PROJECT REPORT

PROJECT NAME	SMART SOLUTION FOR RAILWAYS
TEAM ID	PNT2022TMID06140
COLLEGE NAME	ALAGAPPA CHETTIAR GOVERNMENT COLLEGE OF ENGINEERING AND TECHNOLOGY.
TEAM MEMBERS	S.ANU ARCHANA P.BHARATHI S.SHRUTHIKA SHABNAM K.R.VENNILA

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

13.1 Source Code

13.2 GitHub & Project Demo Link

1. INTRODUCTION

1.1 PROJECT REVIEW:

SMART SOLUTIONS FOR RAILWAYS

As trains are one of the most preferred modes of transportation among middle class and impoverished people as it attracts for its amenities. Simultaneously there is an increase at risk from thefts and accidents like chain snatching, derailment, fire accident. In order to avoid or in better words to stop all such brutality we came up with a solution by providing an application which can be accessed by the user after booking their tickets. With a single click this app addresses issues by sending a text message to TC and RPF as an alert. In our project we use Node-Red service, app-development, IBM cloud platform to store passenger data.

1.2 PURPOSE :

Internet is basically system of interconnected computers through network. But now its use is changing with changing world and it is not just confined to emails or web browsing. Today's internet also deals with embedded sensors and has led to development of smart homes, smart rural area, e-health care's etc. and this introduced the concept of IoT . Internet of Things refers to interconnection or communication between two or more devices without humanto-human and human-to-computer interaction.

Connected devices are equipped with sensors or actuators perceive their surroundings. IOT has four major components which include sensing the device, accessing the device, processing the information of the device, and provides application and services. In addition to this it also provides security and privacy of data . Automation has affected every aspect of our daily lives. More improvements are being introduced in almost all fields to reduce human effort and save time. Thinking of the same is trying to introduce automation in the field of ticket booking. Ticket booking is an integral part , we have to provide it with some necessary functionalities. Problems that occur due to complications in the process of booking tickets need to be overcome. The latest method used by our Indian railways requires lot of time.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

1. A Web page is designed for the public where they can book tickets by seeing the available seats.
2. After booking the train, the person will get a QR code which has to be shown to the Ticket Collector while boarding the train.
3. The ticket collectors can scan the QR code to identify the personal details.
4. A GPS module is present in the train to track it. The live status of the journey is updated in the Web app continuously
5. All the booking details of the customers will be stored in the database with a unique ID and they can be retrieved back when the Ticket Collector scans the QR Code.

2.2 REFERENCES

1. Passenger Monitoring Model for easily Accessible Public City Trams/Trains	Roman Khoeblal, Teeravisit Laohapensaeng,Roungsan Chaisricharoen	2015	Passenger monitoring, passenger controlRFID distance reading, ticket control, RFID ticket inspection	It is possible to travel cross countrywith a single public transportationcard, using transport systems of several transport operators.	Applicable only for passenger monitoring.
2. Application of smart computing in Indian Railway Systems	Parag Chatterjee Asoke Nath	2014	By Interlinking uniqueidentification system with train ticket reservation systemby using video surveillance,rail sensors, biometric input devices and multimedia displays.	Reduces manual effortin passenger data entry Provides security verification.	Significant investment is needed. Risk of database.

3. Android Suburban RailwayTicketing with GPS as TicketChecker.	Sana Khoja, Maithili Kadam	2012	Android, SQ lite, Cloud Database, ASR,QR Code	E-Ticket facility, enabling reuse and replacement of components	QR Codes before the user enters or leaves thestation, where the user can have access which is risk in ticket booking.
4. NovelApproach forSmart Indian Railways.	Sujith Kumar, K.M.Yatheendra Parvan, V.Sumathy, Thejeswari C.K	2017	Digitalization, Smart Railways, Aadhar Card, Smartphone, Identity Verification	Employ a mobileapplication through whichpassengers can accessvarious ticketing options in user friendly and efficient manner.	Biometric database isrisk of hacking.

2.3 PROBLEM STATEMENT DEFINITION

Mithra who is a software developer at MNC, which is located little far from her house needs to take her travel through train daily in which she wants to save her time and get into her office on time.

The problem mainly affects the passengers in the train.

The problem mainly occurs in the railway bookings in person and manual works at railways.

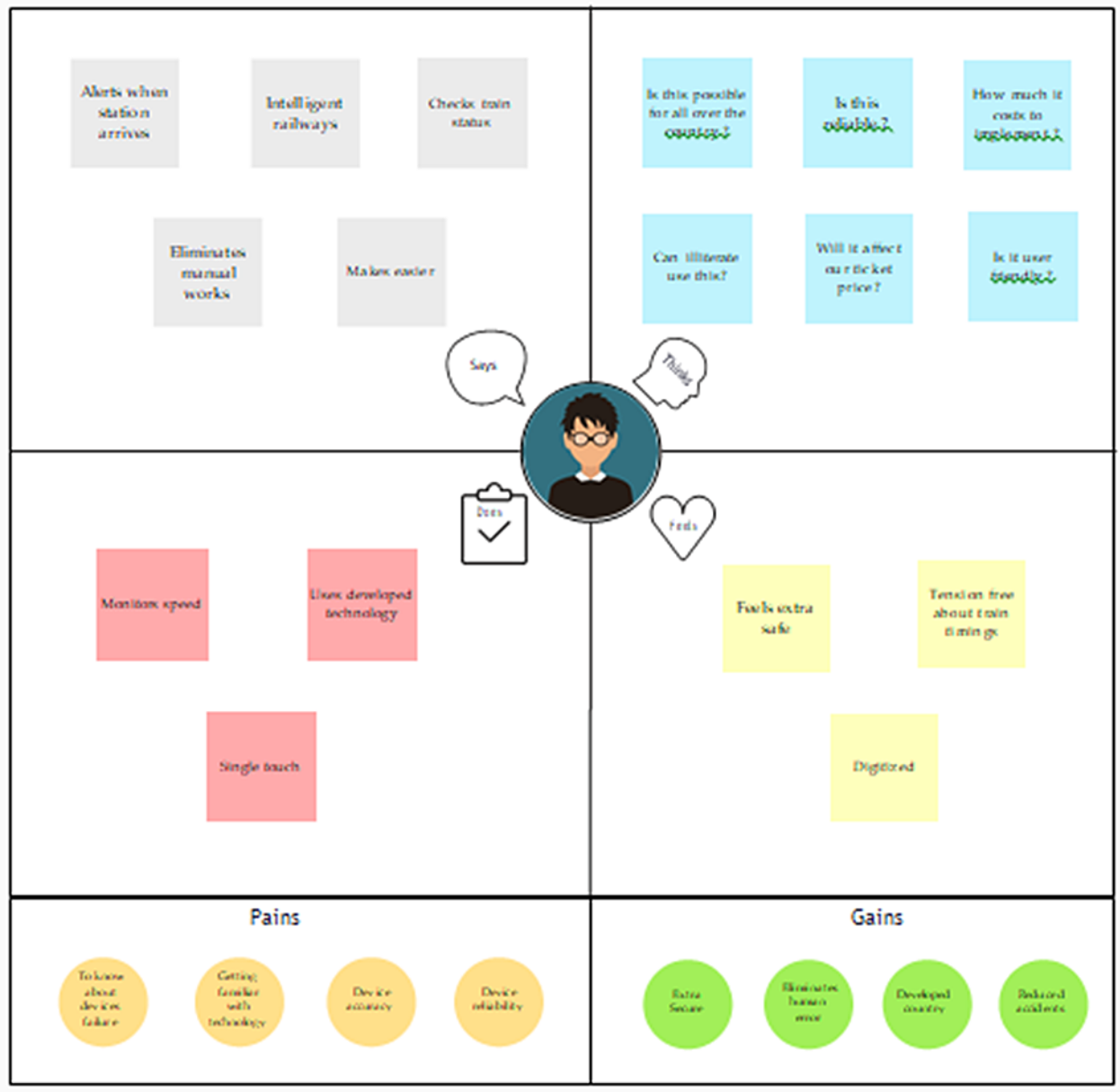
When Passenger wants to book ticket, they wait in long que and waste time. Sometimes they may miss their destination or miss their train due to unaware of timings.

The issue occurs when there is a fault in manual monitoring.

