INTELLIGENT RESTAURANT WITH SMART BEACONS

NAME: P .SARAH
REGISTRATION NUMBER: 19BEC0532

INTRODUCTION

OVERVIEW:

With Food and beverages comprising 17.1% of the retail business alone, technology has given a new meaning to the technological advancement that's happening at the restaurant front.Globally, technologies like beacon technology is a mega hit at restaurants. **Beacons** are small, battery-operated wireless devices that transmit coded messages to nearby paired smart-phones using **Bluetooth Low Energy (BLE)** which have helped in engaging guests and increasing repeat visits. **Customised mobile app** can show the food menu on user's smartphone as soon as he reaches your outlet and whenever your customer place orders from his mobile, it inform kitchen staff the table number along with the food order and storing the order placed for further reference. This feature makes the process more efficient and reduces the requirement of service peoples in the restaurant.

PURPOSE:

* Marketing messaging
* More customised experience for the customers
* Delivering a tailored digital food menu to customers
* Food Ordering from Dine-In Table which will be displayed in the IoT device
* Storing the food order in the cloudant for further reference

LITERATURE SURVEY

EXISTING PROBLEM

A waiter often makes errors while the note making the order or sending the order to the kitchen. Writing wrong table number,wronginstructions or some specific orders cause delays of the food and make customer having unsatisfied experience. Manual order placing is time consuming and is complex both for the customer and the staff.

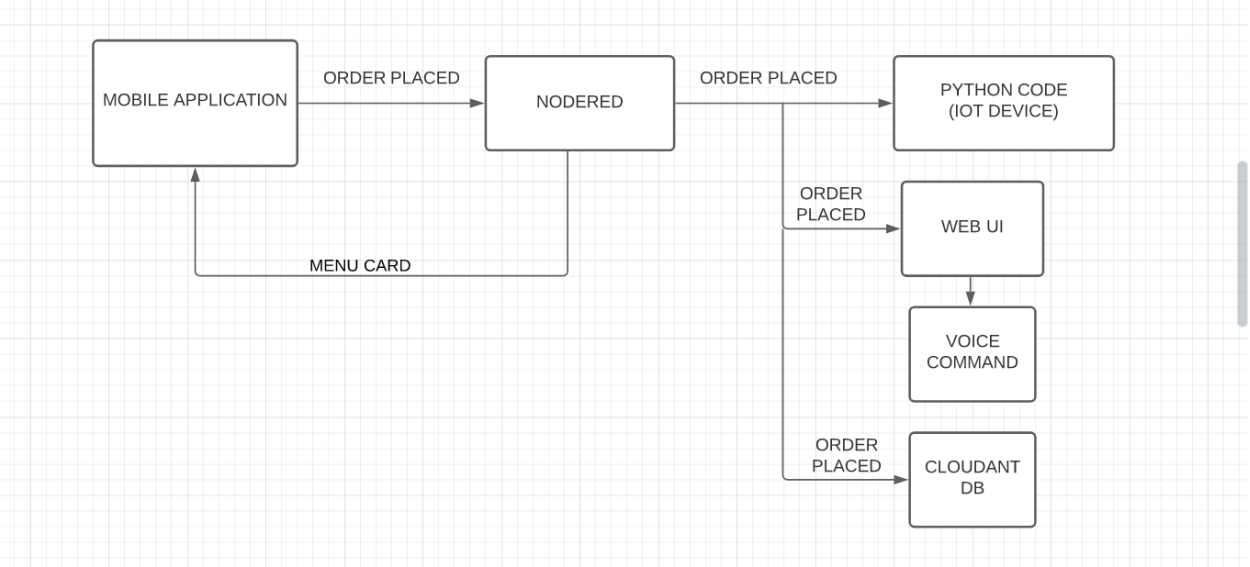
PROPOSED SOLUTION

A good order taking mechanism in a restaurant ensures speedy delivery of orders that is accompanied by efficiency.We propose integrating restaurents with smart beacons for a smart and easy way of food ordering .A centralized restaurant ordering system with

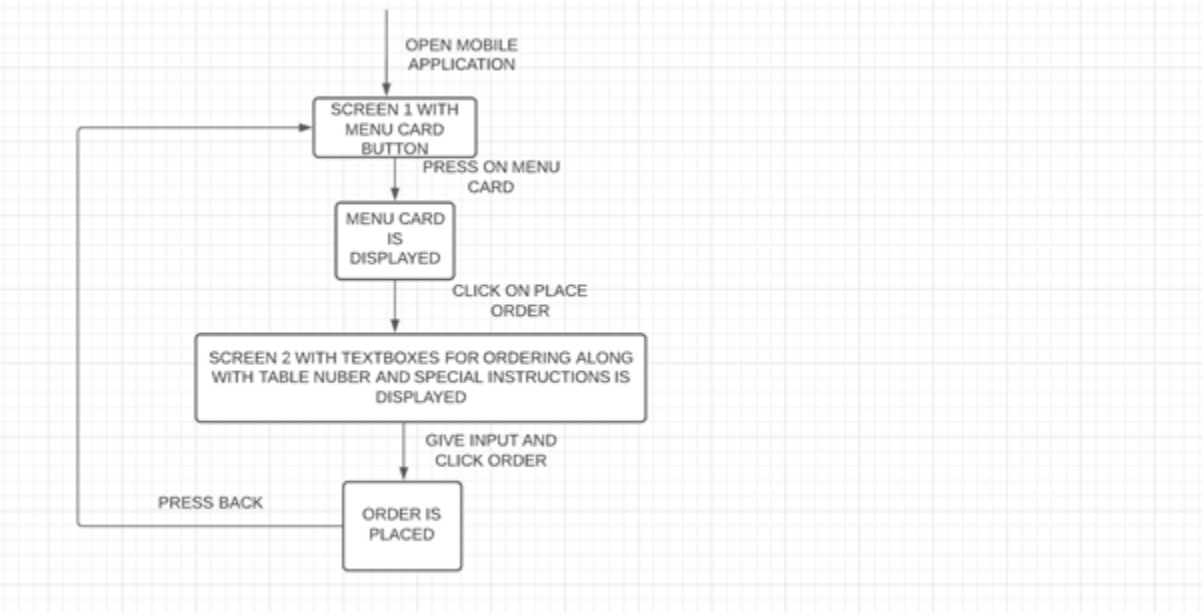
beacons **decreases** the chances of **manual errors**.Taking Orders Via **Restaurant's Website/ Mobile App** is the most **efficient** way. All the orders received online are pushed in the system directly. The restaurant attendants have access to the entire menu and table numbers on their screens.The orders placed are displayed in the kitchen screens using web application and voice commands of the order are given which makes it easy for the chef **.Marketing messages** are sent to users mobile as soon as they enter the restaurent premisis to increase the chance of the visits from the customer.

THEORITICAL ANALYSIS

BLOCK DIAGRAM ON FLOW OF THE PROJECT



BLOCK DIAGRAM SHOWING THE WORKING OF MOBILE APPLICATION



HARDWARE/SOFTWARE DESIGNING

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:

Create a device in IBM Watson IoT platform

Create Node-red application

Develop a python code to subscribe and get the data (menu details) from the IBM IoT Platform and also develop a python code to generate voice commands.

