**Phase 4**

Project Title: SMART PARKING

**Project ID:**  proj\_223739\_Team\_5

**College Code:** 6208

**College:** Gnanamani College of Technology

**Branch:** B.Tech-Information Technology

**Team Members & ID:**

ABISHEK R - (au620821205002)

SANJAY S - (au620821205047)

PRIYADHARSAN B -(au620821205303)

KARTHICKRAJA M -(au620821205025)

## PHASE 4 :

Creating a complete app that receives and displays parking availability data from a Raspberry Pi using Python and the Flutter framework is a complex project that involves both frontend and backend development. I can provide you with a basic outline of the steps involved and some code snippets to get you started:

1. Set up the Raspberry Pi:

- Connect sensors or cameras to the Raspberry Pi to collect parking availability data.

- Use Python on the Raspberry Pi to process this data and expose it through an API.

2. Create a Flutter App:

- Set up a Flutter development environment.

- Create a new Flutter project.

- Add necessary dependencies for making HTTP requests and building the UI.

3. Implement Flutter UI:

- Create a user interface to display parking availability data. This might include a map, list, or any suitable visualization.

- Design the UI to show the data dynamically as it's received.

4. Create API Requests in Flutter:

- Use Flutter's HTTP package (such as `http` or `dio`) to make API requests to the Raspberry Pi.

- Retrieve parking availability data from the Raspberry Pi using these requests.

5. Update UI with Real-time Data:

- Implement a mechanism to continuously fetch and update parking availability data from the Raspberry Pi.

- Update the UI to reflect the real-time data.

Here's a simplified Python script to serve as a starting point on the Raspberry Pi:

```python

# Import necessary libraries (e.g., Flask for creating a web API)

from flask import Flask, jsonify

app = Flask(\_\_name)

# Simulated parking availability data

parking\_data = {

'available\_spots': 50,

'total\_spots': 100,

}

@app.route('/get\_parking\_data', methods=['GET'])

def get\_parking\_data():

return jsonify(parking\_data)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=5000)

```

In your Flutter app, you can make HTTP requests to the Raspberry Pi's endpoint (e.g., `http://raspberry\_pi\_ip:5000/get\_parking\_data`) to retrieve parking availability data. You'll need to parse the JSON response and update your app's UI accordingly.

Remember that this is just a simplified example. In a real-world scenario, you would likely use sensors, databases, and more advanced techniques to collect and manage parking availability data.

Building a complete app like this is a substantial project, and you may need to further refine and expand the codebase to meet your specific requirements.