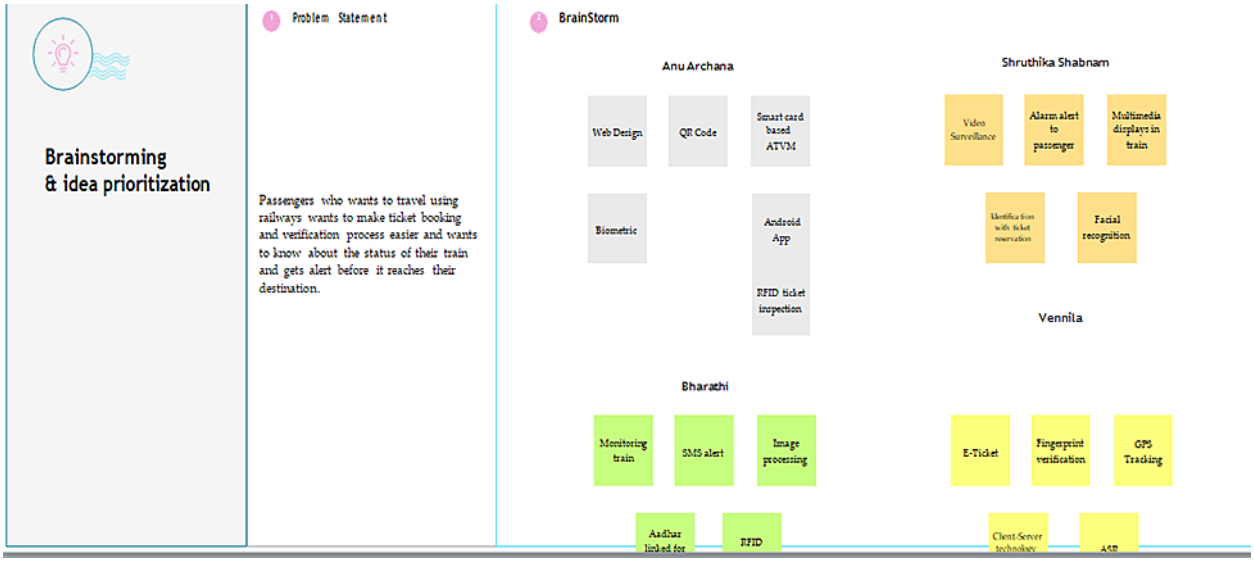
Its important to fix the problem in order to make aware of the passenger about timings and their stations and to avoid waiting in que for long time for booking tickets.

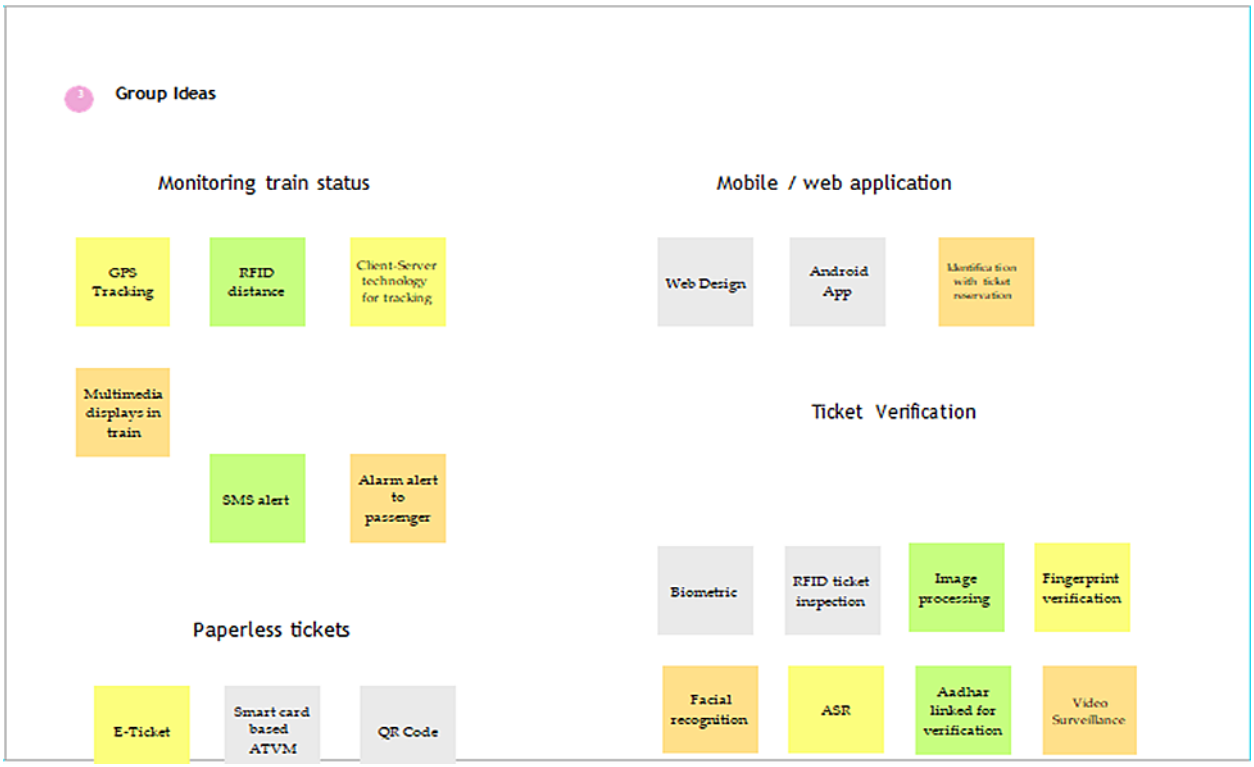
3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