PYTHON CODE:

```
import wiotp.sdk.device
import time
import random
myConfig = {
   "identity": {
      "orgId": "glif1g",
```

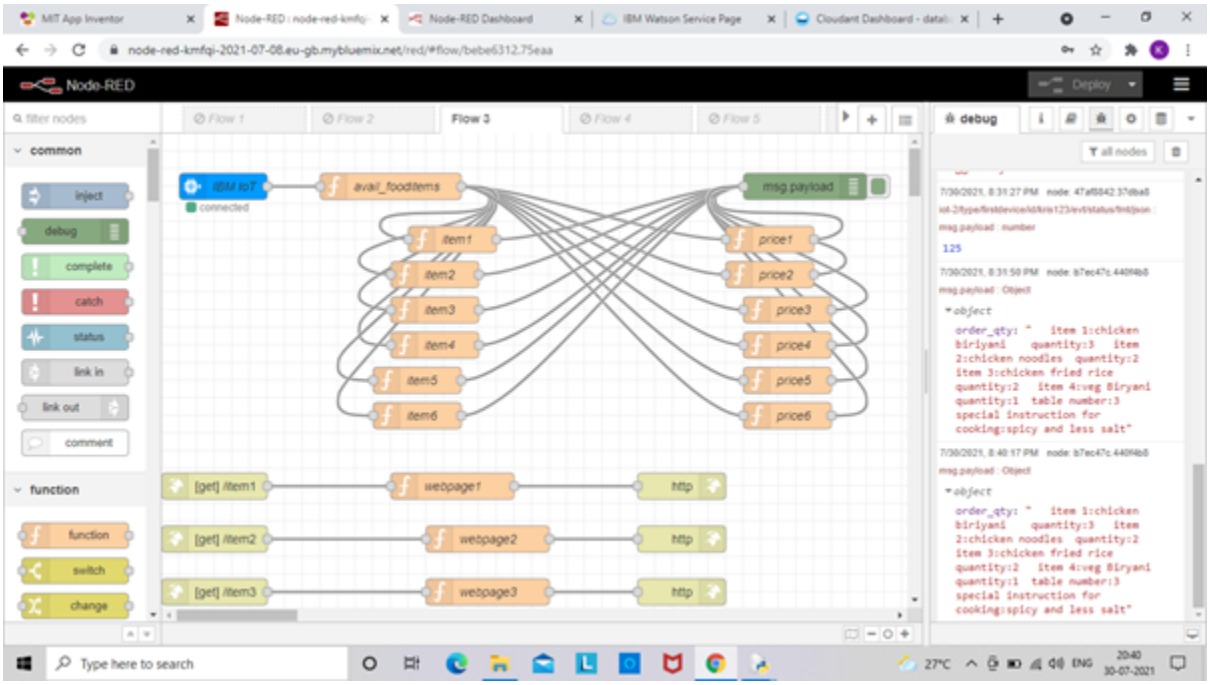
```
        "typeId": "sarahdevice",
        "deviceId":"060801"
    },
    "auth": {
        "token": "06082001"
    }
}
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
import playsound

authenticator = IAMAuthenticator('PjC0bvAGJ1nFseEZGlUNUfZjO9ntqUkrYdFwS065OWVa')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)

text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instances/761a9
9f7-59be-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)
#sending the data to IBM IoT platform which will store the menu card in the cloudant
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
    break
client.disconnect()

def myCommandCallback(cmd):
#receiving the data from the IBM IoT platform which will be recieving the order placed from the MIT App
```

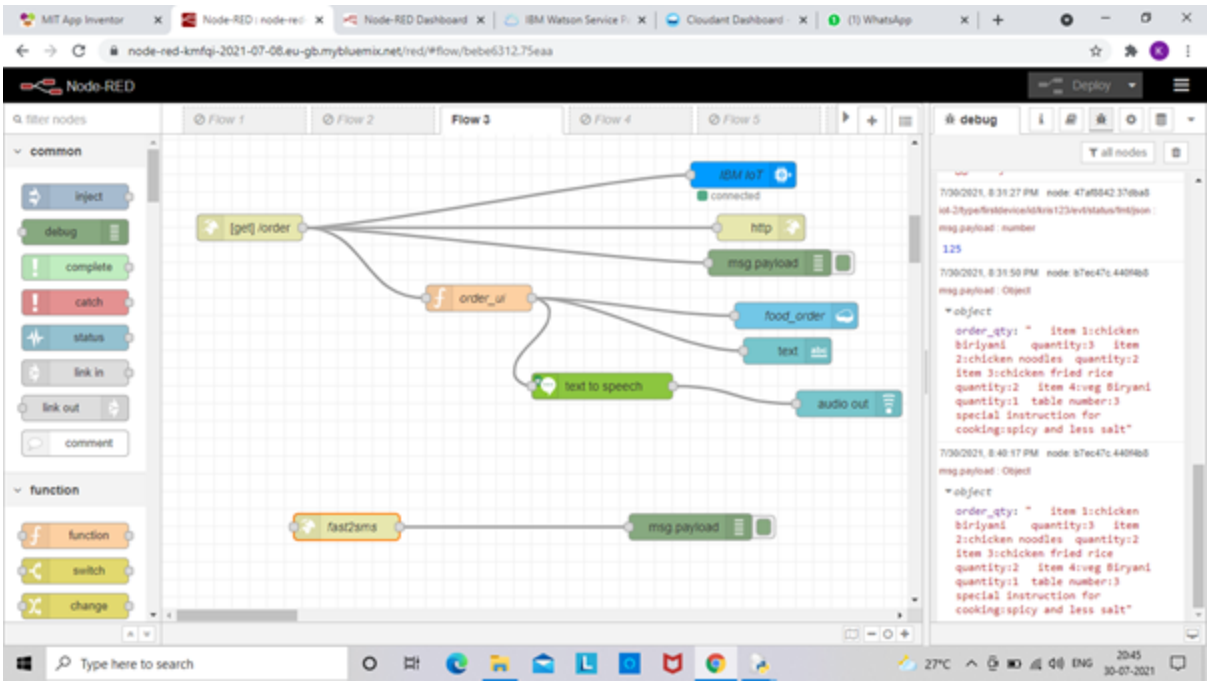
```
print("Food order received from IBM IoT Platform: %s" % cmd.data['order_qty'])
with open('order.mp3', 'wb') as audio_file:
#voice command of the order placed
   audio_file.write(
      text_to_speech.synthesize(
         cmd.data['order_qty'],
         voice='en-US_AllisonV3Voice',
         accept='audio/mp3'
      ).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()
```

Building web application

➤ `Node-RED flow to send the data of menu to the MIT app and recieve the order placed from the application back to Node-RED which is again send back to IoT device(python code) and will be stored at the cloudant database.

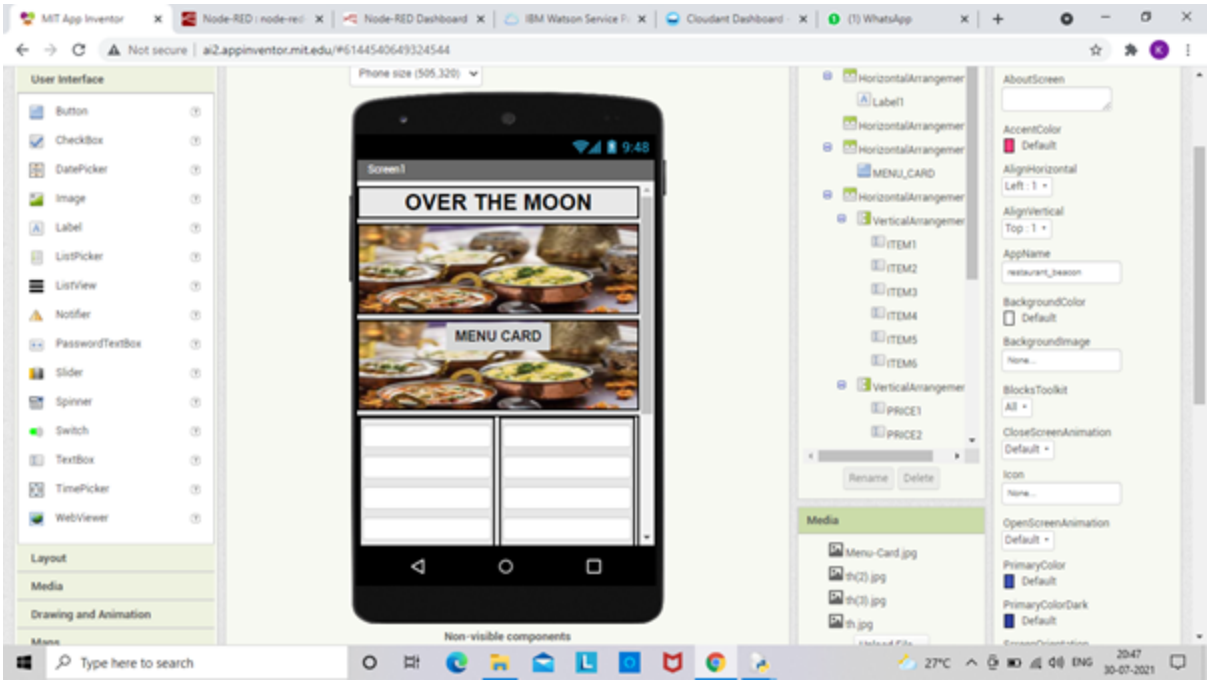
➤ Create a web UI to display the orders in the kitchen along with voice command

