**Internet of Things**

**Phase-4 : Development Part 2**

**Project Title : Public Transportation Optimization**

**Sub Title : Continue building the project by developing the real-time transit information platform. Use web development technologies (e.g. , HTML, CSS, JavaScript) to create a platform that displays real-time transit information. Design the platform to receive and display real-time location, ridership, and arrival time data from IoT sensors word document.**

**Submitted by :**

1. **P.Karthikeyan**
2. **N.Udhayakumar**
3. **K.Mohamed Imran**
4. **A.Karuppasamy**
5. **G.Deva**

**Real-Time Transit Information Platform Development Plan**

**Introduction**

* This document outlines the development plan for a real-time transit information platform. The platform’s goal is to provide users with real-time location, ridership, and arrival time data from IoT sensors, improving the overall commuter experience

**Project Overview**

* The real-time transit information platform aims to enhance the accessibility and convenience of public transportation. By providing up-to-the-minute data, it empowers commuters to make informed decisions and reduces uncertainty in transit services.

**Project Scope**

* Display real-time transit information for various routes and transportation modes.
* Collect, process, and display real-time location, ridership, and arrival time data.
* Implement user authentication and authorization for a personalized experience.

**Technologies Used**

* **Front-End Development:** HTML, CSS, JavaScript
* **Back-End Development:** Node.js
* **Database:** MySQL

**Front-End Development**

* Create an intuitive user interface for displaying real-time transit information.
* Design responsive layouts for seamless access on various devices.
* Implement interactive features for a user-friendly experience.

**Back-End Development**

* Develop server-side components to process data received from IoT sensors.
* Set up and manage a MySQL database for real-time data storage.

**IoT Sensors**

* Ultrasonic Sensors
* Magnetic Sensors
* Infrared Sensors
* Camera
* Ground Loop Sensors
* Pavement Sensors
* Wireless Sensors
* Radar Sensors
* GPS Sensors
* Weather Sensors
* Traffic Density Sensors

**Integration with IoT Sensors**

* Establish secure connections with IoT sensors, utilizing protocols like MQTT and HTTP.
* Design and create APIs to receive real-time data from sensors.

**Real-Time Data Processing**

* Implement data processing logic to continuously update location, ridership, and arrival time data.
* Utilize Web Socket technology for real-time data transmission to the front-end.

**User Authentication and Authorization**

* Implement a robust user authentication system to manage access control for platform features and data.

**Testing**

* Conduct extensive testing to ensure the platform functions as expected.
* Perform stress testing to evaluate performance under high user and data loads.

**Deployment**

* Deploy the platform on a web server or cloud hosting service.
* Implement monitoring and error tracking systems to maintain system stability.

**User Interface Refinement**

* Continuously gather user feedback and make necessary improvements to the user interface.

**Documentation and Training**

* Create comprehensive documentation for users and administrators.
* Provide training for staff responsible for platform maintenance and support.

**Maintenance and Updates**

* Schedule regular updates to introduce new features and address any identified issues.
* Stay current with evolving IoT technologies and web standards to ensure ongoing compatibility.

**Scalability**

* Plan for scalability to accommodate a growing number of sensors and users.

**Program**

**HTML :**

<!DOCTYPE html>

<html lang=”en”>

<head>

<meta charset=”UTF-8”>

<meta name=”viewport” content=”width=device-width, initial-scale=1.0”>

<title>Real-Time Transit Information</title>

<link rel=”stylesheet” href=”styles.css”>

</head>

<body>

<header>

<h1>Real-Time Transit Information Platform</h1>

</header>

<nav>

<ul>

<li><a href=”#live-map”>Live Map</a></li>

<li><a href=”#sensor-data”>Sensor Data</a></li>

</ul>

</nav>

<section id=”live-map”>

<h2>Live Map</h2>

<div id=”map”></div>

</section>

<section id=”sensor-data”>

<h2>Sensor Data</h2>

<div id=”sensor-display”></div>

</section>

<script src=”script.js”></script>

</body>

</html>

**CSS :**

Body {

Font-family: Arial, sans-serif;

Margin: 0;

Padding: 0;

Background-color: #f0f0f0;

}

Header {

Background-color: #333;

Color: #fff;

Text-align: center;

Padding: 1rem 0;

}

Nav {

Background-color: #444;

Color: #fff;

Text-align: center;

Padding: 1rem 0;

}

Nav ul {

List-style: none;

