THE PROBLEM ASSIGNMENT

The problem assignment has three components. The first component is the frontend development. The second component is deployment. The third component is documentation.

1. **Development: Please develop an API and a web app.**

Instruction: Your API should make the data.csv file available to your web app. Your web app should query your API, and interactively visualize the data on screen. The visualization should depend on the user's choices, enabling the user to meaningfully interact with the data. The visualization itself can rest on a basemap such as the basemaps available from ArcGIS (please refer to the pdf file that describes the data). The web app should be intuitive, self-explanatory, interactive, and aesthetically appealing. Specifically, interaction with the data should not require any additional instruction. You can use node.js to develop the API as a REST or GraphQL API, and you can use React or Angular to develop the web app. Nonetheless, you can also use any other technology that you feel comfortable with or see fit for the purpose. In developing your web app and API, please use classes, interfaces, or modules wherever possible to allow easy expandability.

2. **Deployment: To deploy your web app and your API on the web, please use Docker.**

Instruction: Please deploy the web app through Github React pages: https://github.com/gitname/react-gh-pages. For each of the two, the web app and API, please also provide to us the docker, docker-compose.yml, action.yml, and dependabot.yml files. The libraries that you use for the API and web app should receive regular updates. The containers running the web app and API should follow the twelve factors of cloud apps and therefore allow the possibility of cloud deployment later. Please store your code (React app, API, and the remaining files) into a dedicated Github repository. Please let us know the name of the repository and grant us access.

3. **Documentation: The dedicated Github repository should include a README file. The README file suffices for the documentation.**

Instruction: The README file should list functional and non-functional requirements of your web app and API. The key functional requirement is a brief description of what your web app does in enabling the user to interact with the dataset visually. The non-functional requirements should list the tools, frameworks, and programming languages you used for the coding. They should also be versioned.