INDUSTRIAL INTERNSHIP REVIEW
Winter Semester 2021-2022

Name: Sarnitha G U
Registration Number:19BEC0575

Organization: Smartbridge Educational Services
Course: Internet of Things

Course Completion Certificate

Smart Internz

N·E·A·T
National Educational Alliance for Technology

Externship Certificate

This to certify that

Sarnitha G U

has successfully completed the externship program on **Internet of Things (IoT) powered by IBM** from 28 June 2021 to 31 July 2021 and fulfilled the project work requirements.

Certificate ID: Ext-IOT-2022-4426

August 10, 2021
Issued Date

Jayaprakash. Ch
Program Manager

Table of contents

INTRODUCTION

The Internet of Things is the concept of connecting any device (so long as it has an on/off switch) to the Internet and to other connected devices. The IoT is a giant network of connected things and people – all of which collect and share data about the way they are used and about the environment around them.

That includes an extraordinary number of objects of all shapes and sizes – from smart microwaves, which automatically cook your food for the right length of time, to self-driving cars, whose complex sensors detect objects in their path, to wearable fitness devices that measure your heart rate and the number of steps you've taken that day, then use that information to suggest exercise plans tailored to you. There are even connected footballs that can track how far and fast they are thrown and record those statistics via an app for future training purposes.

OVERVIEW OF THE INTERNSHIP

Learnt to use various application in implementing an IoT system. The applications which were use are:

1. Python IDLE

Python is a popular programming language that can be used to create web applications, create workflows and rapid prototyping. It is open sourced, is an interpreted language (executed one line at a time), cross platform language (can run on any operating system) and is an object oriented language. It is also extensible. Python also supports the feature of dynamic memory allocation.

2. IBM IoT Platform

IBM IoT Platform is platform offered by IBM services. It integrates a bundled set of services to connect, capture, register, analyze, and archive IoT devices and data. With the help of the device management service, we can create and manage devices and perform bulk device addition or removal. The data from the device is moved to this platform using MQTT protocol, which is then transferred to web or mobile applications using API services. This platform where we can manage devices is useful to access all the current and previously sent or stored data.

3.Node Red

Node-red is an open sourced, flow based platform. It can be used as an IoT platform and a dashboard. It supports the building of web/mobile applications by joining nodes using a web interface. The part named flow is where all the nodes and wires are gathered and worked upon. It uses the JSON format (Javascript Object Notation) to describe its metadata.

4.MIT App Inventor

App Inventor is a free, cloud-based service that allows us to make our own mobile apps using a block based programming language. We access App Inventor using a web browser (Chrome, Firefox, Safari). A mobile application can be built using the blocks available here, and this is integrated with the node red flow that is in turn integrated with IBM Watson platform.

PROJECT: INTELLIGENT RESTAURANT WITH SMART BEACONS

PURPOSE:

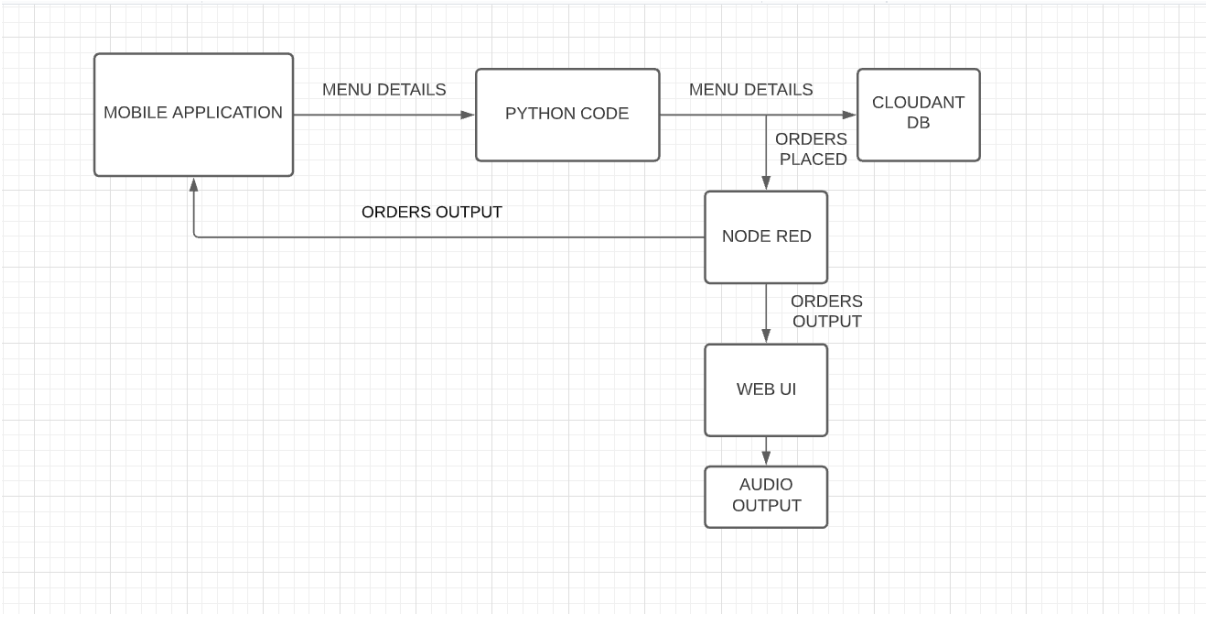
Delivering a tailored digital food menu to customers

customised mobile app can show the food menu on user's smartphone as soon as he reaches your outlet.

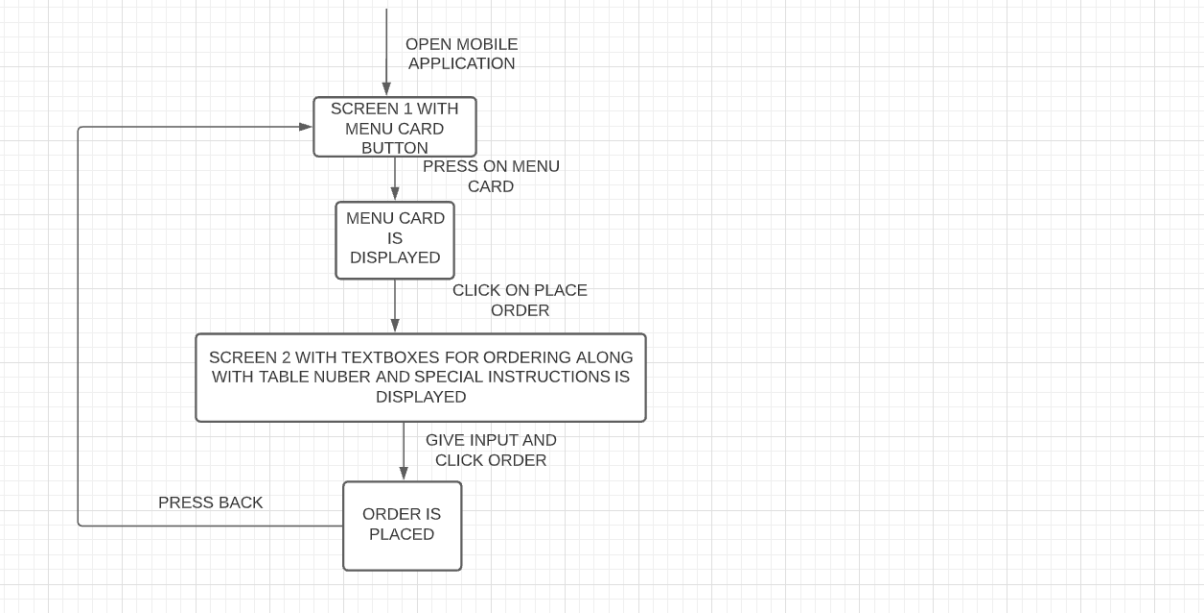
Food Ordering from Dine-In Table

Whenever the customer place orders from the his mobile, inform kitchen staff the table number along with the food order.

