**TRAFFIC MANAGEMENT**

Web Development:

* For the web platform, you can use HTML for the structure, CSS for styling, and JavaScript for interactive elements and real-time data fetching. Here's a basic example of how to structure the web app:

<!DOCTYPE html>

<html>

<head>

<title>Real-Time Traffic Information Platform</title>

<!-- Add CSS for styling -->

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

</head>

<body>

<div id="map"></div>

<div id="traffic-info">

<!-- Real-time traffic information will be displayed here -->

</div>

<!-- Include JavaScript for fetching and displaying real-time data -->

<script src="script.js"></script>

</body>

</html>

Then, you can style the elements and add interactive components using CSS and JavaScript in separate files (styles.css and script.js).

**Mobile App Development:**

For the mobile apps, you can use native technologies such as Swift for iOS and Java/Kotlin for Android, or you can use cross-platform frameworks like React Native or Flutter to develop for both platforms simultaneously. Here's a basic example for an iOS app using Swift

import UIKit

class TrafficInfoViewController: UIViewController {

override func viewDidLoad() {

super.viewDidLoad()

// Create UI elements and set up real-time data fetching

}

}

FOR AN ANDROID APP USING JAVA

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

public class TrafficInfoActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// Set up UI elements and real-time data fetching

}

}

To integrate real-time data fetching and route recommendations in both the web platform and mobile apps. Additionally, ensure a user-friendly interface and intuitive design for a better user experience. Always test thoroughly and consider user feedback to make improvements.