

PHASE 5: Project Documentation & Submission

PROJECT OBJECTIVE:

The main objectives of the project are to design and implement a real-time parking availability system that benefits drivers and alleviates parking issues. This system involves several components, including IoT Raspberry Pi Pico sensor setup, mobile app development using HTML, CSS, JavaScript, and Flask, Raspberry Pi integration, and the use of IBM Cloud DB2 to store data received from the Wokwi Raspberry Pi.

COMPONENTS OF THE PROJECT:

IoT Raspberry Pi Pico on Wokwi Sensor Setup:

- Configure and set up a network of IoT devices using Raspberry Pi Pico microcontrollers.
- Implement sensors and actuators to monitor parking spaces in real-time.
- Connect the Raspberry Pi Pico boards to a central server for data collection and processing.

Mobile App Development (HTML, CSS, JavaScript, Flask):

- Create a mobile app that allows users to access real-time parking availability information.
- Design the user interface using HTML and CSS for an intuitive and user-friendly experience.
- Develop the app's functionality using JavaScript for interactivity.
- Implement a Flask-based server to handle communication between the app and the IoT devices.

Raspberry Pi Integration:

- Integrate Raspberry Pi boards with sensors (e.g., ultrasonic sensors) to detect vehicle presence in parking spaces.
- Use Wi-Fi or another suitable communication protocol to transmit data from the Raspberry Pi boards to the central server.

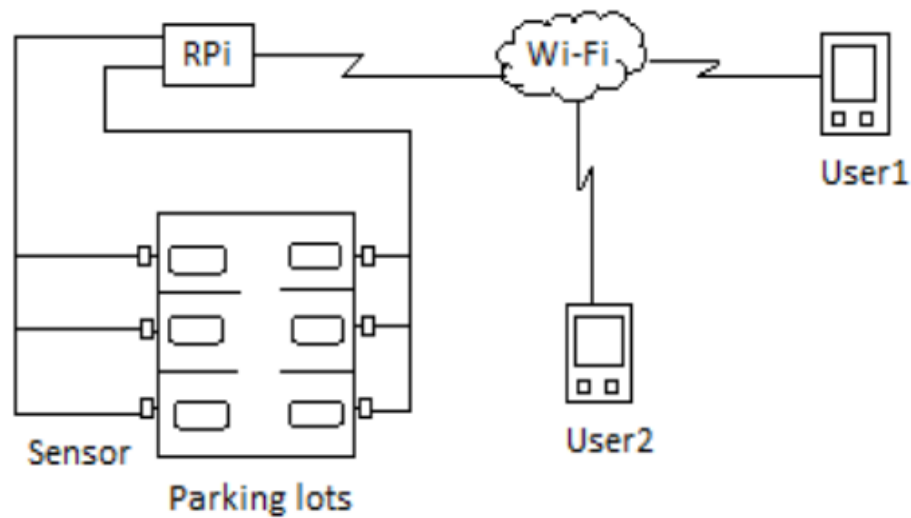
IBM Cloud DB2 Integration:

- Use IBM Cloud DB2 or a similar database service to store data received from the Wokwi Raspberry Pi.
- Create a database schema to store information about parking spaces, their availability, and relevant metadata.

Code Implementation:

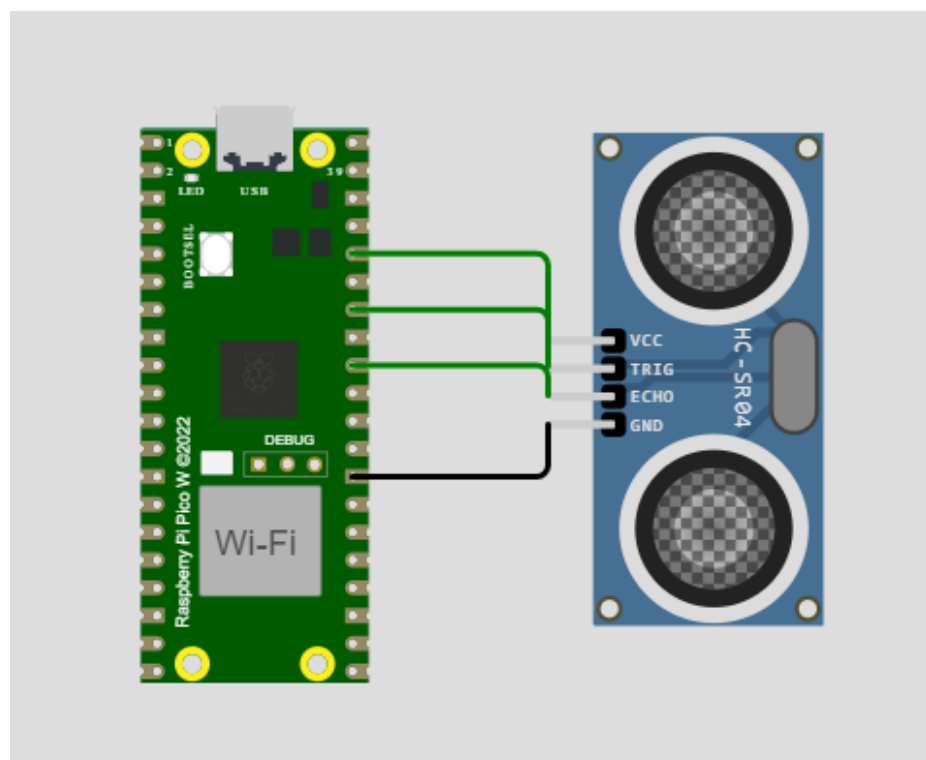
- Write code for the Raspberry Pi Pico to capture sensor data and transmit it to the server.
- Develop the server-side code to handle incoming data, update the database, and serve the data to the mobile app.
- Implement the mobile app's functionality for users to interact with the system.

DIAGRAM:



“The above is the working diagram of,0 How the entire project work on . It includes sensors of the parking slot, Cloud connection to receive data and an application to make all together”.

CIRCUIT DIAGRAM :



“The above circuit is done in WOKWI platform”

Connection of IoT sensors to detect parking space occupancy.

The screenshot displays the WOKWI platform interface. On the left, a code editor shows a MicroPython script for an ultrasonic sensor. The script defines pins, a measurement function, and a loop that prints distance and pulse duration. On the right, a simulation window shows a Raspberry Pi connected to an ultrasonic sensor module. Below the simulation, a terminal window displays the output of the script, showing distance measurements around 403 cm and pulse durations around 23525 us.

```
1 from machine import Pin
2 import time
3
4 # Ultrasonic sensor pins
5 TRIG_PIN = 28 # GPIO Pin for Trigger (Pin 2 in Wokwi)
6 ECHO_PIN = 27 # GPIO Pin for Echo (Pin 3 in Wokwi)
7
8 # Function to measure distance using ultrasonic sensor
9 def measure_distance():
10     trig = Pin(TRIG_PIN, Pin.OUT)
11     echo = Pin(ECHO_PIN, Pin.IN)
12
13     trig.off()
14     time.sleep_us(2)
15
16     trig.on()
17     time.sleep_us(10)
18     trig.off()
19
20     pulse_start = pulse_end = 0
21     while echo.value() == 0:
22         pulse_start = time.ticks_us()
23
24     while echo.value() == 1:
25         pulse_end = time.ticks_us()
26
27     pulse_duration = pulse_end - pulse_start
28     distance = (pulse_duration * 0.0343) / 2 # Speed of sound = 343 m/s
29
30     return pulse_duration, distance
```

Simulation output:

```
Distance: 403.30 cm
Pulse Duration: 23526 us
Distance: 403.47 cm
Pulse Duration: 23525 us
Distance: 403.45 cm
```

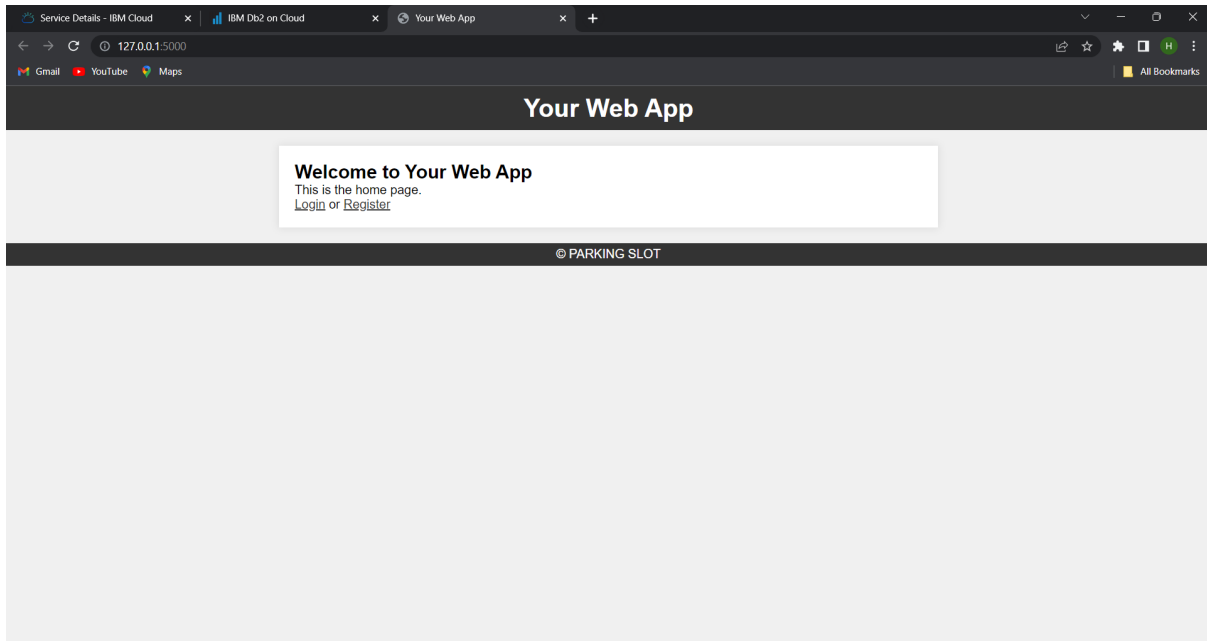
“Since Wokwi’s MicroPython Raspberry Pi does not support IBM Cloud DB2 modules, simulated values are added to the table.”

The screenshot shows the IBM Db2 on Cloud interface. The left sidebar displays a list of data objects, including tables, views, and aliases. The main area shows a table named 'MEASUREMENT' with a single column 'MEASUREMENT'. The table contains 20 rows of simulated data, ranging from 150.0 to 750.0. The interface includes a search bar, a 'Run all' button, and a 'Syntax assistant' toggle.

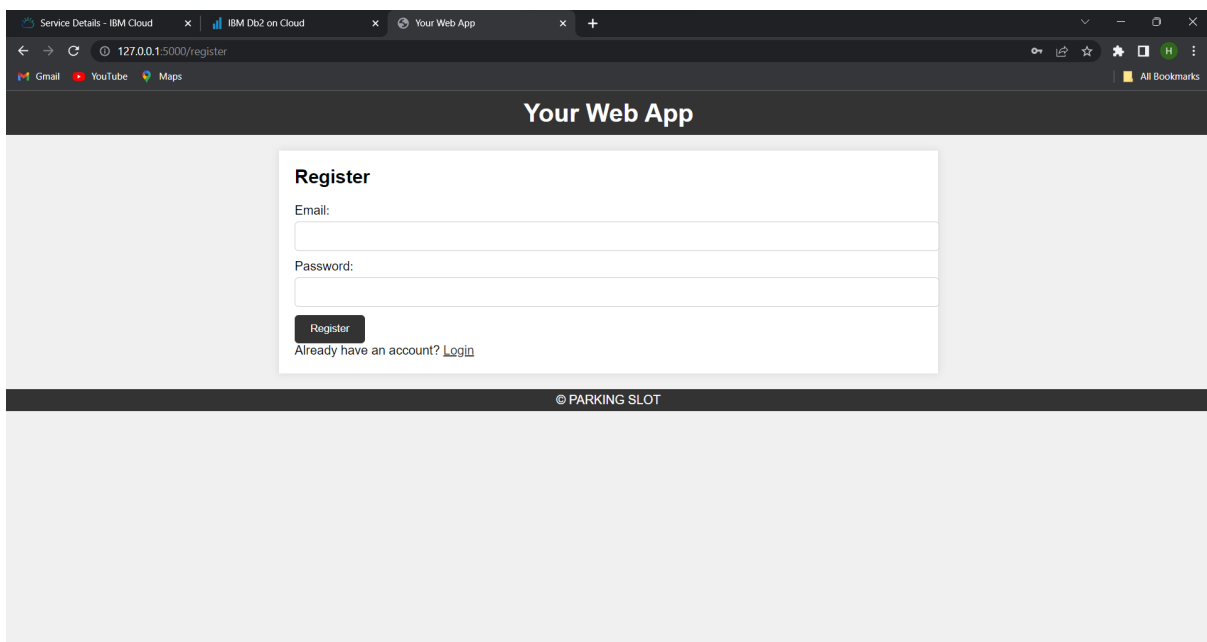
MEASUREMENT
278.0
222.0
350.0
555.0
450.0
750.0
150.0
202.0
420.0
212.0

APPLICATION:

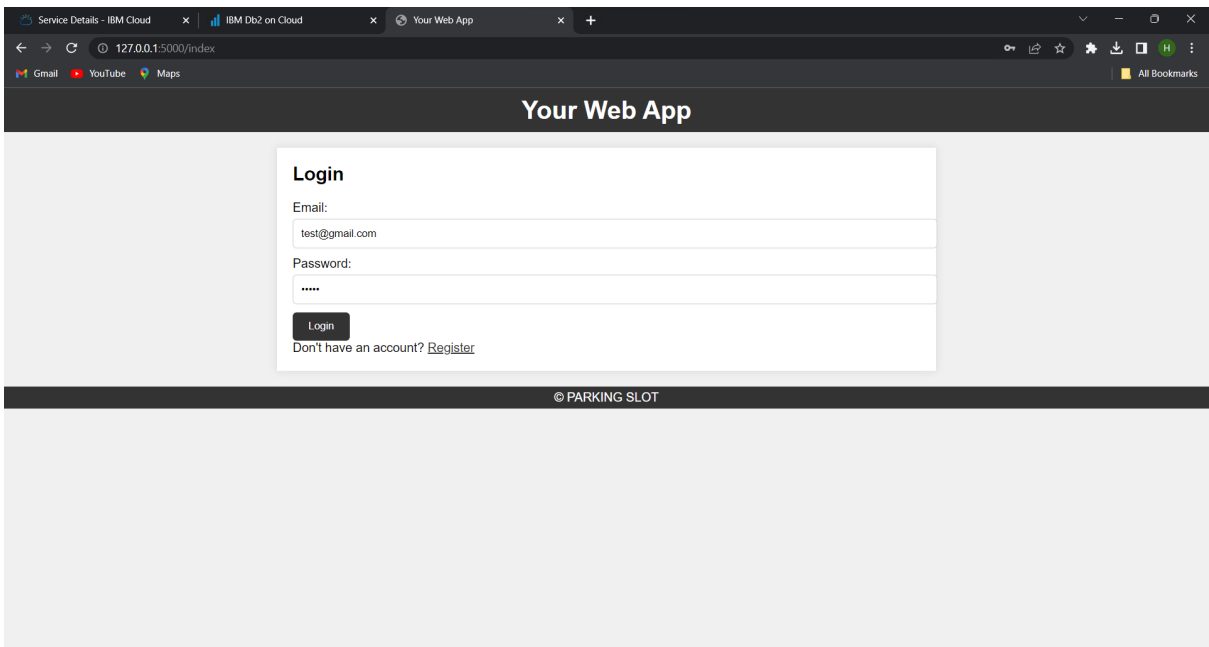
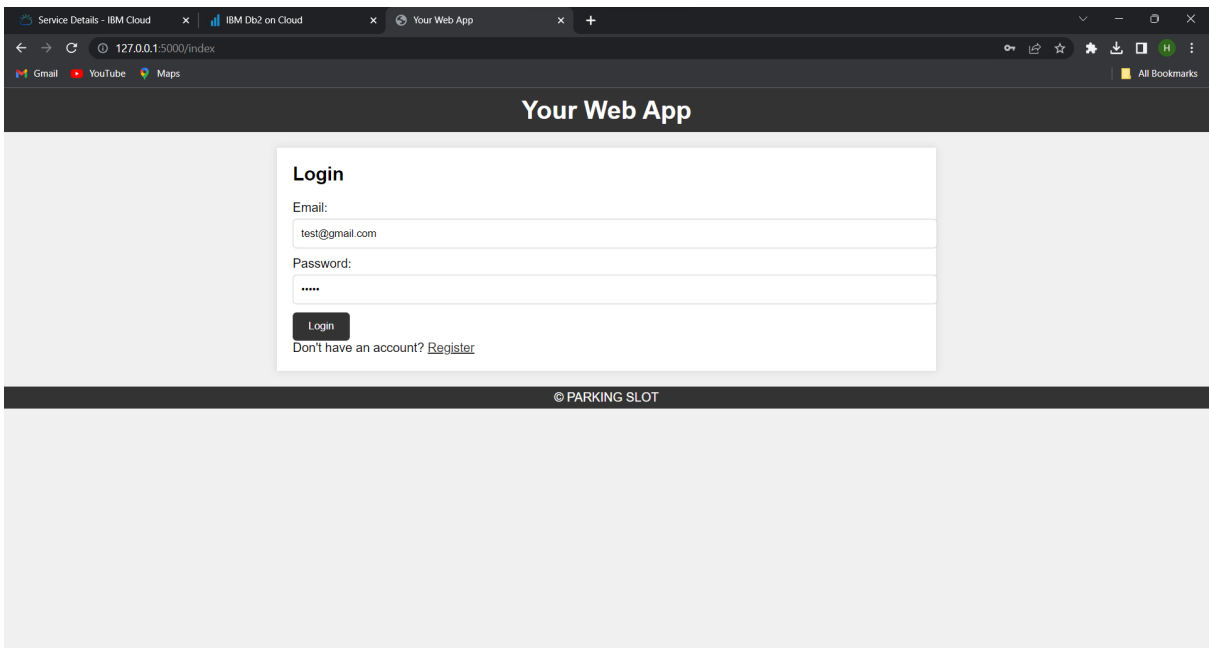
HOME PAGE:



Registration page:



Login page:



Parking Availability:

Service Details - IBM Cloud

IBM Db2 on Cloud

127.0.0.1:5000/index

127.0.0.1:5000/index

Gmail

YouTube

Maps

All Bookmarks

Parking Availability

Slot 1

Slot 2

Slot 3

Slot 4

NotAvailable

Available

NotAvailable

Available

CLOUD CONNECTION:

IBM Db2 on Cloud

Overview

In-flight executions

Connections

Table performance

Real time

Pause data refresh

Refresh

Last collected: 23/11/01 06:53:16 PM

Search application handle or application name

Application name	Client user ID	Client IP address	Application handle	Connection start time	Workload name	Service superclass name	Workload occurren
python.exe		172.30.156.0	20241	Nov 1, 2023 6:53:12 PM	SYSDEFAULTUSER WORKLOAD	SYSDEFAULTUSERCLASS	UOWWAIT
python.exe		172.30.156.0	19754	Nov 1, 2023 6:53:07 PM	SYSDEFAULTUSER WORKLOAD	SYSDEFAULTUSERCLASS	UOWWAIT

“In the previous phases, necessary steps and code have been included .As a result , the above screenshot have been provided”.

Benefits of the Real-time Parking Availability System:

- **Reduced Traffic Congestion:** Drivers can access real-time information about available parking spaces, reducing the time spent searching for parking spots. This leads to less traffic congestion and a more efficient flow of vehicles.
- **Time and Fuel Savings:** Drivers can save time and fuel by quickly finding parking spaces without circling around or waiting for spots to become available.
- **Improved User Experience:** The mobile app provides a convenient and user-friendly way for drivers to plan their parking, reserve spots in advance, and receive alerts when their reserved spot is about to become available.
- **Reduced Environmental Impact:** Less time spent searching for parking results in reduced fuel consumption and emissions, contributing to a more environmentally friendly transportation system.
- **Data-Driven Insights:** The system collects valuable data on parking space usage, allowing city planners and businesses to analyze trends, optimize parking management, and make data-driven decisions for future infrastructure development.

- **Revenue Generation:** Parking space owners can monetize the system by offering reservations and collecting fees from users, creating a new revenue stream.

CONCLUSION:

The real-time parking availability system aims to enhance the overall parking experience for drivers, improve traffic flow, and provide valuable data for urban planning. It represents a practical and data-driven solution to address parking issues in urban areas.

