INTELLEGENT RESTAURANT WITH SMARTBEACONS

1 INTRODUCTION

Technology has been the front-runner in driving engagement and experience in consumers' shopping journey. 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).
Restaurants will use the technology in a different way than retailers, engaging guests and increasing repeat visits.

2 LITERATURE SURVEY

There are existing problem like there can be mistake in taking order, or simply manual error. so in order to avoid such problems we can use smart beacons develop app. making entire process digital/online is our purpose behind this project.

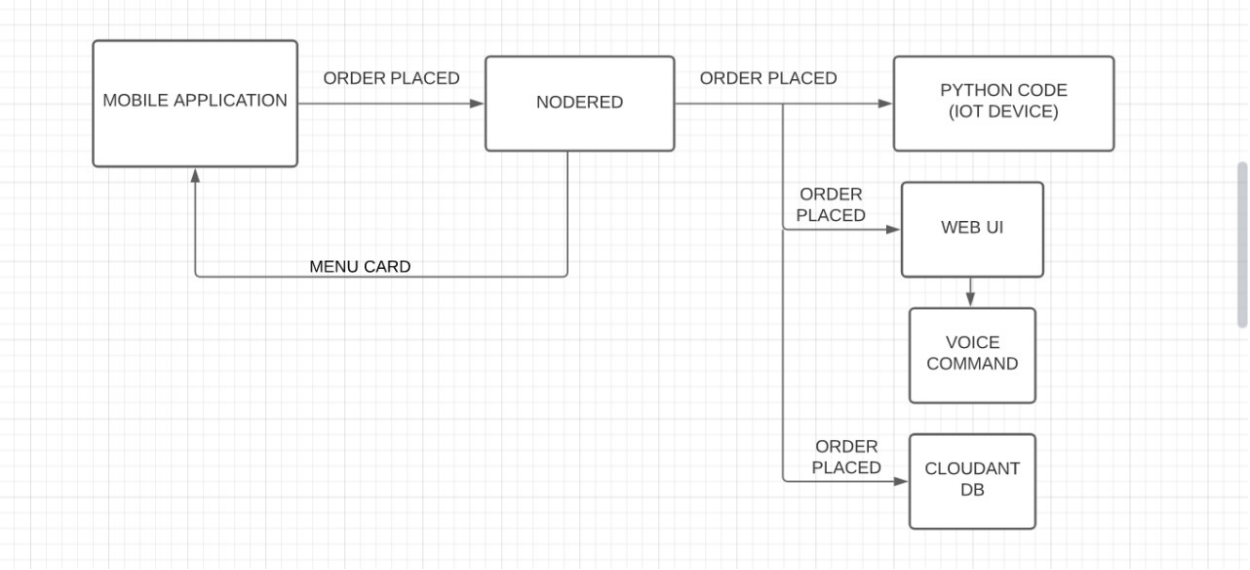
3 THEORITICAL ANALYSIS

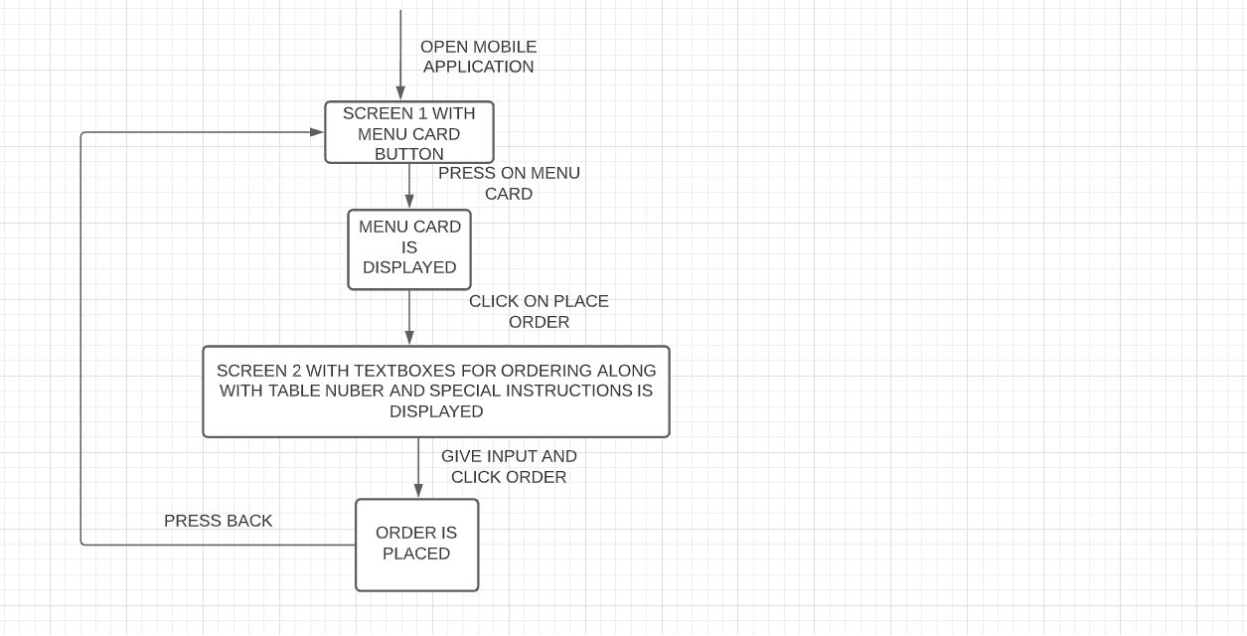
When using manual way of taking order in restaurant more time is taken and also mistake can be done while taking order. If we use smart beacon to convert out restaurant to intellegent restaurant which has facility to order food using mobile app. which will reduce chances of error and also the order will reach directly tp chef so time will also be saved side by side.

4 EXPERIMENTAL INVESTIGATION

For our project on "intellegent restaurant using smart beacons" we have done research on smart beacons, then how to connect node red, then creating app using MIT app inventor, using cloudant database, and how to connect them all to get the desired output.

