PUBLIC TRANSPORTATION OPTIMIZATION USING IOT

Objectives :

The key objectives of a public transportation optimization project using IoT include improving efficiency, reducing operational costs, enhancing passenger experience with real-time information, minimizing environmental impact, and promoting safety through data-driven decision-making.

Iot setup devices :

1.Sensors
2. A microcontroller (such as Arduino or Raspberry pi)
3.Communication module (WIFI or GSM)

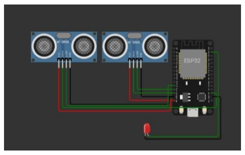
Hardware Specifications:

1.Real-time data collection from GPS trackers
2.Arduino or Raspberry pi
3.ESP32 module
4.OLED Display
5.Power supply
6.Passenger Counting Sensors
7.Cameras
8.Maintenance Sensors
9.Passenger Information Displays
10.Data storage

Software specifications:

1.Embedded Software for Microcontroller: Programs the microcontroller for data collection and transmission.
2.Server-side Software: Manages and analyzes the received data.
3.Database: Stores historical data for analysis.
4.Web/App Interface: Allows users to access and visualize the data.
5.Data Analytics Tools: Process and interpret transport data for meaningful insights.

Circuit Diagram:



The working of a Public Transportation Optimization system using IoT involves the integration of various technologies and components to enhance the efficiency, safety, and user experience of public transportation services. Here's an overview of how such a system typically operates:

1.IoT Devices on Buses :

Buses are equipped with IoT devices, which can include GPS trackers, sensors, and communication modules. These devices continuously collect data during bus operations.

2.Data Collection :

IoT devices on buses collect real-time data, including GPS coordinates, speed, passenger counts, fuel consumption, and environmental conditions (e.g., temperature and humidity).

3.Data Transmission :

The collected data is transmitted to a central server or cloud platform using wireless communication protocols, such as cellular networks or Wi-Fi.

4.Data Processing and Storage :

The central server processes the incoming data, which may involve filtering, cleaning, and aggregating the information. Processed data is stored in a database for further analysis.

5.Real-time Bus Tracking :

Passengers and operators can access real-time bus tracking information through web or mobile applications. They can see the live locations of buses on a map, estimated arrival times, and any delays.

6.Route Optimization :

The system uses real-time and historical data to optimize bus routes. Algorithms consider factors like traffic conditions, passenger demand, and bus capacity to adjust routes and schedules.

7.Scheduling :

Bus schedules are dynamically adjusted to optimize service frequency during peak hours and reduce waiting times for passengers.

8.Passenger Information :

Passengers receive real-time information about bus locations, arrival times, and any service disruptions through the user interface. This helps them plan their journeys more efficiently.

9.Safety and Security :

The system may include safety features such as emergency notifications in case of accidents or breakdowns. Security measures protect data and ensure passenger safety.

10.Predictive Maintenance :

By analyzing sensor data from the buses, the system can predict maintenance needs and schedule servicing to prevent breakdowns, reducing downtime and improving reliability.

11.User Interfaces :

Passengers and operators interact with the system through user-friendly interfaces on web and mobile applications. They can access route planning, ticket purchasing, and receive alerts.

12.Reporting and Analytics :

Operators can generate reports and access analytics to make data-driven decisions for service improvement, cost reduction, and resource allocation.

13.Integration with Local Authorities :

The system collaborates with local transportation authorities to ensure compliance with regulations and share data for planning and coordination.

14. Notifications :

The system sends notifications to passengers about service changes, delays, and other relevant information through the user interfaces.

15.Scalability :

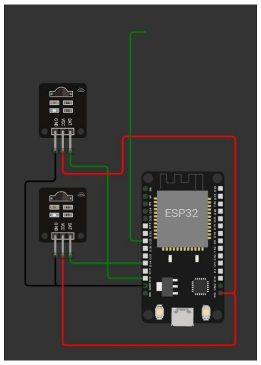
The system can accommodate an increasing number of buses and users as the public transport network expands.

16.Maintenance and Updates :

Ongoing maintenance, software updates, and bug fixes are performed to keep the system running smoothly and up-to-date.

17.Regulatory Compliance :

The system complies with relevant regulations, data privacy laws, and safety standards governing public transportation.



Source code :

HTML (index.html):

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

2.CSS (style.css)

```css
#map {
width: 80%;
height: 400px;
margin: 20px auto;
}
```

3.JavaScript (script.js)

```javascript
// Simulated bus data (replace with actual IoT data)
const buses = [
{ id: 1, lat: 40.7128, lng: -74.0060 },
{ id: 2, lat: 40.7306, lng: -73.9352 },
// Add more bus data here
];
```

```
function initMap() {
const map = new google.maps.Map(document.getElementById("map"), {
zoom: 12,
center: { lat: 40.7128, lng: -74.0060 }, // Default center (New York City)
});
// Display markers for buses
buses.forEach((bus) => {
const marker = new google.maps.Marker({
position: { lat: bus.lat, lng: bus.lng },
map: map,
title: `Bus ${bus.id}`,
});
});
// Update bus positions every 10 seconds (simulate real-time updates)
setInterval(updateBusPositions, 10000);
}
function updateBusPositions() {
// Simulate updating bus positions with new data
buses.forEach((bus) => {
bus.lat += Math.random() * 0.01 - 0.005;
bus.lng += Math.random() * 0.01 - 0.005;
});
// Update marker positions on the map
const markers = Array.from(document.querySelectorAll("div[title^='Bus']"));
markers.forEach((marker, index) => {
marker.position = new google.maps.LatLng(buses[index].lat, buses[index].lng);
marker.title = `Bus ${buses[index].id}`;
marker.setMap(null);
marker.setMap(map);
});
}
```
In this example, we create a simple web page that displays a Google Map with bus markers.
The bus data is simulated, and the markers are updated every 10 seconds to simulate real-time
bus tracking.
For a complete public transport optimization system, you would need to integrate this frontend
with a backend that communicates with IoT devices on buses, processes data, and implements

optimization algorithms. You might also need a database for storing real-time bus data. Additionally, you would need user authentication and more advanced features for passengers
and operators.

Conclusion :

In conclusion, a Public Transportation Optimization project using IoT technologies aims to make public transport more efficient, reliable, and passenger-friendly. By collecting and analyzing real-time data, optimizing routes, and providing accurate information to passengers, this project can lead to a more sustainable and convenient public transportation system. It enhances user experience and helps optimize resources for transport authorities.