ui, Avenir, Helvetica, Arial, sans-serif;
  line-height: 1.5;
  font-weight: 400;

  color-scheme: light dark;
  color: rgba(32, 13, 52, 0.87);
  background-color: #f3f3f3;

  font-synthesis: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
/* Home and movies */
a {
  font-weight: 500;
  color: #e364ff;
  text-decoration: inherit;

}
a:hover {
  color: #e7b7ff;
}

body {
  margin: 0;
  /* display: flex;
  place-items: center; */
  min-width: 320px;
  min-height: 100vh;

}

h1 {
  font-size: 2.5em;
  line-height: 1.1;
  color: rgb(81, 40, 130);
  font-family: 'Vollkorn', serif;
}
/* search button*/
button {
  border-radius: 8px;
  border: 1px solid transparent;
  padding: 0.6em 1.2em;
  font-size: 1em;
  font-weight: 500;
  font-family: inherit;
  background-color: #e3b3ff;
  cursor: pointer;
  transition: border-color 0.25s;

}

/* Button for movie */
button:hover {
  border-color: #9e2ff3;
}
button:focus,
button:focus-visible {
  outline: 4px auto -webkit-focus-ring-color;

}



```

CAP230_Assignment2_Krittaya/src/main.jsx Download

```jsx
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import { BrowserRouter } from 'react-router-dom'
import './index.css'
import App from './App.jsx'

createRoot(document.getElementById('root')).render(
  <StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </StrictMode>
)

```

CAP230_Assignment2_Krittaya/src/App.jsx

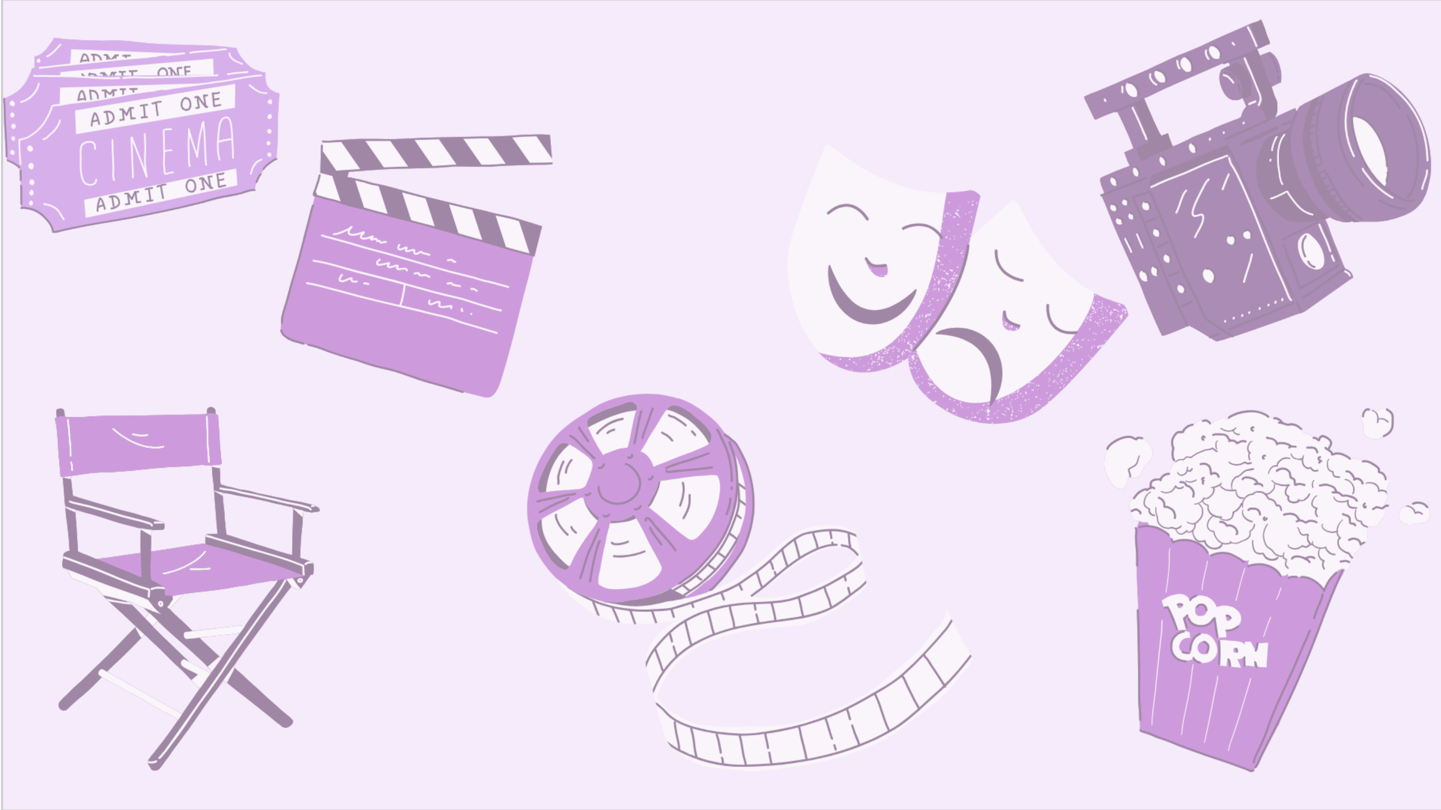
Download

```
import React, { useState, useEffect } from 'react';
import { Routes, Route, Link, useNavigate } from 'react-router-dom';
import HomePage from './pages/HomePage.jsx';
import MoviesPage from './pages/MoviesPage.jsx';
import MovieDetailPage from './pages/MovieDetailPage.jsx';
import PersonDetailPage from './pages/PersonDetailPage.jsx';
import LoginPage from './pages/LoginPage.jsx';
import RegisterPage from './pages/RegisterPage.jsx';
import { getToken, getRefreshToken, clearToken, setToken } from './utils/auth'; // Import functions
import './App.css';

function App() {
  const [isAuthenticated, setIsAuthenticated] = useState(false);
  const [email, setEmail] = useState(localStorage.getItem("email"));
  const [loading, setLoading] = useState(true);
  const navigate = useNavigate();

  // Function to check if the token is expired
  const isTokenExpired = (token) => {
    if (!token) return true;

    const parts = token.split('.');
    if (parts.length !== 3) return true; // Invalid JWT

    const payload = JSON.parse(atob(parts[1])); // Decode the token's payload
    const expirationTime = payload.exp * 1000; // Expiry time is in seconds, convert to milliseconds

    return Date.now() >= expirationTime; // Compare expiration time with current time
  };