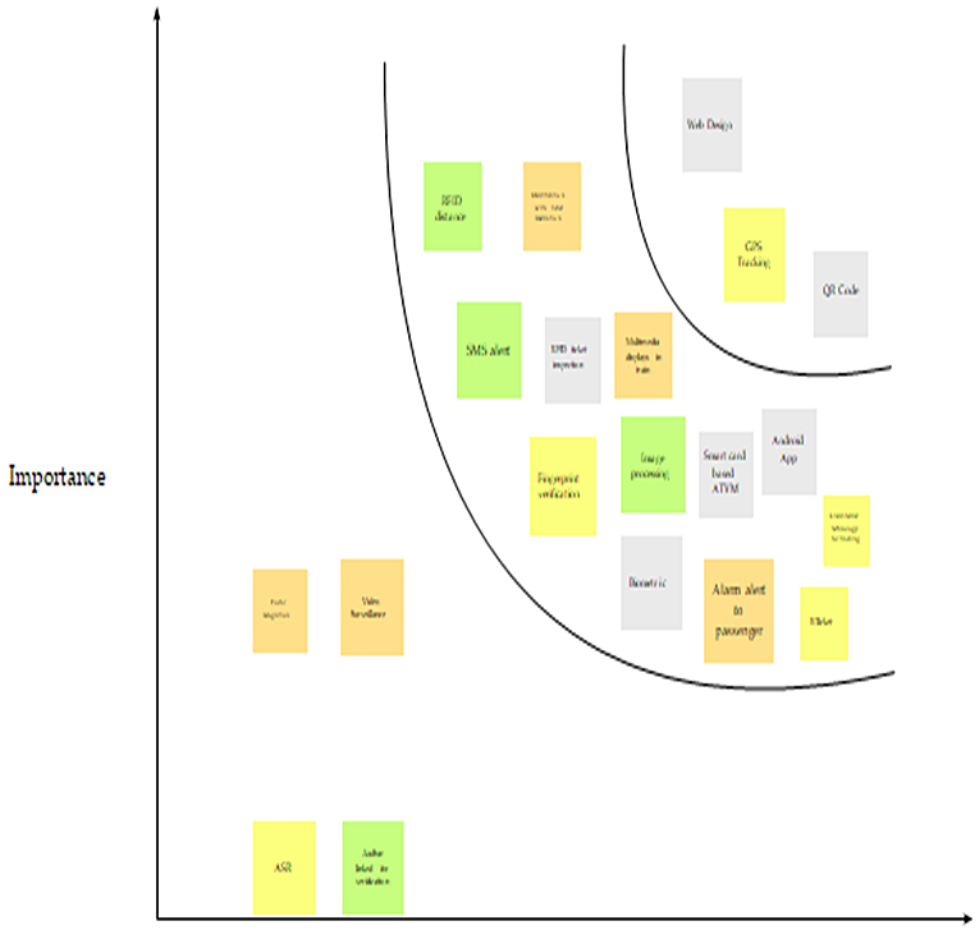


3.2 IDEATION AND BRAINTSTORMING





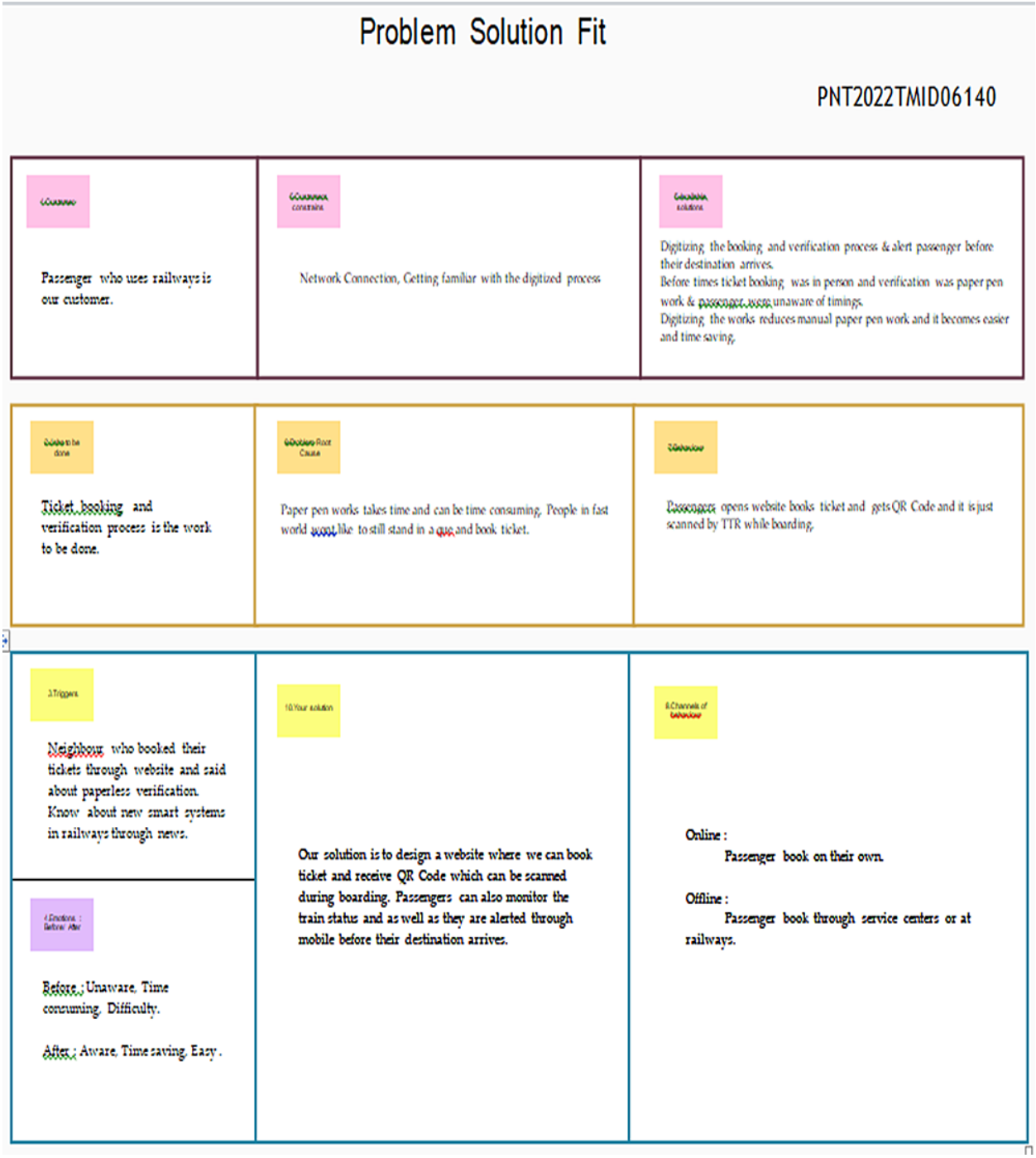
Prioritize


Importance
Web Design
RFID distance
GPS Tracking
QR Code
SMS alert
Android App
Smart card based ATVM
Biometric
Alarm alert to passenger
ASR

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1	Problem Statement (Problem to besolved)	To make ticket booking and verification processeasier and alert, make aware aboutpassenger's destination before it arrives.
2	Idea / Solution description	We have come up with the idea of designing a website to book tickets and to see other information about train. After booking passenger receive QR code and it is scanned byticket collector during boarding to verify identity. Unique ID database is stored in the cloud. Through GPS tracking live status of traincan be monitored and alerts passenger before their destination arrives.
3	Novelty / Uniqueness	Passenger is alerted though mobile phonebefore destination arrives.
4	Social Impact/ Customer Satisfaction	People feel relaxed and tension free abouttickets and arrival of destination.
5	Business Model(Revenue Model)	Digitizing the paper works minimizes the cost, and people more likely prefer this and it is costefficient and secure.
6	Scalability of the Solution	It supports adding new features and it is userfriendly with high security.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Booking	Book tickets through website/App.
FR-2	User Confirmation	Confirmation via OTP.
FR-3	User’s Ticket	Receiving QR-code throughmail.
FR-4	User’s Remainder	One day beforetheir journey viaSMS.
FR-5	User’s Destination Remainder	SMS Alert before15 mins of their departure.

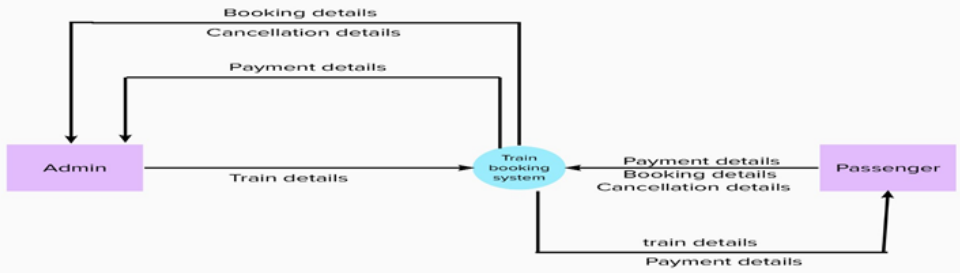
4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Simple booking procedure and user can use website easily.
NFR-2	**Security**	Personal information of the user is secured in database.
NFR-3	**Reliability**	Multiple numberof users can access withoutany failure.
NFR-4	**Performance**	Uder friendly
NFR-5	**Availability**	24/7 availability of website.
NFR-6	**Scalability**	Can have the feature of selecting the required favourite seat.

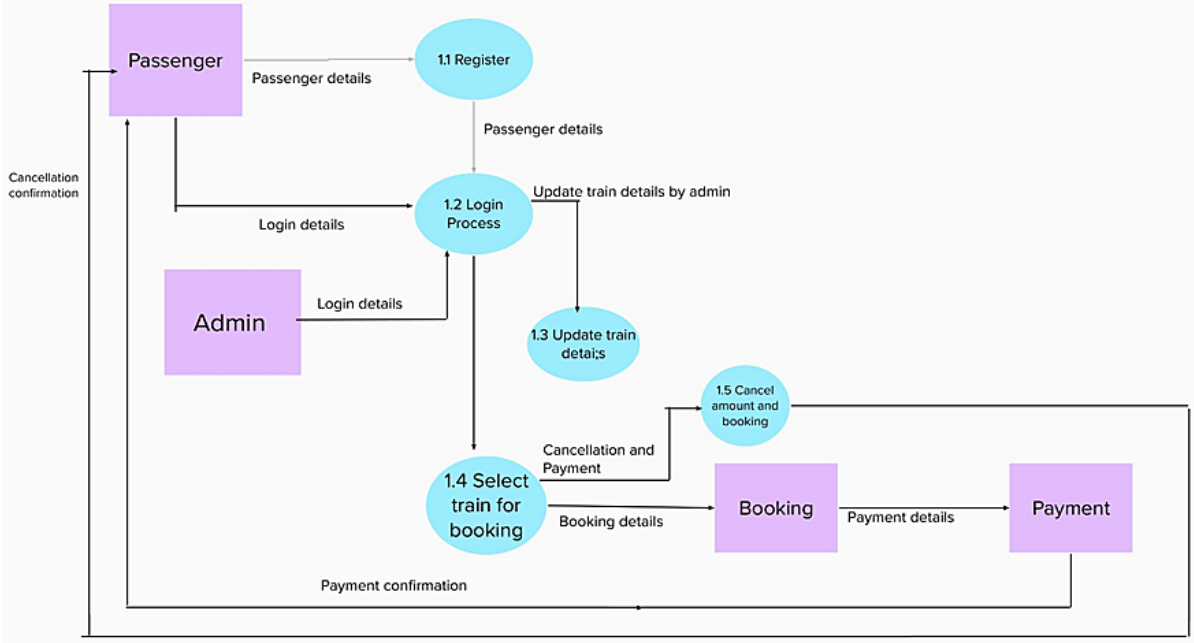
5. PROJECT PLANNING AND DESIGN

5.1 DATA FLOW DIAGRAMS

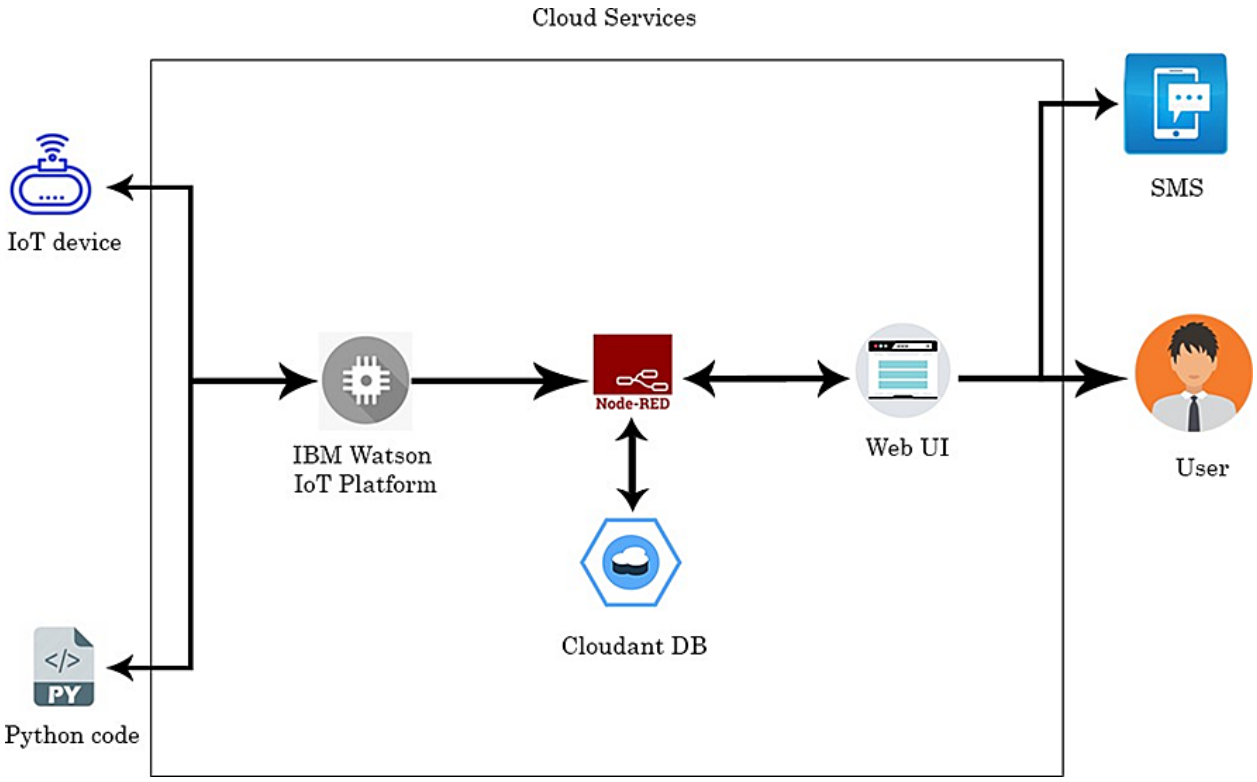
DFD level 0:



