# Histogram Chart

## Front End:

### Softwares required:

**Visual studio Code** : use below link to install.

<https://code.visualstudio.com/>

**Node** : Use below link to install latest Node version.

<https://nodejs.org/en/download>

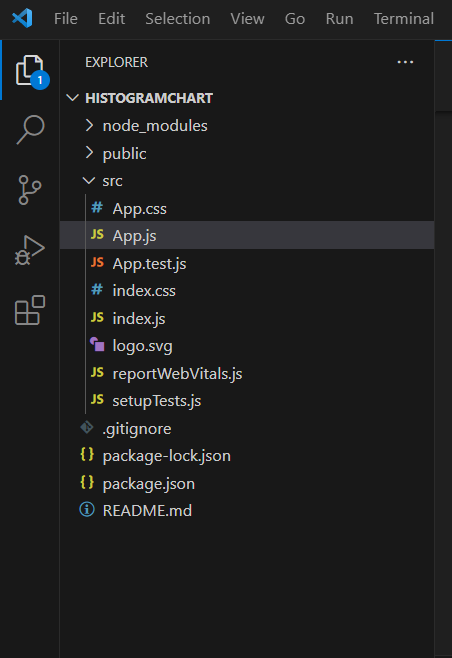
### Follow Below steps to create a project :

* Load Visual studio code and open a new terminal.
* Create a folder in local drive .EX : D:\MeritConnects\Histogram\_project
* Navigate to this path in terminal. To navigate use below command

Cd D:\MeritConnects\Histogram\_project

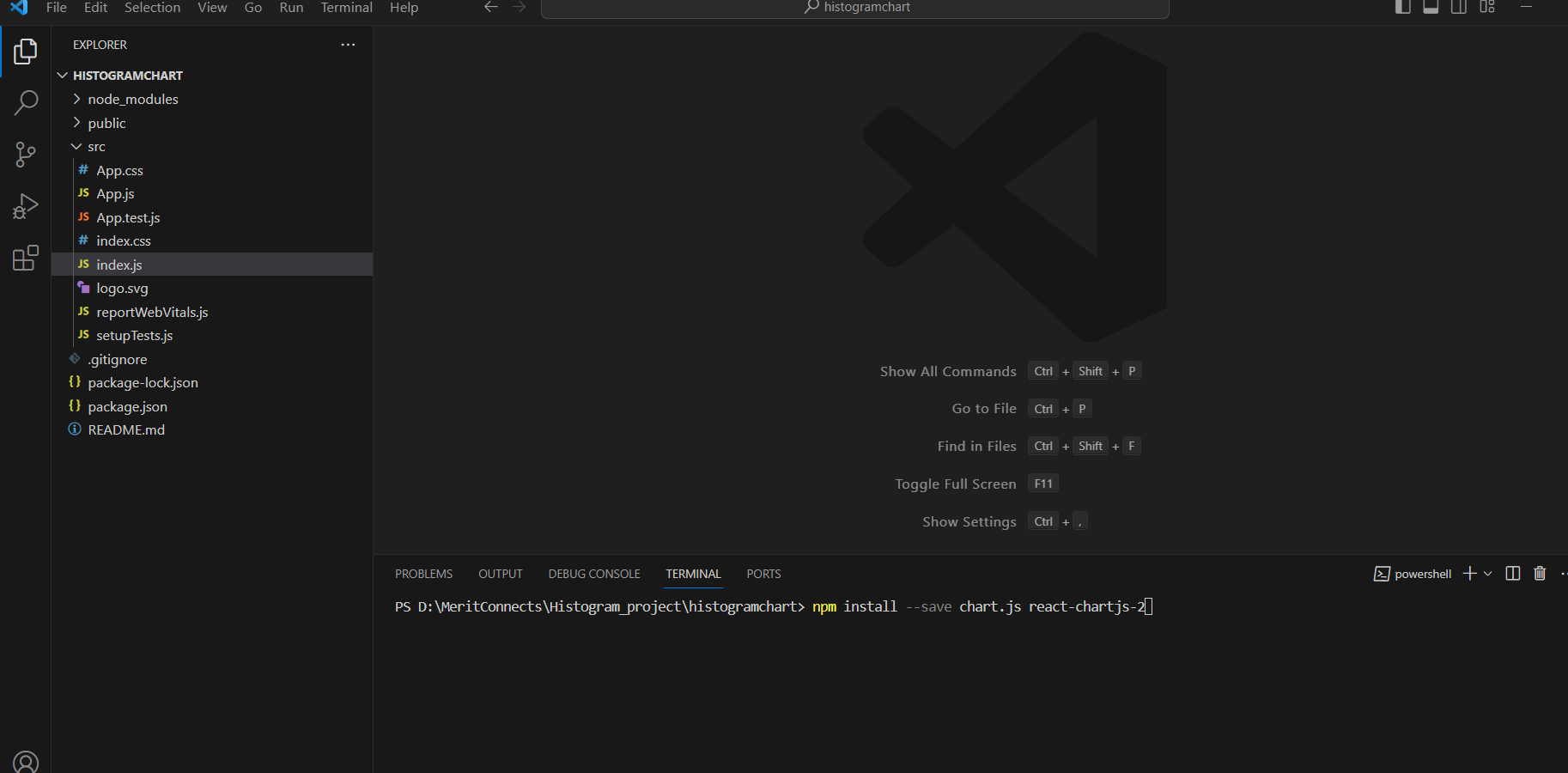
* Create a new Project using below command