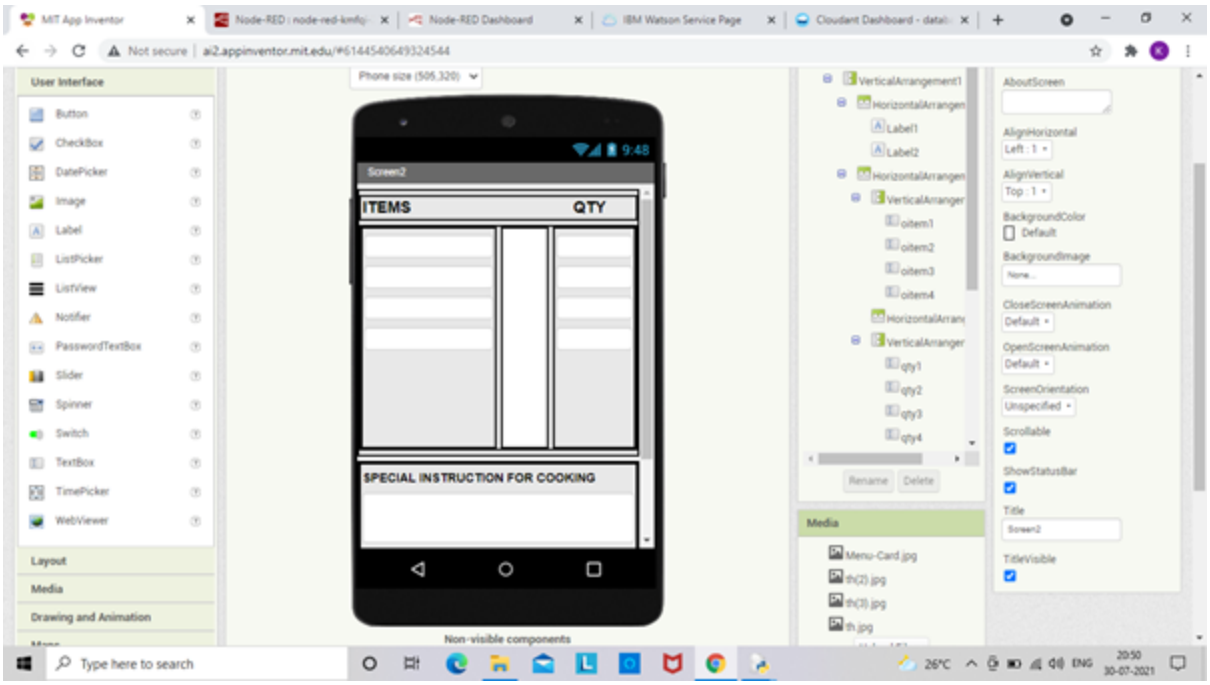




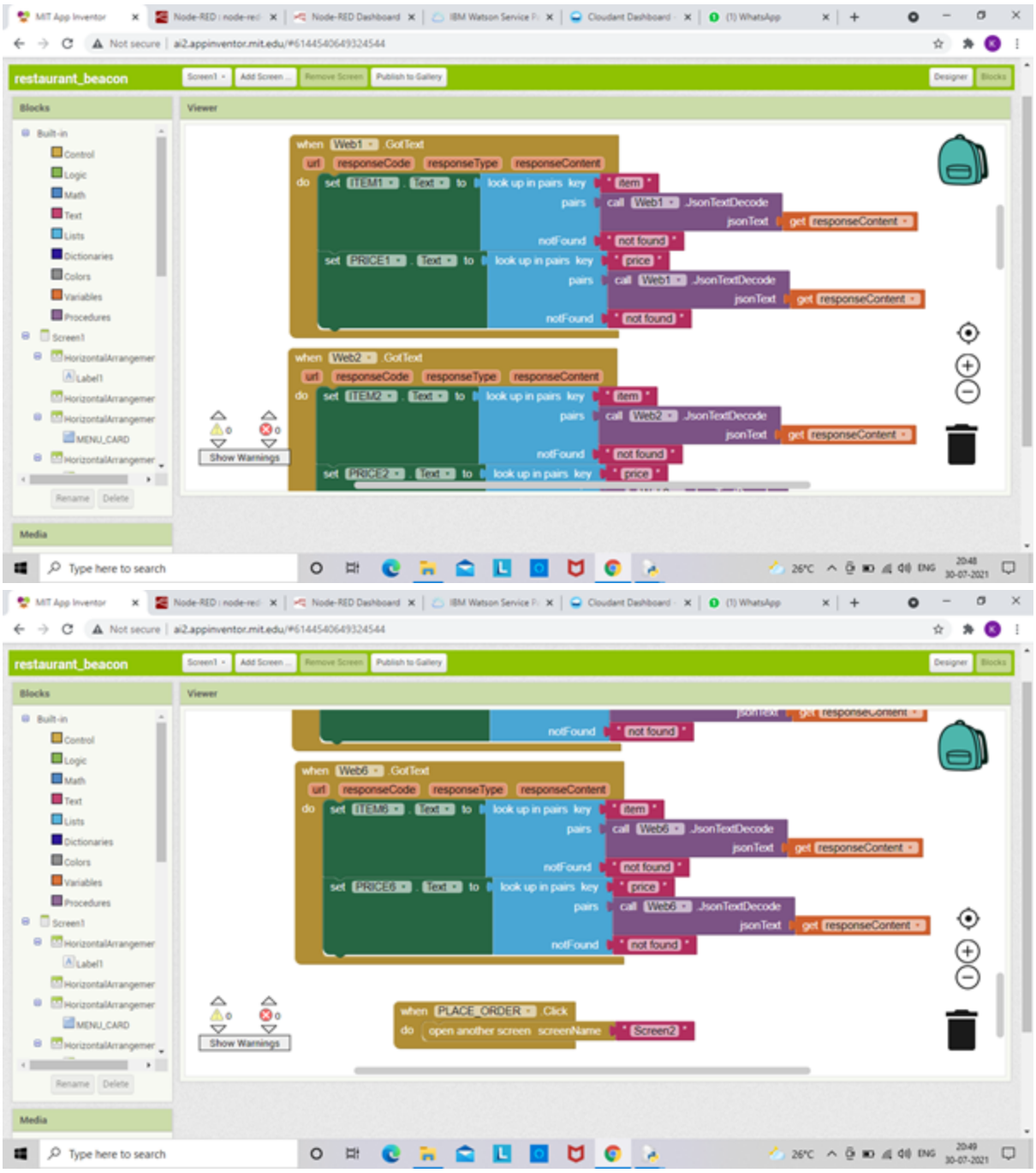
Create a mobile application to display the received menu and also integrate a text box to take order from user.

