



National University of Computer and Emerging Sciences

Assignment 2

Web Engineering

GITHUB LINK: [HERE](#)

- The project contains the following Files:
- dashboard.css
- dashboard.js
- dashboard.html
- tables.css
- tables.html
- tables.js

Code shared in both:

- Both pages contain a navigation bar with 2 button to switch between the two pages.
- Both pages contain an input text box for getting city name and search the data related to it.
- Both use the method of using Open Weather API's current weather url to extract the longitude and latitude of city provided, and then use it to in the forecast url to get forecast data.

Code In Dashboard Page:

- The dashboard page contains the XML response data of forecast and current weather received via Ajax.
- Each current data is extracted from it's respective tag Name and it's respective value from attribute, that is entered into the `<p>` tag according to the ids.
- Then data of dates and temperature and weather is stored into array, by turning node lists into arrays by the use of `Array.from` and `.map`.
- The 2d attribute is extracted from each empty table in the HTML via their ids and then the labels, and data sets are added.
- The data sets are directly added to line and bar graph and both are assigned delay and drop animations as provided in the official website.
- In the doughnut graph, the data sets are first compared to another default data set containing all the values of main weather dataset and the days are counted accordingly, then the weather conditions with zero days of occurrence are filtered from the data set. The final values of dataset are used into creating the doughnut graph with delay animation.

Code In Tables Page:

- The tables page contains the XML response data of forecast and current weather received via Ajax.
- Each forecast data is extracted from it's respective tag Name and it's respective value from attribute using index values.
- Each data from extracted dataset and index value is entered as `<td>` tag of table according to the ids of `<tr>` tags corresponding to the data extracted, using a for loop of 10 and each `<td>` is given id through the corresponding dataset name and index number in for loop i.e `"data_" + i`.

- When ever user enter a new city, the data is first removed using a for loop of 10 and adding that to corresponding string name of dataset (e.g: "time_" + i, "temperature" + i) along with using the .remove function.
- For the Chat bot the Gemini Api is used. The user data is entered into a input text box and extracted when he click search. Then in a string, all the data from the table is entered via for loop and 10, using ids based of name of dataset and index number.
- After that the question Is also appended to the string along with a warning to only answer weather related questions and if there are some other questions then please say sorry I can't help you with that.
- Then the string is sent via ajax HTTPS and the response is given to the response <p> tag.