WDD330 – week 4 - project plan – Ben Johansen

Temperature checker

Overview:

Prompts the user for a longitude and latitude, then displays the nearest town/city and its current temperature. It also has the option to use the user’s current location, which can be found using the IP Geolocation API from geoapify. I use a weather app on my phone and the feature to see the current temperature in my area is one of the features that I use the most.

Target Audience:

Anyone who needs to know what the current temperature is

Major Functions:

- **getLocationFromIP**

  uses the IP geolocation API to get the longitude and latitude from the user’s IP address

- **getCityFormLocation**

  uses the longitude and latitude to get the nearest town/city

- **getLocationFromAddress**

  uses the address to get the longitude and latitude

- **getWeatherData**

  gets the weather data for the longitude and latitude

- **getData**

  gets the data from an API and handles errors

- **displayData**

  displays the nearest town/city and the current temperature

- **handleGetWeatherBtn**

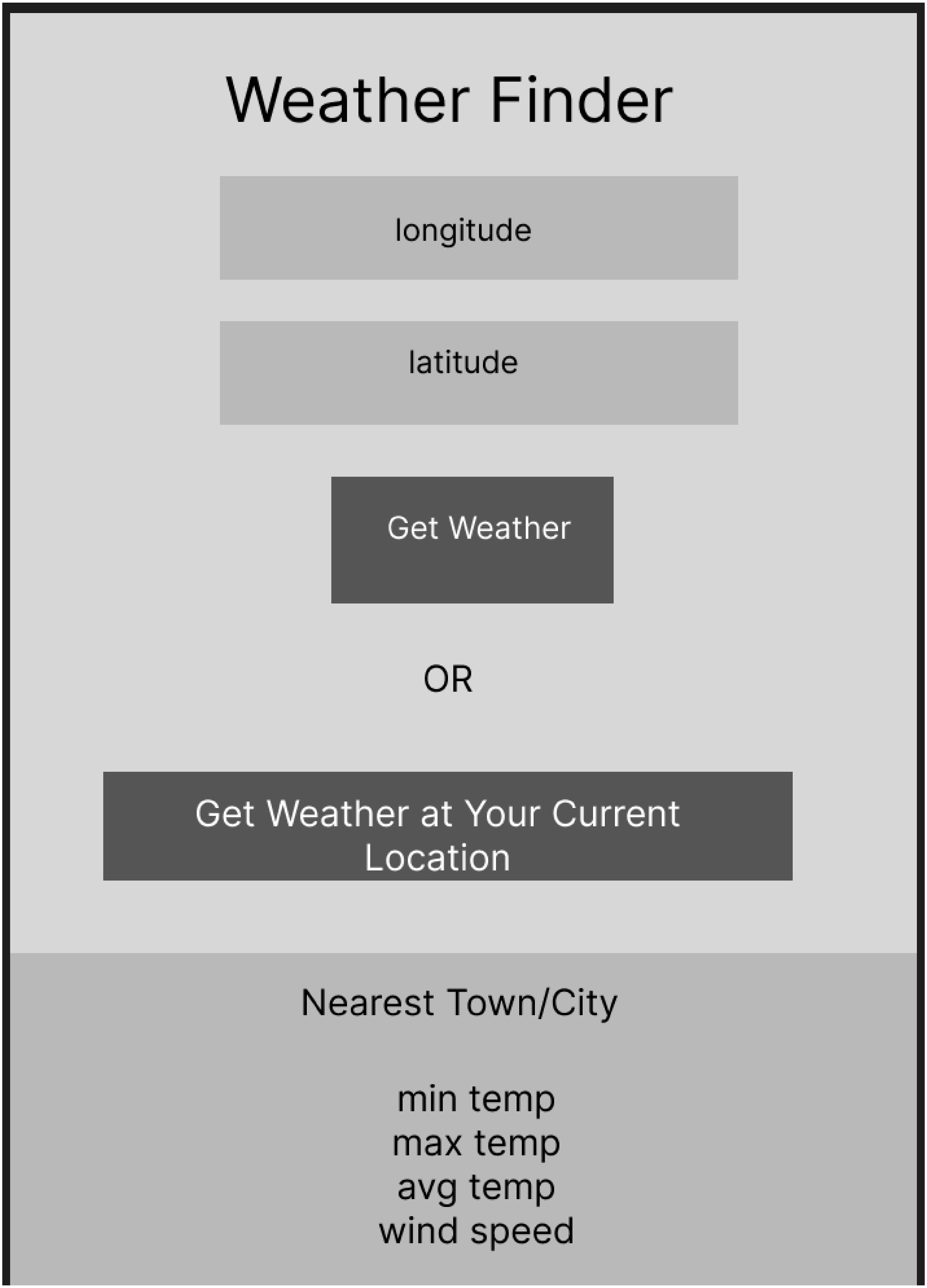
  gets called when the getWeather button is clicked. It calls the functions to find the nearest town/city and weather info and display it