BLOCK DIAGRAM ON FLOW OF THE PROJECT



BLOCK DIAGRAM SHOWING THE WORKING OF MOBILE APPLICATION



Required services and softwares:

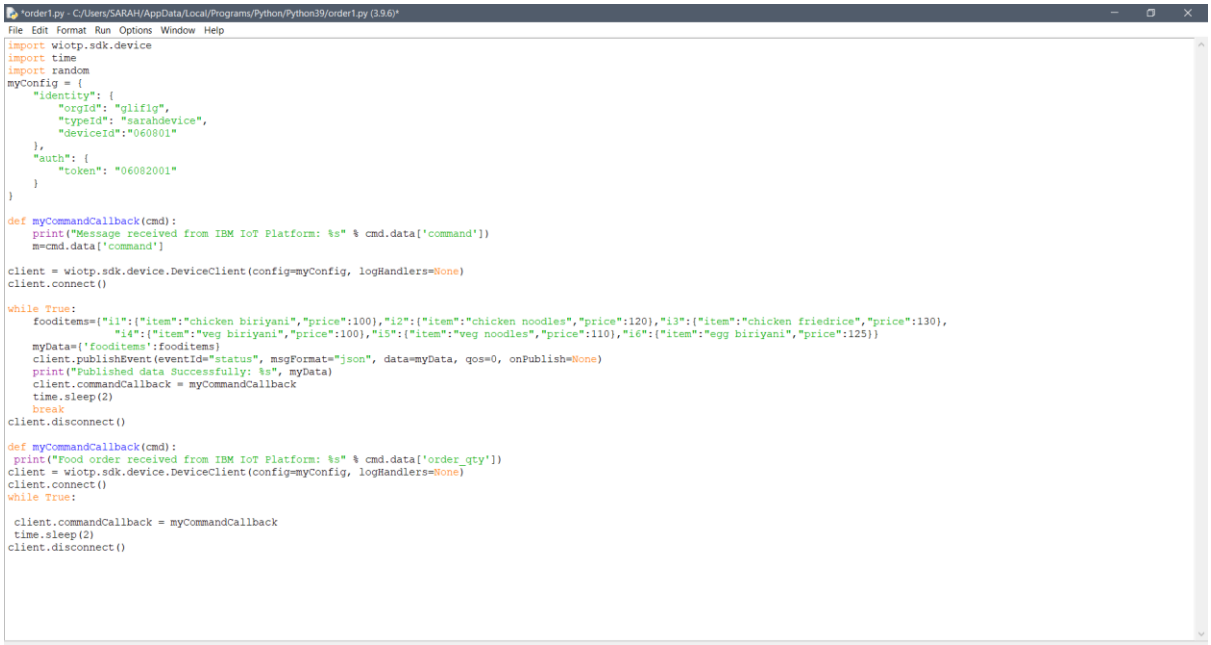
1. IBM cloud Account
2. IBM IoT Platform
3. NodeRed
4. MIT App Inventor
5. IDLE Python
6. IBM Text to speech
7. Fast2sms

Activities:

SETUP ENVIRONMENT:

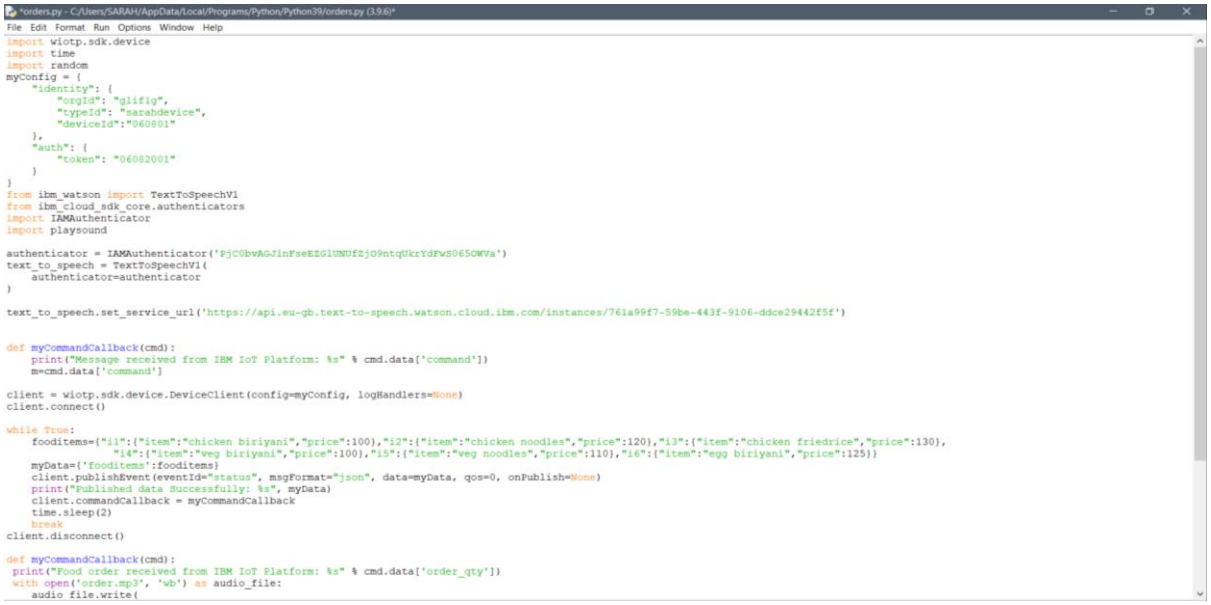
1. Create a device in IBM Watson IoT platform
2. Create Node-red application
3. Develop a python code to subscribe and get the data (menu details) from the IBM IoT Platform
4. Develop a python code to generate voice commands.
5. Create a web UI to display the orders in the kitchen
6. Create a mobile application to display the received menu and also integrate a text box to take order from user.

1.PYTHON CODE USED TO GET THE MENU DETAILS (DATA) FROM THE IBM IoT PLATFORM



```
*order1.py - C:/Users/SARAH/AppData/Local/Programs/Python/Python39/order1.py (3.9.6)*
File  Edit  Format  Run  Options  Window  Help
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "gliflg",
        "typeId": "sarahdevice",
        "deviceId":"060801"
    },
    "auth": {
        "token": "06082001"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    fooditems={"i1":{"item":"chicken biriyani","price":100},"i2":{"item":"chicken noodles","price":120},"i3":{"item":"chicken friedrice","price":130},
               "i4":{"item":"veg biriyani","price":100},"i5":{"item":"veg noodles","price":110},"i6":{"item":"egg biriyani","price":125}}
    myData={'fooditems':fooditems}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
    break
client.disconnect()

def myCommandCallback(cmd):
 print("Food order received from IBM IoT Platform: %s" % cmd.data['order_qty'])
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
while True:

 client.commandCallback = myCommandCallback
 time.sleep(2)
client.disconnect()
```