DFD level 1:



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Passenger (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1
	Booking	USN-4	As a user, I can book tickets with ID proof	I can receive successful payment notification	Medium	Sprint-1
	Cancellation	USN-4	As a user, I can cancel my ticket at any time	I can receive cancel notification	Medium	Sprint-2
Passenger (Web user)	Register	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password		High	Sprint-1
	Booking	USN-3	As a user, I can book tickets with ID proof	I can receive successful payment notification	Medium	Sprint-1
	Cancellation	USN-3	As a user, I can cancel my ticket at any time	I can receive cancel notification	Medium	Sprint-2
Administrator	Login	Nil	As a administrator, I can login with email id and password	I can update entire system	High	Sprint-1

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User StoryNumber	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I canregister through phone number,gmail.	1	High	Anu Archana
Sprint-1	Registration	USN-2	As a user, I can register through the form byfilling the details.	2	High	Bharathi
Sprint-1	Confirmation	USN-3	As a user, I can receive conformation throughemail, OTP once registration is successful.	2	Low	Shruthika shabnam
Sprint-1	Login	USN-4	As a user,I can login via login id and password Or through OTP received on registered phone number.	2	Medium	Vennila
Sprint-1	Display train details	USN-5	As a user, I can enter the start and destinationplaceto get the list of trains available	1	High	Anu Archana
Sprint-2	Booking	USN-6	As a user,I can provide basic details such as name,age, gender etc	2	High	Bharathi
Sprint-2	Booking	USN-7	As a user, I can choose the class, seat, berthand availability	1	Medium	Shruthka shabnam

Sprint-2	Payment	USN-8	As a user, I can chooseto pay through credit/debit card or UPI.	1	High	Vennila
Sprint-2	Display	USN-9	As a user, Ican see my ticket	1	High	Anu Archana
Sprint-3	Ticket generation	USN-10	As a user, i can download the generated e ticketfor my journey along with the QR code which is used for authentication duringmy journey.	1	High	Bharathi

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story/ Task**	**Story Points**	**Priority**	**Team Members**
Sprint-3	Remainder	USN-11	As a user, I get remember about my journey aday before actualjourney.	1	High	Shruthi ka shabn am
Sprint-3	Journey tracking	USN-12	As a user, I can track the train using GPS and can get information suchas Current stopanddelay	1	High	Vennila
Sprint-4	Alert	USN-13	As a user, I get alert message when my destination arrives.	1	High	Anu Archana
Sprint-4	Cancellation	USN-14	As a user, i can cancel my tickets if there's is achange of plan	1	High	Bharathi
Sprint-4	Feed details	USN-15	As a user, I can share the feedback about my journeylike train delays, extra addition of seatsby adding extra compartments	2	Low	Shruthi ka shabn am

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA

7. CODING AND SOLUTIONING

7.1 FEATURE 1

- IoT device
- IBM Watson Platform
- Node red
- Cloudant DB
- Web UI
- Python code

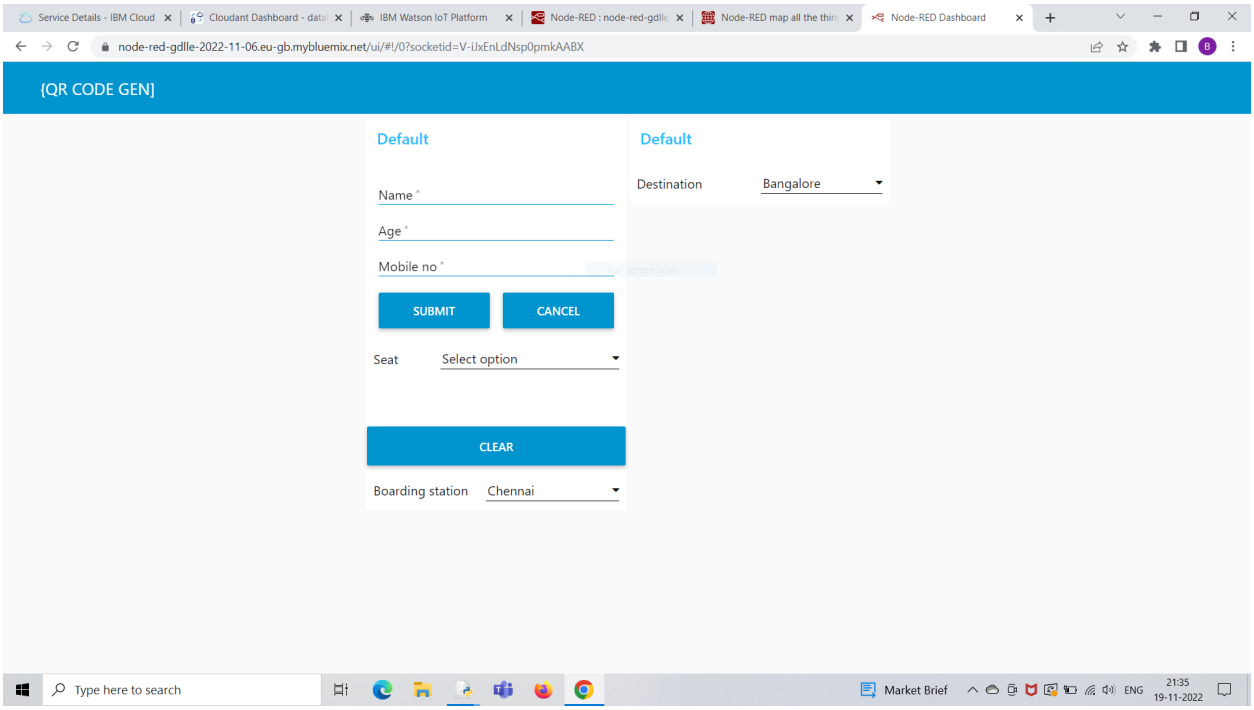
7.2 FEATURE 2

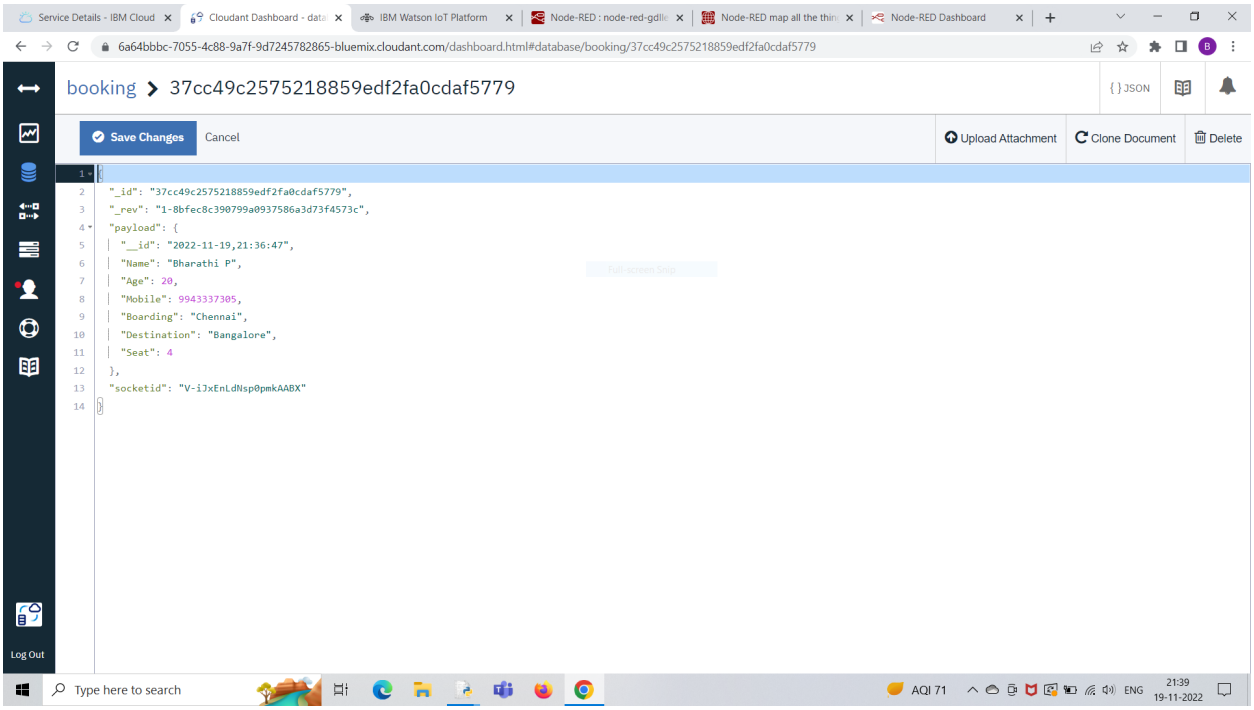
1. Login
2. Ticket Booking
3. QR Generation
4. Verification