- **alert**

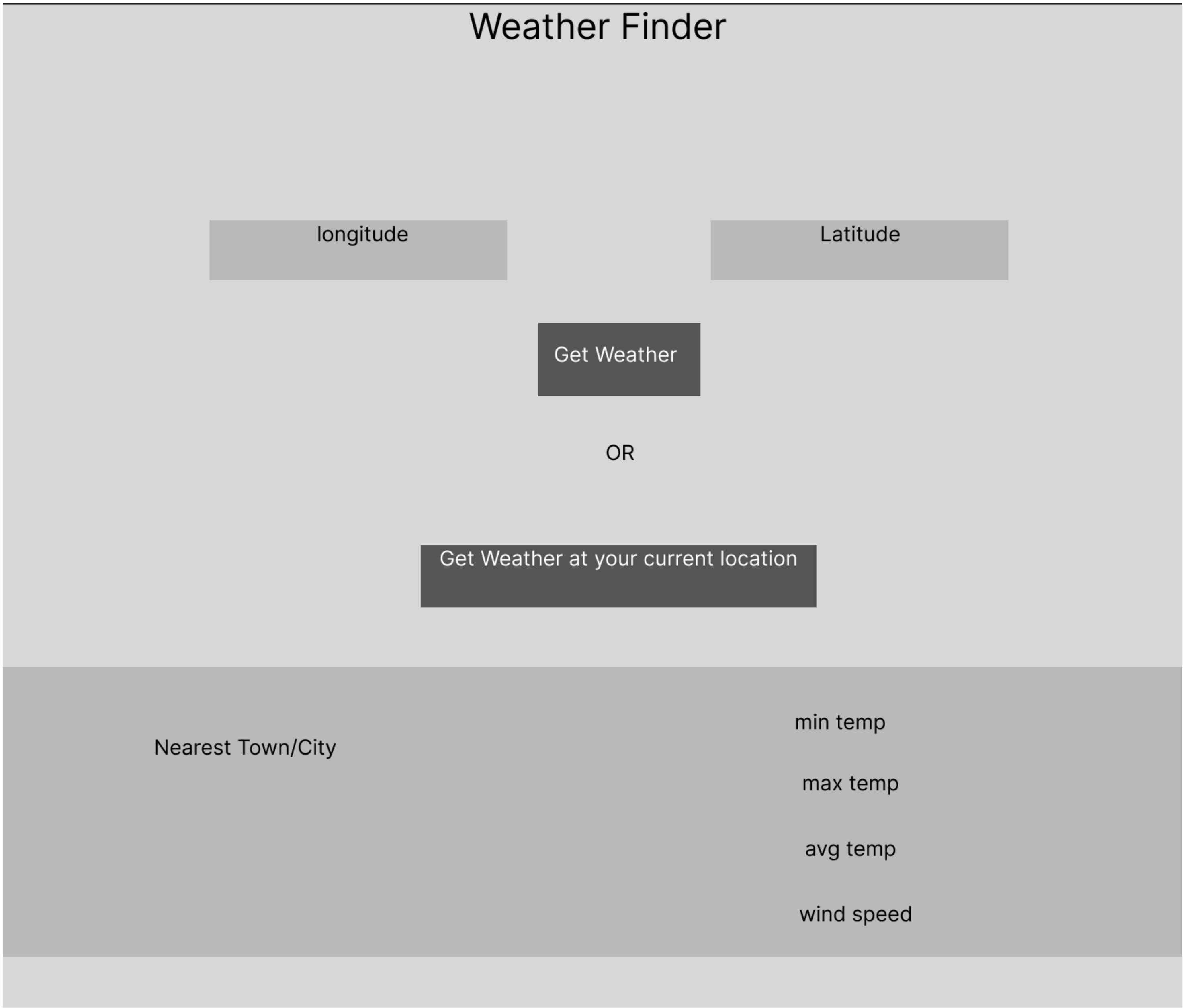
creates an alert at the top of the screen. It will mostly be used to let the user know if there was an error with getting the API data.

Wireframes:

Small Wireframe


Weather Finder
longitude
latitude
Get Weather
OR
Get Weather at Your Current Location
Nearest Town/City
min temp
max temp
avg temp
wind speed

Large Wireframe



External Data:

- [Forcast API]()

  ```
  {
      maxTemp_c,
      minTemp_c,
      avgTemp_c,
      maxTemp_f,
      minTemp_f,
      avgTemp_f,
      maxWind_mph,
      maxWind_kph
  }
  ```

- **IP Geolocation API**

  ```
  {
      city: {
          name
      },
      country: {
          name,
          iso_code
      },
      continent: {
          name
      },
      state: {
          name
      },
      location: {
          longitude,
          latitude
      }
  }
  ```

- **Places API**

  ```
  {
      name,
      country,
      state,
      city
  }
  ```

Module List

- **index.js**

  handles the javascript related to the home page

- **locationAPI.js**

  handles data retrieval from the geoapify API

- **weatherAPI.js**

  handles data retrieval from the weather API

Graphic Identity:

Colors

- backgrounds

  ⬜

- headers

⬛

- backgrounds and accents

  🟦

Fonts

Open Sans

Timeline & Project Planning:

The trello board has 1 list for each week. I will hopefully be finished before week 7. I'm planning to do more than half of the work in week 5, so I don't have to worry about it as much in week 6.

Trello Board

Challenges:

- Making sure that I follow my schedule and don't procrastinate
- Formatting the address that the user inputs in a way that the API can read it