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

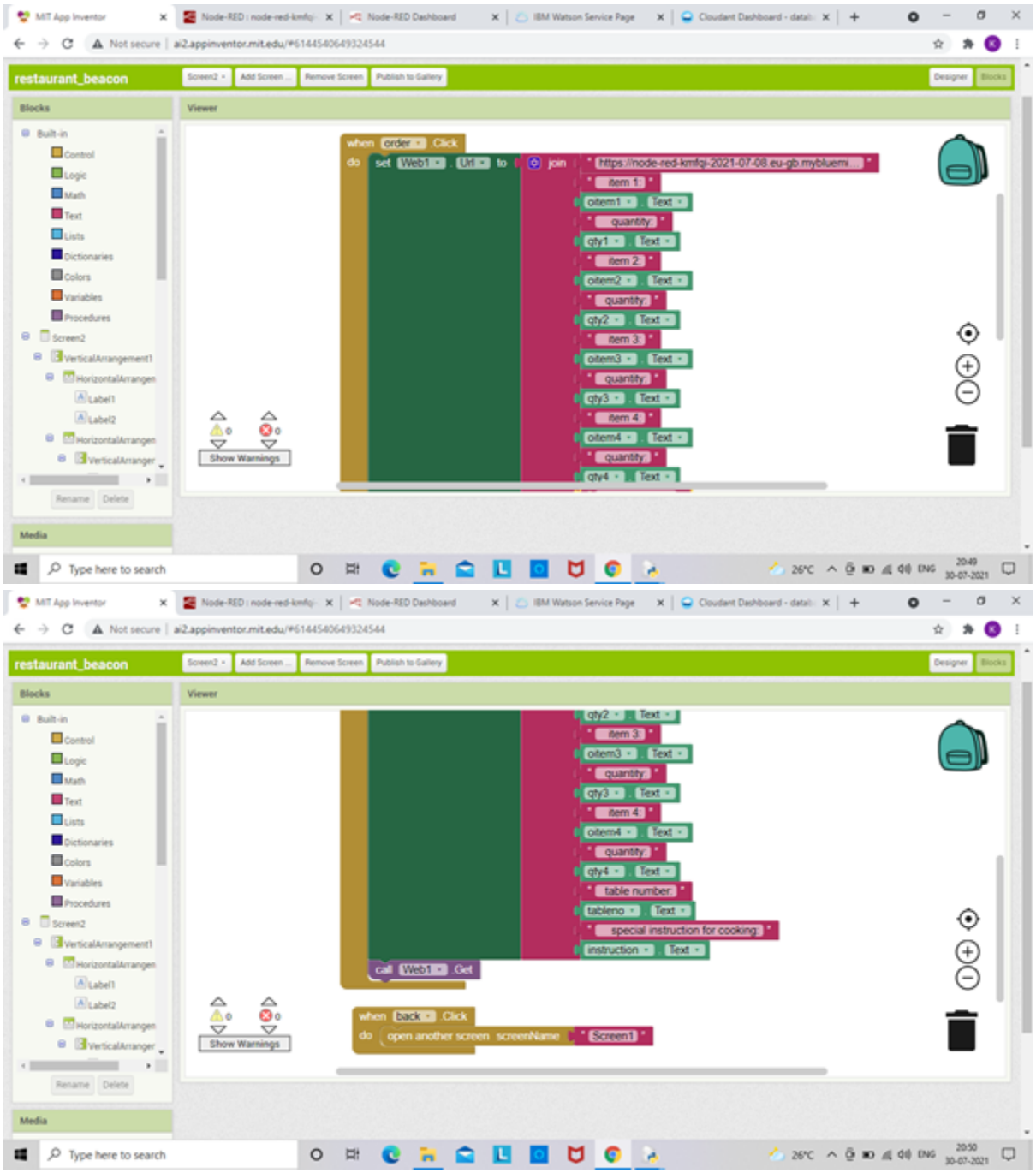




Blocks for the above designed mobile application
screen 1(for showing the menu card)



blocks for screen 2(for placing the order)



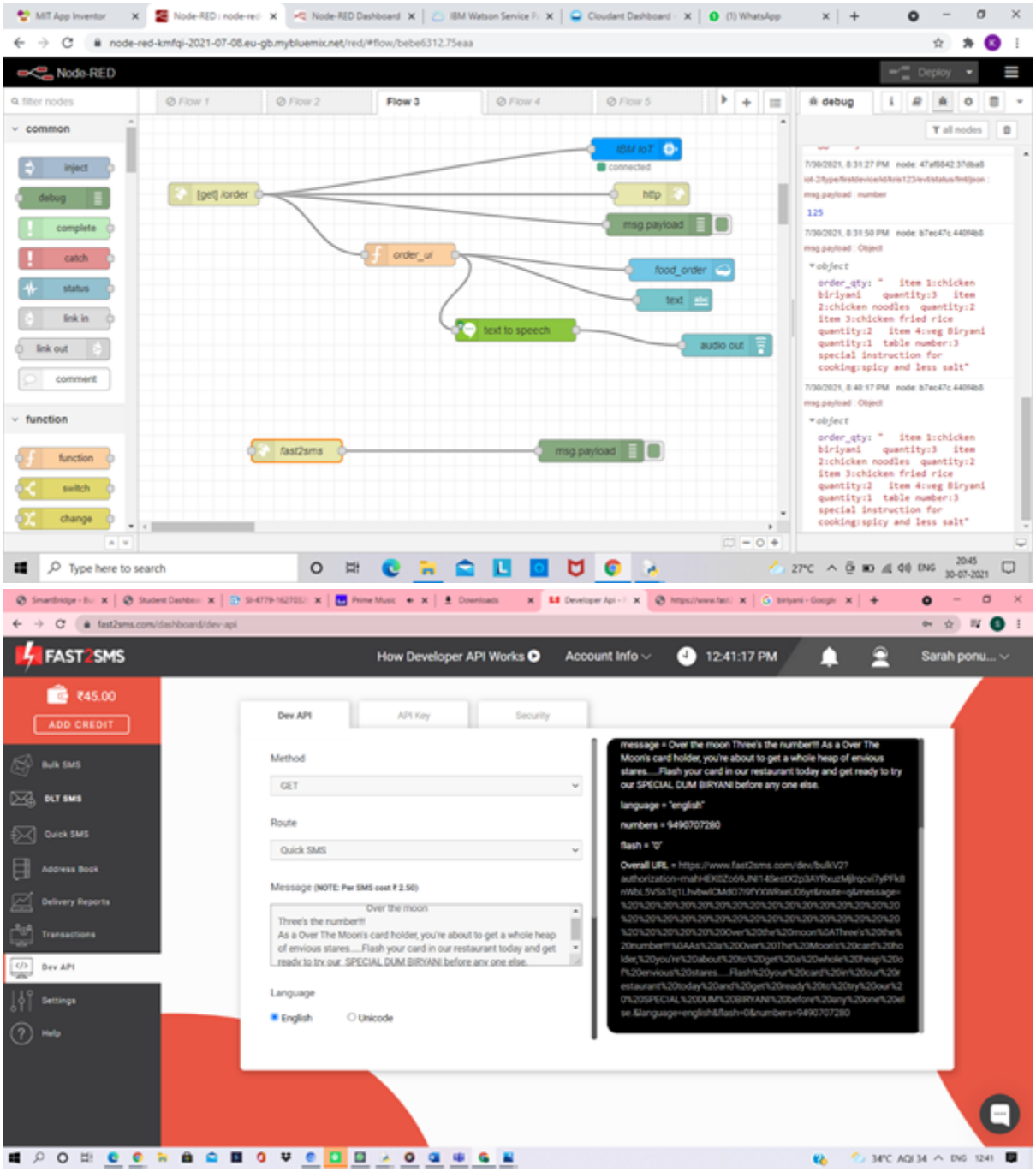
Configure the application to receive the data from the cloud and to send the menu details:

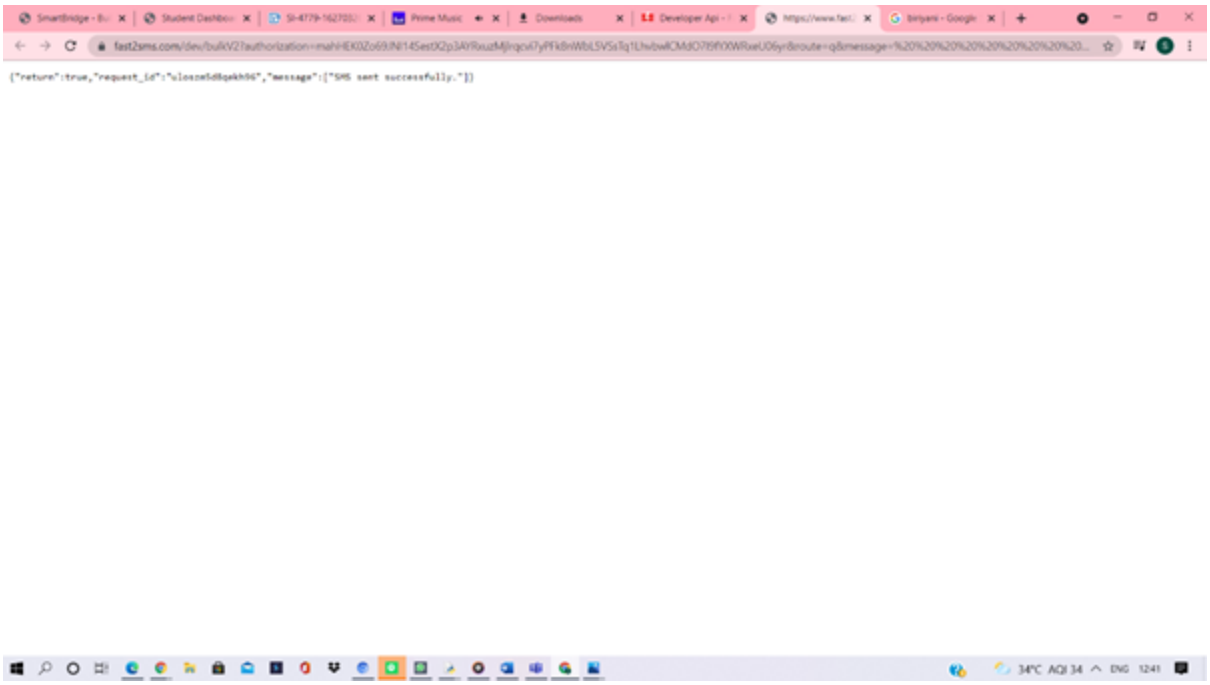
To get the data from the webpage, we add the WEB component from the Connectivity which is present in the Palette

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

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

Sending offer and discount related SMS through FAST2SMS

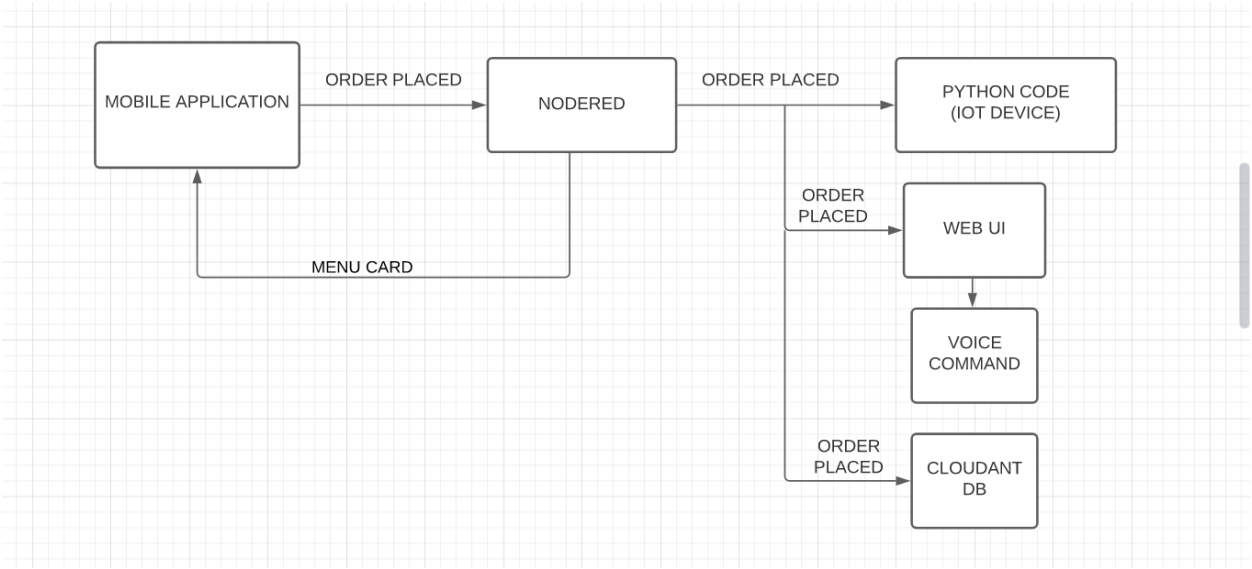




EXPERIMENTAL INVESTIGATIONS:

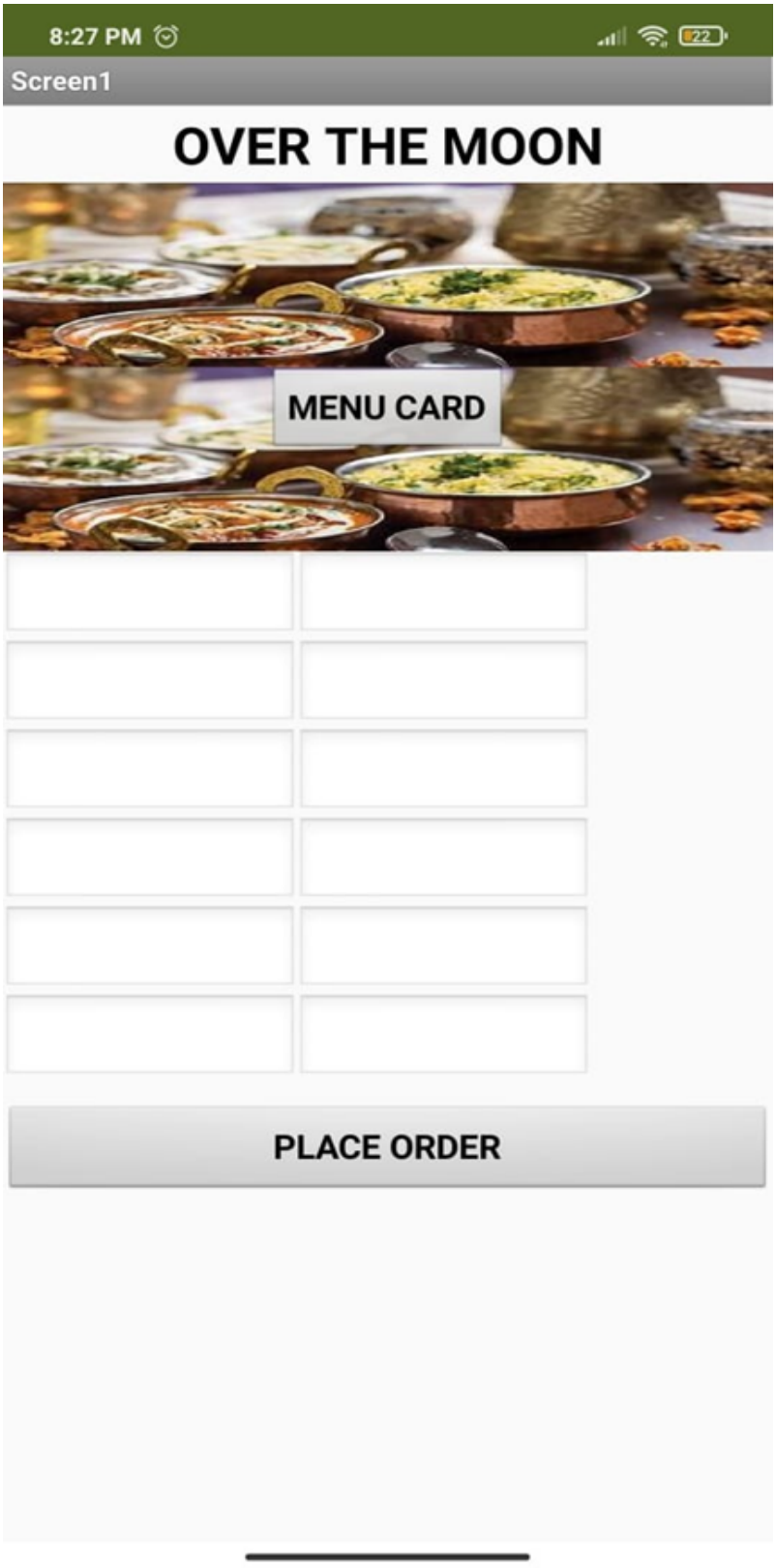
We did a brief research on smart beacon to know their basic functioning and working. We learned how to integrate Node-Red with MIT app as well as the IoT device(python code) to send and recieve data between eachother. We also did a research on how to store and recieve the data from the cloudant database. We also worked with text to speech services to send voice commands.