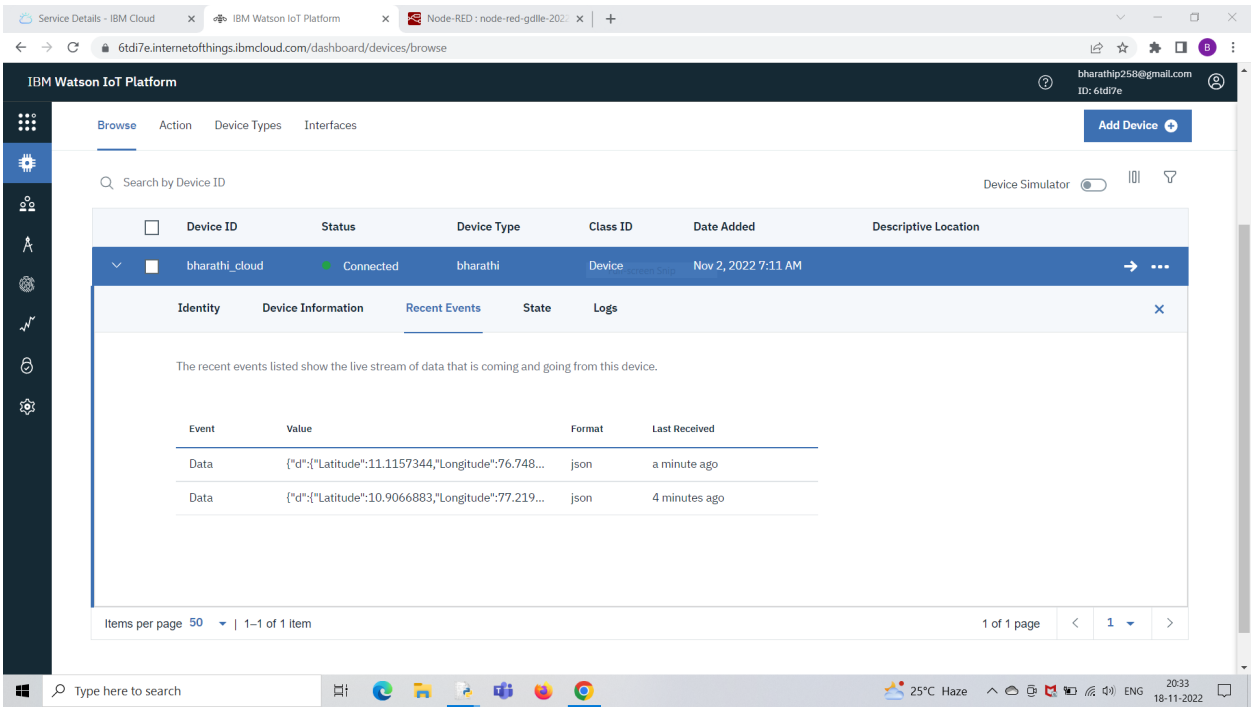
8. TESTING

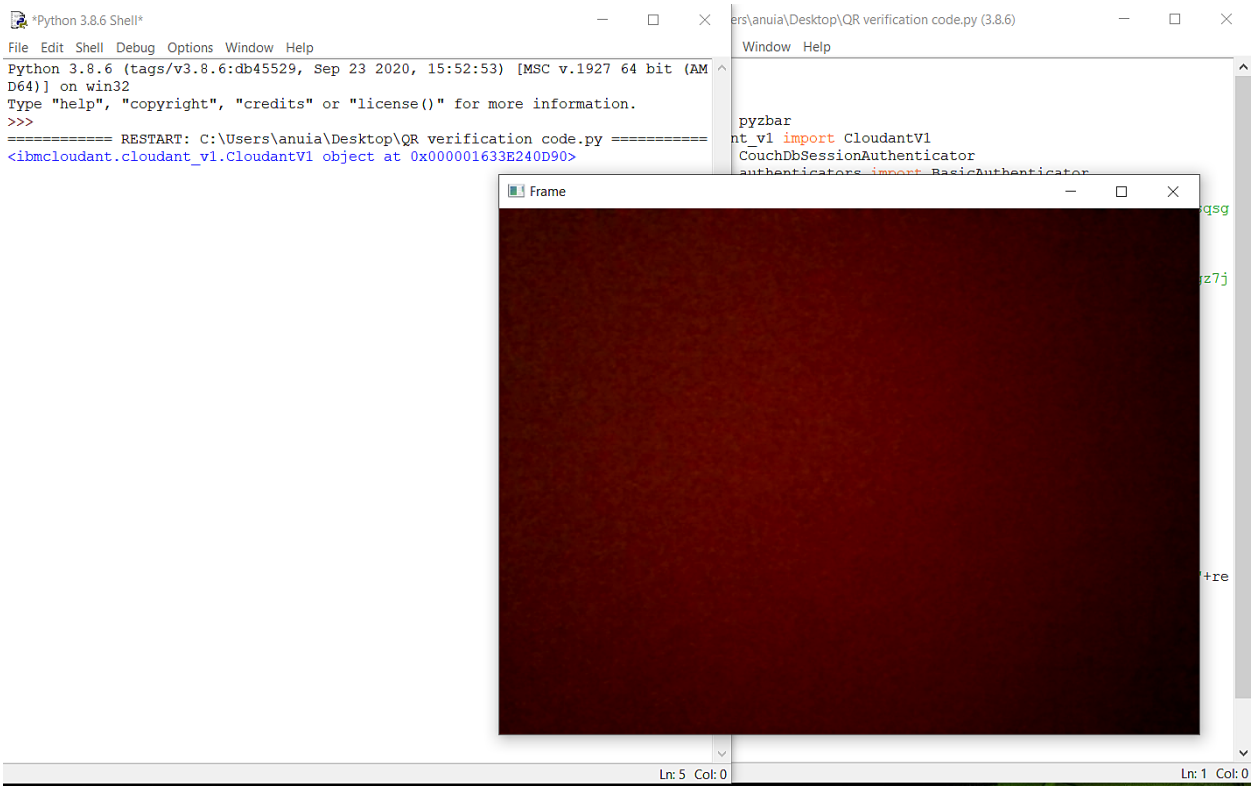
It has passed all the test cases.

9. RESULTS


Service Details - IBM Cloud
Cloudant Dashboard - data
IBM Watson IoT Platform
Node-RED : node-red-gdlle
Node-RED map all the thin
Node-RED Dashboard
node-red-gdlle-2022-11-06.eu-gb.mybluemix.net/ui/#!/0?socketid=V-iJxEnLdNsp0pmkAABX
{QR CODE GEN}
Default
Name *
Age *
Mobile no *
SUBMIT
CANCEL
Seat
Select option
CLEAR
Boarding station
Chennai
Default
Destination
Bangalore
Type here to search
Market Brief
ENG
21:35
19-11-2022


Service Details - IBM Cloud
Cloudant Dashboard - data
IBM Watson IoT Platform
Node-RED : node-red-gdlle
Node-RED map all the thin
Node-RED Dashboard
6a64bbbc-7055-4c88-9a7f-9d7245782865-bluemix.cloudant.com/dashboard.html#database/booking/37cc49c2575218859edf2fa0cdaf5779
booking > 37cc49c2575218859edf2fa0cdaf5779
{ } JSON
Save Changes
Cancel
Upload Attachment
Clone Document
Delete
{
"_id": "37cc49c2575218859edf2fa0cdaf5779",
"_rev": "1-8bfec8c390799a0937586a3d73f4573c",
"payload": {
"__id": "2022-11-19,21:36:47",
"Name": "Bharathi P",
"Age": 20,
"Mobile": 9943337305,
"Boarding": "Chennai",
"Destination": "Bangalore",
"Seat": 4
},
"socketid": "V-iJxEnLdNsp0pmkAABX"
}
Log Out
Type here to search
AQI 71
ENG
21:39
19-11-2022


Service Details - IBM Cloud
IBM Watson IoT Platform
Node-RED : node-red-gdlle-202
6tdi7e.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM Watson IoT Platform
bharathip258@gmail.com
ID: 6tdi7e
Browse
Action
Device Types
Interfaces
Add Device
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
bharathi_cloud
Connected
bharathi
Device
Nov 2, 2022 7:11 AM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
Data
{"d":{"Latitude":11.1157344,"Longitude":76.748...
json
a minute ago
Data
{"d":{"Latitude":10.9066883,"Longitude":77.219...
json
4 minutes ago
Items per page 50
1–1 of 1 item
1 of 1 page
1
Type here to search
25°C Haze
ENG
2033
18-11-2022


Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\anuia\Desktop\QR verification code.py ==========
<ibmcloudant.cloudant_v1.CloudantV1 object at 0x000001633E240D90>
ers\anuia\Desktop\QR verification code.py (3.8.6)
Window Help
pyzbar
nt_v1 import CloudantV1
CouchDbSessionAuthenticator
Frame
Ln: 5 Col: 0
Ln: 1 Col: 0

10. ADVANTAGES AND DIADVANTAGES

10.1 ADVANTAGES

1. The passengers can use this application, while they are travelling alone to ensure their safety.
2. It is easy to use.
3. It has minimized error rate.
4. Time saving.