Npx create-react-app appname

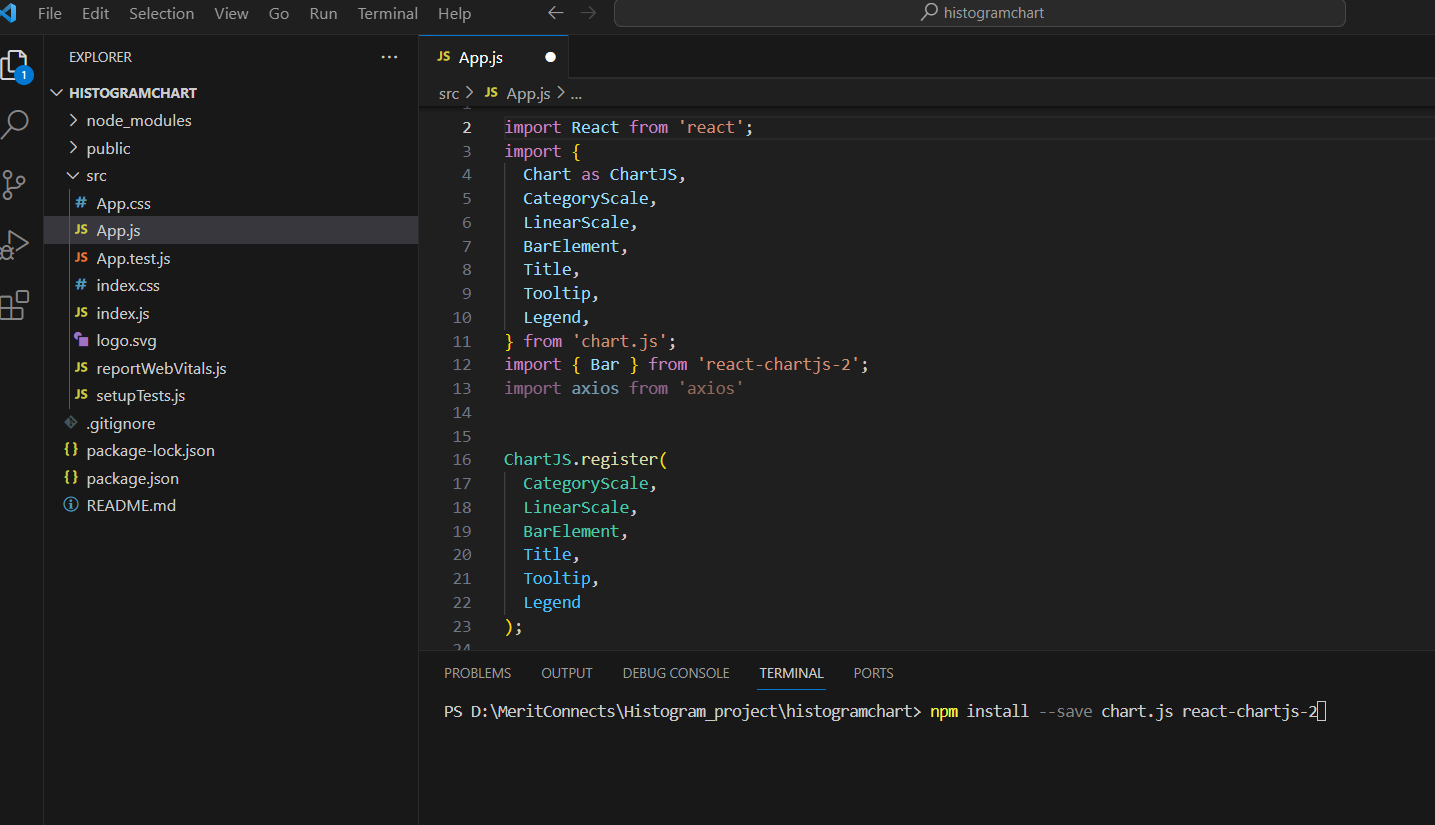


* Once the project is created ,install charts