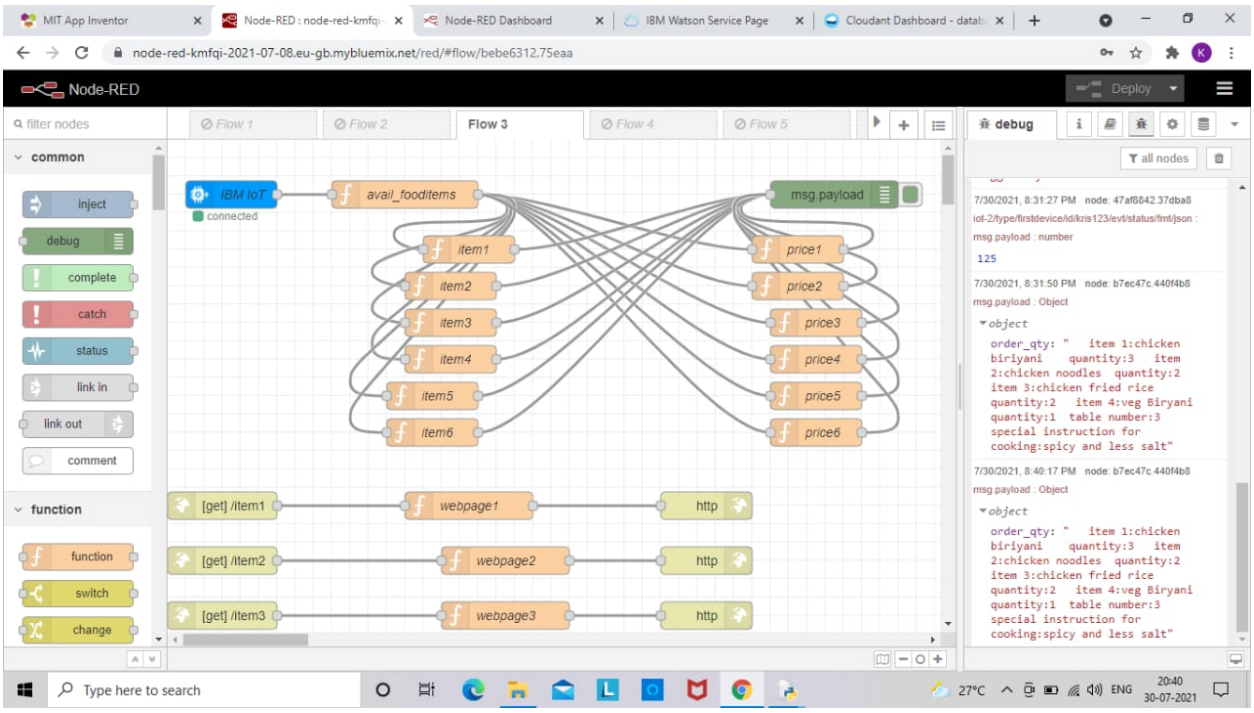
5 FLOWCHART

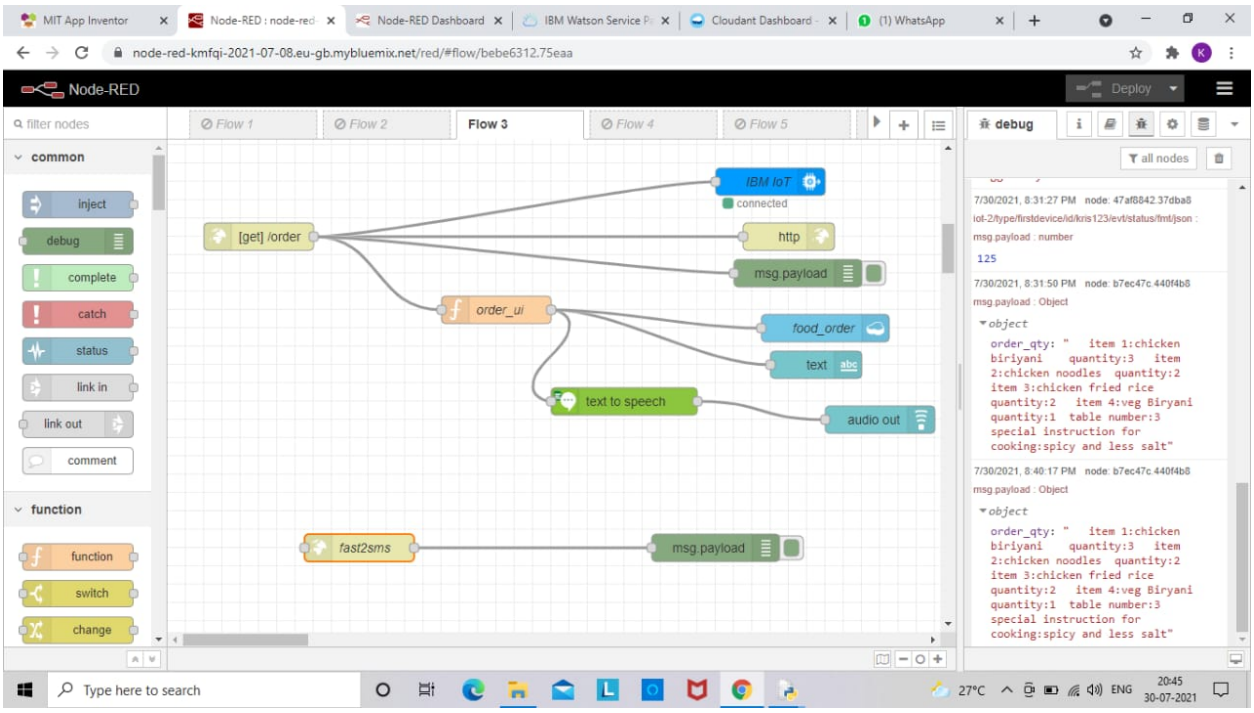


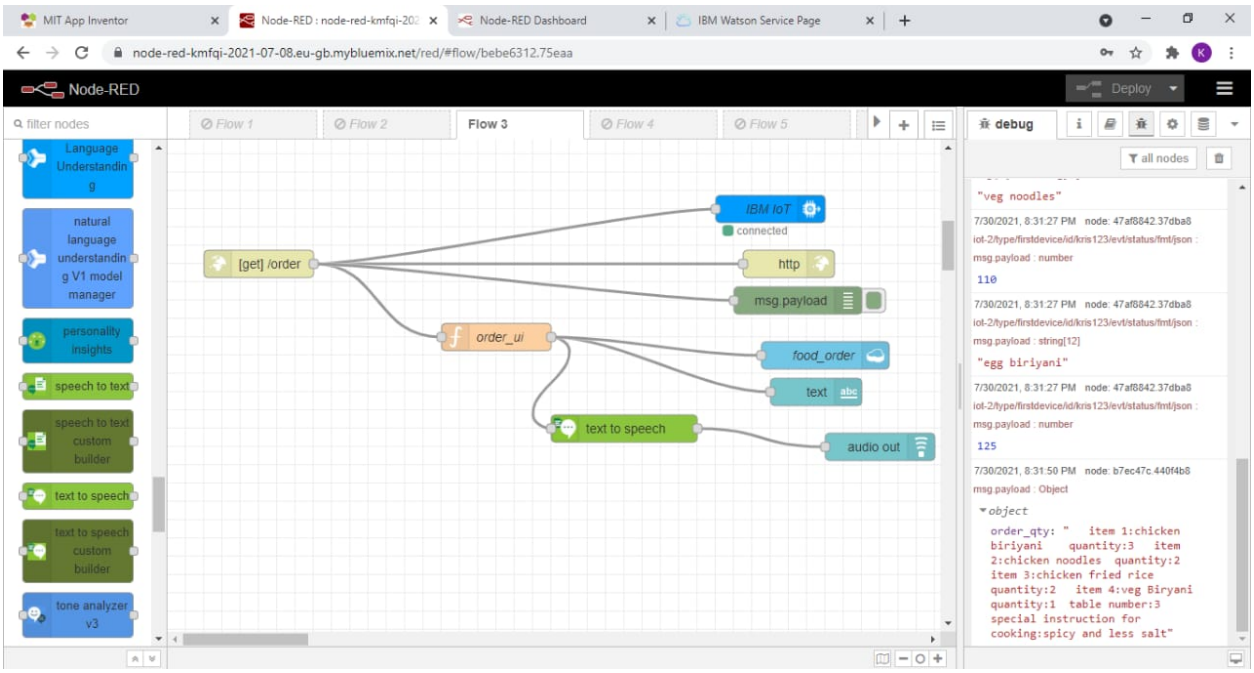


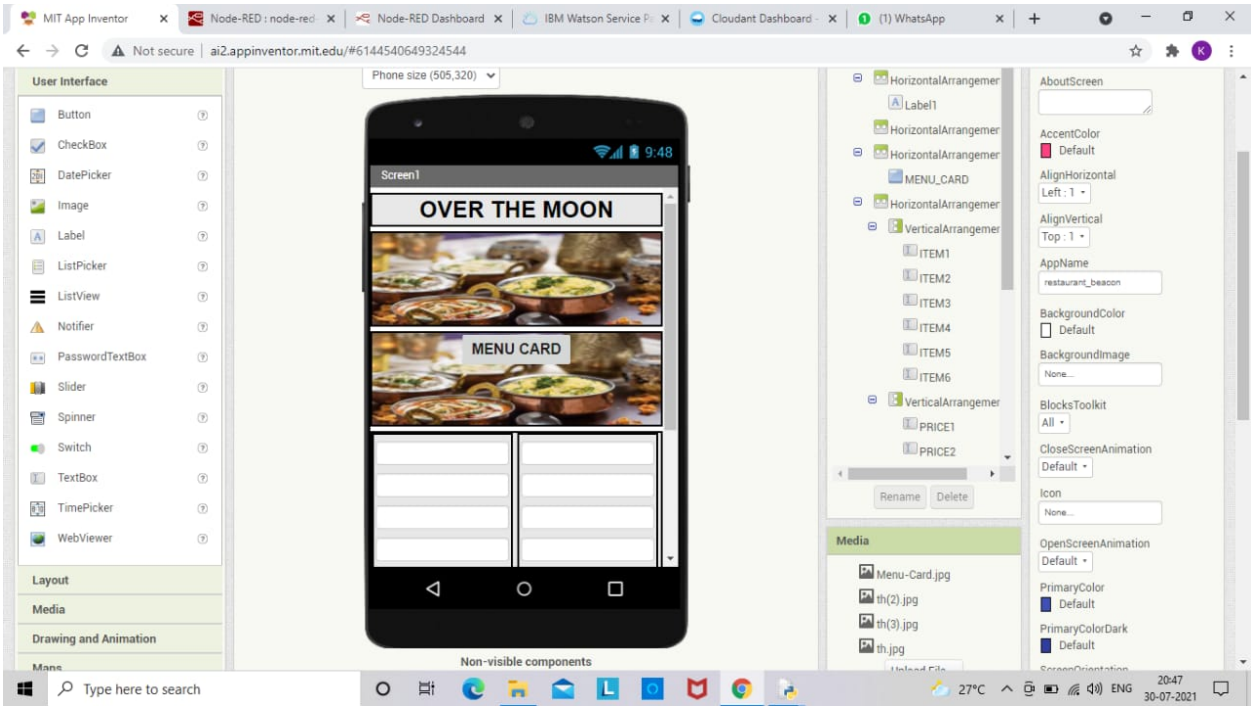
6 RESULT

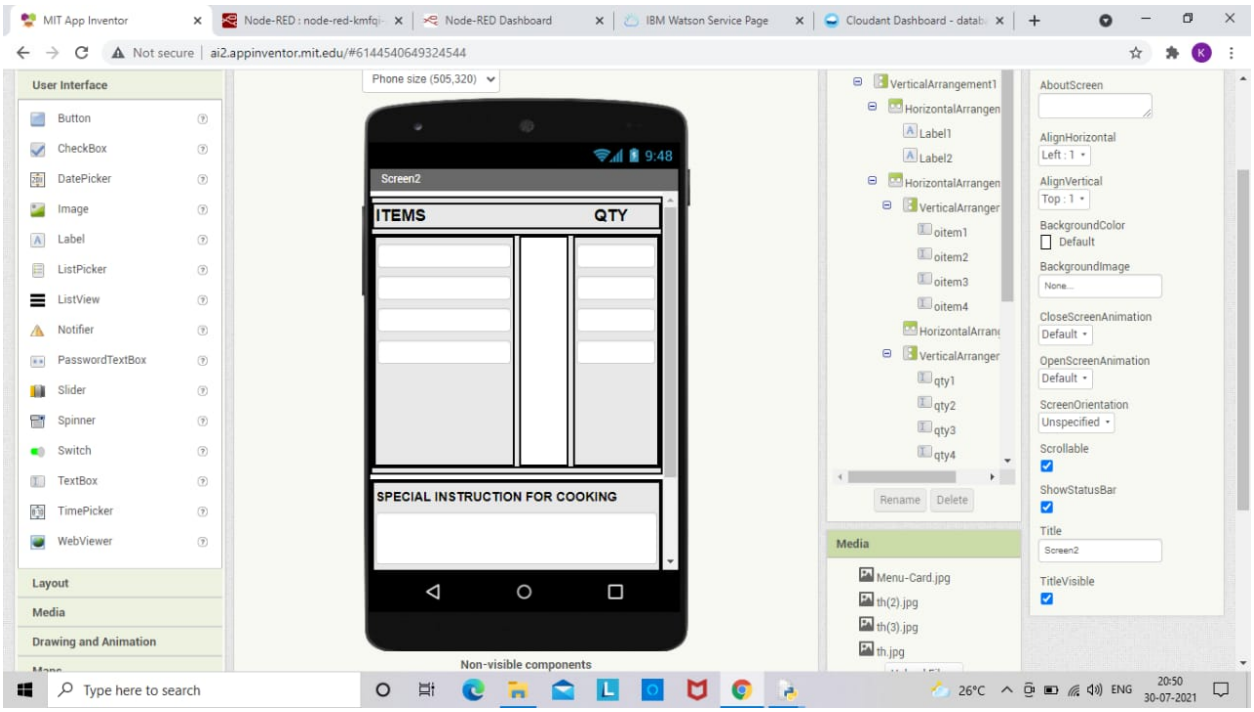
We have successfully completed our project on intellegent restaurant with smart beacons. I am attaching the screenshots related to various platforms used and the output obtained.

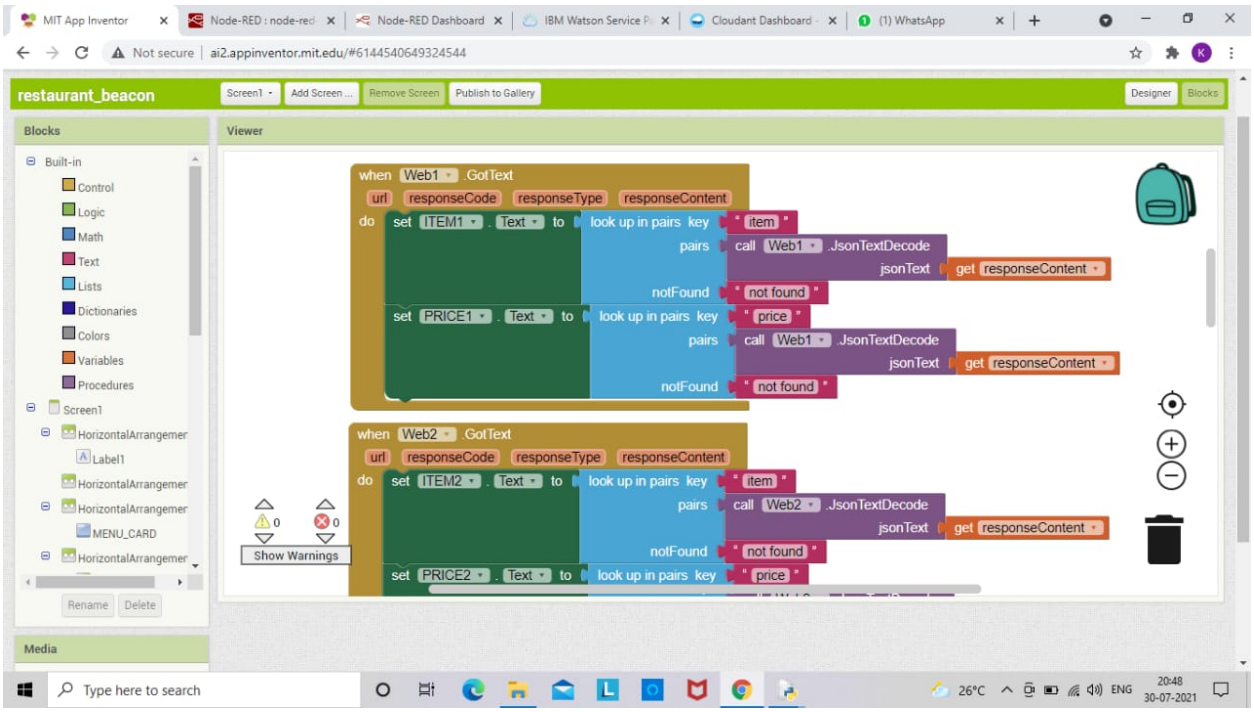

MIT App Inventor
Node-RED : node-red-kmfqi
Node-RED Dashboard
IBM Watson Service Page
Cloudant Dashboard - datab
node-red-kmfqi-2021-07-08.eu-gb.mybluemix.net/red/#flow/bebe6312.75eaa
Node-RED
Deploy
filter nodes
Flow 1
Flow 2
Flow 3
Flow 4
Flow 5
debug
all nodes
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
IBM IoT
connected
avail_fooditems
msg.payload
item1
item2
item3
item4
item5
item6
price1
price2
price3
price4
price5
price6
[get] /item1
[get] /item2
[get] /item3
webpage1
webpage2
webpage3
http
7/30/2021, 8:31:27 PM node: 47af8842.37dba8
iot-2/type/firstdevice/id/kris123/evt/status/fmt/json : msg.payload : number
125
7/30/2021, 8:31:50 PM node: b7ec47c.440f4b8
msg.payload : Object
object
order_qty: " item 1:chicken biriyani quantity:3 item 2:chicken noodles quantity:2 item 3:chicken fried rice quantity:2 item 4:veg Biryani quantity:1 table number:3 special instruction for cooking:spicy and less salt"
7/30/2021, 8:40:17 PM node: b7ec47c.440f4b8
msg.payload : Object
object
order_qty: " item 1:chicken biriyani quantity:3 item 2:chicken noodles quantity:2 item 3:chicken fried rice quantity:2 item 4:veg Biryani quantity:1 table number:3 special instruction for cooking:spicy and less salt"
Type here to search
27°C
ENG
20:40
30-07-2021