10.2.DISADVANTAGES

1. Network issue may arise

11. CONCLUSION:

Almost all the countries across the globe strive to meet the demand for safe, fast, and reliable rail services. Lack of operational efficiency and reliability, safety, and security issues, besides aging railway systems and practices are haunting various countries to bring about a change in their existing rail infrastructure. The global rail industry struggles to meet the increasing demand for freight and passenger transportation due to lack of optimized use of rail network and inefficient use of rail assets. Often, they suffer from the lack in smart technologies and latest technological updates to provide the most efficient passenger services. This is expected to induce rail executives to build rail systems that are smarter and more efficient. The passenger reservation system of Indian Railways is one of the world's largest reservation models. Daily about one million passengers travel in reserved accommodation with Indian Railways. Another sixteen million travel with unreserved tickets in Indian Railways. In this vast system, it is a herculean task to efficiently handle the passenger data, which is a key point of consideration now-a-days. But the implementation of the latest technological updates in this system gradually turns inevitable due to increasing demand for providing the most efficient passenger services. Handling the passenger data efficiently backed by intelligent processing and timely retrieval would help backing up the security breaches. Here we've explored different issues of implementing smart computing in railway systems pertaining to reservation models besides pointing out some future scopes of advancement.

Most significant improvements have been evidenced by more informative and user-friendly websites, mobile applications for real-time information about vehicles in motion, and e-ticket purchases and timetable information implemented at stations and stops. With the rise of Industry, railway companies can now ensure that they are prepared to avoid the surprise of equipment downtime. Like above mentioned, the developed application of our project can lead the passenger who travel can travel safely without any fear

12. FUTURE SCOPE

This application is ensured for safety for the passengers while they are travelling alone as well as they travel with their family or friends. In future, this application may also be used by passengers who travel through bus. By further enhancement of the application the passengers can explore more features regarding their safety.

13. APPENDIX

13.1 SOURCE PROGRAM

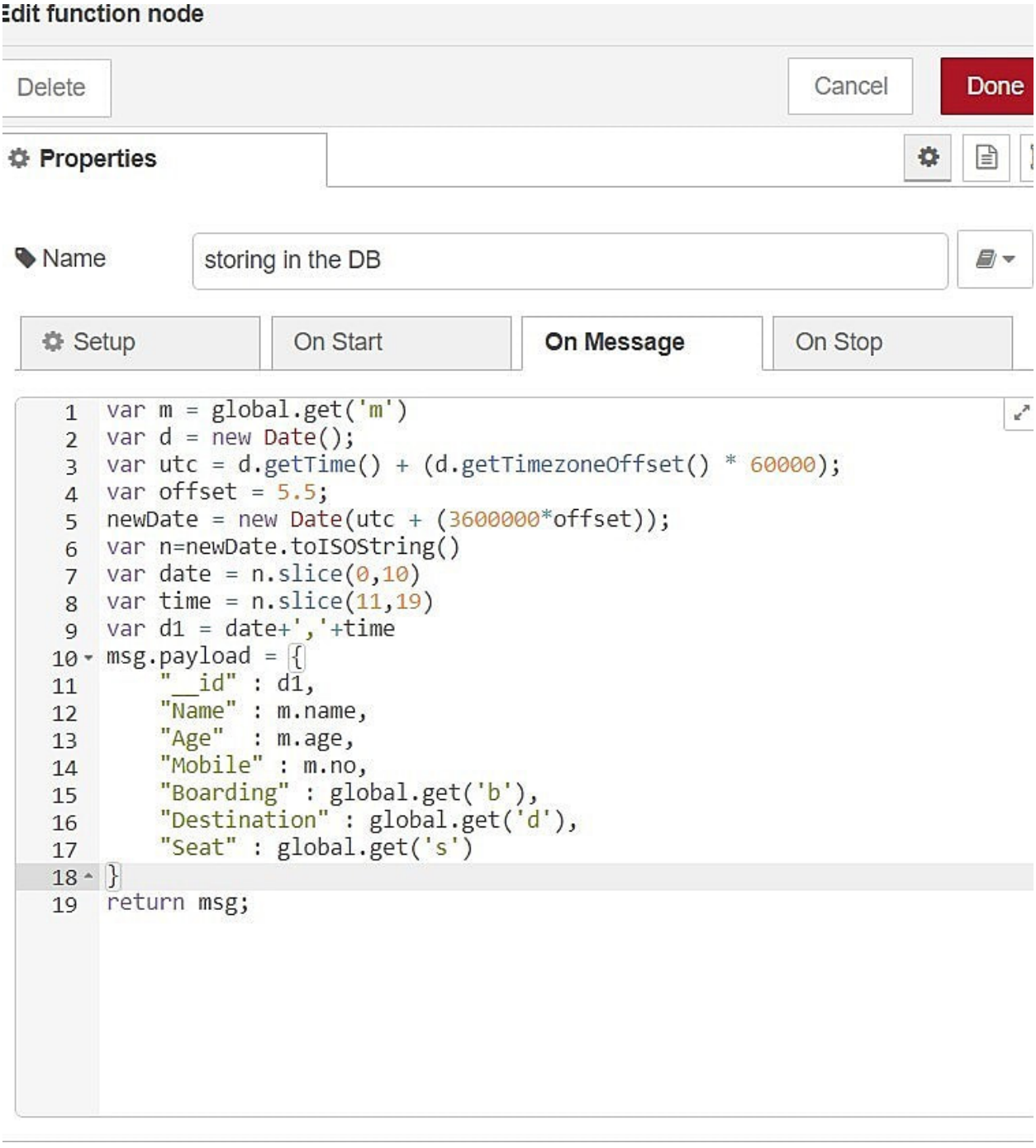
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import requests
import json

#Provide your IBM Watson Device Credentials
organization = "be2fcf"
deviceType = "GPS"        #Credentials of Watson IoT sensor
simulator
deviceId = "TrainGPS"
authMethod = "token"
authToken = "pa2tTjP1b3VRKFFP7R"