  // Handle user logout
  const handleLogout = async () => {
    const refreshToken = localStorage.getItem("refreshToken");

    if (!refreshToken) {
      // If no refresh token, force the user to log in again
      console.log("No refresh token found, forcing re-login.");
      clearToken();
      setIsAuthenticated(false);
      setEmail(null);
      localStorage.removeItem("email");
      navigate("/login");
      return;
    }

    try {
      const response = await fetch("http://4.237.58.241:3000/user/logout", {
        method: "POST",
        headers: {
```

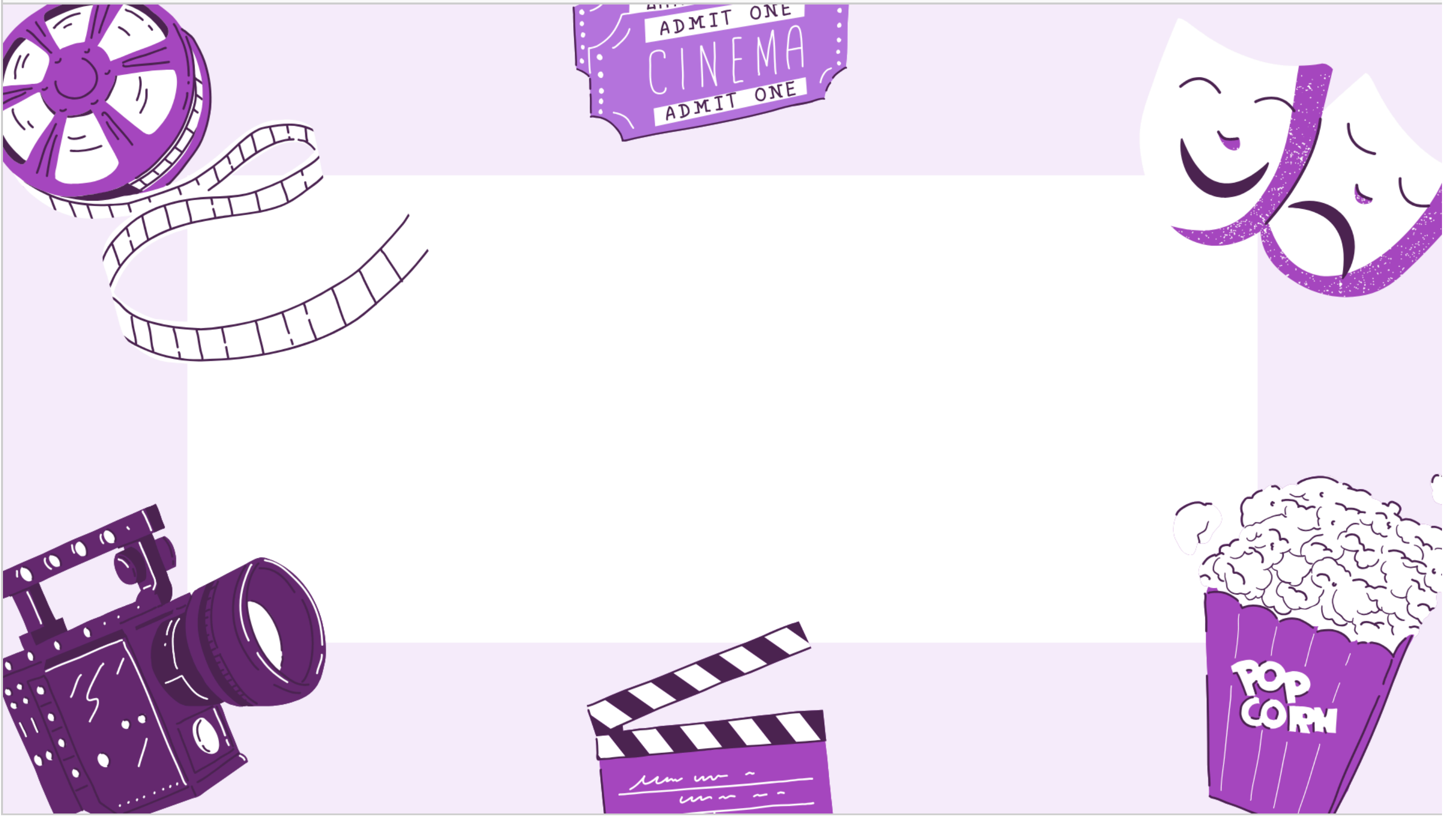
```
        "Content-Type": "application/json",
      },
      body: JSON.stringify({ refreshToken }),
    });

    if (!response.ok) {
      // If the logout request fails due to an expired token, force the user to log in again
      console.error("Failed to logout due to expired token, forcing re-login.");

      clearToken(); // Clear tokens from localStorage
      setIsAuthenticated(false);
      setEmail(null);
      localStorage.removeItem("email");

      // Redirect to login page
      navigate("/login");
      return;
    }

    // If logout is successful (i.e., refresh token was valid), clear tokens
    clearToken();
    setIsAuthenticated(false);
    setEmail(null);
    localStorage.removeItem("email");

    navigate("/login"); // Redirect to login page
  } catch (error) {
    console.error("Logout error:", error);
    clearToken(); // Clear tokens in case of any error
    navigate("/login"); // Redirect to login page
  }
};

// Function to refresh the token using the refresh token
const refreshToken = async () => {
  const refreshToken = getRefreshToken();
  if (!refreshToken) {
    clearToken();
    return;
  }

  try {
    const res = await fetch("http://4.237.58.241:3000/user/refresh", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify({ refreshToken }),
    });

    const data = await res.json();
    if (res.ok && data.bearerToken && data.refreshToken) {
      setToken(data.bearerToken.token, data.refreshToken.token);
      setIsAuthenticated(true);
    } else {
      clearToken();
    }
  } catch (err) {
    console.error("Token refresh error:", err);
    clearToken();
  }
};

// Check if the user is authenticated (i.e., if a token exists)
useEffect(() => {
  const token = getToken();
  if (token) {
    setIsAuthenticated(true); // Token exists, user is authenticated
  }
  setLoading(false); // Stop loading when the check is complete
}, []);

if (loading) return <div>Loading...</div>; // You can show a loading spinner here.

return (
  <div>
    <nav className="navbar">
      <Link className="navbar-link" to="/">Home</Link>
      <Link className="navbar-link" to="/movies">Movies</Link>
      {isAuthenticated ? (
        <>
          <Link className="navbar-link" to="/login" onClick={handleLogout}>
            Logout
          </Link>
          <span className="navbar-email">👤 {email}</span>
        </>
      ) : (
        <>
          <Link className="navbar-link" to="/login">Login</Link>
          <Link className="navbar-link" to="/register">Register</Link>
        </>
      )}
    </nav>

    <Routes>
      <Route path="/" element={<HomePage />} />
      <Route path="/movies" element={<MoviesPage />} />
      <Route path="/movies/:imdbID" element={<MovieDetailPage />} />
      <Route path="/people/:id" element={<PersonDetailPage />} />
      <Route path="/login" element={<LoginPage />} />
      <Route path="/register" element={<RegisterPage />} />
    </Routes>
    {/* Footer Section */}
    <footer className="footer">
```

```
        {/* <p>All data is from API</p> */}
        <p>&copy; 2025 Krittaya Kruapat. All rights reserved.</p>
      </footer>
    </div>
  );
}
export default App;

```

CAP230_Assignment2_Krittaya/dist/assets/MoviepageBG-DZbmksjN.png Download



▾ CAP230_Assignment2_Krittaya/dist/assets/MovieHero-BFbWySRe.png Download



▾ CAP230_Assignment2_Krittaya/dist/assets/index-BQOiJvqv.css Download