related packages using below;

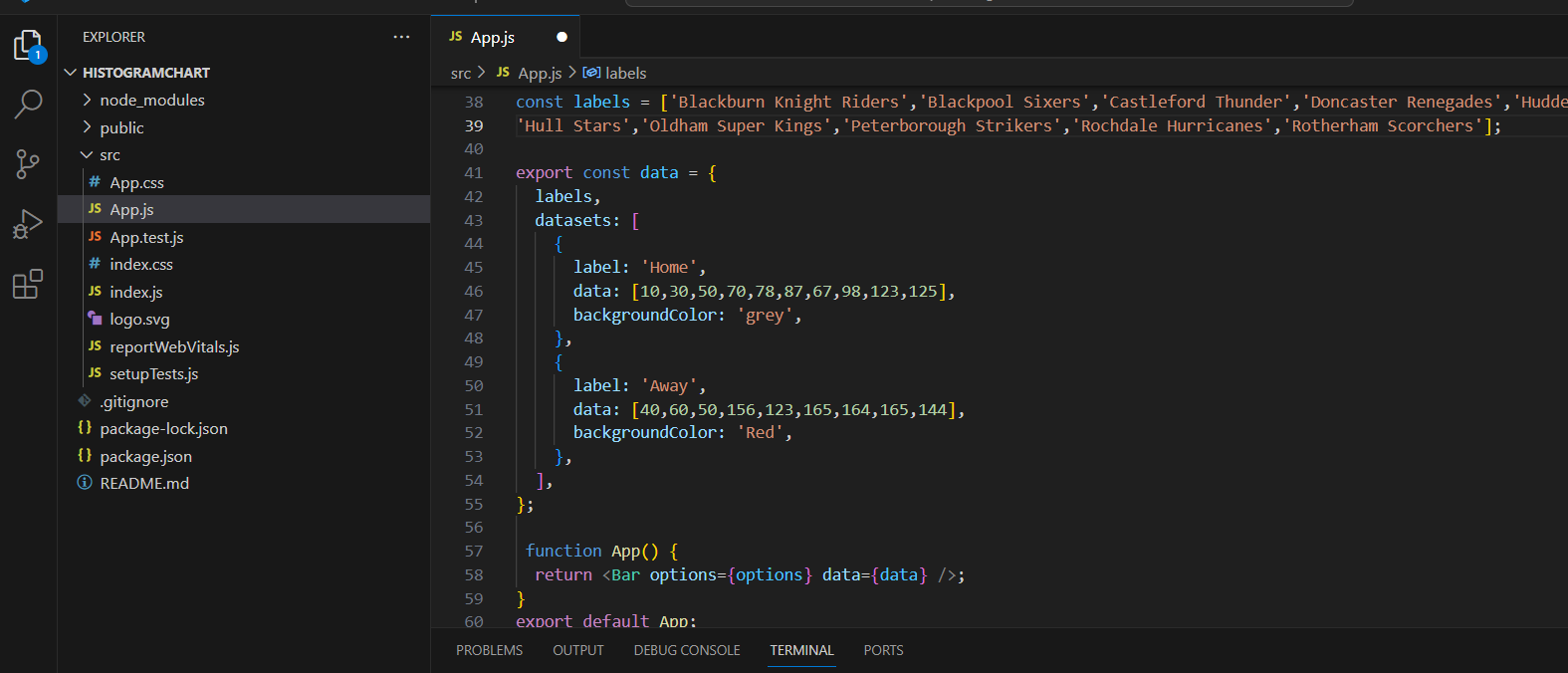
npm install --save chart.js react-chartjs-2



