Noise Pollution Monitoring

Objectives:

Continue building the project by developing the real-time transit information platform. Use web development technologies to create a platform that displays real-time transit information. Monitor current noise levels in various locations. Design the platform to display real-time location from IoT sensors

Introduction:

In the age of data-driven decision-making, real-time monitoring of noise levels has become increasingly important for a variety of applications, from ensuring workplace safety to managing urban noise pollution. To address this need, we have developed a cutting-edge platform that offers real-time noise level data display. This platform empowers users with instant access to accurate and up-to-the-minute information on noise levels in their environments. Whether it's for personal comfort, environmental assessment, or compliance monitoring, our platform provides a user-friendly and informative solution for managing noise in real time.

HTML Coding:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="refresh" content="20" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Noise Pollution Monitoring</title>
    <link rel="stylesheet" href="css/bs.css">
    <link rel="stylesheet" href="css/font-awesome.min.css">

</head>
<body class="bg-dark ">
    <style>
        *{
            color: white;
        }
        th{
            text-align: center;
        }
        td{
            text-align: center;
        }
        a{
            text-decoration: none;
```

Creating a real-time noise level display in HTML involves integrating the necessary components like JavaScript for data collection and updating, as well as HTML

CSS Coding:

```css
@charset "UTF-8";
/*!
 * Bootstrap v5.0.2 (https://getbootstrap.com/)
 * Copyright 2011-2021 The Bootstrap Authors
 * Copyright 2011-2021 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE)
 */
:root {
  --bs-blue: #0d6efd;
  --bs-indigo: #6610f2;
  --bs-purple: #6f42c1;
  --bs-pink: #d63384;
  --bs-red: #dc3545;
  --bs-orange: #fd7e14;
  --bs-yellow: #ffc107;
  --bs-green: #198754;
  --bs-teal: #20c997;
  --bs-cyan: #0dcaf0;
  --bs-white: #fff;
  --bs-gray: #6c757d;
  --bs-gray-dark: #343a40;
  --bs-primary: #0d6efd;
  --bs-secondary: #6c757d;
  --bs-success: #198754;
  --bs-info: #0dcaf0;
  --bs-warning: #ffc107;
  --bs-danger: #dc3545;
  --bs-light: #f8f9fa;
  --bs-dark: #212529;
  --bs-font-sans-serif: system-ui, -apple-system, "Segoe UI", Roboto, "Helvetica Neue", Arial, "Noto Sans", "Liberation Sans", sans-serif, "Apple Color Emoji",
```

You can add some CSS styles to make the noise level display visually appealing

JAVASCRIPT Coding:

```javascript
var database = firebase.database();

//To send data to firebase
/*
database.ref('data').set({
  name: 'Raja',
  age: 20
});
*/


    database.ref('noise_pollution_monitoring').once('value', function(snapshot) {
        var firebaseData = snapshot.val();
        const lastUpdated = {};

    // Iterate through the Firebase data
    for (const key in firebaseData) {
        const item = firebaseData[key];
        const latLon = `${item.lat},${item.lon}`;

        // Check if we haven't seen this latLon pair before or the current item's datetime is greater
        if (!lastUpdated[latLon] || item.datetime > lastUpdated[latLon].datetime) {
            lastUpdated[latLon] = item;
        }
    }


    firebaseData = lastUpdated;
```

To display real-time noise data, you'll need a source for this data. It could be an external API or a local sensor. we use a simple JavaScript function to simulate changing noise levels at regular intervals.

Firebase Coding:

```javascript
const firebaseConfig = {
    apiKey: "AIzaSyDkN4wcQjdhWiX64pof-UcOSHCr3j1v8D4",
    authDomain: "noise-pollution-monitori-7a445.firebaseapp.com",
    databaseURL: "https://noise-pollution-monitori-7a445-default-rtdb.firebaseio.com",
    projectId: "noise-pollution-monitori-7a445",
    storageBucket: "noise-pollution-monitori-7a445.appspot.com",
    messagingSenderId: "554067503555",
    appId: "1:554067503555:web:d44e6bf5b91ce46c5ef576",
  };


firebase.initializeApp(firebaseConfig);
```

To create a real-time noise level monitoring system using Firebase, you'll need to use Firebase's real-time database feature and integrate it with a front-end web application.

Mobile Applications:

To create mobile application to display real-time noise level data involves several steps including data acquisition, user interface design, data processing, and presentation.

Key features:

Data acquisition:

To see real-time noise level data, you will need to access the noise level measurements from the sensor. This can be achieved by various methods, such as using the device's built-in microphone or connecting to an external noise level sensor. If you use your device's microphone, you'll need to perform audio capture and signal processing to convert audio into noise level data.

User Interface Design:

If you're using an external sensor, you'll need to integrate it with your mobile device. This typically involves using communication protocols such as Bluetooth or USB to collect sensor data and then interpreting that data in your application.

Data processing:

Once you have acquired the raw noise level data, you may need to process and filter it to obtain meaningful measurements like averaging, peak detection, or noise classification to understand and categorize different noise levels.

Real Time updates:

Users receive real time updates on noise pollution levels

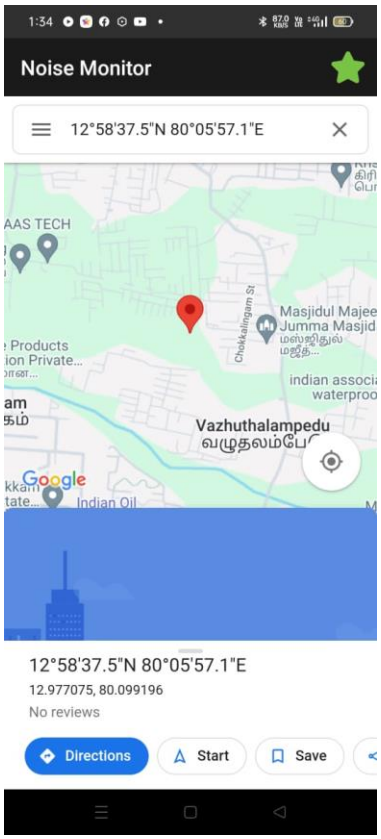
Usage of mobile applications:

- Install app and open the app
- Receive real time noise level updates
- Explore sensor locations on Google Maps for more information.

Noise Pollution Creation:

- To initialize the screen

- Output in user mobile phones



Conclusions:

The IoT Noise Pollution Monitoring System's web platform and Android app provide an accessible and user-friendly way to stay informed about real-time noise pollution data in various locations. These tools can be instrumental in addressing noise pollution issues, influencing urban planning, and promoting community involvement