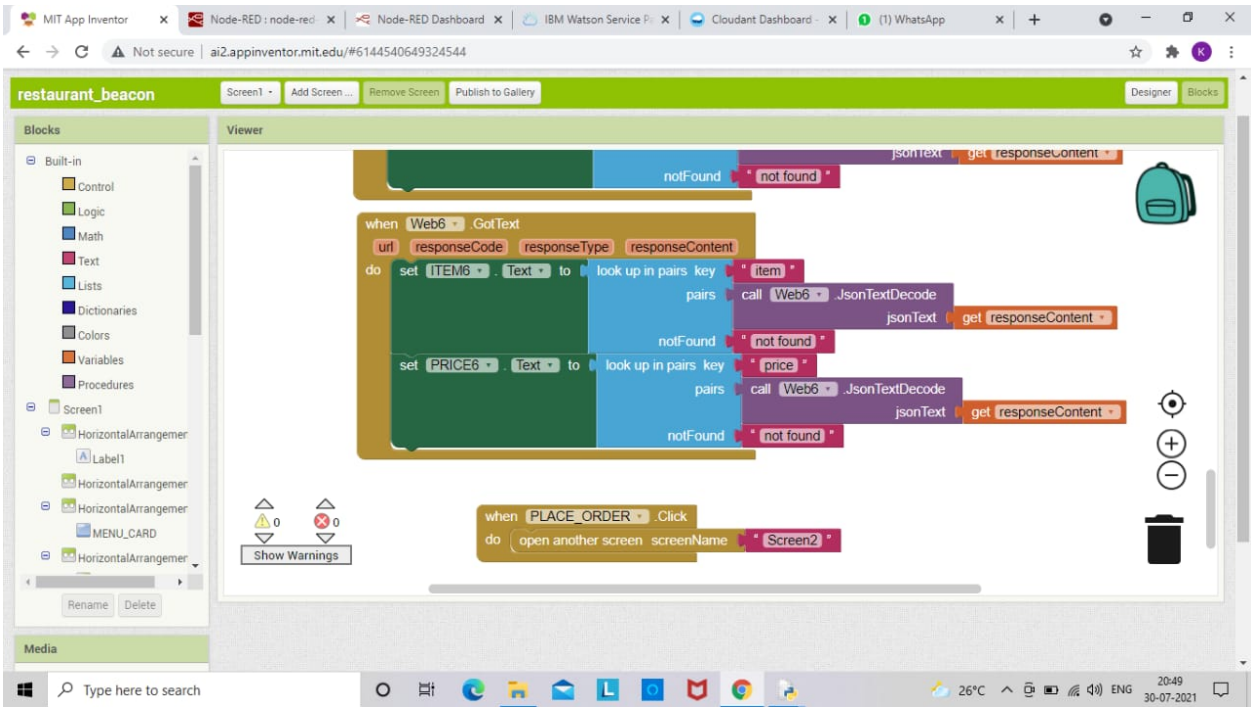

MIT App Inventor
Node-RED : node-red
Node-RED Dashboard
IBM Watson Service P
Cloudant Dashboard
(1) WhatsApp
node-red-kmfqi-2021-07-08.eu-gb.mybluemix.net/red/#flow/bebe6312.75eaa
Node-RED
Deploy
filter nodes
Flow 1
Flow 2
Flow 3
Flow 4
Flow 5
debug
all nodes
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
IBM IoT
connected
[get] /order
http
msg.payload
order_ui
food_order
text
text to speech
audio out
fast2sms
msg.payload
7/30/2021, 8:31:27 PM node: 47af8842.37dba8
iot-2/type/firstdevice/id/kris123/evt/status/fmt/json : msg.payload : number
125
7/30/2021, 8:31:50 PM node: b7ec47c.440f4b8
msg.payload : Object
object
order_qty: " item 1:chicken biriyani quantity:3 item 2:chicken noodles quantity:2 item 3:chicken fried rice quantity:2 item 4:veg Biryani quantity:1 table number:3 special instruction for cooking:spicy and less salt"
7/30/2021, 8:40:17 PM node: b7ec47c.440f4b8
msg.payload : Object
object
order_qty: " item 1:chicken biriyani quantity:3 item 2:chicken noodles quantity:2 item 3:chicken fried rice quantity:2 item 4:veg Biryani quantity:1 table number:3 special instruction for cooking:spicy and less salt"
Type here to search
27°C
ENG
20:45
30-07-2021

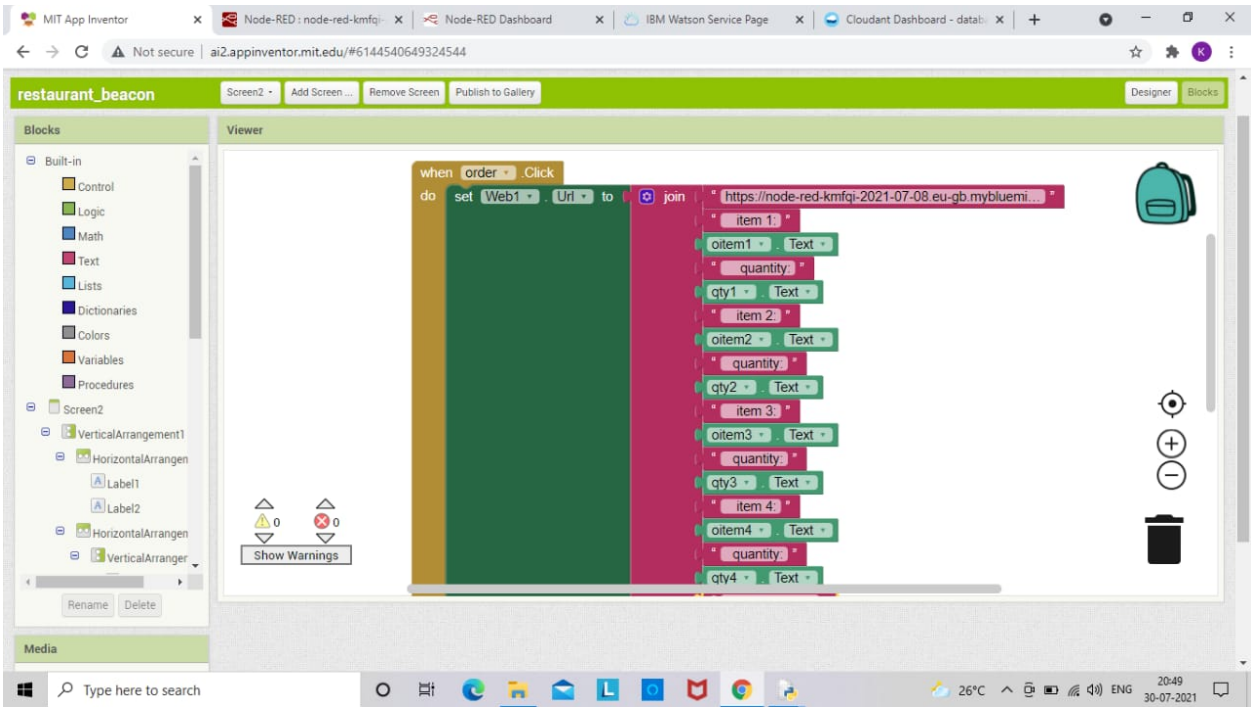

MIT App Inventor
Node-RED : node-red-kmfqi-202
Node-RED Dashboard
IBM Watson Service Page
node-red-kmfqi-2021-07-08.eu-gb.mybluemix.net/red/#flow/bebe6312.75eaa
Node-RED
Deploy
filter nodes
Flow 1
Flow 2
Flow 3
Flow 4
Flow 5
debug
all nodes
Language Understanding
natural language understanding V1 model manager
personality insights
speech to text
speech to text custom builder
text to speech
text to speech custom builder
tone analyzer v3
[get] /order
IBM IoT
connected
http
msg.payload
order_ui
food_order
text
text to speech
audio out
"veg noodles"
7/30/2021, 8:31:27 PM node: 47af8842.37dba8
iot-2/type/firstdevice/id/kris123/evt/status/fmt/json : msg.payload : number
110
7/30/2021, 8:31:27 PM node: 47af8842.37dba8
iot-2/type/firstdevice/id/kris123/evt/status/fmt/json : msg.payload : string[12]
"egg biriyani"
7/30/2021, 8:31:27 PM node: 47af8842.37dba8
iot-2/type/firstdevice/id/kris123/evt/status/fmt/json : msg.payload : number
125
7/30/2021, 8:31:50 PM node: b7ec47c.440f4b8
msg.payload : Object
object
order_qty: " item 1:chicken biriyani quantity:3 item 2:chicken noodles quantity:2 item 3:chicken fried rice quantity:2 item 4:veg Biryani quantity:1 table number:3 special instruction for cooking:spicy and less salt"