FLOWCHART:

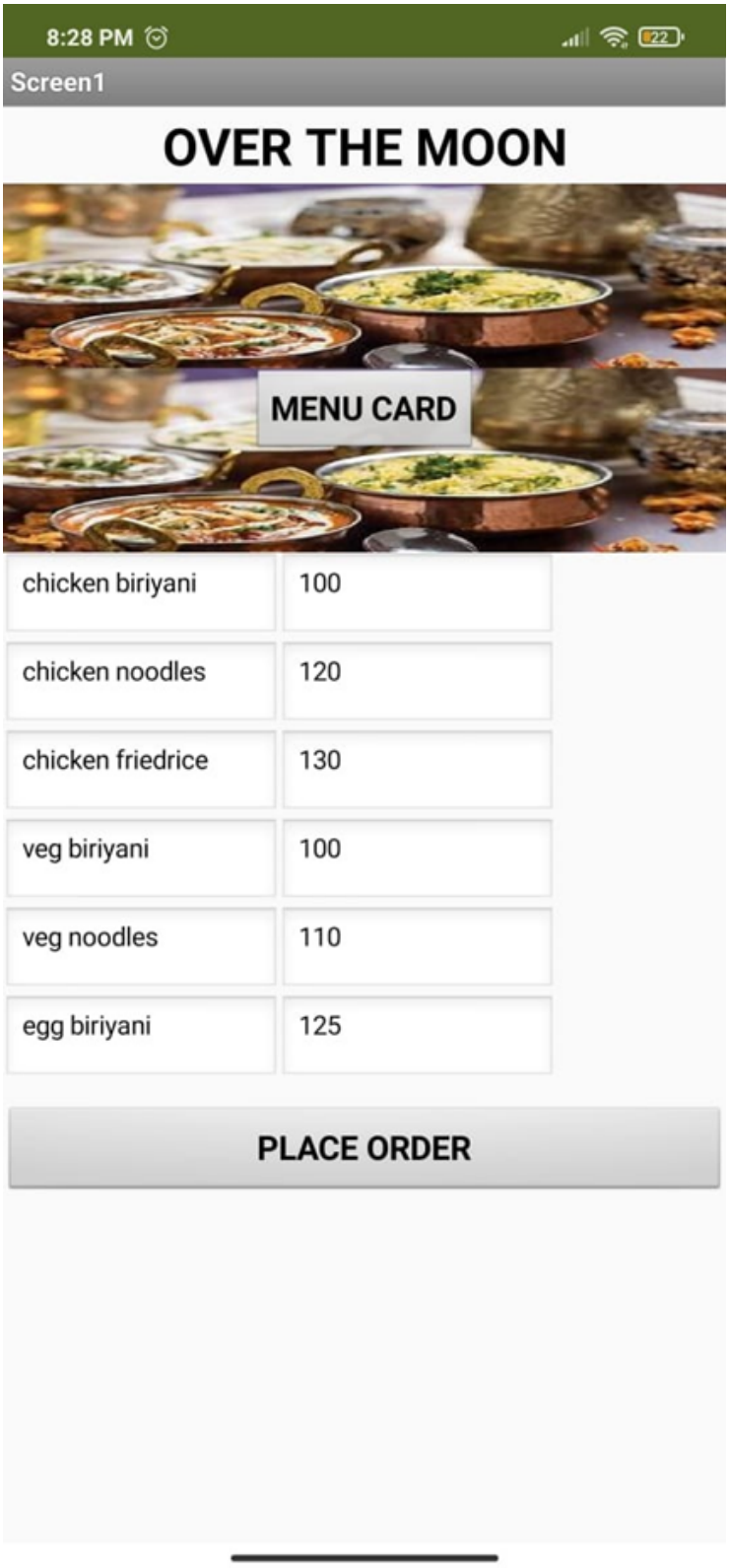


RESULTS:

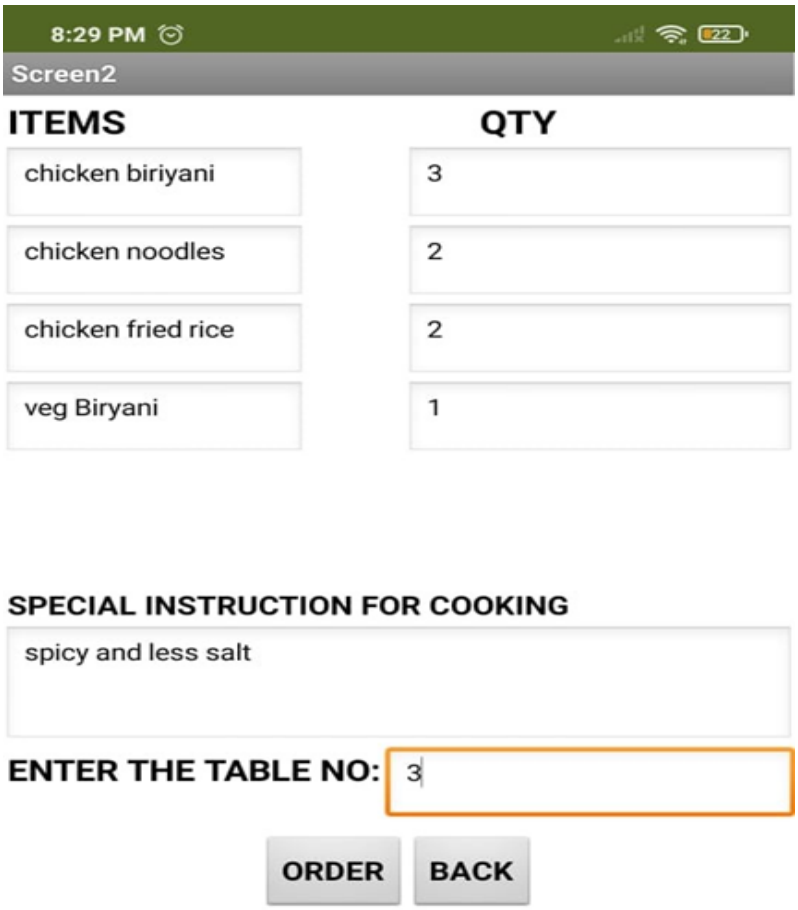
When the application is opened on the mobile.The below screen will be visible.



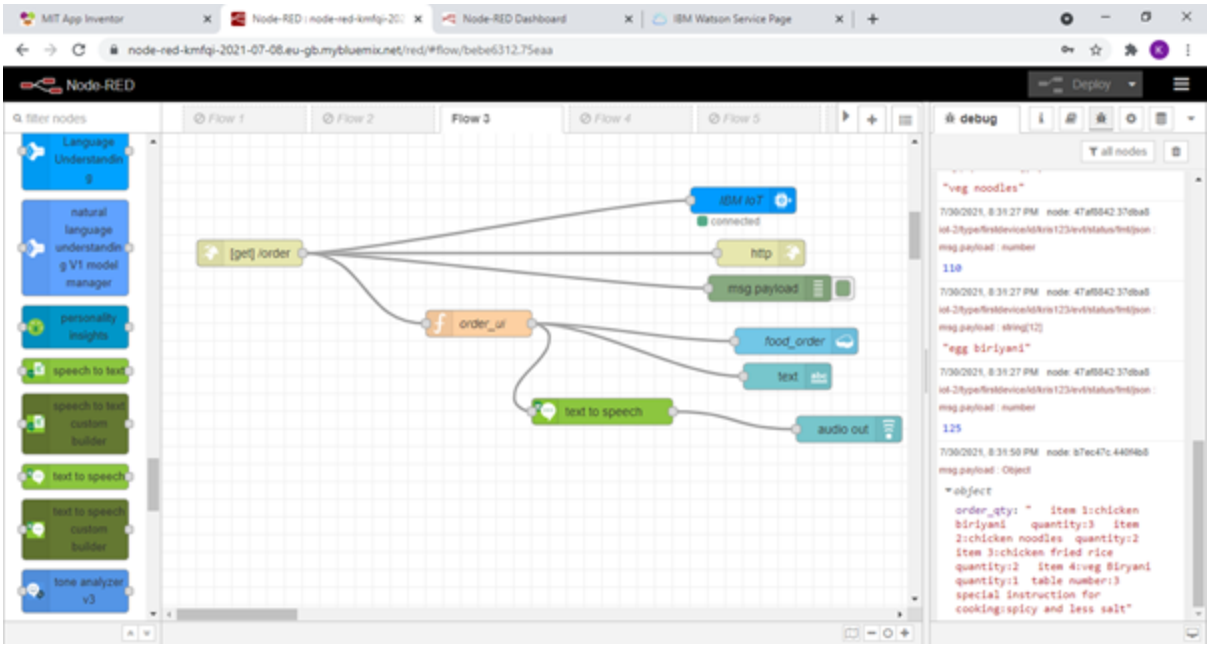
The Menu card is displayed when the user selects the option "MENU CARD"



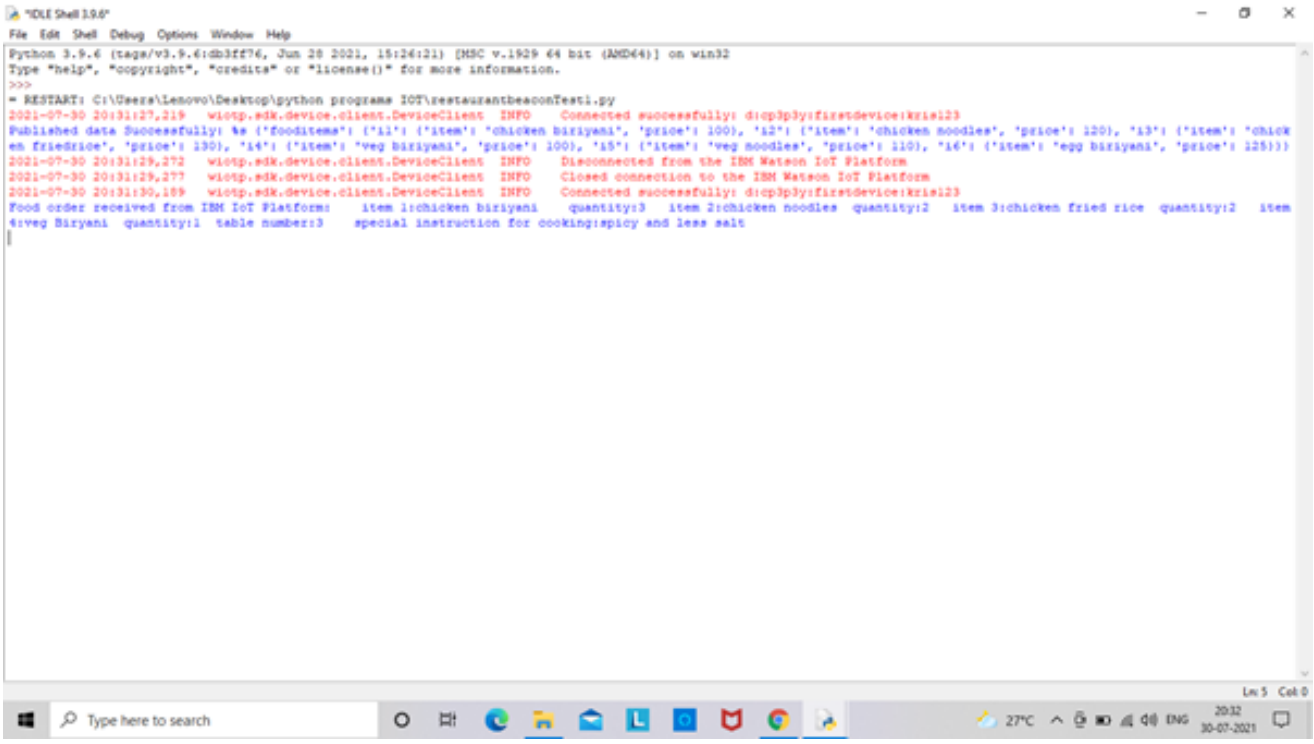
When the user presses the "PLACE ORDER" button, it is redirected to screen 2.
here, the user can give the order along with the inputs of table number and special instructions. The order will be place once the button "ORDER" is clicked.
if the user presses "back" button, it is redirected back to screen1