```
@import"https://fonts.googleapis.com/css2?family=Vollkorn:wght@400;700&display=swap";@import"htt
bottom:60px;padding:20px;background-color:#ffc4fb;border-radius:10px;box-shadow:0 4px 8px #0000(
8;base64,d09GMgABAAAAABMUAAsAAAAAKMwAABLCAAEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAHIlWB
weight:400;font-style:normal}.ag-theme-alpine,.ag-theme-alpine-dark,.ag-theme-alpine-auto-dark{--ag-a
theme-alpine-auto-dark input[class^=ag-][type=text],.ag-theme-alpine-auto-dark input[class^=ag-][type=
child,.ag-theme-alpine-auto-dark .ag-charts-format-sub-level-group-item:last-child{margin-bottom:0}.ag-
theme-alpine-dark .ag-filter-active .ag-icon-filter,.ag-theme-alpine-auto-dark .ag-filter-active .ag-icon-filte

```

▾ CAP230_Assignment2_Krittaya/dist/assets/index-_GoSY5hR.js Download

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

CAP230_Assignment2_Krittaya/src/utils/auth.js Download

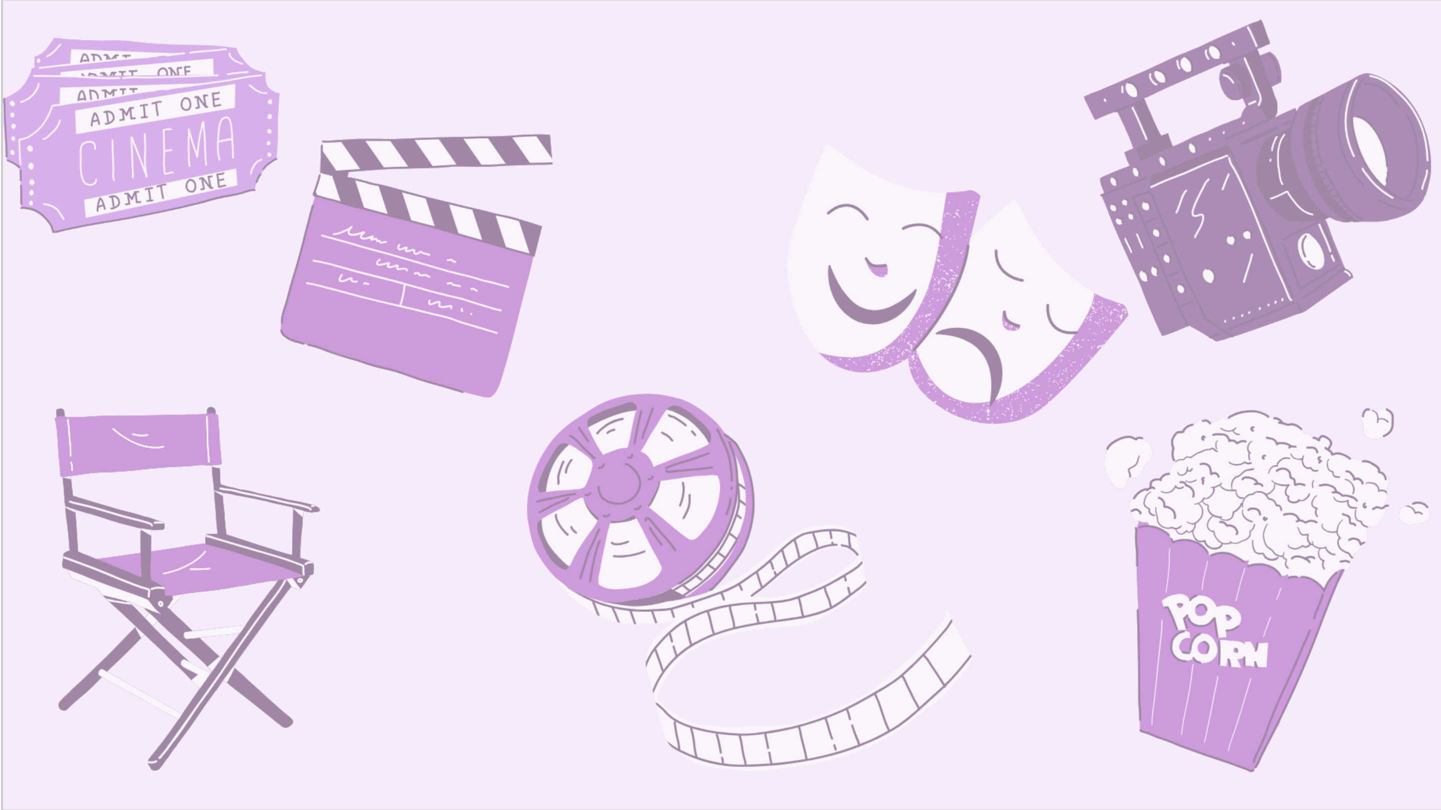
```
// Save the token and refresh token to localStorage
export function setToken(token, refreshToken) {
  if (!token || !refreshToken) {
   console.error("setToken error: Both token and refreshToken must be provided.");
   return;
  }

  localStorage.setItem("token", token);
  localStorage.setItem("refreshToken", refreshToken);
  console.log("Tokens saved:", token, refreshToken); // Log tokens for debugging

 }

 // Retrieve the bearer token from localStorage
 export function getToken() {
  const token = localStorage.getItem("token");
  if (!token) {
   console.log("getToken : No token found in localStorage.");
  }
  return token;
 }

 // Retrieve the refresh token from localStorage
 export function getRefreshToken() {
  const refreshToken = localStorage.getItem("refreshToken");
  if (!refreshToken) {
   console.error("getRefreshToken error: No refresh token found in localStorage.");
  }
  return refreshToken;
 }

 // Clear both the token and refresh token from localStorage
 export function clearToken() {
  localStorage.removeItem("token");
  localStorage.removeItem("refreshToken");
 }

```

▾ **CAP230_Assignment2_Krittaya/src/utils/fetchwithauth.js** Download

```
// utils/fetchWithAuth.js

export const fetchWithAuth = async (url, options = {}) => {
  const token = localStorage.getItem('token');

  if (!token) {
   throw new Error('No token found');
  }

  let response = await fetch(url, {
   ...options,
   headers: {
    ...options.headers,
    Authorization: `Bearer ${token}`,
   },
  });

  // If the token is expired (401 Unauthorized), refresh the token
  if (response.status === 401) {
   console.log('Token expired, refreshing...');
   await refreshToken(); // Call refresh function