# Initialize the device client.
L=0

try:
        deviceOptions = {"org":organization , "type": deviceType ,
"id":deviceId , "auth-method":authMethod , "auth-token":authToken}
```

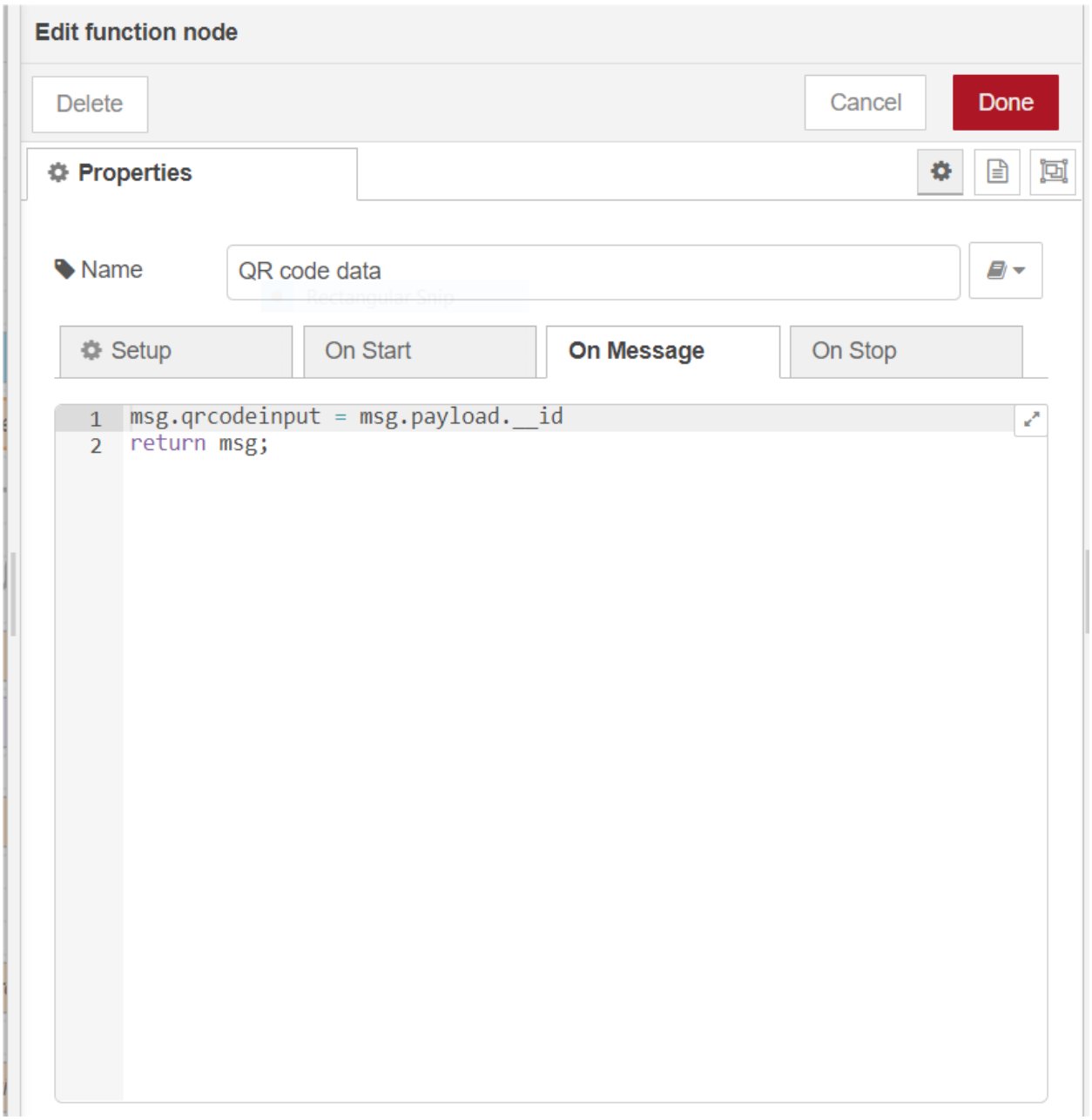
```
        deviceCli = ibmiotf.device.Client(deviceOptions)
        #.....................................

except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
        sys.exit()

# Connect and send a datapoint "hello" with value "world" into the
cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
    overpass_url = "http://overpass-api.de/api/interpreter"
    overpass_query = """
    [out:json];area[name="India"];(node[place="village"](area););out;
    """

    response = requests.get(
    overpass_url,
    params={'data': overpass_query}
    )

    coords = []
    if response.status_code == 200:
        data = response.json()
        places = data.get('elements', [])
        for place in places:
            coords.append((place['lat'], place['lon']))
```

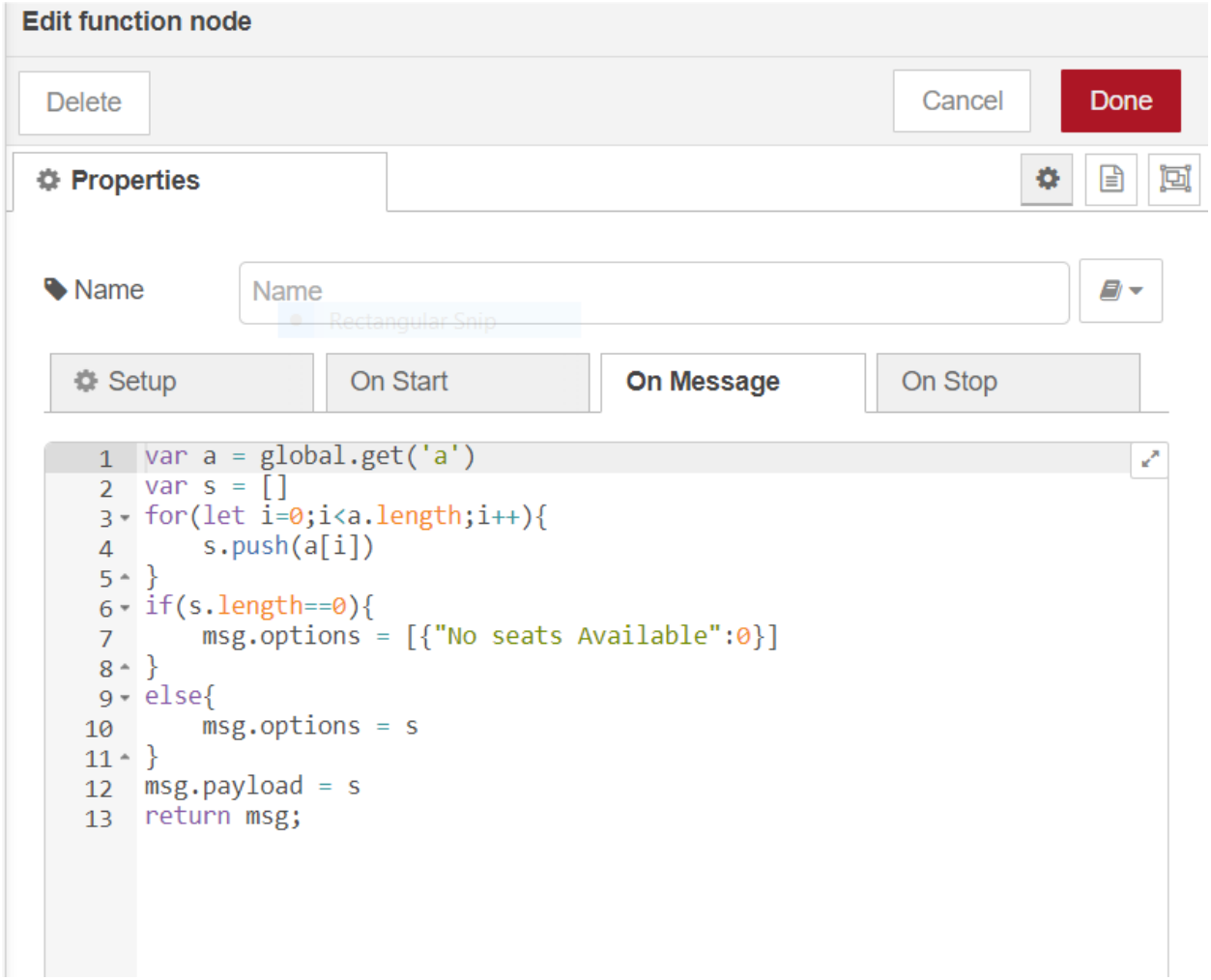
```
        print ("Got %s village coordinates!" % len(coords))
        print (coords[0])
    else:
        print("Error")

    i = random.randint(1,100)
    L = coords[i]
    #Send random gprs data to node-red to IBM Watson
    data = {"d":{ 'Latitude' : L[0], 'Longitude' : L[1]}}
    #print data
    def myOnPublishCallback():
        print("Published gprs location = ", L, "to IBM Watson")

    success = deviceCli.publishEvent("Data", "json", data, qos=0,
on_publish=myOnPublishCallback)
    time.sleep(12)
    if not success:
        print("Not connected to IoTF")
    time.sleep(1)

deviceCli.disconnect()


import cv2
import time
import numpy as np
import pyzbar.pyzbar as pyzbar
```

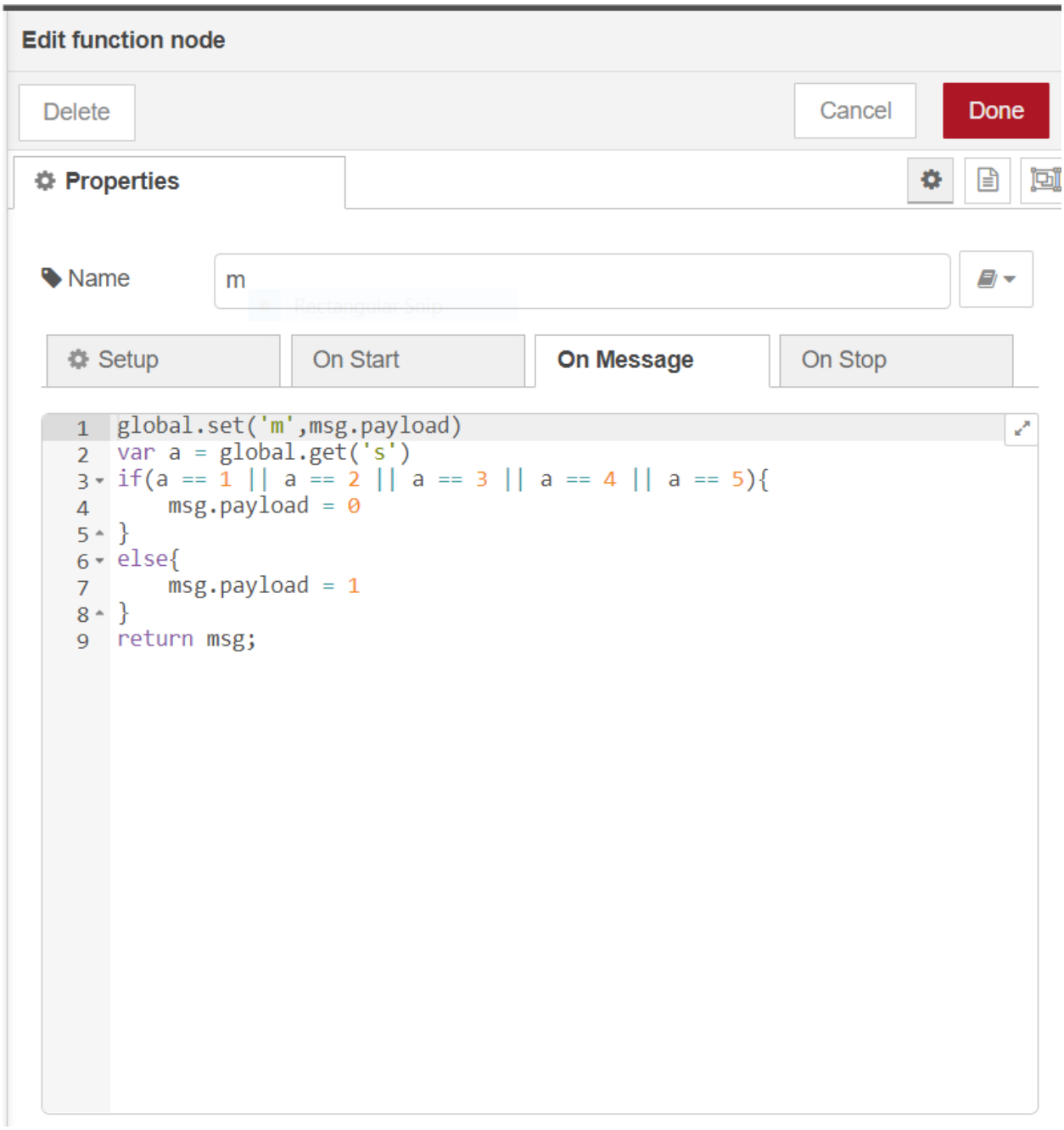
```
from ibmcloudant.cloudant_v1 import CloudantV1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import
BasicAuthenticator

