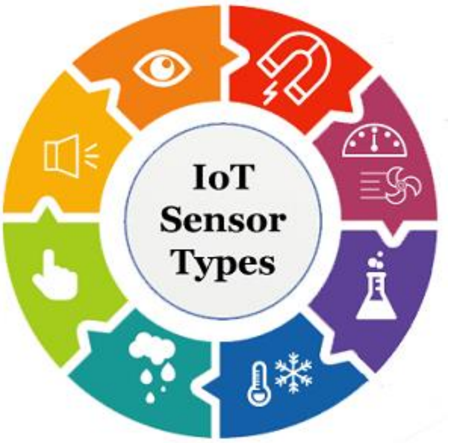
PHASE 5 : SMART PARKING

1. **Phase Real-time Monitoring:** Continuously monitor parking spaces to provide real-time data on parking availability.
2. **User-Friendly Interface:** Develop a mobile app to display parking information to drivers.
3. **Raspberry Pi Integration:** Use Raspberry Pi as a central controller to collect data from IoT sensors and communicate with the mobile app.
4. **Reduced Congestion:** Help drivers quickly find available parking, reducing traffic congestion and emissions.
5. **Improved User Experience:** Enhance the overall parking experience for drivers by reducing search time and frustration.

IoT Sensor Setup:

Deploy IoT sensors in parking areas to monitor parking space occupancy. These sensors typically include:

- **Ultrasonic Sensors:** Detect the presence of vehicles in parking spaces.
- **Raspberry Pi:** Acts as a central controller to collect data from the sensors and communicate with the mobile app.
- **Wi-Fi/Cellular Connectivity:** Allows sensors to transmit data to the Raspberry Pi and, subsequently, to the mobile app.
- **Power Supply:** Sensors are powered by a stable power source, such as a battery or a wired connection.



Mobile App Development:

Develop a mobile app for drivers to check real-time parking availability. The app includes:

- **Map Interface:** A map showing parking areas with color-coded markers indicating available spaces.
- **Search Functionality:** Drivers can search for parking based on location or criteria like proximity to a destination.
- **Navigation Integration:** Integration with navigation apps to guide drivers to the chosen parking spot.

- **Real-time Updates:** Continuous updates on parking availability.
- **Payment Integration:** Optionally, drivers can make payments or reserve parking spots through the app.

Raspberry Pi Integration:

Raspberry Pi serves as the central hub for data collection and communication. It integrates with IoT sensors, collects data on parking space occupancy, and communicates with the mobile app.

Code Implementation:

The code for the IoT sensors involves programming the Raspberry Pi to collect data from the sensors and transmit it to the mobile app. The mobile app is developed using programming languages such as Swift for iOS or Java/Kotlin for Android.

Benefits to Drivers and Alleviation of Parking Issues:

1. **Time and Stress Savings:** Drivers can quickly identify available parking spaces, saving time and reducing the stress of searching for parking.
2. **Reduced Congestion:** By guiding drivers to available parking spots, the system helps reduce traffic congestion, especially in urban areas.
3. **Fuel and Emission Reduction:** As drivers spend less time searching for parking, there is a reduction in fuel consumption and emissions, benefiting the environment.
4. **Improved Urban Mobility:** The system enhances urban mobility by making parking more efficient, contributing to better traffic flow.
5. **Enhanced User Experience:** Drivers enjoy a smoother parking experience, which can positively impact their overall impression of a location.
6. **Revenue Generation:** In cases where payments or reservations are integrated into the app, it can also generate revenue for parking authorities.

In conclusion, a real-time parking availability system benefits drivers by saving time, reducing stress, and alleviating parking issues, while also contributing to reduced traffic congestion and environmental benefits. It offers a win-win solution for drivers and parking management, improving the overall urban mobility experience.

PROGRAM:

```
Import ‘package:flutter/material.dart’;

Import ‘package:http/http.dart’ as http;

Import ‘dart:convert’;


Void main() => runApp(MyApp());


Class MyApp extends StatelessWidget {
```

PHASE 5 : SMART PARKING

```
  @override
  Widget build(BuildContext context) {
    Return MaterialApp(
      Title: ‘Parking Availability’,
      Theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      Home: MyHomePage(title: ‘Parking Availability’),
    );
  }
}

Class MyHomePage extends StatefulWidget {
  MyHomePage({Key key, this.title}) : super(key: key);

  Final String title;

  @override
  _MyHomePageState createState() => _MyHomePageState();
}

Class _MyHomePageState extends State<MyHomePage> {
  String _parkingAvailability = ‘Loading…’;

  Future<void> _getParkingAvailability() async {
    Final response = await http.get(‘http://<raspberry_pi_ip_address>/parking_availability’);
    Final data = json.decode(response.body);
    setState(() {
      _parkingAvailability = data[‘availability’];
    });
  }
```

PHASE 5 : SMART PARKING

```
 @override
 Widget build(BuildContext context) {
  Return Scaffold(
   appBar: AppBar(
    title: Text(widget.title),
   ),
   Body: Center(
    Child: Column(
     mainAxisAlignment: MainAxisAlignment.center,
     children: <Widget>[
      Text(
       'Parking Availability:',
      ),
      Text(
       '$_parkingAvailability',
       Style: Theme.of(context).textTheme.headline4,
      ),
     ],
    ),
   ),
   floatingActionButton: FloatingActionButton(
    onPressed: _getParkingAvailability,
    tooltip: 'Refresh',
    child: Icon(Icons.refresh),
   ),
  );
 }
}
```