PUBLIC TRANSPORTATION OPTIMIZATION USING IOT

Project Objectives:

Real-time Transit Information: Developing a real-time transit information system that provides passengers with up-to-date information about public transportation, including bus and train arrivals, delays, and route information.

IoT Sensor Deployment: Installed a network of IoT sensors at key transit hubs and on public transportation vehicles to collect real-time data on vehicle locations, passenger loads, and environmental conditions.

Transit Information Platform: Building a robust and scalable cloud-based platform to aggregate and process data from IoT sensors, generate real-time transit information, and provide APIs for passenger-facing applications.

Code Implementation: Developing software applications, including mobile apps, web interfaces, and data visualization tools, to present real-time transit information to passengers and transportation authorities.

IoT Sensor Deployment:

The IoT sensor deployment involves equipping buses, trams, and train cars with sensors to collect data. Sensors may include GPS modules, accelerometers, environmental sensors (temperature, humidity, air quality), passenger count sensors (e.g., cameras, weight sensors), and vehicle health sensors. These sensors transmit data to a centralized server over a wireless network (e.g., cellular, Wi-Fi) for real-time monitoring and analysis.

IoT Sensor Deployment

Platform Development:

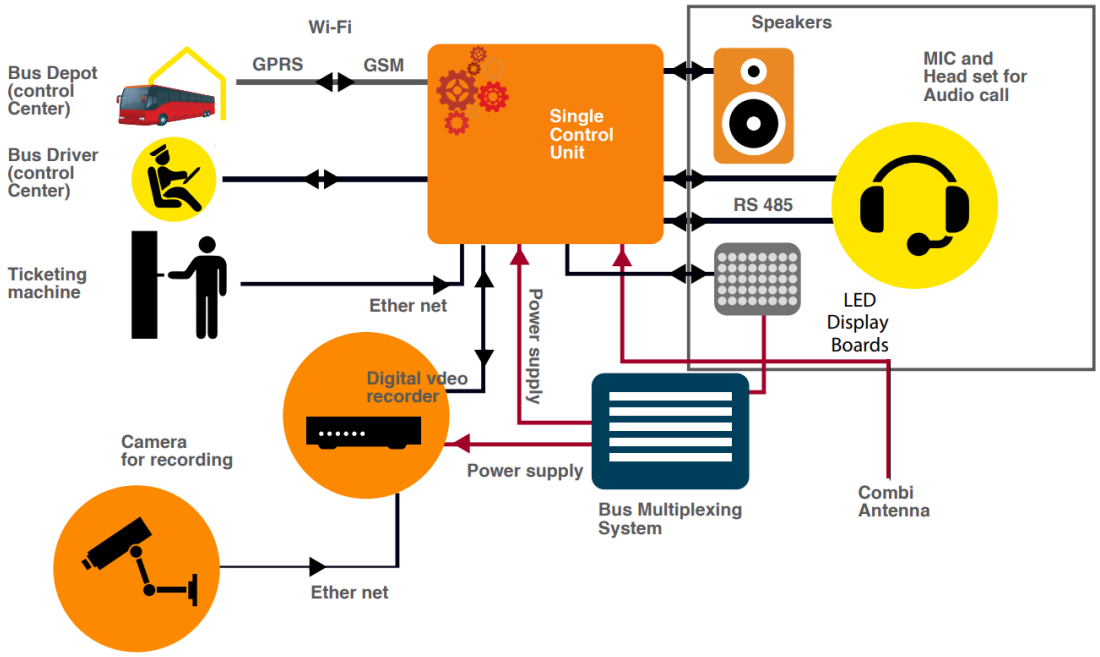
The platform development consists of a cloud-based system that processes the data collected by IoT sensors, performs real-time data analysis, and provides APIs for the frontend applications.

Data Ingestion: IoT sensors send data to a central cloud server.
Data Processing: Raw sensor data is processed, cleaned, and structured for analysis.

Real-time Analytics: Advanced algorithms are applied to the data for route optimization, delay prediction, and passenger load balancing.