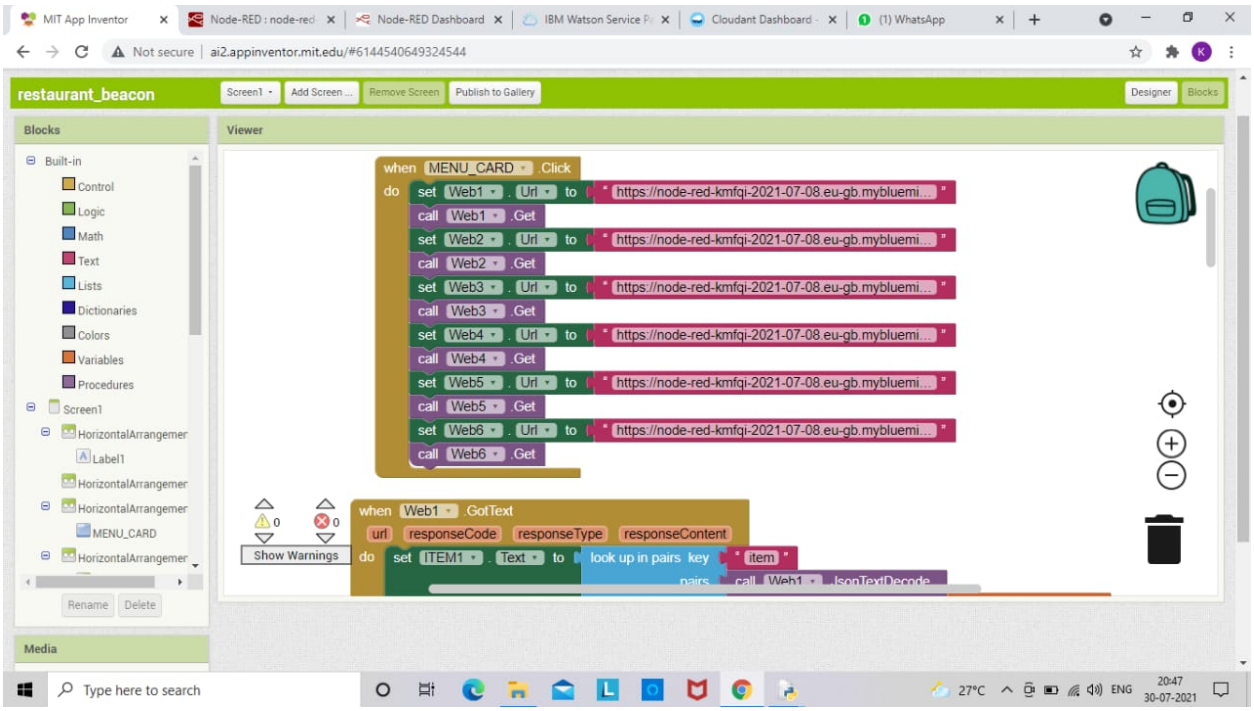

MIT App Inventor
Node-RED : node-red
Node-RED Dashboard
IBM Watson Service P
Cloudant Dashboard
(1) WhatsApp
Not secure
ai2.appinventor.mit.edu/#6144540649324544
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Maps
Phone size (505,320)
9:48
Screen1
OVER THE MOON
MENU CARD
Non-visible components
HorizontalArrangemer
Label1
MENU_CARD
VerticalArrangemer
ITEM1
ITEM2
ITEM3
ITEM4
ITEM5
ITEM6
PRICE1
PRICE2
Rename
Delete
Media
Menu-Card.jpg
th(2).jpg
th(3).jpg
th.jpg
AboutScreen
AccentColor
Default
AlignHorizontal
Left : 1
AlignVertical
Top : 1
AppName
restaurant_beacon
BackgroundColor
Default
BackgroundImage
None...
BlocksToolkit
All
CloseScreenAnimation
Default
Icon
None...
OpenScreenAnimation
Default
PrimaryColor
Default
PrimaryColorDark
Default
Type here to search
27°C
ENG
20:47
30-07-2021

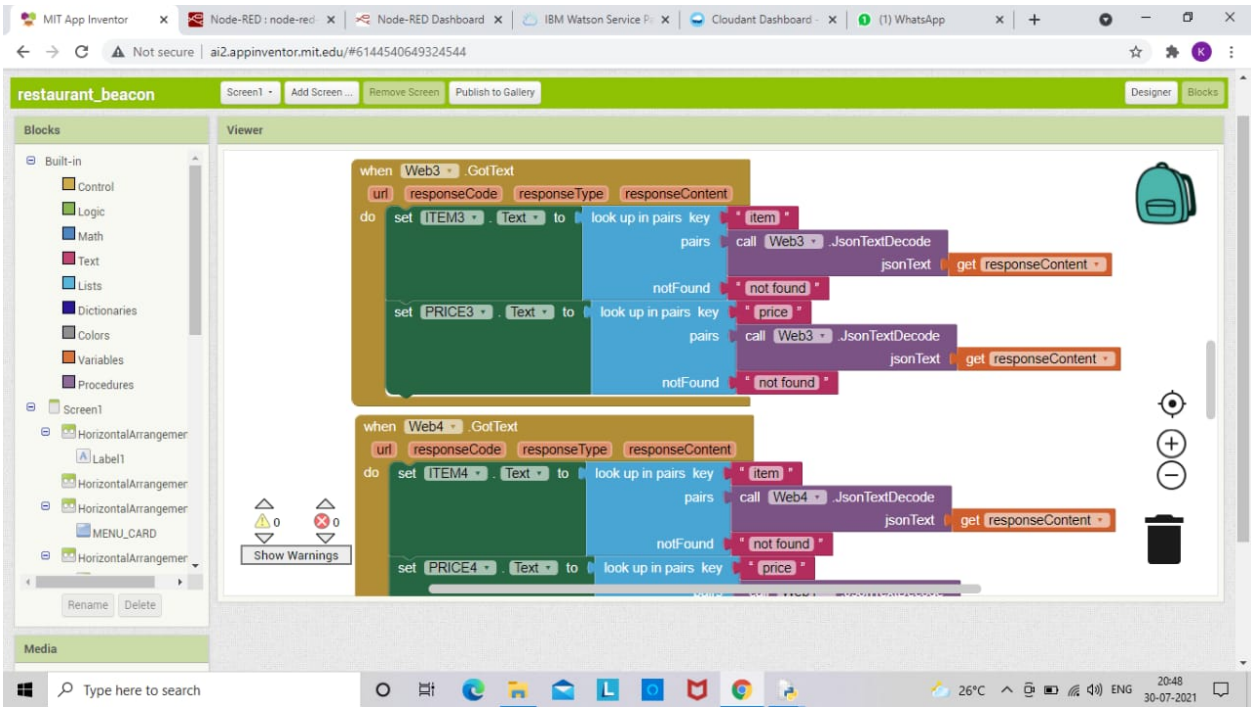

MIT App Inventor
Node-RED : node-red-kmfqi
Node-RED Dashboard
IBM Watson Service Page
Cloudant Dashboard - datab
Not secure | ai2.appinventor.mit.edu/#6144540649324544
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Phone size (505,320)
9:48
Screen2
ITEMS
QTY
SPECIAL INSTRUCTION FOR COOKING
Non-visible components
VerticalArrangement1
HorizontalArrangen
Label1
Label2
HorizontalArrangen
VerticalArranger
oitem1
oitem2
oitem3
oitem4
HorizontalArran
VerticalArranger
qty1
qty2
qty3
qty4
Rename
Delete
Media
Menu-Card.jpg
th(2).jpg
th(3).jpg
th.jpg
AboutScreen
AlignHorizontal
Left : 1
AlignVertical
Top : 1
BackgroundColor
Default
BackgroundImage
None...
CloseScreenAnimation
Default
OpenScreenAnimation
Default
ScreenOrientation
Unspecified
Scrollable
ShowStatusBar
Title
Screen2
TitleVisible
Type here to search
26°C
ENG
20:50
30-07-2021