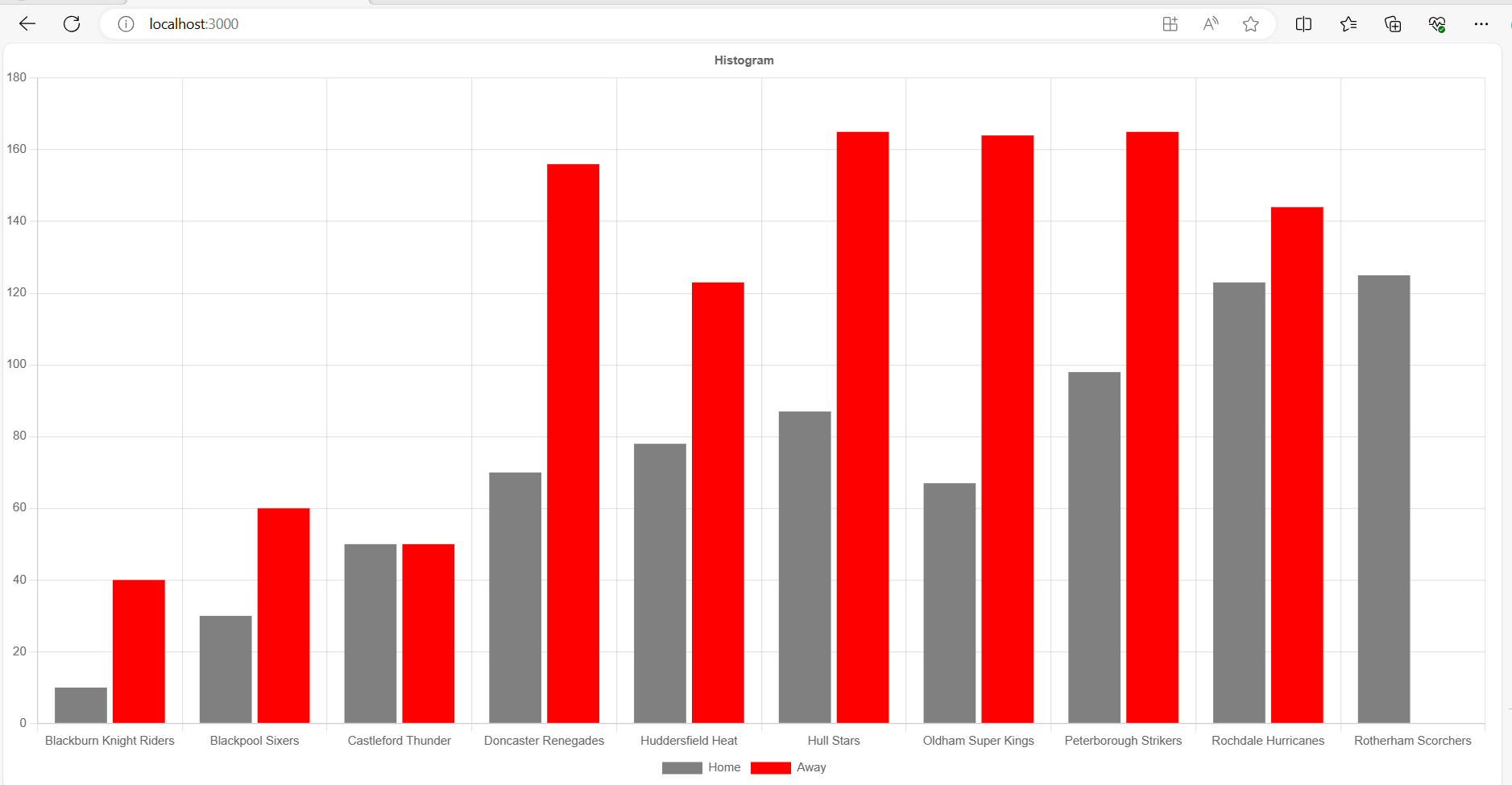
* Import Charts relates packages and register them as below :



* Declare labels and datasets as below and return default function.