APIs: The platform exposes APIs for passenger-facing applications.

Platform Architecture



Code Implementation:

The code implementation involves developing various applications and interfaces.

Passenger Mobile App: Provides real-time transit information, route planning, and alerts. It can show estimated arrival times and delays.

Code

Dart

```
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: BusTrackingApp(),
    );
  }
}

class BusTrackingApp extends StatefulWidget {
  @override
  _BusTrackingAppState createState() => _BusTrackingAppState();
}

class _BusTrackingAppState extends State<BusTrackingApp> {
  List<Bus> buses = [];

  Future<void> fetchBusData() async {
    final response = await http.get(Uri.parse('YOUR_API_ENDPOINT_HERE'));

    if (response.statusCode == 200) {
      final data = json.decode(response.body);
      setState(() {
        buses = List<Bus>.from(data.map((bus) => Bus.fromJson(bus)));
      });
    } else {
      // Handle API request error
      print('Failed to fetch bus data.');
    }
  }

  @override
  void initState() {
    super.initState();
    fetchBusData();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
```

```
      appBar: AppBar(
        title: Text('Bus Tracking App'),
      ),
      body: ListView.builder(
        itemCount: buses.length,
        itemBuilder: (context, index) {
          return ListTile(
            title: Text('Bus ID: ${buses[index].id}'),
            subtitle: Text('Latitude: ${buses[index].latitude}, Longitude: ${buses[index].longitude}'),
          );
        },
      ),
    );
  }
}

class Bus {
  final String id;
  final double latitude;
  final double longitude;

  Bus({
    required this.id,
    required this.latitude,
    required this.longitude,
  });

  factory Bus.fromJson(Map<String, dynamic> json) {
    return Bus(
      id: json['id'],
      latitude: json['latitude'],
      longitude: json['longitude'],
    );
  }
}
```

In this Code, the Flutter app fetches real-time bus location data from an API and displays it in a simple list view.

Web Interface: A user-friendly web portal for desktop users to access transit information.

Visualization Tools: Dashboards for transportation authorities to monitor the system's performance and transit data in real time.

Code

Index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Public Transport Optimization Web Interface</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <h1>Real-time Bus Tracking</h1>
    <div id="map"></div>
    <script src="script.js"></script>
</body>
</html>
```

Style.css

```
body {
    font-family: Arial, sans-serif;
    text-align: center;
}

#map {
    width: 100%;
    height: 500px;
}
```

Script.js

```
const map = L.map('map').setView([YOUR_INITIAL_LATITUDE, YOUR_INITIAL_LONGITUDE], 13);
// Set initial coordinates and zoom level
const accessToken = 'YOUR_MAPBOX_ACCESS_TOKEN'; // Replace with your Mapbox access token

// Create a Mapbox map layer
L.tileLayer(`https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token=${accessToken}`, {
    maxZoom: 18,
    id: 'mapbox/streets-v11', // You can change the map style
```

```
    tileSize: 512,
    zoomOffset: -1
}).addTo(map);

// Function to update bus locations
function updateBusLocations() {
    // Fetch real-time bus data from your API
    fetch('YOUR_API_ENDPOINT_HERE')
        .then(response => response.json())
        .then(data => {
            // Clear existing markers
            if (busMarkers) {
                busMarkers.clearLayers();
            }

            // Create new markers for each bus
            for (const bus of data) {
                const marker = L.marker([bus.latitude, bus.longitude])
                    .bindPopup(`Bus ID: ${bus.id}`)
                    .addTo(busMarkers);
            }

            // Add bus markers to the map
            busMarkers.addTo(map);
        })
        .catch(error => {
            console.error('Failed to fetch bus data: ', error);
        });
}

// Initialize a layer group for bus markers
const busMarkers = L.layerGroup().addTo(map);

// Set an interval to update bus locations every 10 seconds (adjust as needed)
setInterval(updateBusLocations, 10000);

// Initial bus location update
```

updateBusLocations();

1. Route Planning:

For route planning, used algorithms like Dijkstra's algorithm, A* search. Here's route planning function using Python:

python
Copy code