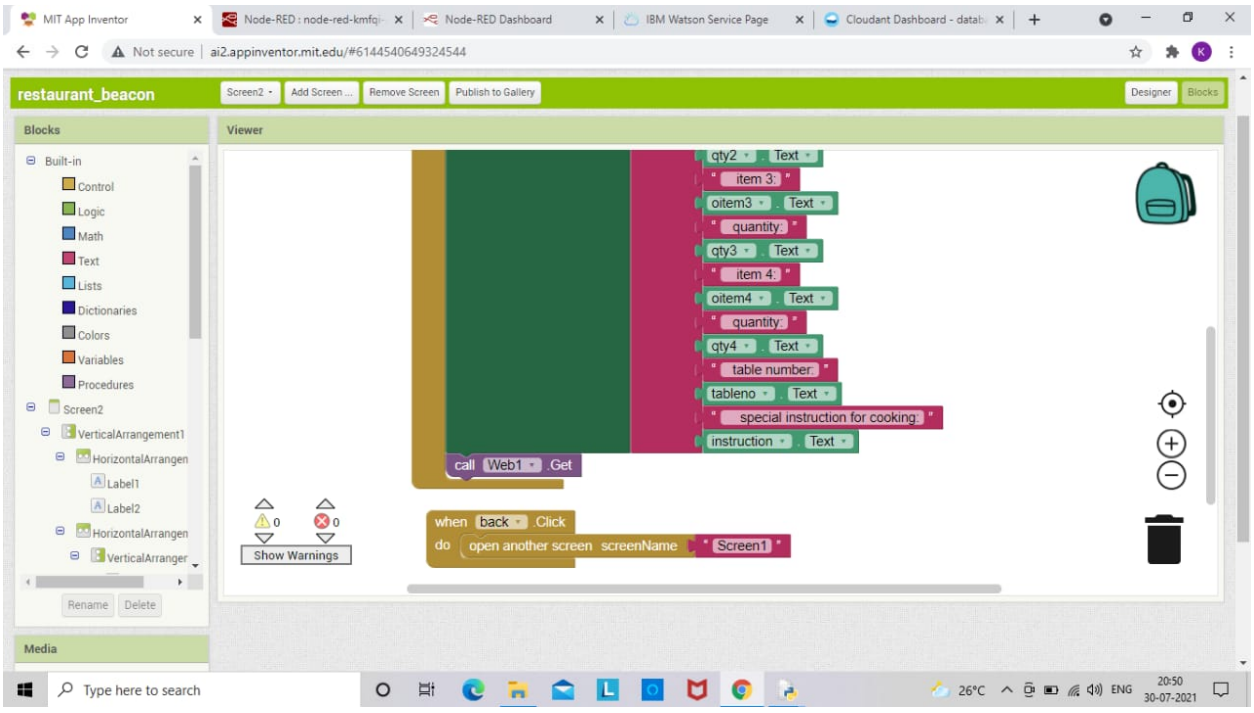

MIT App Inventor
Node-RED : node-red
Node-RED Dashboard
IBM Watson Service P
Cloudant Dashboard
(1) WhatsApp
Not secure | ai2.appinventor.mit.edu/#6144540649324544
restaurant_beacon
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
HorizontalArrangemer
Label1
HorizontalArrangemer
HorizontalArrangemer
MENU_CARD
HorizontalArrangemer
Rename
Delete
Media
Viewer
when Web1 .GotText
url responseCode responseType responseContent
do set ITEM1 . Text to look up in pairs key " item "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
set PRICE1 . Text to look up in pairs key " price "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
when Web2 .GotText
url responseCode responseType responseContent
do set ITEM2 . Text to look up in pairs key " item "
pairs call Web2 .JsonTextDecode
jsonText get responseContent
notFound " not found "
set PRICE2 . Text to look up in pairs key " price "
Show Warnings
Type here to search
26°C
ENG
20:48
30-07-2021


MIT App Inventor
Node-RED : node-red
Node-RED Dashboard
IBM Watson Service P
Cloudant Dashboard
(1) WhatsApp
Not secure | ai2.appinventor.mit.edu/#6144540649324544
restaurant_beacon
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
HorizontalArrangemer
Label1
MENU_CARD
Rename
Delete
Media
notFound
not found
when Web6 .GotText
url responseCode responseType responseContent
do set ITEM6 . Text to look up in pairs key item
pairs call Web6 .JsonTextDecode
jsonText get responseContent
notFound not found
set PRICE6 . Text to look up in pairs key price
pairs call Web6 .JsonTextDecode
jsonText get responseContent
notFound not found
when PLACE_ORDER .Click
do open another screen screenName Screen2
Show Warnings
Type here to search
26°C
ENG
20:49
30-07-2021


MIT App Inventor
Node-RED : node-red-kmfqi
Node-RED Dashboard
IBM Watson Service Page
Cloudant Dashboard - datab
Not secure | ai2.appinventor.mit.edu/#6144540649324544
restaurant_beacon
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen2
VerticalArrangement1
HorizontalArrangen
Label1
Label2
VerticalArranger
Rename
Delete
Media
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
Type here to search
26°C
ENG
20:49
30-07-2021


MIT App Inventor
Node-RED : node-red
Node-RED Dashboard
IBM Watson Service
Cloudant Dashboard
(1) WhatsApp
Not secure | ai2.appinventor.mit.edu/#6144540649324544
restaurant_beacon
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
HorizontalArrangemer
Label1
HorizontalArrangemer
HorizontalArrangemer
MENU_CARD
HorizontalArrangemer
Rename
Delete
Media
Viewer
when MENU_CARD Click
do set Web1 . Url to " https://node-red-kmfqi-2021-07-08.eu-gb.mybluemi... "
call Web1 .Get
set Web2 . Url to " https://node-red-kmfqi-2021-07-08.eu-gb.mybluemi... "
call Web2 .Get
set Web3 . Url to " https://node-red-kmfqi-2021-07-08.eu-gb.mybluemi... "
call Web3 .Get
set Web4 . Url to " https://node-red-kmfqi-2021-07-08.eu-gb.mybluemi... "
call Web4 .Get
set Web5 . Url to " https://node-red-kmfqi-2021-07-08.eu-gb.mybluemi... "
call Web5 .Get
set Web6 . Url to " https://node-red-kmfqi-2021-07-08.eu-gb.mybluemi... "
call Web6 .Get
Show Warnings
when Web1 .GotText
url responseCode responseType responseContent
do set ITEM1 . Text to look up in pairs key " item "
Type here to search
27°C
ENG
20:47
30-07-2021


MIT App Inventor
Node-RED : node-red
Node-RED Dashboard
IBM Watson Service
Cloudant Dashboard
(1) WhatsApp
Not secure | ai2.appinventor.mit.edu/#6144540649324544
restaurant_beacon
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen1
HorizontalArrangemer
Label1
HorizontalArrangemer
HorizontalArrangemer
MENU_CARD
HorizontalArrangemer
Rename
Delete
Media
Viewer
when Web3 .GotText
url responseCode responseType responseContent
do set ITEM3 . Text to look up in pairs key " item "
pairs call Web3 .JsonTextDecode
jsonText get responseContent
notFound " not found "
set PRICE3 . Text to look up in pairs key " price "
pairs call Web3 .JsonTextDecode
jsonText get responseContent
notFound " not found "
when Web4 .GotText
url responseCode responseType responseContent
do set ITEM4 . Text to look up in pairs key " item "
pairs call Web4 .JsonTextDecode
jsonText get responseContent
notFound " not found "
set PRICE4 . Text to look up in pairs key " price "
Show Warnings
Type here to search
26°C
ENG
20:48
30-07-2021


MIT App Inventor
Node-RED : node-red-kmfqi
Node-RED Dashboard
IBM Watson Service Page
Cloudant Dashboard - datab
Not secure | ai2.appinventor.mit.edu/#6144540649324544
restaurant_beacon
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen2
VerticalArrangement1
HorizontalArrangen
Label1
Label2
HorizontalArrangen
VerticalArranger
Rename
Delete
Media
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 "
0
0
Show Warnings
Type here to search
26°C
ENG
20:50
30-07-2021

8:29 PM

Screen2

ITEMS	QTY
chicken biriyani	3
chicken noodles	2
chicken fried rice	2
veg Biryani	1

SPECIAL INSTRUCTION FOR COOKING

spicy and less salt

ENTER THE TABLE NO: 3

ORDER BACK

8:28 PM

Screen1

OVER THE MOON

chicken biriyani	100
chicken noodles	120
chicken friedrice	130
veg biriyani	100
veg noodles	110
egg biriyani	125