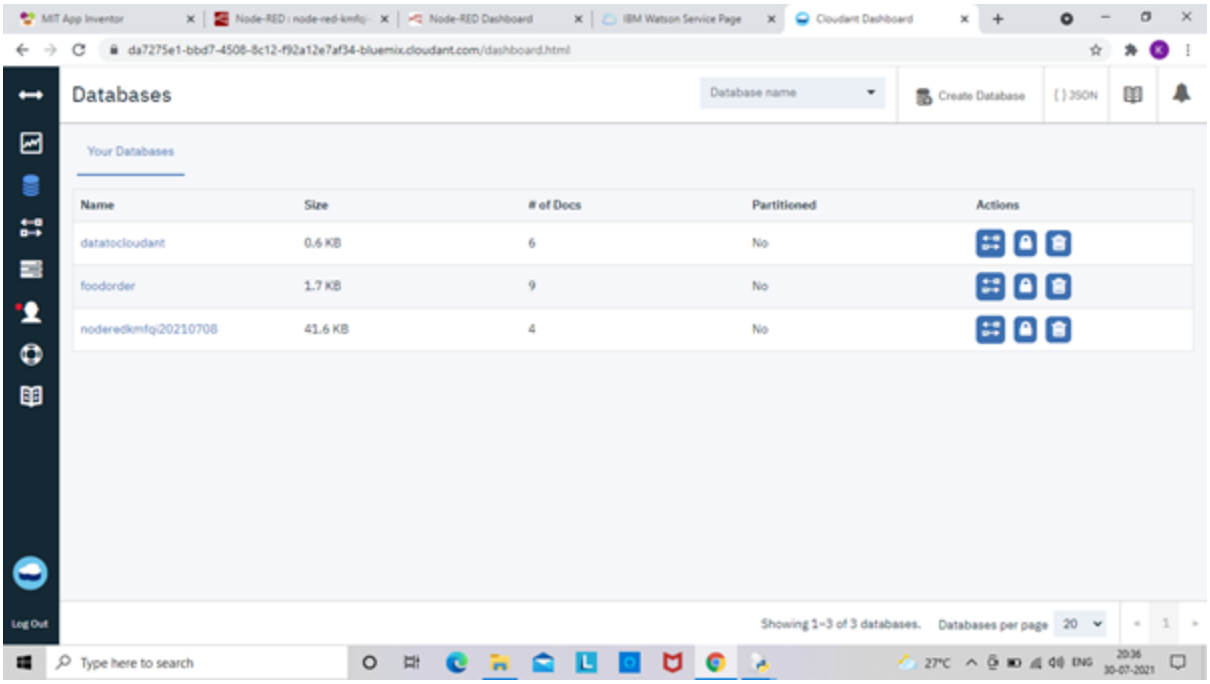
food orders printed in node red

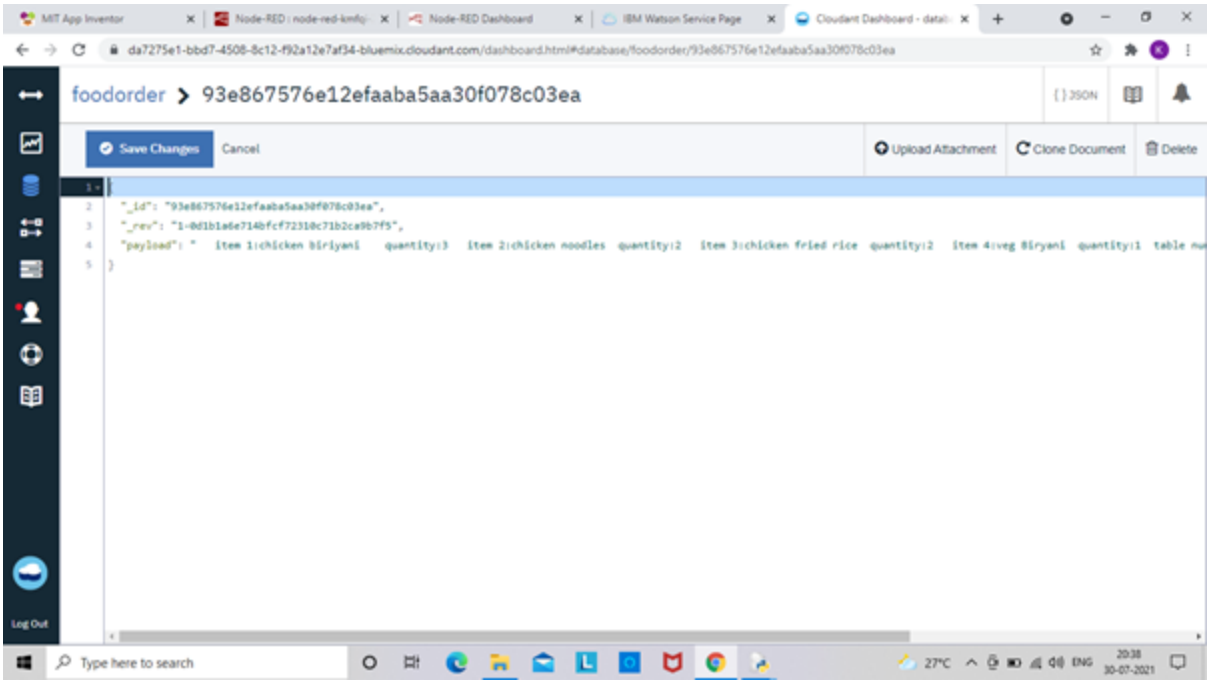


output of food orders printed in IDLE shell



orders stored in the cloud





Marketing Message recieved by user to his/her mobile

ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

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

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.

The orders placed at the tables are displayed in a web app in the kitchen along with voice commands for the chef whichsaves time since there will be no human errors made as everything is online.

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 RESTAURENTS

Check-ins / Check-out: You can greet your customer with some personalized message.

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.

Automate your order processing system

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

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.

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:

Letting customers order-ahead

Let customers know how crowded the restaurant is, before arrival

BIBILIOGRAPHY:

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/

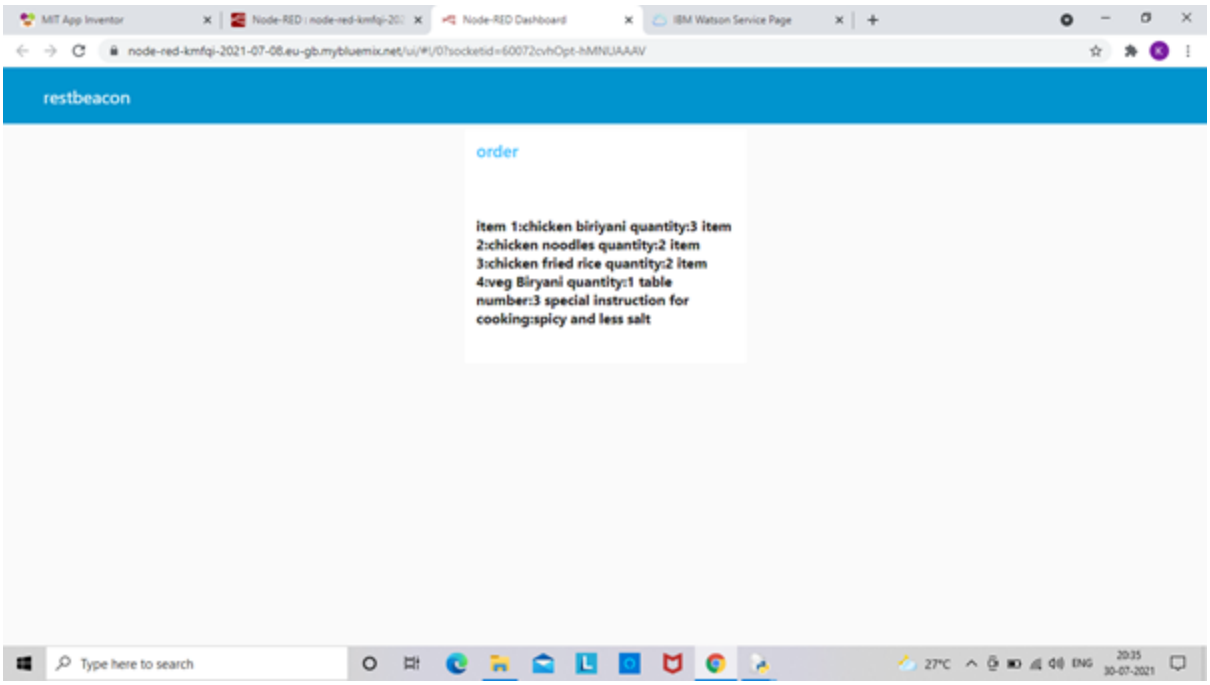
APPENDIX:

source code:

```
import wiotp.sdk.device
import time
import random
myConfig = {
   "identity": {
      "orgId": "glif1g",
      "typeId": "sarahdevice",
      "deviceId":"060801"
   },
   "auth": {
      "token": "06082001"
   }
}
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
import playsound

authenticator = IAMAuthenticator('PjC0bvAGJ1nFseEZGlUNUfZjO9ntqUkrYdFwS065OWVa')
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-d
dce29442f5f')


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)
#sending the data to IBM IoT platform which will store the menu card in the cloudant
   print("Published data Successfully: %s", myData)
   client.commandCallback = myCommandCallback
   time.sleep(2)
   break
client.disconnect()

def myCommandCallback(cmd):
#receiving the data from the IBM IoT platform which will be recieving the order placed from the MIT App
print("Food order received from IBM IoT Platform: %s" % cmd.data['order_qty'])
with open('order.mp3', 'wb') as audio_file:
```

```
#voice command of the order placed
   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()
```

UI output screenshot



Displaying placed food order in webpage (ui) that is shown in kitchen along with voice command.