```
def plan_route(start_location, end_location):
    # Implement your route planning algorithm here
    # Return a list of waypoints for the optimal route
    pass
```

2. Passenger Information:

A mobile app or web interface can provide real-time information to passengers. Here's an example of a function to retrieve bus schedule information:

python
Copy code

```
def get_bus_schedule(bus_id):
    # Implement a function to fetch bus schedule from a database
    pass
```

3. Environmental Monitoring:

we used IoT sensors to monitor environmental conditions and create alerts. Here's an environmental monitoring function:

python
Copy code

```
def monitor_environment(sensor_data):
    # Implement code to process and analyze sensor data
    # Generate alerts for adverse conditions
    pass
```

4. User Authentication:

implemented user authentication to secure our system. Depending on our technology stack, used libraries like Firebase Authentication, Passport.js, or Django authentication for web apps. Here's user authentication using Python and Flask:

python

```
from flask import Flask, request, session, redirect, url_for, flash, render_template

app = Flask(__name)
app.secret_key = 'your_secret_key'

@app.route('/login', methods=['POST'])
def login():
    # Implement user authentication logic here
    pass

@app.route('/logout')
def logout():
    session.pop('user_id', None)
    return redirect(url_for('index'))
```

5.User Experience (UX) design is a process that involves creating design guidelines, user interfaces, and a user-friendly experience.

Index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Public Transport Optimization</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <header>
        <h1>Welcome to Public Transport Optimization</h1>
    </header>

    <main>
```

```
        <section id="route-planner">
            <h2>Plan Your Trip</h2>
            <form id="trip-planner-form">
                <label for="start-location">Start Location:</label>
                <input type="text" id="start-location" placeholder="Enter your start location" required>

                <label for="end-location">End Location:</label>
                <input type="text" id="end-location" placeholder="Enter your end location" required>

                <button id="plan-route-button">Plan Route</button>
            </form>
        </section>

        <section id="real-time-info">
            <h2>Real-time Information</h2>
            <div id="real-time-updates"></div>
        </section>
    </main>

    <footer>
        <p>&copy; 2023 Public Transport Optimization</p>
    </footer>

    <script src="script.js"></script>
</body>
</html>
```

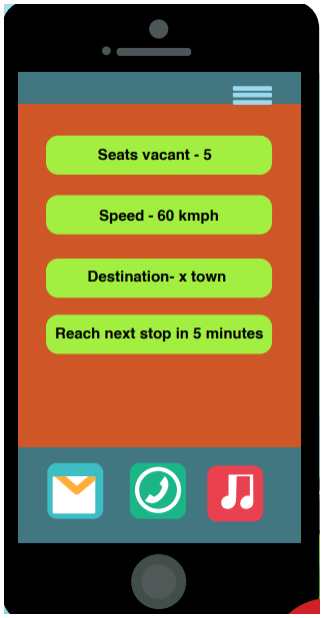
Styles.css

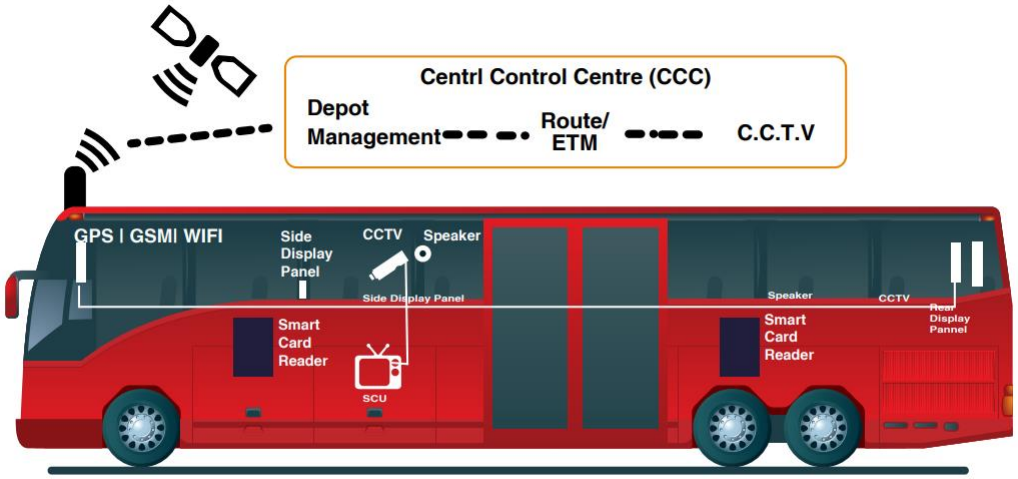
```
body {
    font-family: Arial, sans-serif;
    text-align: center;
    background-color: #f4f4f4;
}

header {
    background-color: #007BFF;
    color: #fff;
    padding: 20px;
```