PLACE ORDER

8:27 PM

Screen1

OVER THE MOON

PLACE ORDER

8:28 PM

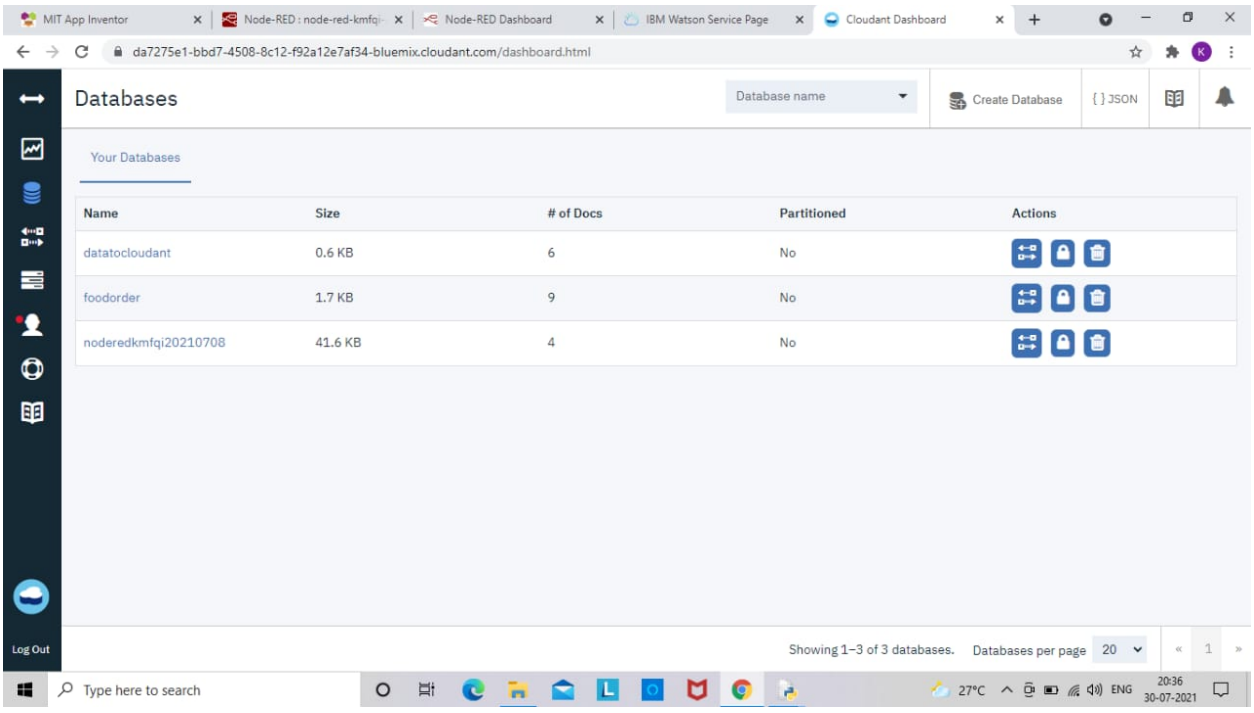
Screen2

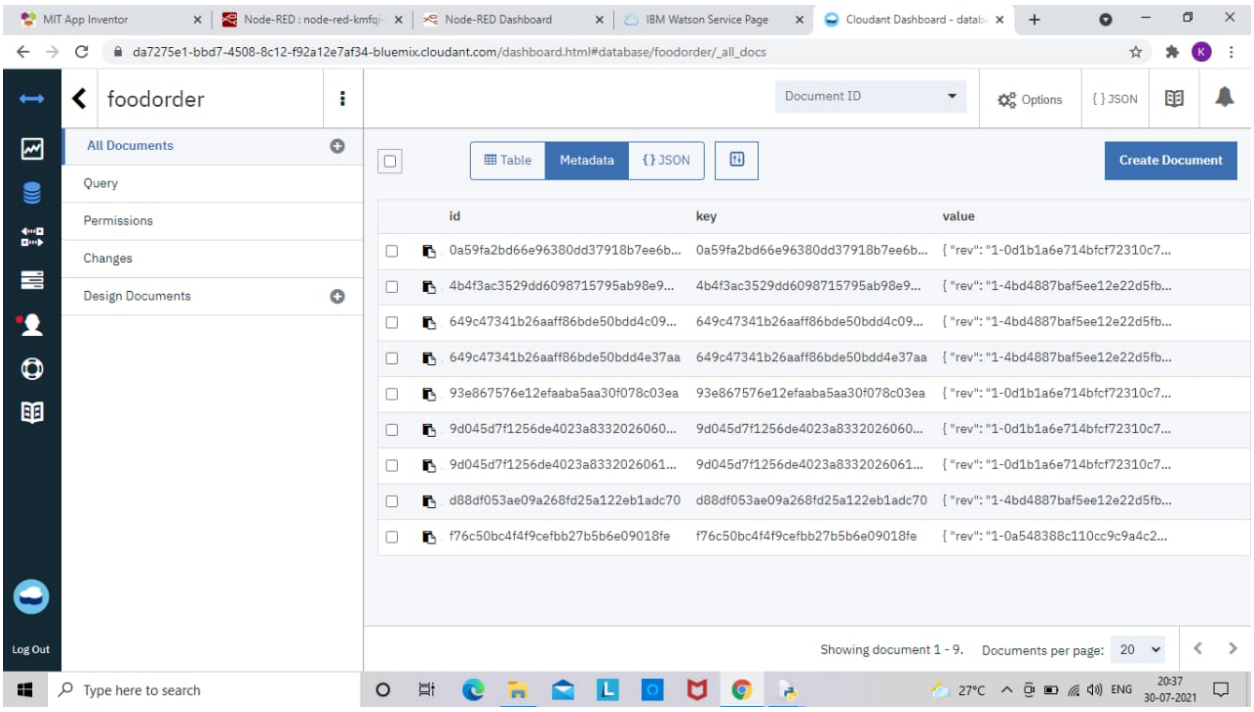
ITEMS	QTY

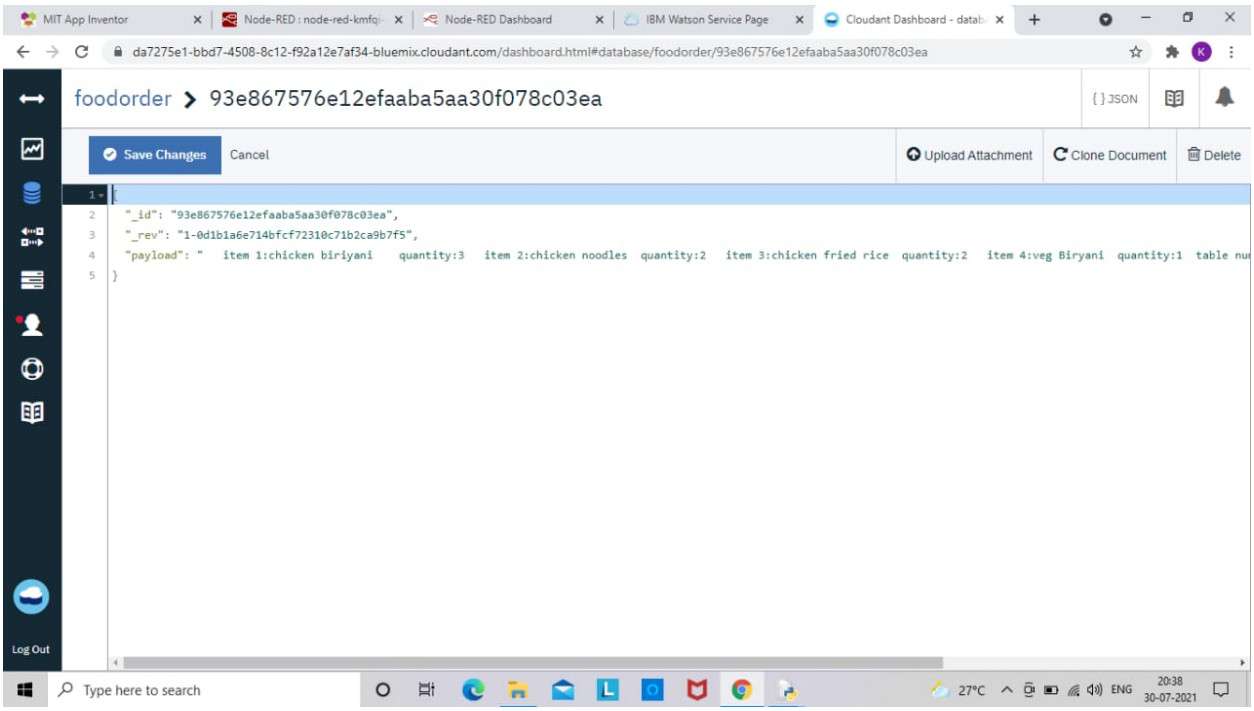
SPECIAL INSTRUCTION FOR COOKING

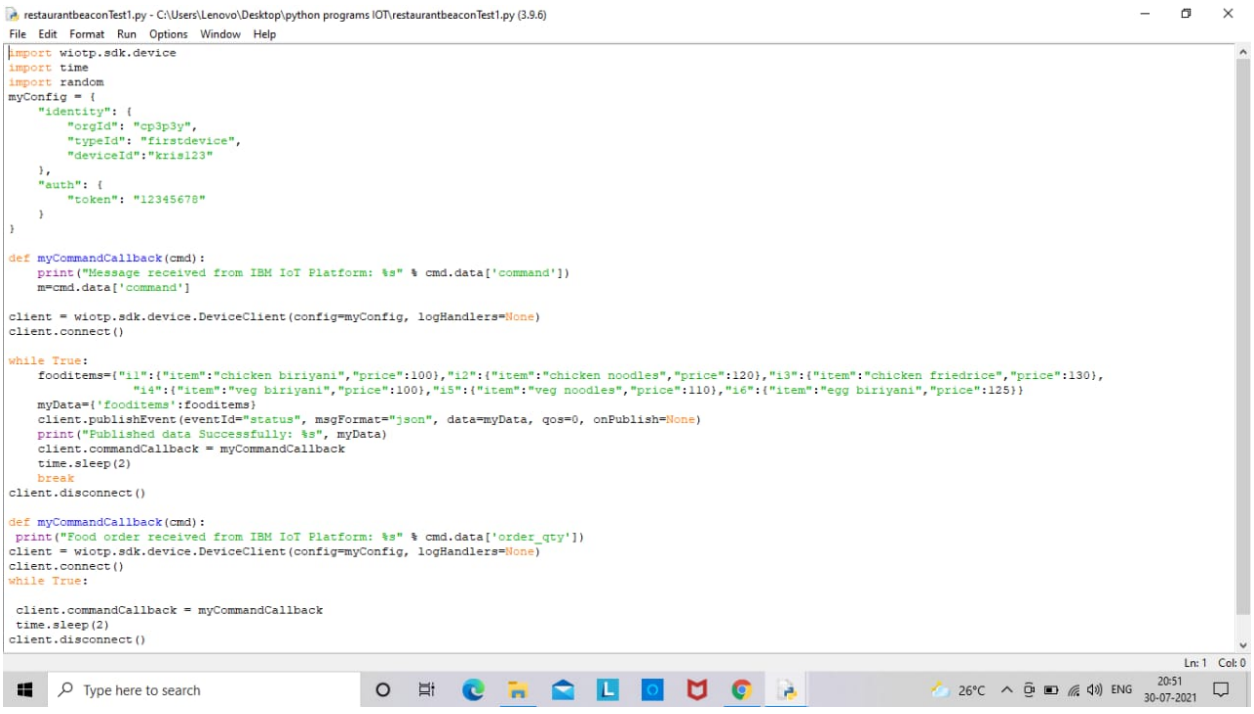
ENTER THE TABLE NO:

