

## **PHASE -4**

### **DEVELOPMENT PART -2**

#### **OBJECTIVE:**

To build the traffic management platform and mobile apps for iOS and Android ,we will need a combination of web development technologies and mobile app development tools. Here's a high-level overview of the steps to achieve this,

#### **TRAFFIC MANAGEMENT PLATFORM (WEB):**

- **WEB DEVELOPMENT :-**

Create a web application using HTML, CSS, and JavaScript for displaying real-time traffic information. Utilize popular web frameworks like React, Angular, or Vue.js for front-end development to create an interactive and responsive user interface.

- **REAL-TIME TRAFFIC DATA :-**

Integrate with your IoT-based traffic management system to receive real-time traffic data. Use JavaScript libraries or frameworks like Axios or Fetch API to make API requests to your backend server for data updates.

- **DATA PRESENTATION:-**

Display traffic data using interactive maps, charts, and tables. Provide information on traffic flow, congestion, road closures, and any accidents or incidents. Implement features like zoom, pan, and filter options for users to explore the data.

- **USER AUTHENTICATION:**

Implement user authentication and authorization to allow registered users to personalize their experience. Consider using technologies like JWT (JSON Web Tokens) for secure authentication.

- **CROSS-PLATFORM DEVELOPMENT:-**

Consider using cross-platform app development frameworks like React Native or Flutter to build mobile apps for both iOS and Android simultaneously, saving time and effort.

- **APP FEATURES:**

Create mobile apps that provide users with access to real-time traffic updates and route recommendations. Implement user-friendly interfaces with features such as:

1. Real-time traffic maps with markers and overlays.
2. GPS-based navigation and route planning.
3. Alerts for traffic incidents and congestion.
4. User preferences and saved routes.

- **API INTEGRATION:**

Utilize the same APIs used for the web platform to fetch real-time traffic data for mobile apps. Ensure that the mobile apps can display this data seamlessly and in a mobile-friendly format.

- **PUSH NOTIFICATIONS:**

Implement push notifications to alert users about critical traffic events, such as accidents or road closures.

- **TESTING AND DEPLOYMENT:**

Thoroughly test the mobile apps on iOS and Android devices to ensure compatibility and functionality . Deploy the apps to their respective app stores (Apple App Store and Google Play Store).

## Html code:

```
<!DOCTYPE html>
<html>
<head>
  <title>Traffic Management System</title>
</head>
<body>
  <header>
    <h1>Welcome to the Traffic Management System</h1>
  </header>
  <section>
    <h2>Real-time Traffic Information</h2>
    <p>Current traffic conditions: Moderate</p>
    <p>Average speed: 45 mph</p>
    <p>Incidents: None</p>
  </section>
  <section>
    <h2>Plan Your Route</h2>
    <form>
      <label for="origin">Origin:</label>
      <input type="text" id="origin" name="origin" required>

      <label for="destination">Destination:</label>
      <input type="text" id="destination" name="destination" required>

      <input type="submit" value="Get Directions">
    </form>
  </section>
</section>
```

```
<h2>Contact Information</h2>
<address>
  Email:
<a href="mailto:info@trafficmanagement.com">info@trafficmanagement.com</a><br>
  Phone: 123-456-7890
</address>
</section>
<footer>
  <p> 2023 Traffic Management System</p>
</footer>
</body>
</html>
```

## Css code:

```
body, h1, h2, p, form {
  margin: 0;
  padding: 0;
}

body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
  color: #333;
  line-height: 1.6;
}

header {
  background-color: #333;
  color: #fff;
  text-align: center;
  padding: 10px;
}

h1 {
  font-size: 24px;
}

section {
  background-color: #fff;
  margin: 10px;
  padding: 15px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

h2 {
  font-size: 20px;
}

label {
```

```
    display: block;
    margin-bottom: 5px;
}

input[type="text"] {
    width: 100%;
    padding: 5px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    border-radius: 3px;
}

input[type="submit"] {
    background-color: #333;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

input[type="submit"]:hover {
    background-color: #555;
}

address {
    font-style: normal;
    margin-top: 10px;
}

footer {
    text-align: center;
    background-color: #333;
    color: #fff;
    padding: 10px;
    font-size: 14px;
}
```

## JS Code:

```
document.addEventListener("DOMContentLoaded", function () {
    // Sample traffic data
    const trafficData = {
        conditions: "Moderate",
        speed: "45 mph",
        incidents: "None",
    };

    // Function to update the traffic information
    function updateTrafficInfo() {
        const trafficInfo = document.getElementById("traffic-info");
```

```

        trafficInfo.innerHTML = `
            <p>Current traffic conditions: ${trafficData.conditions}</p>
            <p>Average speed: ${trafficData.speed}</p>
            <p>Incidents: ${trafficData.incidents}</p>
        `;
    }

    // Function to handle form submission (for route planning)
    const form = document.querySelector("form");
    form.addEventListener("submit", function (event) {
        event.preventDefault();
        const origin = document.getElementById("origin").value;
        const destination = document.getElementById("destination").value;
        planRoute(origin, destination);
    });

    // Function to simulate route planning (replace with actual logic)
    function planRoute(origin, destination) {
        setTimeout(() => {
            alert(`Routing from ${origin} to ${destination} complete.`);
        }, 1000);
    }

    // Simulate real-time traffic data updates
    setInterval(() => {
        trafficData.speed = (Math.random() * 60 + 20).toFixed(2) + " mph";
        trafficData.incidents = Math.random() < 0.2 ? "Accident" : "None";
        updateTrafficInfo();
    }, 5000); // Simulate every 5 seconds

    // Initial update of traffic information
    updateTrafficInfo();
});

```