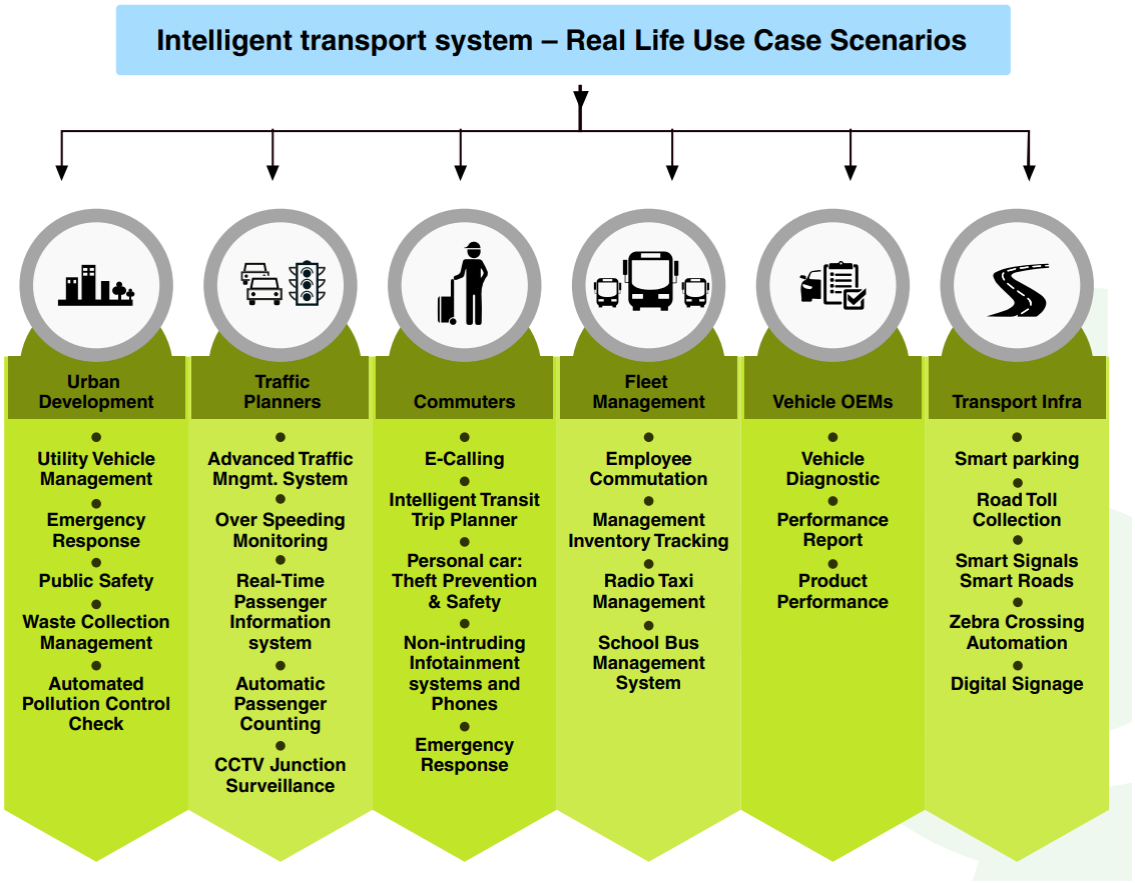
```
}

main {
    padding: 20px;
}

#route-planner {
    background-color: #fff;
    padding: 20px;
    margin-bottom: 20px;
}

#trip-planner-form {
    display: flex;
    flex-direction: column;
    align-items: center;
}

label {
    font-weight: bold;
    margin: 10px 0;
}

input {
    width: 100%;
    padding: 10px;
    margin: 5px 0;
}

button {
    background-color: #007BFF;
    color: #fff;
    padding: 10px 20px;
    border: none;
    cursor: pointer;
}
```

```
#real-time-info {
    background-color: #fff;
    padding: 20px;
}

footer {
    background-color: #007BFF;
    color: #fff;
    padding: 10px;
}
```

