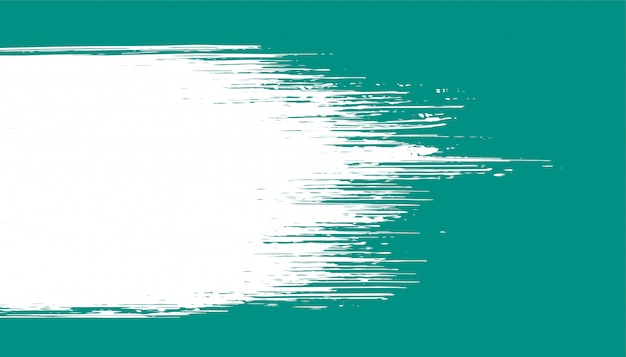
PHASE-4



sumitted by:

dubbala adi kesava reddy

reg no:723921244017

email:ADIKESAVAREDDYDUBBALA@GAMIL.COM

Certainly! Here's a step-by-step guide on how to create a mobile app using Flutter to display real-time parking availability data received from the Raspberry Pi.

1. Set Up Flutter Project:

Create a new Flutter project using the following command in your terminal or command prompt:

flutter create parking\_availability\_app

cd parking\_availability\_app

2. Design the App Interface:

Replace the content of `lib/main.dart` with the following code to design the basic interface of the app:

import 'package: flutter/material.dart';

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

class MyApp extends StatelessWidget {

@override

Widget build (BuildContext context) {

return MaterialApp (

title: 'Parking Availability App',

home: ParkingAvailabilityScreen (),

);

}

}

class ParkingAvailabilityScreen extends StatefulWidget {

@override

\_ParkingAvailabilityScreenState createState () =>

\_ParkingAvailabilityScreenState ();

}

class \_ParkingAvailabilityScreenState extends State<ParkingAvailabilityScreen> {

// TODO: Implement functions to receive and display parking availability data

// from Raspberry Pi.

@override

Widget build (BuildContext context) {

return Scaffold (

appBar: AppBar (

title: Text ('Parking Availability'),

),

body: Center (

child: Text ('Real-time parking availability will be displayed here.'),

),

);

}

}

3. Receive Data from Raspberry Pi:

You can use packages like http to fetch real-time data from the Raspberry Pi. Add this to your pubspec.yaml file:

dependencies:

flutter:

sdk: flutter

http: ^0.13.3 # Add the latest version of the http package

Then, fetch data from your Raspberry Pi inside the \_ParkingAvailabilityScreenState` class using the http package:

import 'package: http/http.dart' as http;

class \_ParkingAvailabilityScreenState extends State<ParkingAvailabilityScreen> {

String availabilityData = 'Loading...';

Future<void> fetchParkingAvailability () async {

final response =

await http.get (Uri.parse('YOUR\_RASPBERRY\_PI\_API\_ENDPOINT'));

if (response. status Code == 200) {

setState (() {

availabilityData = response. body;

});

} else {

throw Exception ('Failed to load parking availability data');

}

}

@override

void initState () {

super. initState ();

fetchParkingAvailability ();

}

@override

Widget build (BuildContext context) {

return Scaffold (

appBar: AppBar (

title: Text ('Parking Availability'),

),

body: Center (

child: Text ('Parking Availability: $availabilityData'),

),

);

}

}

Replace YOUR\_RASPBERRY\_PI\_API\_ENDPOINT with the actual endpoint URL where your Raspberry Pi is serving the parking availability data.

4. Run the App:

Connect your mobile device or emulator and run the app using the following command:

flutter runs

Now, your Flutter app will display real-time parking availability data received from the Raspberry Pi. Make sure your Raspberry Pi is running and accessible via the specified API endpoint for this setup to work properly.