* Run “Npm Start “ command to load the application and result will be displayed as below:



## Docker Setup

docker-compose.yml

|  |
| --- |
| version: '3'  services:    web:      build: ./web      ports:        - "3000:3000"      links:        - api  # Flask App    api:      build: ./api      ports:        - "5000:5000"      deploy:        resources:          limits:            cpus: '0.25'   # Limit to 25% of one CPU core            memory: 1024M  # Limit to 512 megabytes of RAM |

API Dockerfile

|  |
| --- |
| FROM python  WORKDIR /api  COPY requirements.txt ./  RUN pip install --no-cache-dir -r requirements.txt  COPY . .  CMD [ "python", "app.py" ] |

API requirements

|  |
| --- |
| Flask  pandas  requests |

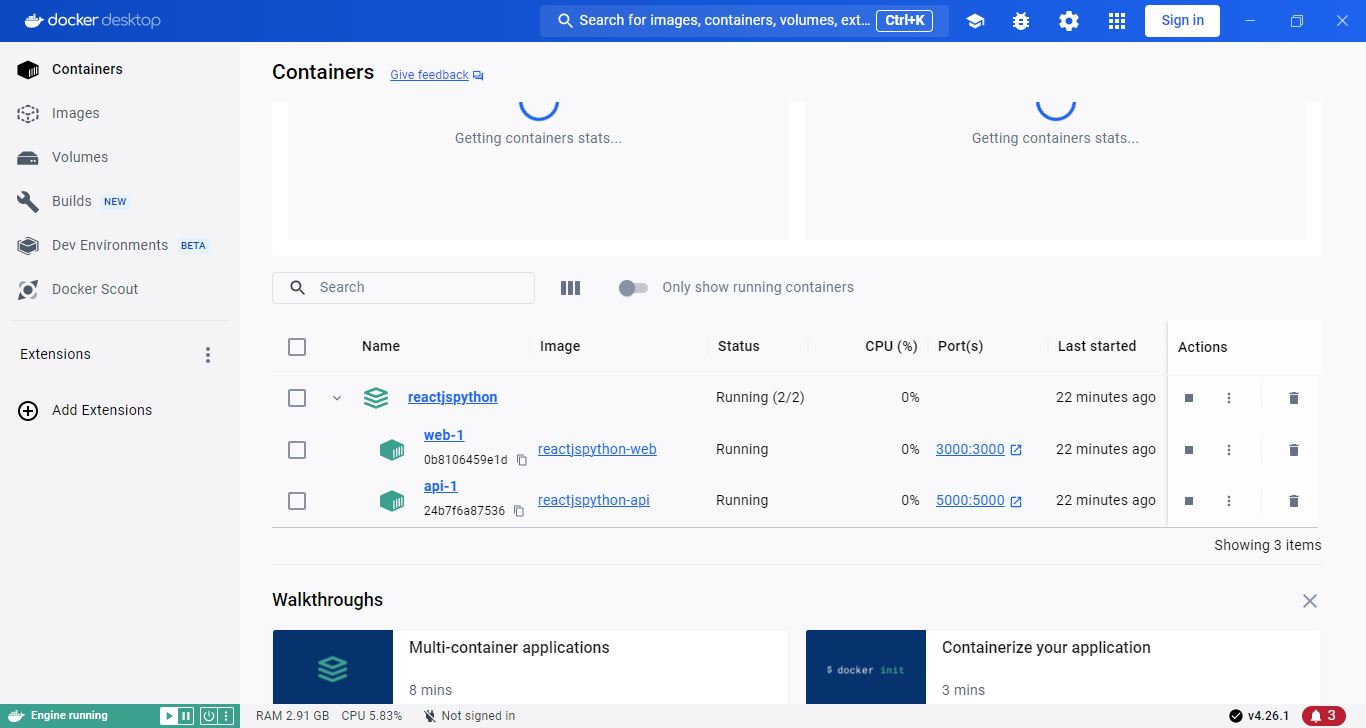
WEB Dockerfile

|  |
| --- |
| # Use an official Node runtime as a parent image  FROM node  # Set the working directory in the container  WORKDIR /src  # Copy package.json and package-lock.json to the working directory  COPY package\*.json ./  # Install dependencies  RUN npm install  # Copy the local code to the container  COPY . .  # Expose the port the app runs on  EXPOSE 3000  # Define the command to run app  CMD ["npm", "start"] |