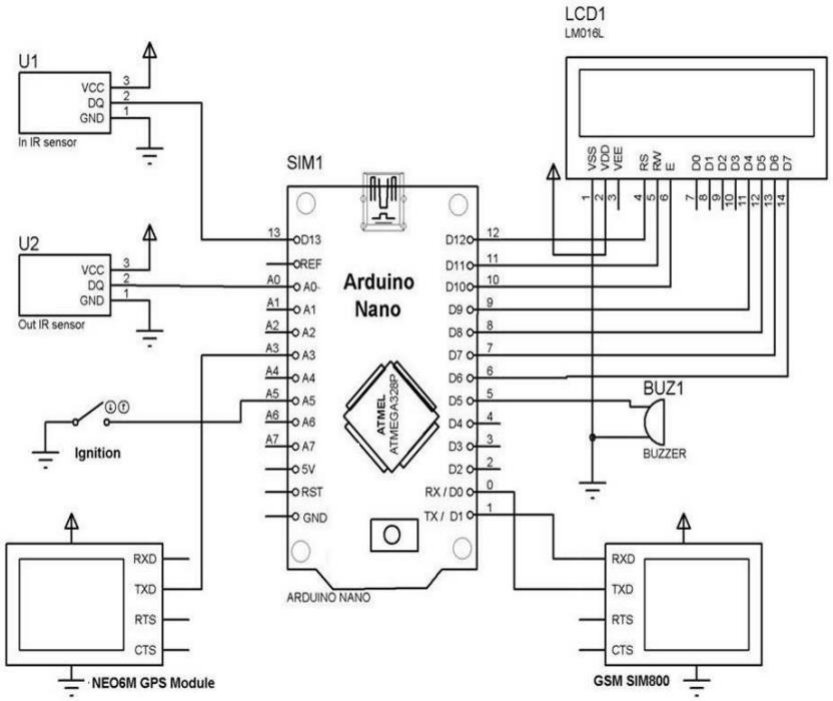
Passanger Mobile app



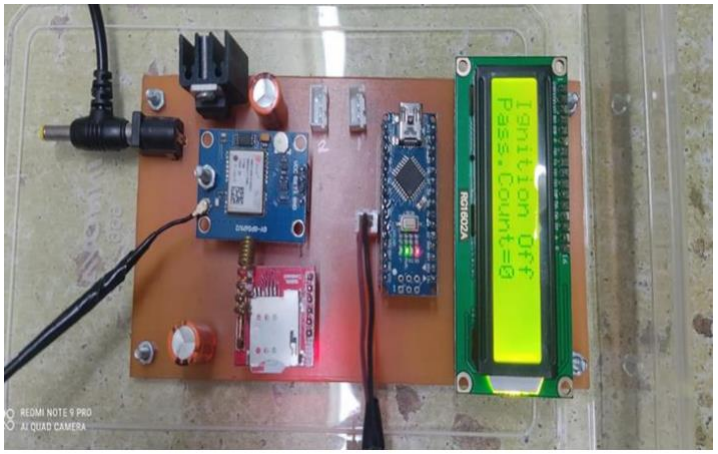




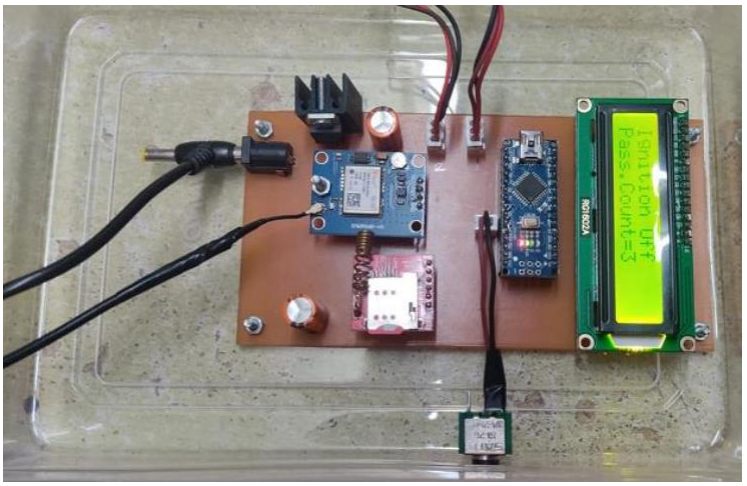
Block design



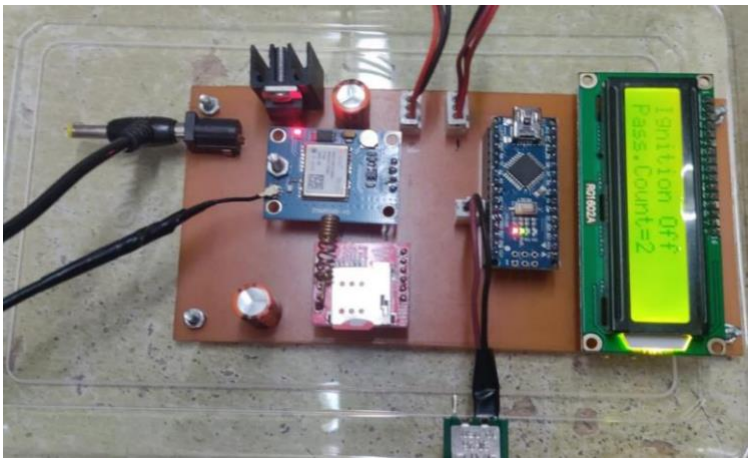
Output



This shows the output of our project. and shows count of passengers which is increasing one by one.as well as the status of ignition.



This shows the output of our project. and shows count of passengers which is increasing one by one.as well as the status of ignition.it is very useful for students and job employee to get exact location and time of bus. And shows the real time location of bus.



This system is very secured and smart assisted transport. It’s more secure, smart and advanced. The system is smart and advanced as it has various features GPS tracking, IoT acknowledgement.The objectives of this are, to design a system that will get the position of bus using GPS module, count the number of passengers in the bus, count the number of passengers in the bus, count the number of passengers in the bus. To design a system that will update all the data of the bus to the web page.

Benefits of the Real-time Transit Information System:

The real-time transit information system offers several benefits for public transportation:

Improved Passenger Experience: Passengers have access to accurate, real-time information, reducing uncertainty and wait times. They can plan their routes and time more efficiently.

Increased Ridership: Providing real-time information can attract more passengers to public transportation, as it enhances convenience and reliability.

Operational Efficiency: Transportation authorities can optimize routes, schedules, and resources based on real-time data, reducing costs and improving service quality.

Environmental Benefits: By optimizing routes and encouraging more ridership, the system can contribute to reducing traffic congestion and greenhouse gas emissions.

Data-Driven Decision Making: Transportation authorities can make informed decisions and respond proactively to disruptions, reducing the impact of delays and incidents on passengers.

In conclusion, a real-time transit information system can significantly improve public transportation services and passenger experience. By deploying IoT sensors, developing a comprehensive platform, and implementing user-friendly applications, the system benefits both passengers and transportation authorities, leading to more efficient and sustainable public transportation.