Python code to get voice commands

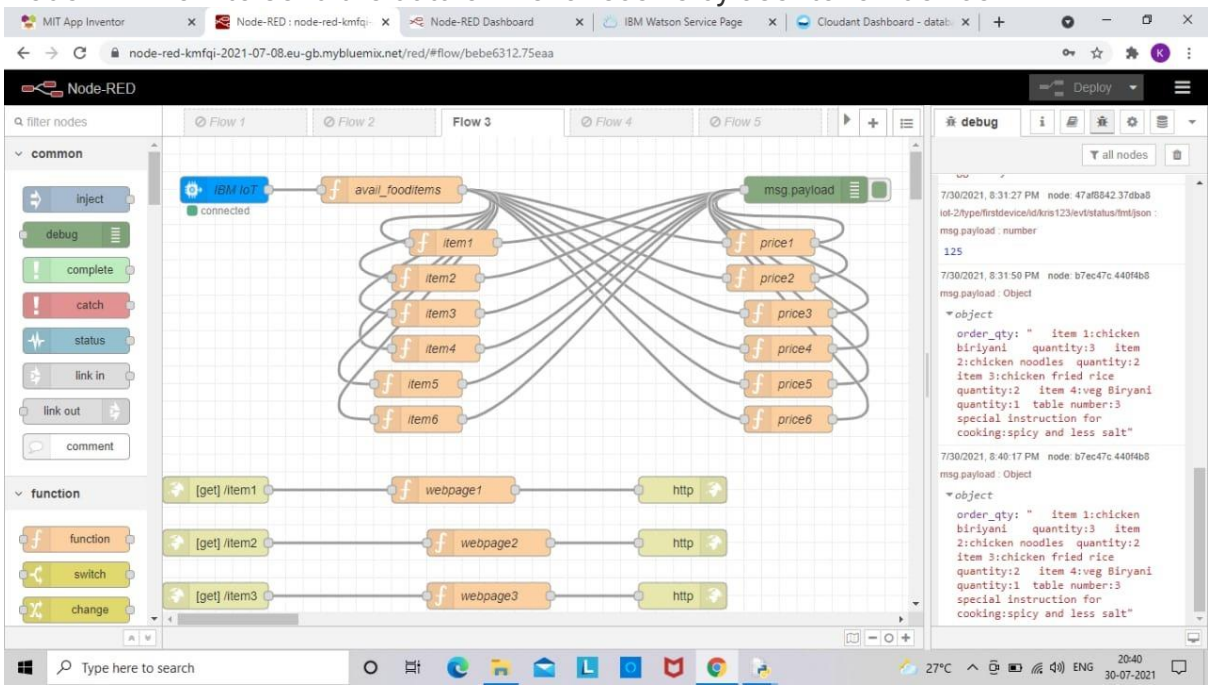


```
*orders.py - C:/Users/SARAH/AppData/Local/Programs/Python/Python39/orders.py (3.9.6)*
File  Edit  Format  Run  Options  Window  Help
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "gliflg",
        "typeId": "sarahdevice",
        "deviceId":"060801"
    },
    "auth": {
        "token": "06082001"
    }
}
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators
import IAMAuthenticator
import playsound

authenticator = IAMAuthenticator('PjC0bvAGJlnFseEZGlUNUfZjO9ntqUkrYdfWS065OWVa')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)

text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/761a99f7-59be-443f-9106-ddce29442f5f')


def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    fooditems={"i1":{"item":"chicken biriyani","price":100},"i2":{"item":"chicken noodles","price":120},"i3":{"item":"chicken friedrice","price":130},
               "i4":{"item":"veg biriyani","price":100},"i5":{"item":"veg noodles","price":110},"i6":{"item":"egg biriyani","price":125}}
    myData={'fooditems':fooditems}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
    break
client.disconnect()

def myCommandCallback(cmd):
 print("Food order received from IBM IoT Platform: %s" % cmd.data['order_qty'])
 with open('order.mp3', 'wb') as audio_file:
    audio_file.write(
```

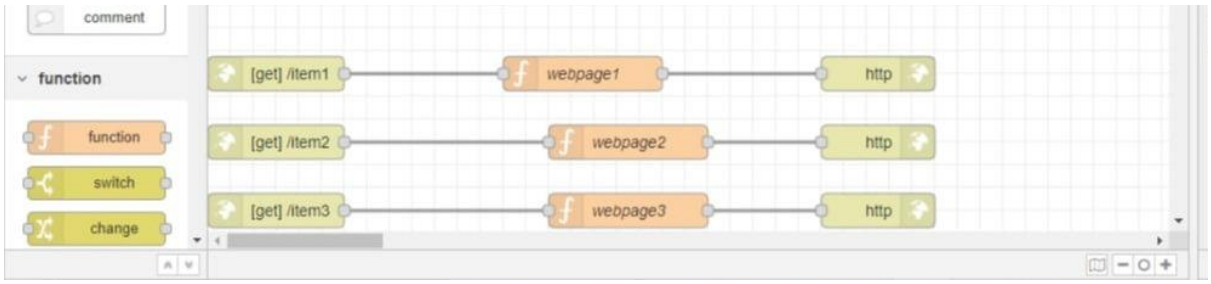
```
*orders.py - C:/Users/SARAH/AppData/Local/Programs/Python/Python39/orders.py (3.9.6)*
File  Edit  Format  Run  Options  Window  Help
from ibm_cloud_sdk_core.authenticators
import IAMAuthenticator
import playsound

authenticator = IAMAuthenticator('PjC0bvAGJlnFseEZGlUNUfZjO9ntqUkrYdFwS065OWVa')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)

text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/761a99f7-59be-443f-9106-ddce29442f5f')


def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    fooditems={"i1":{"item":"chicken biriyani","price":100},"i2":{"item":"chicken noodles","price":120},"i3":{"item":"chicken friedrice","price":130},
               "i4":{"item":"veg biriyani","price":100},"i5":{"item":"veg noodles","price":110},"i6":{"item":"egg biriyani","price":125}}
    myData={'fooditems':fooditems}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
    break
client.disconnect()

def myCommandCallback(cmd):
 print("Food order received from IBM IoT Platform: %s" % cmd.data['order_qty'])
 with open('order.mp3', 'wb') as audio_file:
    audio_file.write(
        text_to_speech.synthesize(
            cmd.data['order_qty'],
            voice='en-US_AllisonV3Voice',
            accept='audio/mp3'
        ).get_result().content)
    playsound.playsound('order.mp3')

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
while True:

 client.commandCallback = myCommandCallback
 time.sleep(2)
client.disconnect()
```