   // Retry the request with the new token
   const newToken = localStorage.getItem('token');
   response = await fetch(url, {
    ...options,
    headers: {
     ...options.headers,
     Authorization: `Bearer ${newToken}`,
    },
   });
  }

  return response.json();
 };

const refreshToken = async () => {
  const refreshToken = localStorage.getItem('refreshToken');

  if (!refreshToken) {
   console.error('No refresh token found');
   return;
  }

  try {
   const res = await fetch('http://4.237.58.241:3000/user/refresh', {
    method: 'POST',
    headers: {
     'Content-Type': 'application/json',
        },
        body: JSON.stringify({
          refreshToken,
        }),
      });

      const data = await res.json();

      if (res.ok && data.token && data.refreshToken) {
        setToken(data.token, data.refreshToken); // Save the new token
      } else {
        console.error('Failed to refresh token:', data);
        clearToken(); // Clear localStorage and force re-login
      }
    } catch (err) {
      console.error('Error refreshing token:', err);
      clearToken(); // Clear localStorage and force re-login
    }
};

const setToken = (token, refreshToken) => {
    localStorage.setItem('token', token);
    localStorage.setItem('refreshToken', refreshToken);
    console.log('Tokens saved:', token, refreshToken); // Log tokens for debugging
};

const clearToken = () => {
    localStorage.removeItem('token');
    localStorage.removeItem('refreshToken');
    console.log("Tokens cleared");
};

```

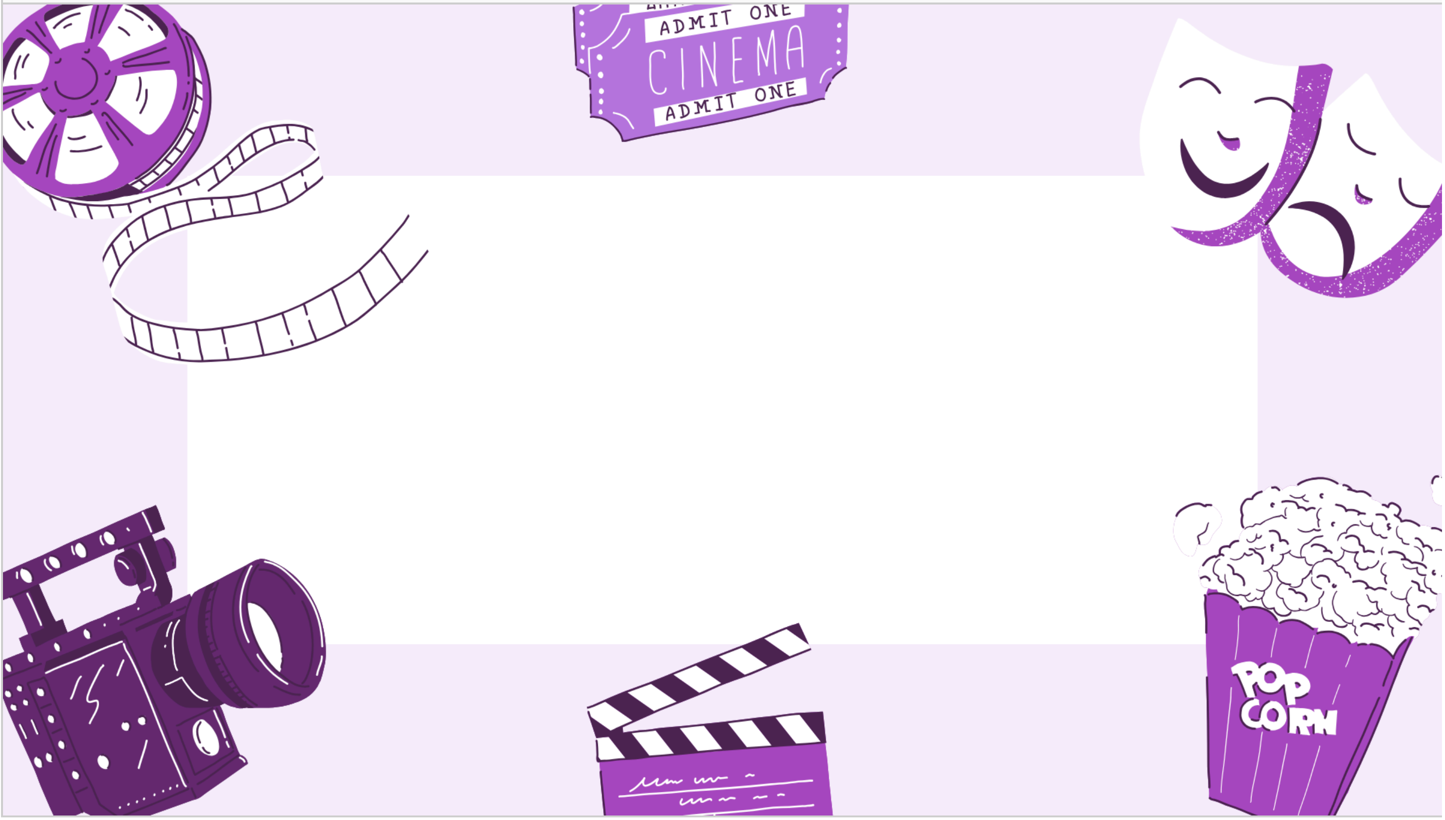
CAP230_Assignment2_Krittaya/src/assets/MoviepageBG.png

Download



CAP230_Assignment2_Krittaya/src/assets/MovieHero.png

Download



CAP230_Assignment2_Krittaya/src/assets/react.svg   Download

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" aria-
hidden="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
import './style.css';

function HomePage() {
  return (
    <div className="hero-section">
    <div className="hero-heading">
    <h1>Welcome Movie Search Website!</h1>
    <h1>By Krittya Kruapat</h1>
    <p>Thank you for visiting! Enjoy browsing and discovering something new!</p>
    </div>
    </div>
  );
}
  export default HomePage;

```

▾ **CAP230_Assignment2_Krittaya/src/pages/MoviesPage.jsx** Download

```jsx
import React, { useState, useEffect } from "react";
import { useNavigate } from "react-router-dom";
import { AgGridReact } from "ag-grid-react";
import { ModuleRegistry } from "ag-grid-community";
import {
  ClientSideRowModelModule,
  PaginationModule,
  ValidationModule
} from "ag-grid-community";

import "ag-grid-community/styles/ag-theme-alpine.css";
import './style.css';


// Register modules
ModuleRegistry.registerModules([
  ClientSideRowModelModule,
  PaginationModule,
  ValidationModule
]);

function MoviesPage() {
  const [query, setQuery] = useState(""); // For title search
  const [selectedYear, setSelectedYear] = useState(""); // For year dropdown
  const [filteredMovies, setFilteredMovies] = useState([]);
  const [allMovies, setAllMovies] = useState([]); // Store all movies for later filtering
  const navigate = useNavigate(); // For navigation on row click

  // Fetch all movies initially to populate the year dropdown
  useEffect(() => {
    async function fetchMovies() {
      try {
        const res = await fetch("http://4.237.58.241:3000/movies/search", {
          headers: { Accept: "application/json" }
        });

        const result = await res.json();
        const data = Array.isArray(result) ? result : result.data || [];

        setAllMovies(data); // Store all fetched movies
        setFilteredMovies(data); // Initially, display all movies
        console.log("First movie object:", data[0]);

      } catch (error) {
        console.error("Failed to fetch movies:", error);
      }
    }

    fetchMovies();
  }, []);

  // Get unique years for the dropdown, convert to numbers for consistency
  const years = [...new Set(allMovies.map((movie) => movie.year))].sort();

  // Ensure the year is a number for comparison
  const handleYearChange = (e) => {
    setSelectedYear(e.target.value ? Number(e.target.value) : "");
  };

  const searchMovies = () => {
    // Filter based on query and selected year
    const filtered = allMovies.filter((movie) => {
      const matchesTitle = movie.title?.toLowerCase().includes(query.toLowerCase());
      const matchesYear = selectedYear ? Number(movie.year) === selectedYear : true; // Convert movie.year to a number
      return matchesTitle && matchesYear;
    });

    setFilteredMovies(filtered);
  };

  const columnDefs = [
    { field: "title", headerName: "Title", sortable: true, flex: 2 },
    { field: "year", headerName: "Year", sortable: true, flex: 1 },
    { field: "imdbRating", headerName: "IMDB Rating", sortable: true, flex: 1 } // Change 'rank' to 'imdbRating'
  ];

  const handleRowClick = (event) => {
    const imdbID = event.data.imdbID;
    if (imdbID) {
      navigate(`/movies/${imdbID}`);
    }
  };

  return (
    <div className="movies-search">
      <h1>Movie Search</h1>
      <div className="search-bar-wrapper">
        <input
          type="text"
          value={query}
          onChange={(e) => setQuery(e.target.value)}
          placeholder="Enter movie title"
        />
      </div>

      {/* Dropdown for year selection */}
      <div>
        <select
          value={selectedYear}
          onChange={handleYearChange}
        >
          <option value="">Select Year</option>
          {years.map((year) => (
            <option key={year} value={year}>
              {year}
            </option>
          ))}
        </select>
      </div>

      <button onClick={searchMovies}>Search</button>

      <div className="ag-theme-alpine ag-grid-container">
      {/* style={{ height: 400, marginTop: 20 }} */}
        <AgGridReact
          rowData={filteredMovies}
          columnDefs={columnDefs}
          pagination={true}
          paginationPageSize={10}
          paginationPageSizeSelector={[10, 20, 50, 100]} // Add 10 to the selector options
          onRowClicked={handleRowClick}
        />
      </div>
    </div>
  );
}

export default MoviesPage;

```

▾ **CAP230_Assignment2_Krittaya/src/pages/style.css** Download

```css
/* MovieDetailPage */
.movie-detail-container {
  color: rgb(81, 40, 130);
  padding: 20px;
  max-width: 1200px;
  margin: auto;
  text-align: center; /* Ensure everything inside is centered */
  display: flex;
  flex-direction: column;
  align-items: center; /* Ensures centering of h2 */
}


.movie-detail-1{
  font-size: 18px;
}
.key-people {
  text-align: center;
  font-size: 24px;
  margin-top: 20px;
  width: 100%; /* Make sure the element takes full width */
  display: block;
}
  .rating {
    font-size: 15px;
    padding: 0px 18px 10px;
    border: #b673de;
    border: 2px solid #ccc; /* Add a solid 2px border with light gray color */
    border-radius: 8px; /* Optional: Add rounded corners */
    background-color: #f9f9f9; /* Optional: Add a background color */
  }
  .movie-poster {
    width: 200px;
    margin-bottom: 20px;
  }
 .movie-detail{
  max-width: 1000px;
  margin-left:60px;
  margin-right:60px;

 }

  .key-people-grid {
    margin-top: 10px;
    font-size: 16px;
    border-radius: 8px;
    overflow: hidden;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
    width: 90%; /* Ensure it takes the full available width */
```

```css
}

.plot {
font-style: italic;
font-size: 18px;
}

.key-people-grid table {
  width: 100%;
  border-collapse: collapse;
  background-color: #ffc4fb;
  font-weight: 600;

}
.key-people-grid table thead{
  color: #4b2c6f;
  font-weight: 800;
  font-size: larger;
}

.key-people-grid th, .key-people-grid td {
  padding: 12px;
  text-align: left;
  border-bottom: 1px solid #ca9bff;
}

.key-people-grid th {
  background-color: #b673de;

}

.key-people-grid tr:nth-child(even) {
  background-color: #ca9bff; /* Default color for even rows */
}


.key-people-grid tr:nth-child(odd) {
  background-color: #ffc4fb;
}


.key-people-grid tr:hover {
  background-color: rgba(255, 255, 255, 0.719);
}

.key-people-grid a {
  color: #642c70;
  font-weight: 800;
}

.key-people-grid a:hover {
  text-decoration: underline;
```

```css
    color: #7f2d91;

  }
  .movie-detail-people a{
    color: #642c70;
    text-decoration: 'none';

  }
  .movies-search h1 {
    font-size: 2.2em;
    font-weight: 800;
  }


  @media (min-width: 750px) {

    .movie-poster {
      margin-right: 20px;
      float: left;
    }

    .movie-detail-1 {
      /* display: block; */
      text-align: left;
      margin-left: 20px;
      justify-content: left;

    }
    .rating{
      margin-right: 20px;
      float: right;
    }
    .rating p{
      text-align: right;
    }
  }

/* ------------------------------------------------------------------------------------------ */

  /* MoviePage */
.movies-search{
  height: 100vh; /* Full height of the page */
  width: 100%; /* Full width of the page */
  background-image: url('../assets/MoviepageBG.png'); /* Background image path */
  background-size: cover; /* Makes sure the background covers the entire container */
  background-position: center center; /* Centers the background */
  background-repeat: no-repeat; /* Ensures the image doesn't repeat */
  padding-top: 40px;
  padding-bottom: 40px;
}
.movies-search h1{

  margin-top: 0;
}
/* Custom styles for ag-grid container */
.ag-grid-container {
  height: 500px;
  margin-top: 40px;
  margin-bottom: 60px;
  padding-left:10%;
  padding-right:10%;

}

/* Custom styles for ag-Grid pagination */
.ag-theme-alpine .ag-paging-panel {
  background-color: #b673de; /* Set background color of the pagination bar */
  color: #330e3b; /* Set text color */
  border-radius: 8px;
  padding: 10px;
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 16px;
}

/* Pagination controls (next, previous, page numbers) */
.ag-theme-alpine .ag-paging-panel .ag-paging-button,
.ag-theme-alpine .ag-paging-panel .ag-page-size-panel select {
  background-color: #642c70; /* Button background */
  color: #fff; /* Button text color */
  border: 2px solid #ca9bff;
  border-radius: 8px;
  padding: 5px 10px;
  margin: 0 5px;
  font-weight: 800;
  cursor: pointer;

}

/* Pagination buttons on hover */
.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 {
  background-color: #7f2d91; /* Change color on hover */
  border-color: #fff;
  text-decoration: underline;


}

/* Active page button */
.ag-theme-alpine .ag-paging-panel .ag-paging-button[aria-pressed="true"] {
  background-color: #ca9bff;
  color: #4b2c6f; /* Color for the active button */
```

```css
  border-color: #b673de;

}

/* Custom page size dropdown */
.ag-theme-alpine .ag-paging-panel .ag-page-size-panel select {
  background-color: #b673de;
  color: #fff;
  border-radius: 8px;
  /* border: 2px solid #ca9bff; */
  padding: 5px;
  font-weight: 600;
  text-decoration: underline;

}

.ag-theme-alpine .ag-paging-panel .ag-page-size-panel select:hover {
  background-color: #7f2d91;
  border-color: #fff;
  text-decoration: underline;

}

/* Pagination dropdown on focus */
.ag-theme-alpine .ag-paging-panel .ag-page-size-panel select:focus {
  outline: none;
  box-shadow: 0 0 5px rgba(75, 75, 75, 0.5);
  text-decoration: underline;


}

/* Add custom styles for ag-Grid */
.ag-theme-alpine {
  --ag-border-color: #ca9bff;
  --ag-header-background-color: #b673de;
  --ag-header-font-weight: 1000;
  --ag-header-font-size: 18px;
  --ag-header-text-align: center;
  --ag-font-family: 'Arial', sans-serif;
  --ag-font-size: 17px;
  --ag-font-color: #4b2c6f;
  --ag-row-hover-color: rgba(255, 255, 255, 0.719);
  --ag-row-border-radius: 8px;
  --ag-background-color: #ca9bff;

}

.ag-theme-alpine .ag-header-cell {
  color: #4b2c6f;

}
```

```css

.ag-theme-alpine .ag-cell {
  color: rgb(81, 40, 130);

}

.ag-theme-alpine .ag-row-even {
  background-color: #ca9bff;
  padding-top: 2px;
  padding-bottom: 2px;
  font-weight: 600;

}

.ag-theme-alpine .ag-row-odd {
  background-color: #ffc4fb;
  padding-top: 2px;
  padding-bottom: 2px;
  font-weight: 600;

}

.ag-theme-alpine .ag-row:hover {
  background-color: rgba(255, 255, 255, 0.719);
  color: #7f2d91;

}
.ag-theme-alpine .ag-cell:hover {
  color: #7f2d91;
  text-decoration: underline;

}

.ag-theme-alpine .ag-cell a {
  color: #33133a;
  font-weight: 1000;
}

.ag-theme-alpine .ag-cell a:hover {
  text-decoration: underline;
  color: #7f2d91;
}



/* ------------------------------------------------------------------------------------------------ */
/*login page and register page*/
/* Styling for the movie title input field */
input[type="text"], input[type="email"], input[type="password"] {
  background-color: #f9f9f9;
  color: #642c70;
  border: 2px solid #b673de;
  padding: 10px;
  font-size: 16px;
  border-radius: 8px;
  width: 250px;
  margin-bottom: 10px;
}

input[type="text"]:focus , input[type="email"]:focus, input[type="password"]:focus {
  outline: none;
  border-color: #7f2d91;
  background-color: #fff;
}


/* Styling for the year select dropdown */
select {
  background-color: #f9f9f9;
  color: #642c70;
  border: 2px solid #b673de;
  padding: 10px;
  font-size: 16px;
  border-radius: 8px;
  width: 250px;
  margin-bottom: 10px;
}

select:focus {
  outline: none;
  border-color: #7f2d91;
  background-color: #fff;
}

/* Styling for the Search button */
button {
  background-color: #b673de;
  color: #fff;
  border: 2px solid #7f2d91;
  padding: 10px 20px;
  font-size: 16px;
  font-weight: bold;
  border-radius: 8px;
  cursor: pointer;
  width: 250px;
  margin-top: 10px;
}

button:hover {
  background-color: #7f2d91;
  border-color: #fff;
  color: #fff;
}

```

```css
/* Wrapper for the search bar */
.search-bar-wrapper {
  text-align: center;
  margin-top: 30px;
}

.movies-search {
  /* margin-top: 20px; */
  font-size: 24px;
  color: #512882;
  font-weight: 800;
}

.login-form {
  display: flex;
  flex-direction: column; /* Stack items vertically */
  align-items: center;
  gap: 5px;
  width: 250px;
  margin: 0 auto;
}

.login-form input, .login-form button {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border-radius: 8px;
}

/* ------------------------------------------------------------------------------------------------ */
/* Homepage */
.hero-section {
  display: flex;
  justify-content: center;
  align-items: center;  /* Aligns the text vertically in the center */
  height: 850px; /* 100% of the viewport height */
  width: 100%; /* Full width of the viewport */
  background-image: url('../assets/MovieHero.png'); /* Set the image as the background */
  background-size: cover; /* Makes the image cover the entire container */
  background-position: center center; /* Centers the background image */
  background-repeat: no-repeat; /* Prevents the image from repeating */
}

.hero-section h1{
  /* display: flex; */
  /* justify-content: center; */
  align-items: center;
  margin: 0;
  margin-block-start: 0em;
  margin-block-end: 0em;
}

```

```css

.hero-heading h1{
  font-size: 3.5rem;
  font-weight: 800;
  color: #7f2d91;
  text-shadow: 2px 2px 5px rgba(0, 0, 0, 0.7); /* Adds shadow for better readability */
  margin: 0; /* Removes any default margin */
  padding: 20px; /* Adds space around the text */
  text-align: center; /* Center-aligns the text */
  width: 100%; /* Ensures the heading stretches across the full width */
}

.hero-heading p{
  font-size: 1.2rem;
  color: #512882;
  margin-top: 30px;
}

/*---------------------------------------------------------------------------------------------- */
/* Common color for both MovieDetailPage and PersonDetailPage */
.common-text-color {
  color: #512882; /* Color used in MovieDetailPage */
}

/* PersonDetailPage styles */
.person-detail-container {
  padding: 20px;
  max-width: 1200px;
  margin: auto;
  display: flex;
  flex-direction: column;
  align-items: center;
}
.person-detail{
  display: flex;
}
.person-detail p{
margin-left: 20px;
margin-right: 20px;
}
.person-detail-container h1,
.person-detail-container h2,
.person-detail-container h3 {
  /* Apply the common color to all header tags */
  color: rgb(81, 40, 130);
  margin-bottom: 10px;
}

.person-detail-container ul {
  margin-top: 10px;
  list-style-type: none;
  padding-left: 0;
}

.person-detail-container li {
  margin-bottom: 5px;
}

/* Custom styles for Roles in Movies grid */
.roles-grid {
  margin-top: 10px;
  font-size: 16px;
  border-radius: 8px;
  overflow: hidden;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
  width: 90%;
  margin-bottom: 40px;
}

.roles-grid table {
  width: 100%;
  border-collapse: collapse;
  background-color: #ffc4fb;
  font-weight: 600;
}
.roles-grid table thead{
  font-size: larger;
}

.roles-grid th,
.roles-grid td {
  padding: 12px;
  text-align: left;
  border-bottom: 1px solid #ca9bff;
}

.roles-grid th {
  background-color: #b673de;


}

.roles-grid tr:nth-child(even) {
  background-color: #ca9bff;
}

/*
.roles-grid tr:first-child {
  background-color: #d4a1ff;
} */

.kroles-gridtr:nth-child(odd) {
  background-color: #ffc4fb;
}
```

```css

.roles-grid tr:hover {
  background-color: rgba(255, 255, 255, 0.719);
}

.roles-grid a {
  color: #512882;
  text-decoration: none;
  font-weight: 800;
}

.roles-grid a:hover {
  text-decoration: underline;
  color: #7f2d91;
}

/* Custom styles for IMDB Rating Distribution chart */
.chart-container {
  margin-top: 20px;
  margin-bottom: 60px;
  padding: 20px;
  background-color: #ffc4fb;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.5);
  width: 80%; /* Ensure it takes full width */
  height: 500px; /* Adjust the height as needed */
}

.chart-container h2 {
  margin-top: 40px;
  text-align: center;
  margin-bottom: 20px;
  color: #56ccf2;
}

.chart-container canvas {
  max-width: 100%;
  height: 100%; /* Adjust the height to 100% of the container */
  height: auto;
  display: block;
  margin: 0 ;
}

/* MovieDetailPage styles */

/* Apply the same color for MovieDetailPage headers */


```

▾ **CAP230_Assignment2_Krittaya/src/pages/LoginPage.jsx** Download

```jsx
import React, { useState, useEffect } from "react";
import { useNavigate, useLocation } from "react-router-dom";
import { setToken } from "../utils/auth"; // Import the setToken function
import './style.css';

function LoginPage() {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [error, setError] = useState("");
  const [expiredSession, setExpiredSession] = useState(false); // Track expired session
  const [isFromPersonDetail, setIsFromPersonDetail] = useState(false); // Track if the user came from PersonDetailPage
  const navigate = useNavigate();
  const location = useLocation();

  // Get the redirectTo URL from query params (this could be the previous page they were trying to access)
  const queryParams = new URLSearchParams(location.search);
  const redirectTo = queryParams.get("redirectTo") || "/"; // Default to home page if not present
  const expired = queryParams.get("expired"); // Check if session expired

  // Determine if the user came from PersonDetailPage by checking for "redirectTo" in the query params
  useEffect(() => {
    if (localStorage.getItem("token")) {
      navigate(redirectTo); // If the user is already logged in, go to the previous page
    }

    // If expired=true, set the expired session state to true
    if (expired === "true") {
      setExpiredSession(true);
    }

    // Check if the user is coming from PersonDetailPage
    if (queryParams.has("redirectTo")) {
      setIsFromPersonDetail(true);
    }
  }, [navigate, redirectTo, expired, location.search]);

  const handleLogin = async (e) => {
    e.preventDefault();
    setError("");

    try {
      const res = await fetch("http://4.237.58.241:3000/user/login", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ email, password }),
      });

      const data = await res.json();

      if (res.ok && data.bearerToken?.token && data.refreshToken?.token) {
        setToken(data.bearerToken.token, data.refreshToken.token);
        localStorage.setItem("email", email);
        navigate(redirectTo); // Redirect to the previous page after successful login
        window.location.href = window.location.href; // Reload the current page
      } else {
        setError("Login failed. Please check your credentials.");
      }
    } catch (err) {
      console.error("Login error:", err);
      setError("Something went wrong. Try again.");
    }
  };

  return (
    <div>
      <h1>Login</h1>
      {/* Display message based on whether the user came from PersonDetailPage or the session has expired */}
      {isFromPersonDetail && !localStorage.getItem("email") && (
        <p>Please login before viewing the details of the person.</p>
      )}
      {expiredSession && localStorage.getItem("email") && (
        <p>Please login again. Your session has expired.</p>
      )}

      <form onSubmit={handleLogin} className="login-form">
        <input
          type="email"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
          placeholder="Email"
          required
        />
        <input
          value={password}
          onChange={(e) => setPassword(e.target.value)}
          placeholder="Password"
          type="password"
          required
        />
        <button type="submit">Login</button>
      </form>
      {error && <p style={{ color: "red" }}>{error}</p>}
    </div>
  );
}

export default LoginPage;
```

▾ **CAP230_Assignment2_Krittaya/src/pages/RegisterPage.jsx** Download

```
import React, { useState } from "react";
import './style.css';

function RegisterPage() {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [confirmPassword, setConfirmPassword] = useState(""); // New state for confirm password
  const [error, setError] = useState("");

  const handleRegister = async (e) => {
    e.preventDefault();
    try {
      const res = await fetch("http://4.237.58.241:3000/user/register", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ email, password }),
      });

      const data = await res.json();

      if (res.ok) {
        alert("Registration successful, please login.");
        window.location.href = "/login";
      } else {
        setError(data.message || "Registration failed");
      }
    } catch (err) {
      setError("Server error");
    }
  };

  return (
    <div>
      <h1>Register</h1>
      {error && <p style={{ color: "red" }}>{error}</p>}
      <form onSubmit={handleRegister}>
        <input
          type="email"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
          placeholder="Email"
          required
        />
        <br />
        <input
          type="password"
          value={password}
          onChange={(e) => setPassword(e.target.value)}
          placeholder="Password"
          required
        />
        <br />
        <input
          type="password"
          value={confirmPassword}
          onChange={(e) => setConfirmPassword(e.target.value)}
          placeholder="Confirm Password"
          required
        />
        <br />
        <button type="submit">Register</button>
      </form>
    </div>
  );
}

export default RegisterPage;

```

CAP230_Assignment2_Krittaya/src/pages/MovieDetailPage.jsx Download

```
import React, { useEffect, useState } from 'react';
import { useParams, Link } from 'react-router-dom';
import './style.css';


function MovieDetailPage() {
  const { imdbID } = useParams();
  const [movie, setMovie] = useState(null);

  useEffect(() => {
    async function fetchMovieDetails() {
      try {
        const res = await fetch(`http://4.237.58.241:3000/movies/data/${imdbID}`, {
          headers: { Accept: "application/json" }
        });

        if (!res.ok) {
          console.error("❌ Failed to fetch movie:", res.statusText);
          return;
        }

        const data = await res.json();
        console.log("🎬 Movie data:", data);
        setMovie(data);
      } catch (err) {
        console.error("⚠️ Error fetching movie details:", err.message);
      }
    }

    if (imdbID) fetchMovieDetails();
  }, [imdbID]);

  if (!movie) return <p>Loading...</p>;

  // Get people for all roles dynamically
  const getPeopleByRole = () =>
    movie.principals?.reduce((acc, person) => {
      if (!acc[person.category]) acc[person.category] = [];
      acc[person.category].push(person);
      return acc;
    }, {}) || {};

  const peopleByRole = getPeopleByRole();

  // Extract ratings
  const imdbRating = movie.ratings?.find(r => r.source.includes('Internet'))?.value
  ? `${movie.ratings.find(r => r.source.includes('Internet')).value}/10`
  : 'Unknown';

```

```
const rtRating = movie.ratings?.find(r => r.source.includes('Rotten'))?.value
  ? `${movie.ratings.find(r => r.source.includes('Rotten')).value}/100`
  : 'Unknown';

const metaRating = movie.ratings?.find(r => r.source.includes('Metacritic'))?.value
  ? `${movie.ratings.find(r => r.source.includes('Metacritic')).value}/100`
  : 'Unknown';

  // Capitalize the first letter of the role
// Capitalize the first letter of each word, remove underscores, and join them
const capitalizeRole = (role) => {
  return role
    .split('_')  // Split the role by underscores
    .map(word => word.charAt(0).toUpperCase() + word.slice(1))  // Capitalize the first letter of each word
    .join(' ');  // Join the words with a space
};

  return (
    <div className="movie-detail-container">
      <h1>{movie.title} ({movie.year})</h1>
      <div className="movie-detail">
      {movie.poster && <img src={movie.poster} alt={movie.title} className="movie-poster" />}
      <div className="rating">
      <h3>⭐ Ratings ⭐</h3>
        <p><strong>IMDb:</strong> {imdbRating}</p>
        <p><strong>Rotten Tomatoes:</strong> {rtRating}</p>
        <p><strong>Metacritic:</strong> {metaRating}</p>
      </div>
      <div className= "movie-detail-1">
      <p><strong>Year:</strong> {movie.year || 'N/A'}</p>
      <p><strong>Runtime:</strong> {movie.runtime && movie.runtime !== 'N/A' ? `${movie.runtime} minutes` : 'Unknown'}</p>
      <p><strong>Genres:</strong> {movie.genres?.join(', ') || 'Unknown'}</p>
      <p><strong>Country:</strong> {movie.country || 'Unknown'}</p>
      <p><strong>Box Office:</strong> {movie.boxoffice ? `$${new Intl.NumberFormat().format(movie.boxoffice)}` : 'Unknown'}</p>
      </div>
      <p className="plot"><strong></strong> {movie.plot || 'Unknown'}</p>


      </div>

      <h2 className="key-people">Key People</h2>
      <div className="key-people-grid">
        <table>
          <thead>
            <tr>
              <th>Role</th>
              <th>Name</th>
              <th>Character</th>
            </tr>
          </thead>
          <tbody>
            {/* Dynamically render all people by role */}
            {Object.entries(peopleByRole).map(([role, people]) => (
              people.map((person) => (
                <tr key={person.id}>
                  <td>{capitalizeRole(role)}</td>{/* Capitalize the role */}
                  <td className="movie-detail-people"><Link to={`/people/${person.id}`} >{person.name}</Link></td>
                  <td>{person.characters?.join(", ") || "N/A"}</td>
                </tr>
              ))
            ))}
          </tbody>
        </table>
      </div>
    </div>
  );
}

export default MovieDetailPage;


```

▾ **CAP230_Assignment2_Krittaya/src/pages/PersonDetailPage.jsx** Download

```jsx
import React, { useState, useEffect } from 'react';
import { useParams, useNavigate, Link } from 'react-router-dom';
import { clearToken } from '../utils/auth'; // Import clearToken function
import { Bar } from 'react-chartjs-2'; // Import the Bar chart component
import { Chart as ChartJS, CategoryScale, LinearScale, BarElement, Title, Tooltip, Legend } from 'chart.js'; // Import necessary Chart.js components
import './style.css';

// Register the necessary Chart.js components
ChartJS.register(CategoryScale, LinearScale, BarElement, Title, Tooltip, Legend);

function PersonDetailPage() {
  const { id } = useParams(); // Get person ID from URL parameters
  const [person, setPerson] = useState(null); // State to store person details
  const [error, setError] = useState(null); // For error handling
  const [ratingData, setRatingData] = useState(null); // For chart data
  const navigate = useNavigate(); // Hook for navigation

  // Function to check if the token is expired
  const isTokenExpired = (token) => {
    const parts = token.split('.');
    const payload = JSON.parse(atob(parts[1]));
    return Date.now() >= payload.exp * 1000;
  };

  useEffect(() => {
    const fetchPersonDetails = async () => {
      const token = localStorage.getItem("token"); // Get the token from localStorage

      if (!token || isTokenExpired(token)) {
        setError("Token expired or not found. Please log in.");
        clearToken(); // Clear the token from localStorage
        navigate(`/login?redirectTo=/people/${id}&expired=true`); // Redirect to login with a "redirectTo" param
        return;
      }

      try {
        const res = await fetch(`http://4.237.58.241:3000/people/${id}`, {
          method: 'GET',
          headers: {
            'Accept': 'application/json',
            'Authorization': `Bearer ${token}`, // Include the token in the header
          },
        });

        if (!res.ok) {
          if (res.status === 401) {
            clearToken();  // Remove the token from localStorage
            setError("Unauthorized access. Please log in again.");
            navigate('/login');  // Redirect to the login page
            return;
          }
          throw new Error(`Failed to fetch person data: ${res.statusText}`);
        }
        const data = await res.json();
        setPerson(data); // Set the fetched data to state

        // Prepare the rating data for the chart
        const imdbRatings = data.roles?.map(role => role.imdbRating).filter(Boolean); // Get IMDB ratings and filter out falsy values

        // Count ratings in specific ranges
        const ratingRanges = [0, 2, 4, 6, 8, 10];
        const ratingCount = new Array(ratingRanges.length - 1).fill(0);

        imdbRatings.forEach(rating => {
          for (let i = 0; i < ratingRanges.length - 1; i++) {
            if (rating >= ratingRanges[i] && rating < ratingRanges[i + 1]) {
              ratingCount[i]++;
              break;
            }
          }
        });

        // Set the chart data
        setRatingData({
          labels: ratingRanges.slice(0, -1).map((r, i) => `${r}-${ratingRanges[i + 1]}`),
          datasets: [
            {
              label: 'IMDB Ratings Count',
              data: ratingCount,
              backgroundColor: 'rgb(141, 75, 192)', // Bar color
              borderColor: 'rgb(141, 75, 192)', // Bar border color
              borderWidth: 1,
            },
          ],
        });

      } catch (err) {
        setError(`Error: ${err.message}`); // Handle other errors
      }
    };

    fetchPersonDetails();
  }, [id, navigate]); // Re-run effect when `id` or `navigate` changes

  if (error) {
    return <div>{error}</div>; // Show error message if there's any
  }

  if (!person) {
    return <p>Loading...</p>; // Show loading state until person data is fetched
  }

  return (
    <div className="person-detail-container">
      <h1>{person.name}</h1>
      <div className="person-detail">
        <p><strong>Born:</strong> {person.birthYear}</p>
        <p><strong>Death:</strong> {person.deathYear || "Unknown"}</p>
      </div>

      <h2>Roles in Movies:</h2>
      <div className="roles-grid">
        <table>
          <thead>
            <tr>
              <th>Role</th>
              <th>Movie Title</th>
              <th>Character(s)</th>
              <th>IMDB Rating</th>
            </tr>
          </thead>
          <tbody>
            {person.roles && person.roles.length > 0 ? (
              person.roles.map((role, idx) => (
                <tr key={idx}>
                  <td>{role.category.charAt(0).toUpperCase() + role.category.slice(1)}</td>{/* Capitalize the role */}
                  <td>
                    <Link to={`/movies/${role.movieId}`} className="movie-link">
                      {role.movieName}
                    </Link>
                  </td>
                  <td>{role.characters?.join(", ") || "N/A"}</td>
                  <td>{role.imdbRating || "N/A"}</td>
                </tr>
              ))
            ) : (
              <tr>
                <td colSpan="4">No roles available.</td>
              </tr>
            )}
          </tbody>
        </table>
      </div>

      <h2>IMDB Rating Distribution:</h2>
      {ratingData ? (
        <div className="chart-container">
          <Bar
            data={ratingData}
            options={{
              responsive: true,
              maintainAspectRatio: false, // Disable aspect ratio to make the chart fill the container
              plugins: {
                title: {
                  display: true,
                  text: 'Distribution of IMDB Ratings',
                },
                legend: {
                  position: 'top',
                },
              },
              scales: {
                x: {
                  beginAtZero: true,
                },
                y: {
                  beginAtZero: true,
                },
              },
            }}
          />

        </div>
      ) : (
        <p>Loading chart...</p>
      )}
    </div>
  );
}

export default PersonDetailPage;

```