Padding: 0;

}

Nav ul li {

Display: inline;

Margin-right: 20px;

}

Nav a {

Text-decoration: none;

Color: #fff;

Font-weight: bold;

}

**JavaScript :**

// Simulated sensor data (replace with real data retrieval logic)

Const sensorData = {

Ultrasonic: “45.2 meters”,

Magnetic: “73 units”,

Infrared: “120 units”,

Camera: “No data available”,

groundLoop: “2 loops detected”,

pavement: “Dry”,

wireless: “Strong signal”,

radar: “Moderate traffic”,

gps: “Latitude: 40.7128, Longitude: -74.0060”,

weather: “Temperature: 70°F, Sunny”,

trafficDensity: “Medium”

};

// Function to display sensor data on the page

Function displaySensorData() {

Const sensorDisplay = document.getElementById(“sensor-display”);

sensorDisplay.innerHTML = “<h3>Sensor Data</h3>”;

for (const sensor in sensorData) {

const sensorValue = sensorData[sensor];

sensorDisplay.innerHTML += `<p>${sensor}: ${sensorValue}</p>`;

}

}

// Call the function to display sensor data

displaySensorData();

**Program**

#define BLYNK\_TEMPLATE\_ID "TMPL26V4fGv5q"

#define BLYNK\_TEMPLATE\_NAME "Test"

#define BLYNK\_AUTH\_TOKEN "XEHxNF\_Ur1Nt2p7wB5B20dNI1ZUwj34P"

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

int duration1 = 0;

int distance1 = 0;

int duration2 = 0;

int distance2 = 0;

int dis1 = 0;

int dis2 = 0;

int dis\_new1 = 0;

int dis\_new2 = 0;

int entered = 0;

int left = 0;

int inside = 0;

#define LED 2

#define PIN\_TRIG1 15

#define PIN\_ECHO1 14

#define PIN\_TRIG2 13

#define PIN\_ECHO2 12

BlynkTimer timer;

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "Wokwi-GUEST";   // your network SSID (name)

char pass[] = "";

#define BLYNK\_PRINT **Serial**

long get\_distance1() {

  // Start a new measurement:

  digitalWrite(PIN\_TRIG1, HIGH);

  delayMicroseconds(10);

  digitalWrite(PIN\_TRIG1, LOW);

  // Read the result:

  duration1 = pulseIn(PIN\_ECHO1, HIGH);

  distance1 = duration1 / 58;

  return distance1;

}

long get\_distance2() {

  // Start a new measurement:

  digitalWrite(PIN\_TRIG2, HIGH);

  delayMicroseconds(10);

  digitalWrite(PIN\_TRIG2, LOW);

  // Read the result:

  duration2 = pulseIn(PIN\_ECHO2, HIGH);

  distance2 = duration2 / 58;

  return distance2;

}

void myTimer() {

**Serial**.println("100");

  dis\_new1 = get\_distance1();

  dis\_new2 = get\_distance2();

  if (dis1 != dis\_new1 || dis2 != dis\_new2){

**Serial**.println("200");

    if (dis1 < dis2){

**Serial**.println("Enter loop");

      entered = entered + 1;

      inside = inside + 1;

      digitalWrite(LED, HIGH);

      Blynk.virtualWrite(V0, entered);

      Blynk.virtualWrite(V2, inside);

      dis1 = dis\_new1;

      delay(1000);

      digitalWrite(LED, LOW);

    }

    if (dis1 > dis2){

**Serial**.println("Leave loop");

      left = left + 1;

      inside = inside - 1;

      Blynk.virtualWrite(V1, left);

      Blynk.virtualWrite(V2, inside);

      dis2 = dis\_new2;

      delay(1000);

    }

  }

}

 void setup() {

**Serial**.begin(115200);

  pinMode(LED, OUTPUT);

  pinMode(PIN\_TRIG1, OUTPUT);

  pinMode(PIN\_ECHO1, INPUT);

  pinMode(PIN\_TRIG2, OUTPUT);

  pinMode(PIN\_ECHO2, INPUT);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 8080);

  timer.setInterval(1000L, myTimer);

}

void loop() {

  Blynk.run();

  timer.run();

}

https://wokwi.com/projects/378449144024167425

**Conclusion**

* This development plan serves as a roadmap for the creation of a real-time transit information platform. Its success will contribute to an improved commuting experience and more efficient public transportation services.

**Thank You**