2. Building web application

Node-RED flow to send the data of menu receive by user to IoT device

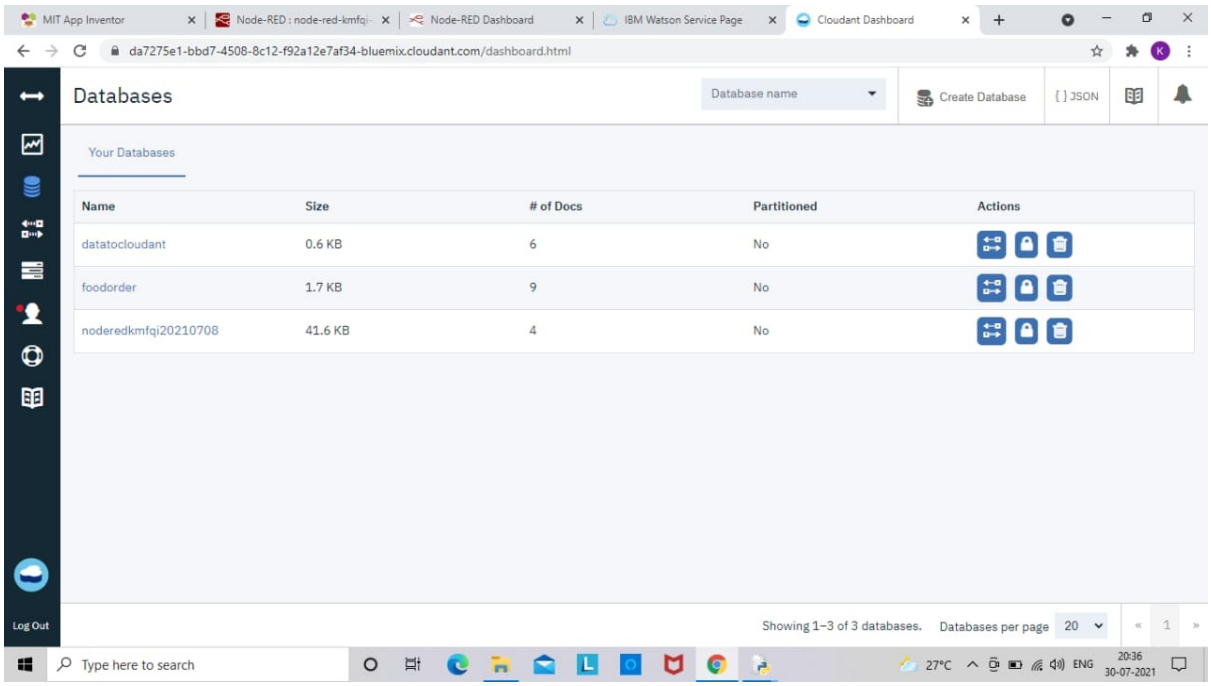


Http Requests To Communicate With Mobile App

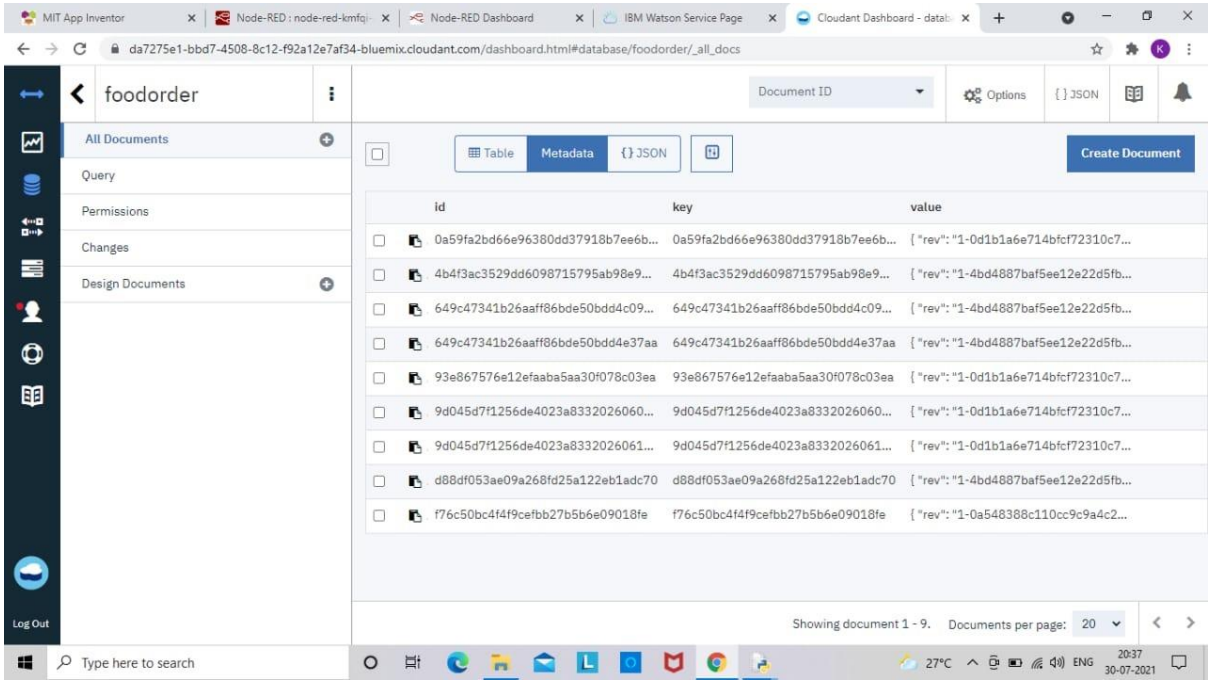


After receiving data from mobile application we store the entire orders information in the cloudant DB so that the entire orders information will be recorded.

A database named "foodorder" is created



Documents stored in the above created database

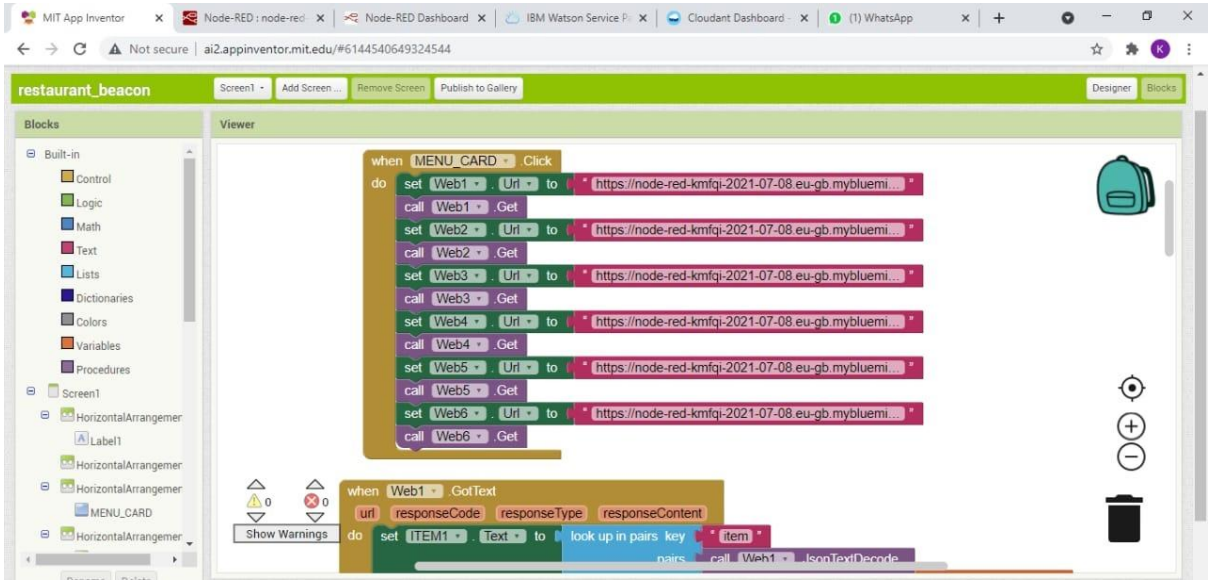


3.Building Mobile Application

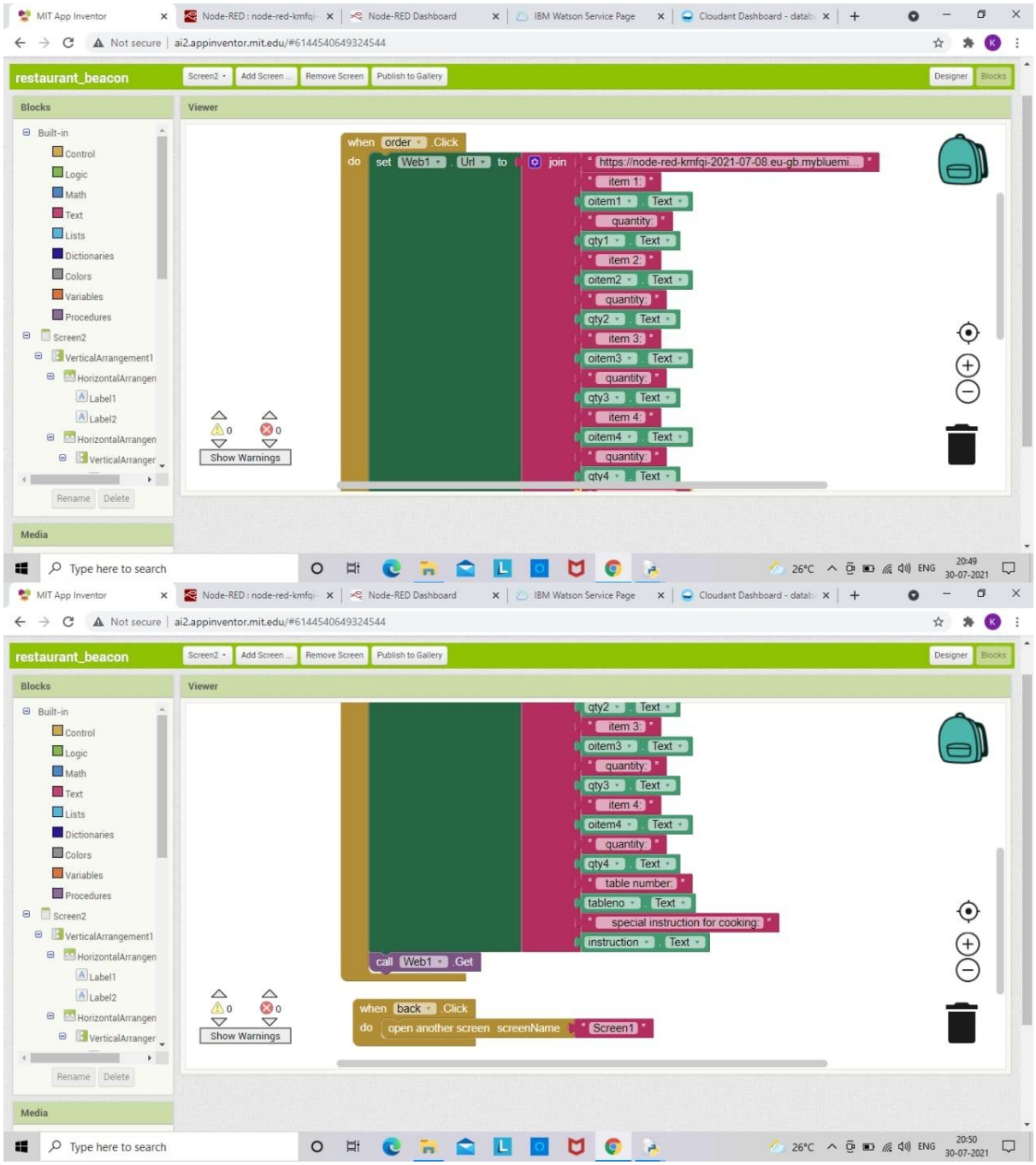
Design of UI to display the received menu integrated with a text box to take order from user.

Blocks for the above designed mobile application

screen 1

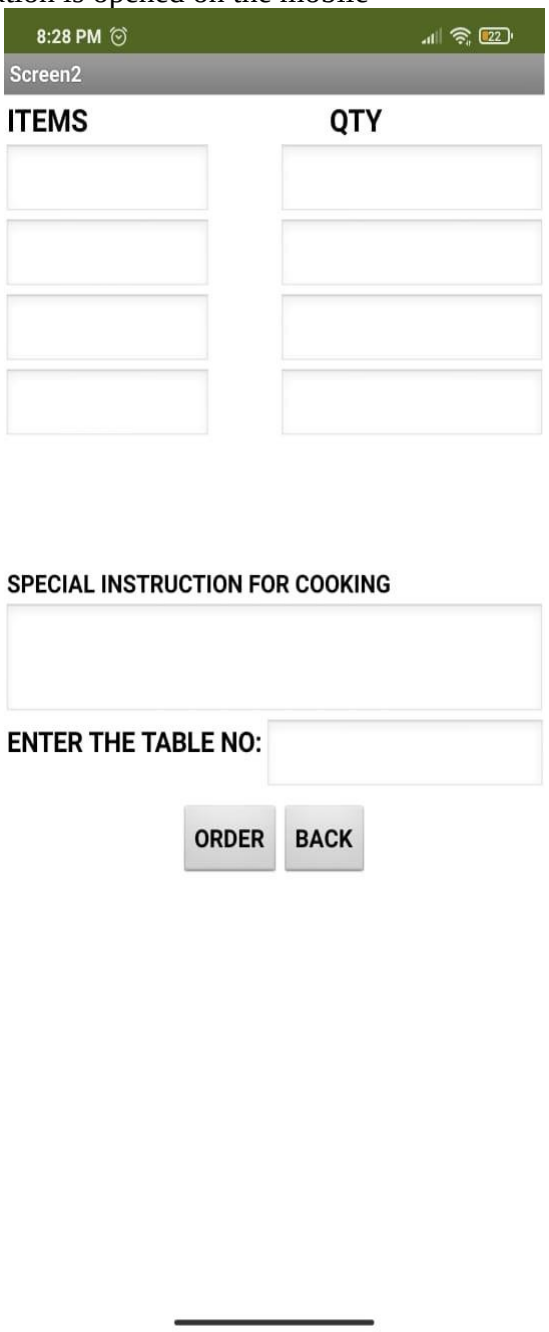


screen 2

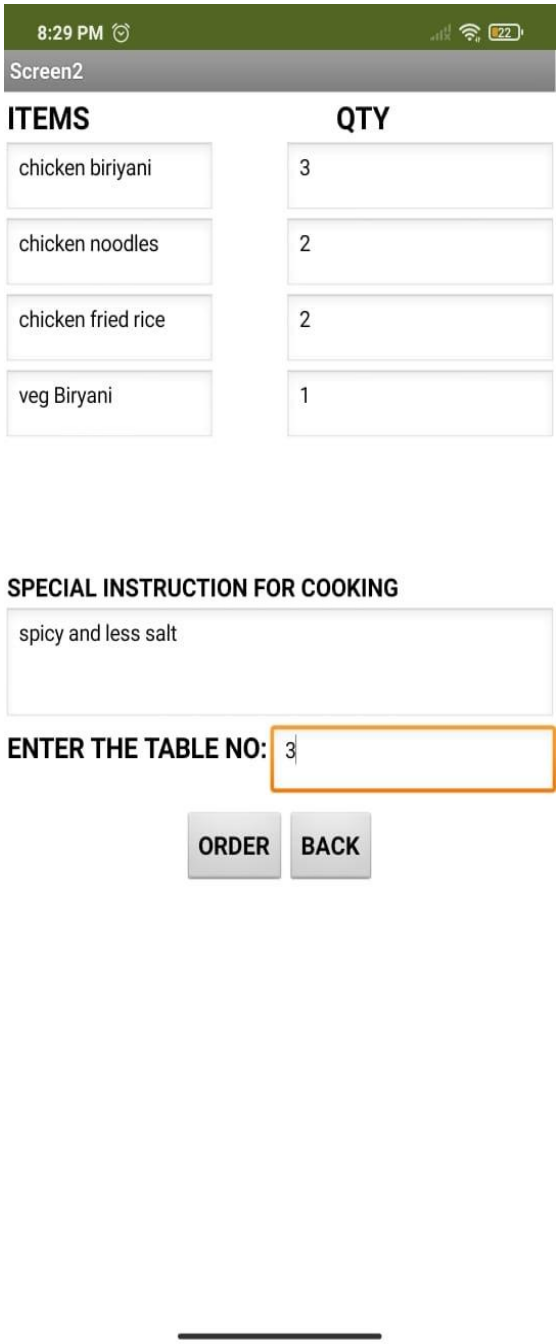

restaurant_beacon
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
when order .Click
do set Web1 . Url to join
https://node-red-kmfqi-2021-07-08.eu-gb.mybluemi...
item 1:
oitem1 . Text
quantity:
qty1 . Text
item 2:
oitem2 . Text
quantity:
qty2 . Text
item 3:
oitem3 . Text
quantity:
qty3 . Text
item 4:
oitem4 . Text
quantity:
qty4 . Text
Show Warnings
Rename
Delete
Media
Type here to search
26°C
ENG
20:49
30-07-2021

restaurant_beacon
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
qty2 . Text
item 3:
oitem3 . Text
quantity:
qty3 . Text
item 4:
oitem4 . Text
quantity:
qty4 . Text
table number:
tableno . Text
special instruction for cooking:
instruction . Text
call Web1 .Get
when back .Click
do open another screen screenName Screen1
Show Warnings
Rename
Delete
Media
Type here to search
26°C
ENG
20:50
30-07-2021

* screen that is displayed when the application is opened on the mobile



* screen that is displayed when the food is ordered



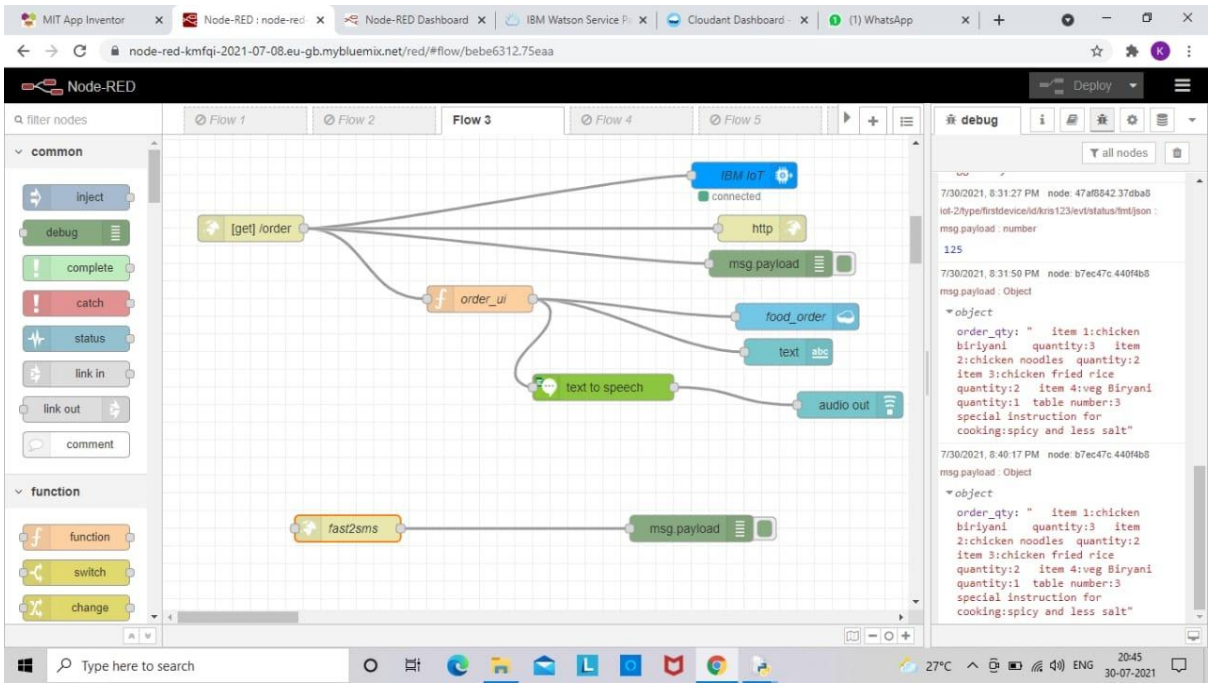
4.Configure the application to receive the data from the cloud and to send the menu details:

* To get the data from the webpage, we add the WEB component from the Connectivity which is present in the Palette

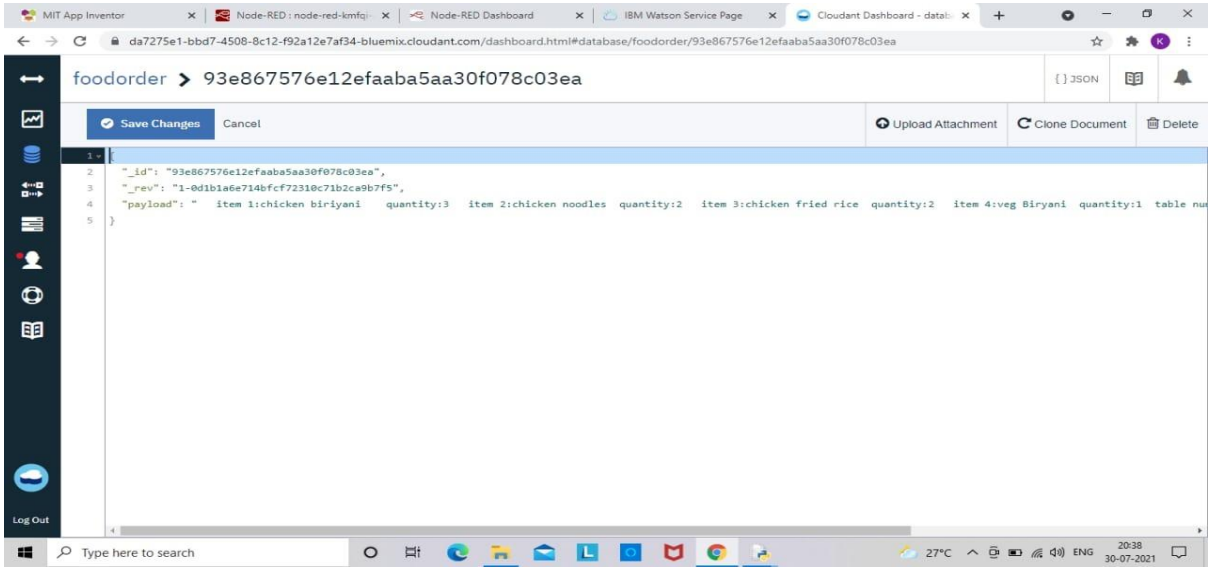
* As it is a non-visible component it will not display on the UI instead, it will be displayed down to the screen

5. The users order is taken and sent to cloud using Http request.

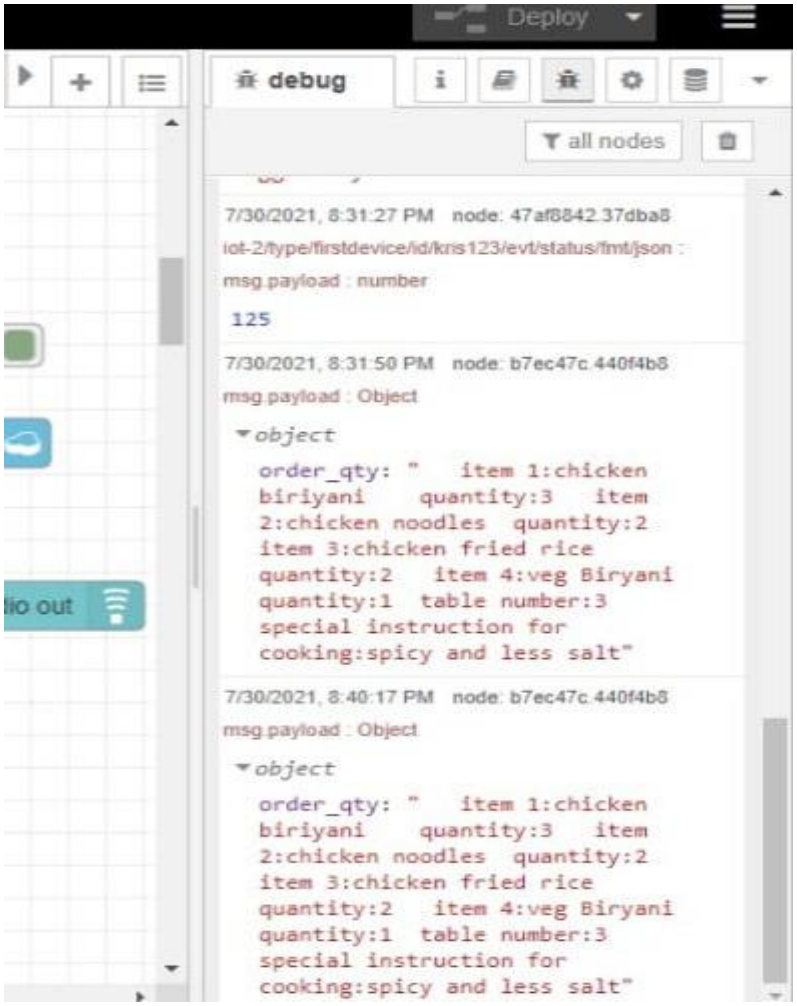
Node red flow to store the orders in cloud using http request and integrating text to speech service to get audio output in the webpage.



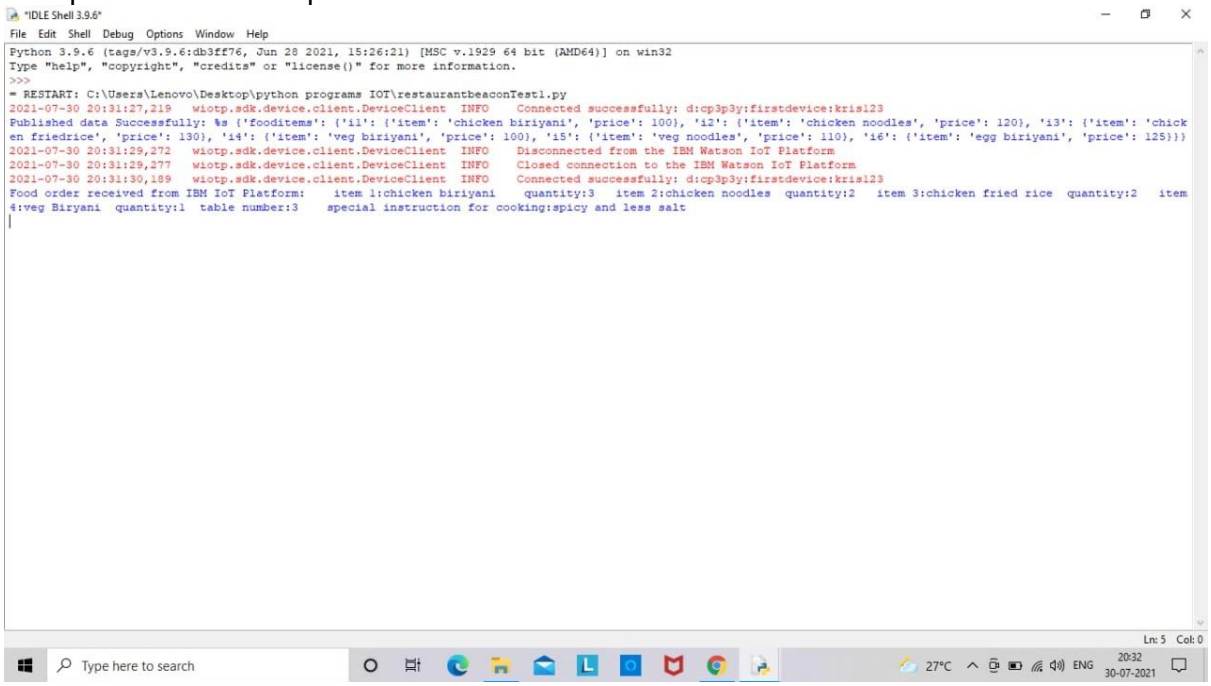
* food order stored in the cloud



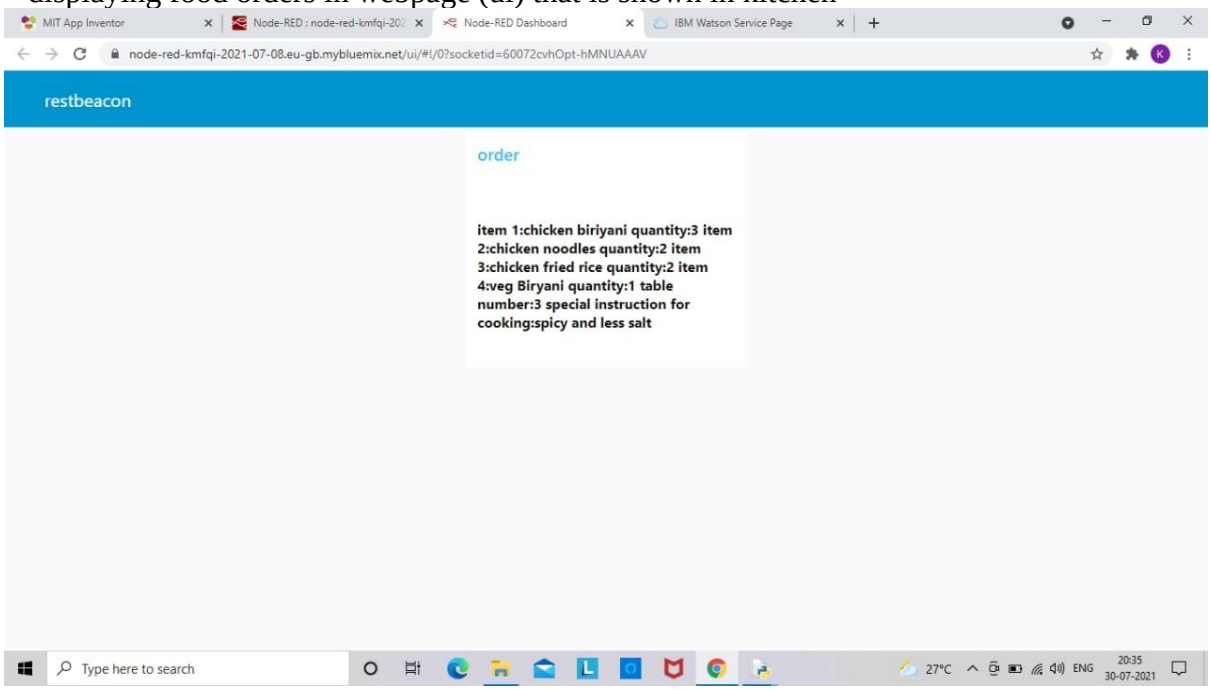
* food orders printed in node red



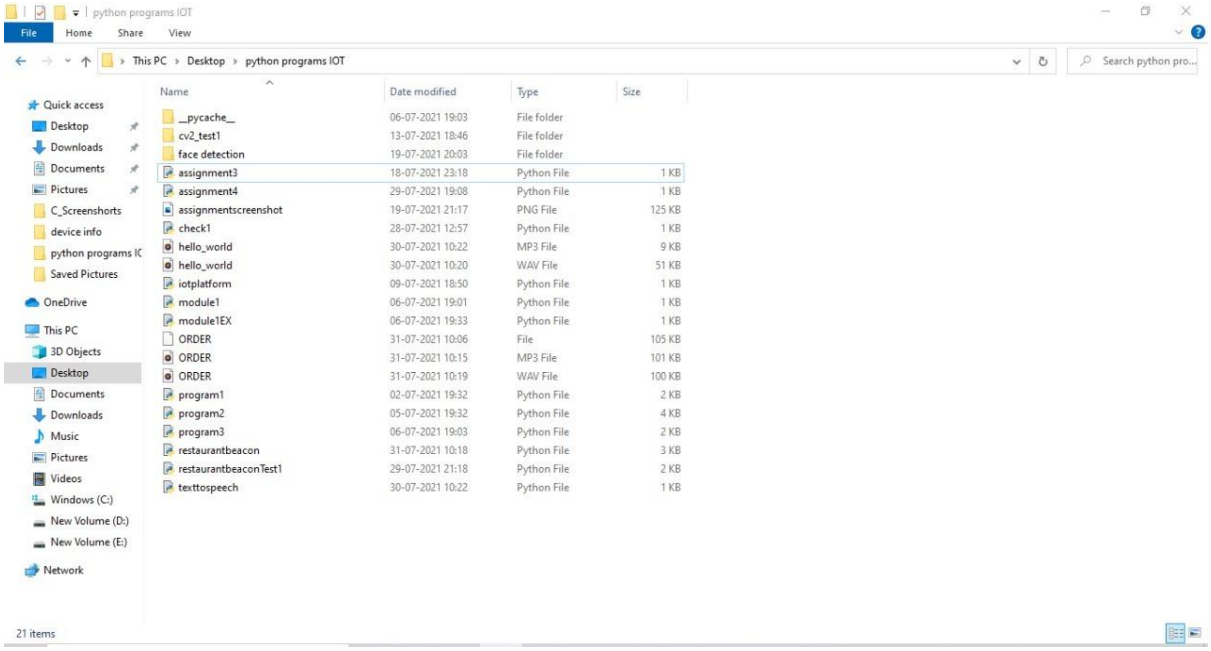
* output of food orders printed in IDLE shell