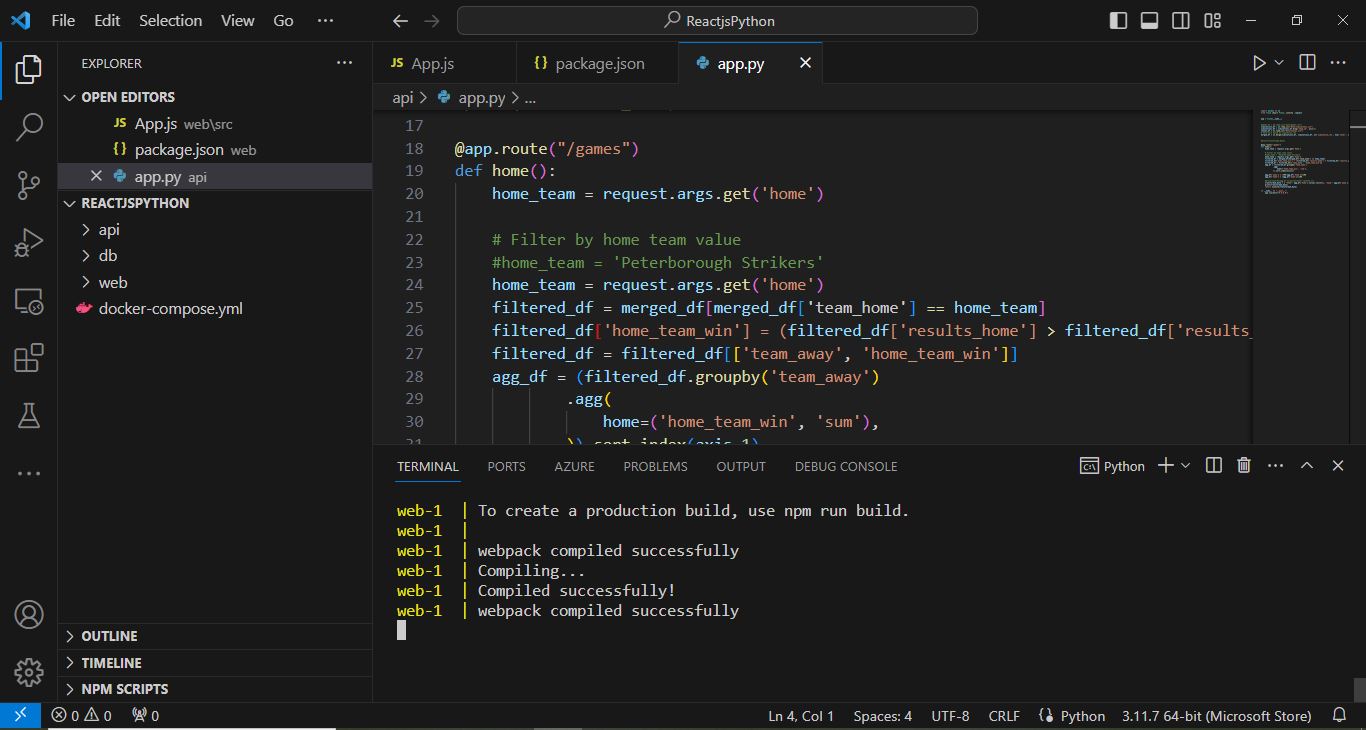
#### Commands used

To Build the image : docker-compose build

To Create the container: docker-compose up

Docker container created   
  


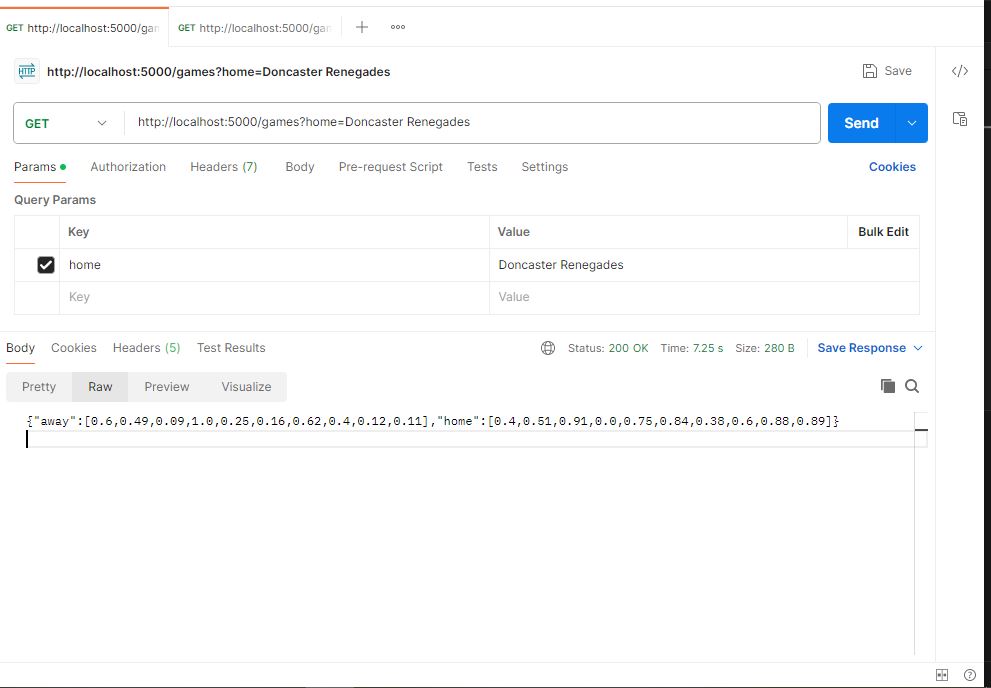
Folder



API Code

|  |
| --- |
| import pandas as pd  from flask import Flask, jsonify ,request  app = Flask(\_\_name\_\_)  #games\_df = pd.read\_csv("data/games.csv")  simulations\_df = pd.read\_csv("data/simulations.csv")  simulations\_df = simulations\_df.drop('team\_id', axis=1)  venues\_df = pd.read\_csv("data/venues.csv")  # Self-join based on simulation\_run  merged\_df = pd.merge(simulations\_df, simulations\_df, on='simulation\_run', how='inner', suffixes=('\_home', '\_away'))    #print(transformed\_data)  @app.route("/games")  def home():      home\_team = request.args.get('home')        # Filter by home team value      #home\_team = 'Peterborough Strikers'      home\_team = request.args.get('home')      filtered\_df = merged\_df[merged\_df['team\_home'] == home\_team]      filtered\_df['home\_team\_win'] = (filtered\_df['results\_home'] > filtered\_df['results\_away']).astype(int)      filtered\_df = filtered\_df[['team\_away', 'home\_team\_win']]      agg\_df = (filtered\_df.groupby('team\_away')              .agg(                  home=('home\_team\_win', 'sum'),              )).sort\_index(axis=1)      agg\_df['away'] = (100-agg\_df['home'])/100      agg\_df['home'] = (agg\_df['home'])/100      #print(jsonify(agg\_df.to\_dict(orient='records')))      transformed\_data = { "home": agg\_df['home'].values.tolist(), "away": agg\_df['away'].values.tolist()}      print(transformed\_data)      return jsonify(transformed\_data)  if \_\_name\_\_ == "\_\_main\_\_":      app.run(host="0.0.0.0") |

Postman request



### Pending Item:

For Inetgration between UI and API, Use Axios package for react.

Use below command and code snippet:

npm i axios

    useEffect(() => {

        axios.get("API url endpoint")

            .then((response) => console.log(response.data));

    }, []);