authenticator = BasicAuthenticator('apikey-v2-
w7kt9ti5ys9p91k2xsuynuhehlbjw5sqsgz7jllg05h','11b2e2c3a290c
04886ab57bf26f3604b')

service = CloudantV1(authenticator=authenticator)
print(service)
service.set_service_url('https://apikey-v2-
w7kt9ti5ys9p91k2xsuynuhehlbjw5sqsgz7jllg05h:11b2e2c3a290c
04886ab57bf26f3604b@6a64bbbc-7055-4c88-9a7f-
9d7245782865-bluemix.cloudantnosqldb.appdomain.cloud')

cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_PLAIN

while True:
    _, frame = cap.read()
    decodedObjects = pyzbar.decode(frame)
    for obj in decodedObjects:
        a=obj.data.decode('UTF-8')
```

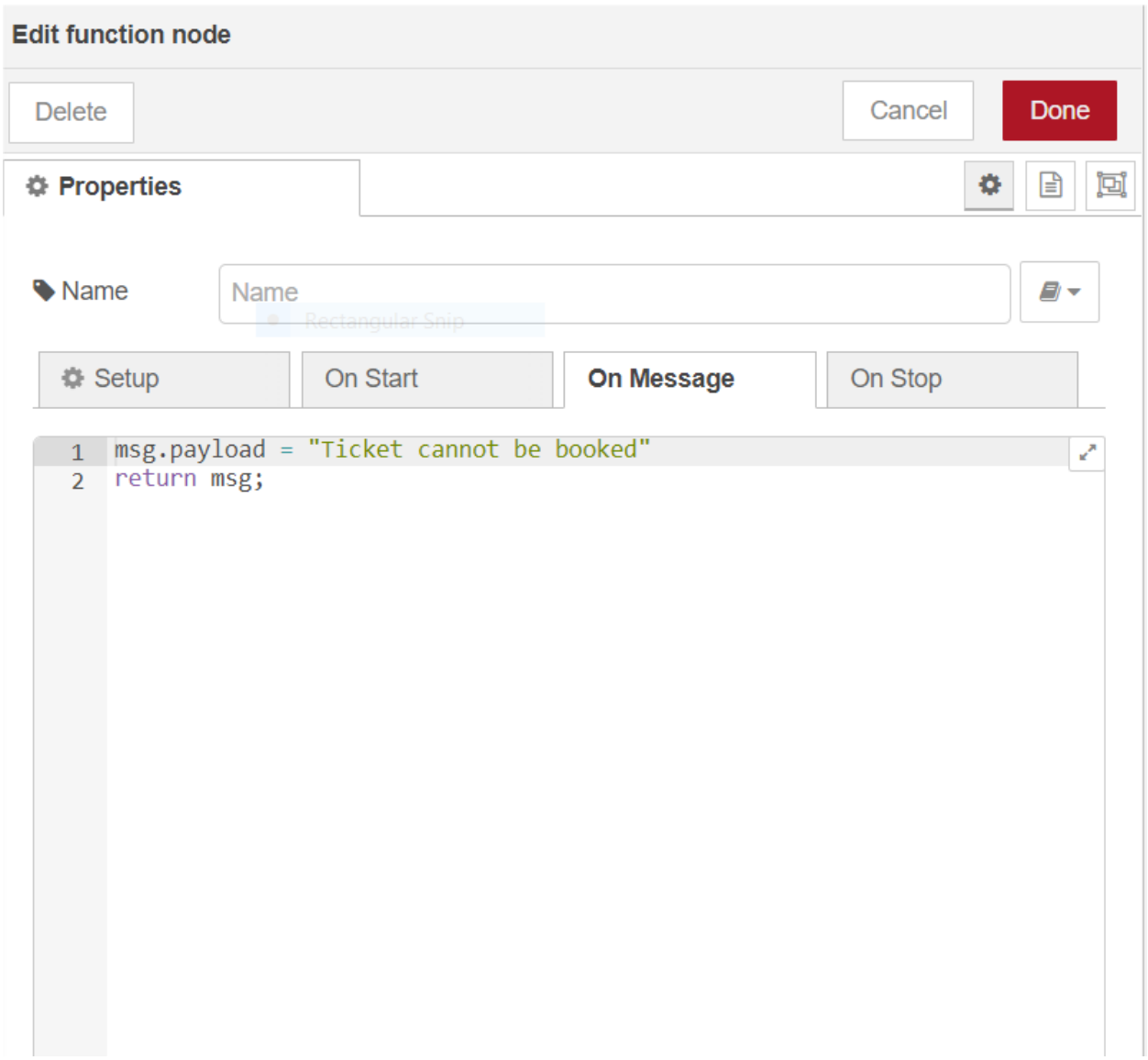
```
        cv2.putText(frame, "Ticket", (50, 50), font, 2,(255, 0, 0), 3)
        try:
            response = service.get_document(
                db='booking-table',
                doc_id = a
                ).get_result()
            print("Passenger Name\t Ticket Count \t Date \t Train NO")
            print(response["pname"]+"\t
\t"+str(response["ticketcount"])+"\t"+response["date"]+"\t"+str(res
ponse["trainNumber"]))
            time.sleep(5)
        except Exception as e :
             print(e)
             print("Not a Valid Ticket")
             time.sleep(5)

    cv2.imshow("Frame", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
client.disconnect()
```


Edit function node
Delete
Cancel
Done
Properties
Name
storing in the DB
Setup
On Start
On Message
On Stop
1 var m = global.get('m')
2 var d = new Date();
3 var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
4 var offset = 5.5;
5 newDate = new Date(utc + (3600000*offset));
6 var n=newDate.toISOString()
7 var date = n.slice(0,10)
8 var time = n.slice(11,19)
9 var d1 = date+','+time
10 msg.payload = {
11 "__id" : d1,
12 "Name" : m.name,
13 "Age" : m.age,
14 "Mobile" : m.no,
15 "Boarding" : global.get('b'),
16 "Destination" : global.get('d'),
17 "Seat" : global.get('s')
18 }
19 return msg;


Edit function node
Delete
Cancel
Done
Properties
Name
QR code data
Setup
On Start
On Message
On Stop
1 msg.qrcodeinput = msg.payload.__id
2 return msg;


Edit function node
Delete
Cancel
Done
Properties
Name
Name
Setup
On Start
On Message
On Stop
1 var a = global.get('a')
2 var s = []
3 for(let i=0;i<a.length;i++){
4 s.push(a[i])
5 }
6 if(s.length==0){
7 msg.options = [{"No seats Available":0}]
8 }
9 else{
10 msg.options = s
11 }
12 msg.payload = s
13 return msg;


Edit function node
Delete
Cancel
Done
Properties
Name
m
Setup
On Start
On Message
On Stop
1 global.set('m',msg.payload)
2 var a = global.get('s')
3 if(a == 1 || a == 2 || a == 3 || a == 4 || a == 5){
4 msg.payload = 0
5 }
6 else{
7 msg.payload = 1
8 }
9 return msg;


Edit function node
Delete
Cancel
Done
Properties
Name
Name
Setup
On Start
On Message
On Stop
1 msg.payload = "Ticket cannot be booked"
2 return msg;

13.2 GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-16943-1659625475