PUBLIC TRANSPORT OPTIMIZATION

Phase 4 Project Submission

Project Title: Public Transport Optimization using IOT

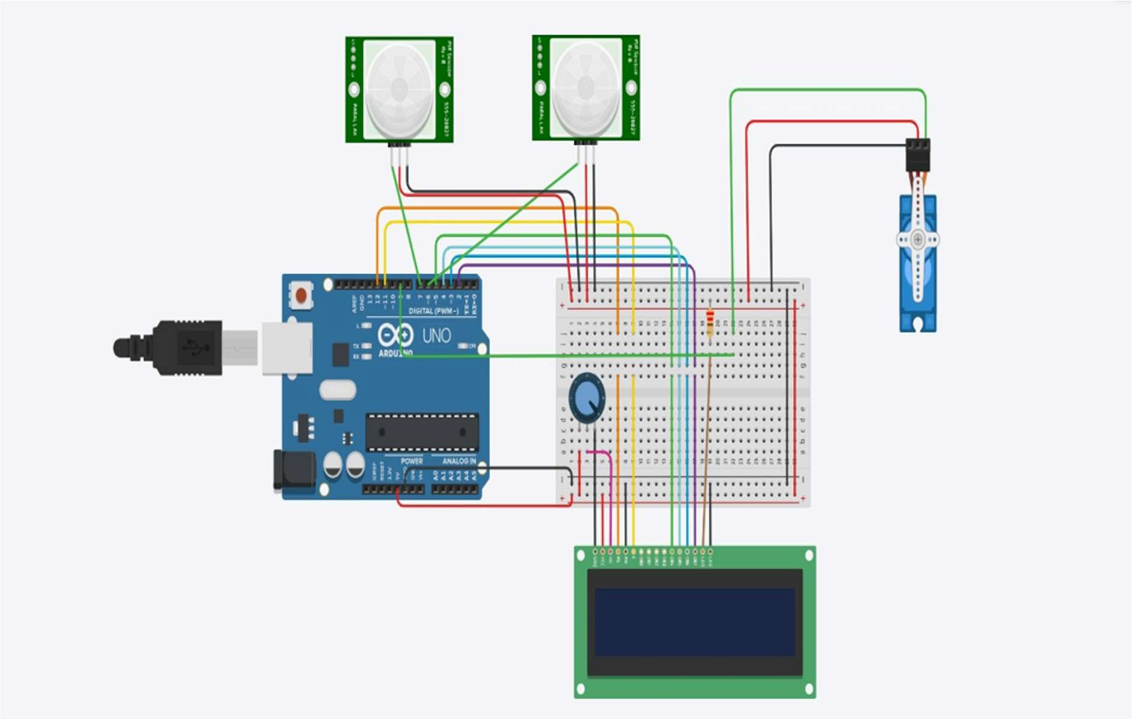
INTRODUCTION :

- ❖ The "Public Transport Optimization" project is a comprehensive initiative to optimize and improve public transportation systems within urban areas.
- ❖ With the growing challenges of traffic congestion, pollution, and the need for more sustainable and efficient transportation.
- ❖ This project focuses on enhancing resident's and commuters' overall public transit experience.

Objective:

- ❖ The primary objective of this project is to optimize public transport systems to make them more reliable, accessible, and sustainable.
- ❖ This entails reducing travel times, increasing the affordability and convenience of public transit, and minimizing the environmental impact of transportation in urban areas.

Circuit Diagram :



PYTHON CODE FOR CONNECTING MOBILE APP WITH ABOVE PROJECT :

```
import 'package:flutter/material.dart';

import 'package:http/http.dart' as http;

import 'dart:convert';

void main() => runApp(MyApp());


class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: VehicleLocations(),
    );
  }
}
class VehicleLocations extends StatefulWidget {
  @override
  _VehicleLocationsState createState() => _VehicleLocationsState();
}
class _VehicleLocationsState extends State<VehicleLocations> {
  String locationData = "";
  Future<void> fetchVehicleLocations() async {
    final response = await http.get('http://your-python-server-url/get_vehicle_location?vehicle_id=bus1');
    if (response.statusCode == 200) {
      setState(() {
        locationData = json.decode(response.body).toString();
      });
    }
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Public Transport Optimization App'),
      ),
      body: Center(
        child: Column(
          children: <Widget>[
            ElevatedButton(
              onPressed: fetchVehicleLocations,
              child: Text('Get Vehicle Location'),
            ),
Text(locationData),
     ],
    ),
   ),
  );
 }
}
```

1. Imports:

   - import 'package: flutter/material.dart';: Imports the Flutter framework for building the UI.
   - import 'package:http/http.dart' as http;: Imports the HTTP package to make HTTP requests.
   - 'dart: convert';: Imports the dart: convert library for JSON decoding.

2. Main Function and App Initialization:

   - void main() => runApp(MyApp());: Entry point of the Flutter app, where the MyApp widget is initialized and run

3. MyApp Class:

   - MyApp is a stateless widget representing the root of the application.
   - build method sets up the basic structure of the app. It creates a MaterialApp widget with a home screen set to VehicleLocations.

4. VehicleLocations Class:

- VehicleLocations is a stateful widget responsible for handling the UI and user interactions related to vehicle locations.
- fetchVehicleLocations is an asynchronous function that makes an HTTP GET request to a specific URL (http://your-python-server-url/get_vehicle_location?vehicle_id=bus1).
- If the response status code is 200 (OK), the returned JSON data is decoded and stored in the locationData variable, triggering a UI update.