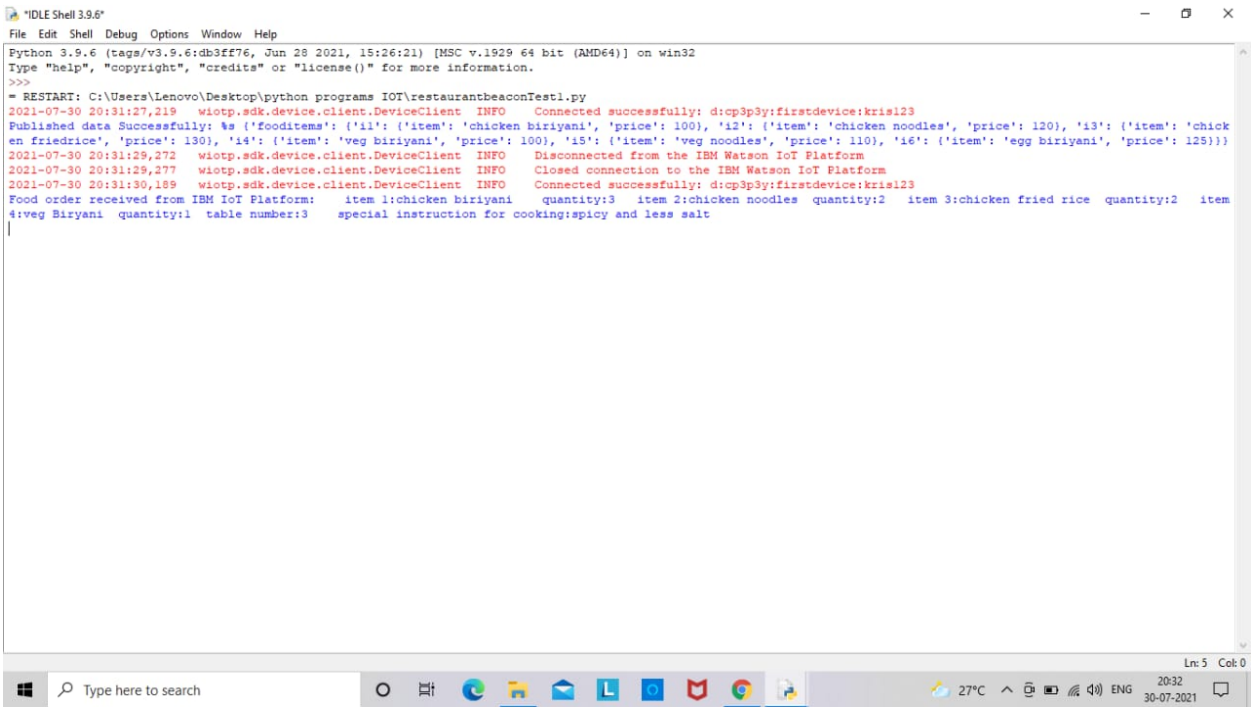
ORDER BACK


MIT App Inventor
Node-RED : node-red-kmfqi
Node-RED Dashboard
IBM Watson Service Page
Cloudant Dashboard
da7275e1-bbd7-4508-8c12-f92a12e7af34-bluemix.cloudant.com/dashboard.html
Databases
Database name
Create Database
{ } JSON
Your Databases
Name
Size
# of Docs
Partitioned
Actions
datatocloudant
0.6 KB
6
No
foodorder
1.7 KB
9
No
noderedkmfqi20210708
41.6 KB
4
No
Log Out
Showing 1–3 of 3 databases.
Databases per page
20
Type here to search
27°C
ENG
20:36
30-07-2021


MIT App Inventor
Node-RED : node-red-kmfqi
Node-RED Dashboard
IBM Watson Service Page
Cloudant Dashboard - datab
da7275e1-bbd7-4508-8c12-f92a12e7af34-bluemix.cloudant.com/dashboard.html#database/foodorder/_all_docs
foodorder
Document ID
Options
{ } JSON
All Documents
Query
Permissions
Changes
Design Documents
Table
Metadata
{ } JSON
Create Document
id
key
value
0a59fa2bd66e96380dd37918b7ee6b...
4b4f3ac3529dd6098715795ab98e9...
649c47341b26aaff86bde50bdd4c09...
649c47341b26aaff86bde50bdd4e37aa
93e867576e12efaaba5aa30f078c03ea
9d045d7f1256de4023a8332026060...
9d045d7f1256de4023a8332026061...
d88df053ae09a268fd25a122eb1adc70
f76c50bc4f4f9cefbb27b5b6e09018fe
{ "rev": "1-0d1b1a6e714bfcf72310c7...
{ "rev": "1-4bd4887baf5ee12e22d5fb...
{ "rev": "1-0a548388c110cc9c9a4c2...
Log Out
Showing document 1 - 9.
Documents per page:
20
Type here to search
27°C
ENG
20:37
30-07-2021


MIT App Inventor
Node-RED : node-red-kmfqi
Node-RED Dashboard
IBM Watson Service Page
Cloudant Dashboard - datal
da7275e1-bbd7-4508-8c12-f92a12e7af34-bluemix.cloudant.com/dashboard.html#database/foodorder/93e867576e12efaaba5aa30f078c03ea
foodorder > 93e867576e12efaaba5aa30f078c03ea
{ } JSON
Save Changes
Cancel
Upload Attachment
Clone Document
Delete
{
"_id": "93e867576e12efaaba5aa30f078c03ea",
"_rev": "1-0d1b1a6e714bfcf72310c71b2ca9b7f5",
"payload": " item 1:chicken biriyani quantity:3 item 2:chicken noodles quantity:2 item 3:chicken fried rice quantity:2 item 4:veg Biryani quantity:1 table nu
}
Log Out
Type here to search
27°C
ENG
20:38
30-07-2021


restaurantbeaconTest1.py - C:\Users\Lenovo\Desktop\python programs IOT\restaurantbeaconTest1.py (3.9.6)
File Edit Format Run Options Window Help
import wiotp.sdk.device
import time
import random
myConfig = {
"identity": {
"orgId": "cp3p3y",
"typeId": "firstdevice",
"deviceId":"kris123"
},
"auth": {
"token": "12345678"
}
}
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()
Ln: 1 Col: 0
Type here to search
26°C
ENG
20:51
30-07-2021



7 ADVANTAGES AND DISADVANTAGES

advantages

- Beacons have great price / value factor compared with other technologies.
- They are easy to use and implement.
- Beacons have multiple applications (engagement, navigation, analysis, etc.)
- Enable to understand customers better by collecting customer behaviour data.
- Enables to retarget customers after they walk away.
- They work in multiple verticals (retail, education, culture, airports, events, offices, hotels etc.)

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. Most beacons are battery-powered, which may be a maintenance issue. You can get USB beacons instead but they need to have a power supply.

8 APPLICATIONS

Following are the applications of smart beacons:

- location
- navigation
- analysis and data collection

- interaction

Beacons are the most robust tool when it comes to collecting data about customers.

Connected with a mobile application they can provide large quantities of data

Number of new visitors

- Number of returning visitors
- How many times a campaign has been viewed
- Conversion
- Visitor location heatmaps
- Visitor dwell time
- Visitor age
- Visitor gender
- How visitors move

9 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.

10 FUTURE SCOPE

In future work can be done on how to reduce network traffice in case of more number of customers. also some other source can be used to avoid installation of app to save storage. in case of network issue we can find some substitue.

11 BIBLIOGRAPHY

For completion of this project we have used referrance from the training videos on topic smart home automation.
Our trainer has guided through the project.
platforms used are:

1. Node red
2. IBM cloudant
3. MIT app inventor
4. python

12 APPENDIX

source code(1):

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "cp3p3y",
        "typeId": "firstdevice",
        "deviceId":"kris123"
    },
    "auth": {
        "token": "12345678"
    }
}

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()
```

source code(2):

```
import wiotp.sdk.device
import time
import random
myConfig = {
   "identity": {
      "orgId": "cp3p3y",
      "typeId": "firstdevice",
      "deviceId":"kris123"
   },
   "auth": {
      "token": "12345678"
   }
}
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-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()
```

UI OUTPUT: