**Project Assignment**

**MOBI6012 – Web Design (ReactJS)**

**Requirements:**

1. Build a web app using **React.js** (allowed to use **create-react-app**, **create-next-app**, or **your own starter template**) that has 4 pages, **a Search Page**, **List Page**, **Detail Page**, and **Favorites Page**, the details will be provided below. Please consider your web app’s **UI/UX** to be appropriate for the customer segment.
2. Fetch data from **a REST API** as your data source. But using **GraphQL** is a **big plus**. You may freely choose any API but in case you don’t know what to choose you can use the spotify api:

* <https://spotify-rest.up.railway.app/> (**REST Wrapper**)
* <https://spotify-graphql.up.railway.app/> (**GraphQL Wrapper**)

If you want to use any of the wrapper locally (provided you have GO lang installed in your computer) feel free to go to:

* <https://github.com/abenbyy/spotify-rest>
* <https://github.com/abenbyy/spotify-graphql>

and follow the steps provided in the README.md file. If you found any error from the API, please contact the Subject Coordinator immediately.

If you’re interested in using other people’s GraphQL here’s a list to help you start <https://github.com/APIs-guru/graphql-apis>

1. Your web app UI/UX should be **mobile first**

**Pages:**

1. **Search Page**

Display a search bar to search data from your API. (ex. Search a Song)

1. **List Page**

Show a list of items in this page, the item can be anything based on the theme of your application. When an item is clicked, it will go to Detail Page.

1. **Detail Page**

Show detailed information of the item clicked. Display a button to add to/remove from the user’s favorites

1. **Favorites Page**

Show a list of items that the user has added to their favorites. Display a button to remove from the user’s favorite

**To be Collected:**

1. Please kindly provide **hosted project repository** (can be GitHub, GitLab, or BitBucket)
2. Doing an extra step and **deploying the web app** can increase your knowledge on deployments and productions (you can use free platforms such as Heroku, Vercel, GithHub pages, Firebase, Netlify, or your very own VPS!)
3. Please **write a simple documentation** of your app, provide an explanation on how your web apps work **with a link to the repository and deployed app** (if you did it).

**Notes:**

1. Good and usable UI/UX is a big plus! We always appreciate you if you can give your best in website designs.
2. Feel free to install any packages that might help you develop the web app such as react-router-dom, UI components, SCSS/SASS, etc.
3. Make your web app as performant as possible. You can use tools such as **Lighthouse** (<https://developers.google.com/web/tools/lighthouse>) to help you audit your web-app
4. Automated test (integration and unit test) is a huge plus and might help you in your college education (NOT MANDATORY FOR THE QUALIFICATION)!
5. Web Platform Engineers in \*\*\*\*\*\*\*\*\* are currently using these stacks, so if you are interested, you can learn and show us your competencies with technologies listed below (NOT MANDATORY FOR THE QUALIFICATION):

* React with React Hooks and React Context
* GraphQL using Apollo Client (<https://www.apollographql.com/>)
* CSS-in-JS using Emotion (<https://www.emotion.sh/>)
* Unit Test using Jest (<https://jestjs.io/>) and React Testing Library (<https://testing-library.com/>)

*If you have any questions, feel free to contact your Subject Coordinator!*

*Remember, no Cheating!*