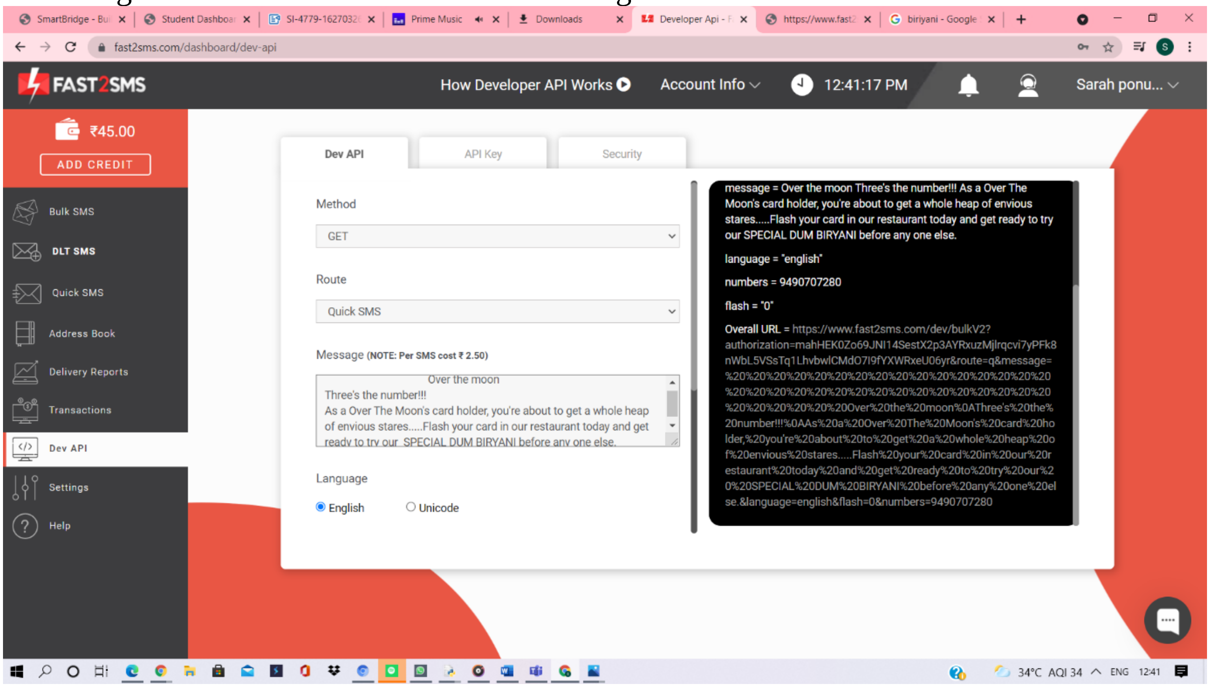
* displaying food orders in webpage (ui) that is shown in kitchen



* voice commands created as audio files with name " ORDER"



6.Sending offer and discount related SMS through FAST2SMS



Message received by user to their mobile

ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1.Adds intelligence to the restaurant by sending Welcome messages and special offers to attract the customers.

2.The beacon broadcaster at each table sends the online Menu card for the customer where they can select the food items and place an order in an efficient way.

3.The orders placed at the tables are displayed in a web app in the kitchen along with voice commands for the chef.

4.saves time since there will be no human errors made as everything is online

5.Data related to orders is stored in the cloud for future references.

DISADVANTAGES:

People need to install an app to be able to experience proximity marketing with beacons. When beacons are not implemented correctly people can get easily annoyed by receiving too many push notifications and may even stop using the app.Whenever there is an issue with the network ,there will be a lag in the app which may not be pleased by the user.

APPLICATIONS :

Beacons not only access us to view digital menu and order online but also has a lot of other applications regarding restaurants.

1. **Check-ins / Check-out**: You can greet your customer with some personalized message as in Push notification to your returning customer to your Restaurant. You can configure to get a feedback form on the Mobile device when they are ready to leave the restaurant.

2. **Reward Points**: . Beacon enable us to make a feature of WireLess check-in & it automatically detects if a user actually in the Restaurant and earn him the reward points on each their visit.

3. **Automate your order processing system**: As you know that by using digital solutions you can optimize your current operations to scale your business. Beacon is a technology which helps at various places to further optimize your processes.

4. **Temperature Beacons**: You can place temperature beacons in your cold storage or where you want to maintain a certain type of necessary temperature. You can get all the temperature

info on your mobile app and it also alarms you if the temperature is over or below required number.

5. **Payment Solution**: Beacon enable technology allows to take payment from cards without the physical handover of the cards. This is more useful at Food takeaways & drive-thrus.

CONCLUSION:

With the help of Beacons, restaurants have the ability to engage customers in a better way through a real-world experience. At the same time, they are also able to understand, analyse, and take necessary actions with regards to consumers needs and interests. Beacon technology could also alert the staff that the customer is nearby and they should begin to prepare the order. This not only allows the restaurant to be quicker and more efficient, but also helps the customer get in and out faster than expected. By leveraging beacon technology to increase consumer churn, restaurants can increase operational efficiencies. Although the technology is still relatively new, restaurants can reap many benefits by being early adopters in this area .Beacon technology has already shown its potential to increase staff and operational efficiency while increasing customer satisfaction and loyalty.

FUTURESCOPE:

Although beacons are popularly considered to be a retailer-focused opportunity, restaurants have a lot to gain from this new proximity-detection technology, as it allows them to have meaningful, personalized conversations with customers. Here are a few ideas on how restaurants can put beacons to use in future:

1. Letting customers order-ahead

Allowing customers to order ahead will help reduce waiting time.at frustrates customers even more, is having to wait for the order when they arrive, even after having ordered ahead via the mobile app. Avoid this by installing a long-range beacon, that will send an alert to kitchen staff once the customer is within a certain distance from the location. This ensures that staff has the order ready.

2.**Let customers know how crowded the restaurant is, before arrival**

Keeping customers informed about how busy a restaurant is at a particular time is one way to allow them to cut down on the time they need to wait.. Deploying beacons around a restaurant will in turn provide real-time information on how many people are actually dining there at a given point in time.

REFERENCES:

https://thesmartbridge.com/documents/projects/SmartHomeAutomationusingIBMCloud.pdf

https://www.retaildive.com/ex/mobilecommercedaily/how-beacon-technology-enhances-the-restaurant-customer-experience

https://www.restaurantindia.in/article/is-beacon-technology-the-next-trend-for-restaurants.12779

https://www.solulab.com/beacons-enhance-restaurant-dining-experience/