Weather-app

This weather app is designed by Somdev Behera. It is a mern based project where the backend is developed in node.js and the frontend is developed in react.js. It uses OpenWeatherMap to get the weather data and leaflet.js to visualize it on the map. This project is a part of the internship task given by Settyl.

The backend

The backend is developed using node.js and express.js. The names of 30 cities were stored inside an array. Using promises the data for each city is fetched and stored in another array.

```
let requests = cities.map((city) => {
  //map function is used to iterate over the array of cities
  return new Promise((resolve, reject) => {
    //promise is used to handle asynchronous operations
    request(
      {
        uri: `http://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${api_key}`, //api call
        method: "GET" //method is GET
      },
      (err, res, body) => {
        if (err) {
          reject(err); //rejecting the promise if there is an error
        }
        resolve(body); //resolving the promise if there is no error
      }
    );
  });
});
```

The backend also uses the pagination technique to send paginated data to the front end.

```
app.get("/_offset=:offset&_limit=:limit", (req, res) => {
  const offset = parseInt(req.params.offset); //offset and limit are the parameters
  const limit = parseInt(req.params.limit);
  const final = offset + limit;
  Promise.all(requests) //promise.all is used to handle multiple promises
    .then((body) => {
      body.forEach((res) => {
        if (res) {
          WeatherData.push(JSON.parse(res)); //pushing the data into the array
        }
      });
      res.send(WeatherData.slice(offset, final)); //sending the data to the client
    })
    .catch((err) => {
```

```
      console.log(err);
    });
});
```

The backend is designed in a RESTful way. When the API is called, the get method toggles the promise and data is fetched from the OpenWeatherMap API and stored inside the weatherData array. Later five of the array elements are sent to the front end.

The front-end

The front-end is designed using react.js. Leaflet js is used to get the map.The ActualMap component calls the backend API to fetch the weather data which is stored inside the weatherData state variable.

```
const ActualMap = () => {
  const [weatherData, setWeatherData] = useState([]); //to store api data
  const [page, setPage] = useState(1); // to store current page
  const [loading, setLoading] = useState(false); //whether in loading or not
  const fetchData = async (offset, limit) => {
    setLoading(true); //setting loading to true
    const url = `http://localhost:9000/_offset=${offset}&_limit=${limit}`;
    try {
      await axios.get(url).then((res) => {
        setWeatherData(res.data);
        setLoading(false); //setting loading to false
      });
    } catch (error) {
      console.log(error);
    }
  }; //function to fetch data
```

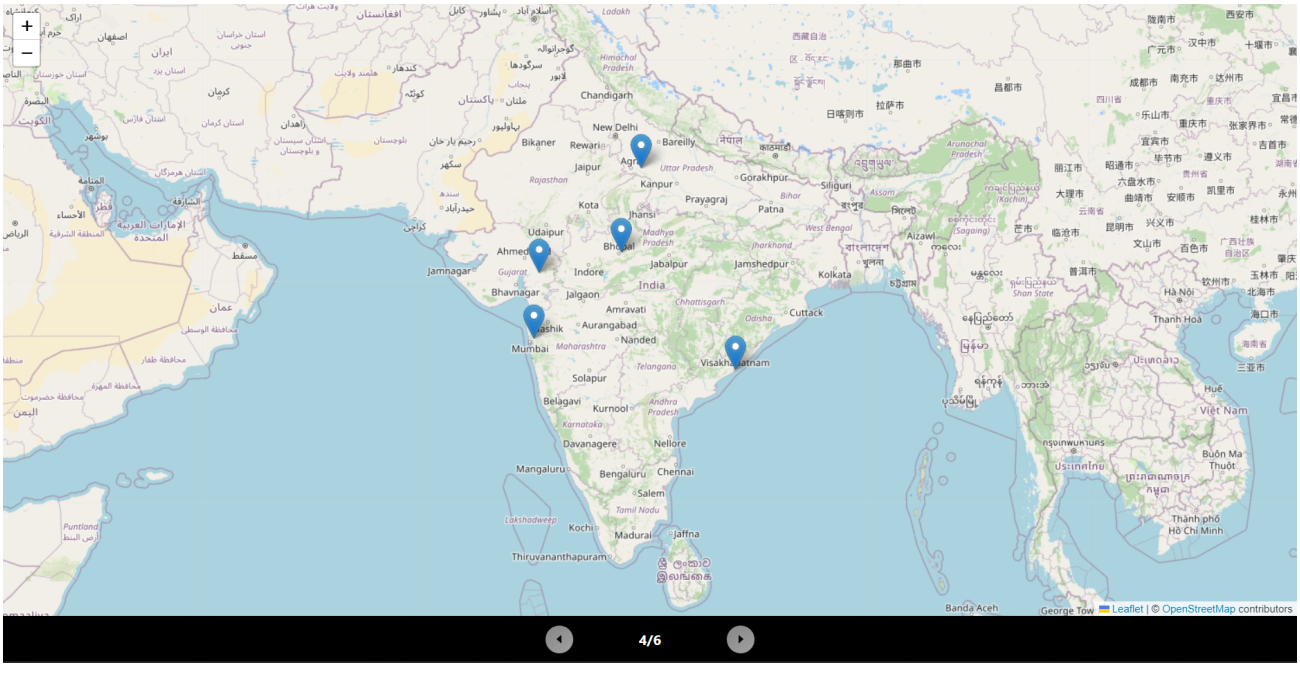
Variable page is used to keep track of the current page and loading is used to check the API fetch status.

Axios is used to fetch data from the backend. The fetchData function is called by useEffect hook which calls with respect to the page variable. The setTimeout function is used to refresh the data every 10 minutes.

```
useEffect(() => {
```

```
  //using useeffect ot call fetchData method
  const offset = (page - 1) * 5;
  fetchData(offset, 5);
  setTimeout(() => fetchData, 10000); //to refresh data
}, [page]);
```

Then the map and markers are developed with weather data fed into them. The site also features a bottom taskbar through which page navigation can be controlled.



Overall working on this assignment was a very good experience. Looking forward to working onboard with the Settyl team.