Cesium Material Application

Tech Stack

- Front React-Application with Bootstrap and CSS
- API Node JS & Express.
- PostGres Database on Local.

How to run the project

Clone the Github-Project and before running the Project, Please install postgres and Create a database called “cesium”. Please run the schema which I have written in database SQL.

```
CREATE DATABASE cesium;

CREATE TABLE materials(
    material_id SERIAL PRIMARY KEY,
    material_name VARCHAR(255),
    material_volume INT,
    material_deliverydate VARCHAR(255),
    material_cost NUMERIC(8,2),
    material_color VARCHAR(255)
);
```

The next thing I would recommend to do is please change the db config file “db.js” to match the Postgres end points.

```
const pool = new Pool({
    user: "adityavarma",
    password : "Messi",
    host : "localhost",
    port : 5432,
    databse : "cesium"
});
```

Now, you are all set with the project, you just need to install the following dependencies.

npm install node

npm install nodemon

npm install react

npm install react-bootsrap

After you are done installing all the dependencies, please go to the root folder and recurse into server first and enter nodemon index.js

This should run the server and watch for any changes.

Next thing you would do is to hop back to the root folder and recurse to Material-app folder and run npm update and npm start, This will run the application on your localhost. The app will be up and running at your localhost port.

Challenges & Time Allocated for Each Pa

- I would say the UI was challenging, I tried my best to replicate the sketch. Especially the Circle which needs to be colored from the getRequest() we get was a bit tricky.
- After Deletion, To reset the state and dynamically updating the DOM was also a bit fun. I used filter and spitted all the rows to the new state which were not matching with the deleted id.
- I used getMaterials() at the end of Add to dynamically update the DOM after adding, I know this is not a good practice, But when I used useEffect() I had some problems with delete. So I did this tradeoff with adding [] at the end of useEffect to stop infinite “getCalls()”.
- Setting the API was the most easy part, It would take more time if I would have done remotely I guess.

SETTING DATABSE AND API – 30MINS

REACT – UI - 120MINS

DOCUMENTATION AND COMMENTS – 30MINS.