DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROJECT BASED EXPERIENTIAL LEARNING PROGRAM (NALAIYA THIRAN)

SMART SOLUTIONS FOR RAILWAYS

A PROJECT REPORT

Submitted by

DHARANI S(912019106004)
VARSHA S(912019106016)
MAHESHWARI S
(912019106009)
PANDIRAMAN (912019106012)

TEAM ID : PNT2022TMID47932

CONTENTS

INTRODUCTION

1.1) PROJECT OVERVIEW

The goal of our project is to develop smart solutions for railways, it has been designed to reduce the work load of the user and also the use of paper. Here in this project, we have all the features shown below.

Features:

- A Web page is designed for the public where they can book tickets by seeing theavailable seats.
- After booking the train, the person will get a QR code which has to be shown tothe Ticket Collector while boarding the train.
- The ticket collectors can scan the QR code to identify the personal details.
- A GPS module is present in the train to track it. The live status of the journey is updated in the Web app continuously
- 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 QRCode.

1.2) PURPOSE

The main purpose behind our project is to speed up the process of ticketing and prevent wastage of time at the ticket counters present at railway stations. This also resolves the issue with the running status of rains where people get confused on where the train is located at. With the help of our service, people can book tickets oreasily locate and track trains from anywhere across the world.

We have also worked on easing the task of ticket collectors to check and verify tickets of passengers. A unique QR code is given to every customer upon a successful transaction which is scanned later by the ticket collector for verification. Once the ticket collector scans the QR code, he gets full details of the person in journey.

LITERATURE SURVEY

2.1) EXISTING PROBLEM

Even after significant improvement in technology, the major problem that still persistsin our railway system is the waiting at ticket counters for long durations and confusion regarding the running of trains whether the passengers were early or missed it.

These problems gave burden to the passengers waiting at the ticketing kiosks by wasting their time which they can use it for their prioritized tasks, and the confusions regarding the train running status gave rise to unnecessary worries to the commuters.

2.2) REFERENCES

1. Roman Khoeblal, Teeravisit Laohapensaeng, Roungsan Chaisricharoen, “PassengerMonitoring Model for easily Accessible Public City Trams/Trains” (2015).

2. Parag Chatterjee, Asoke Nath, “Application of smart computing in IndianRailwaySystems” (2014).

3. Sana Khoja, Maithili Kadam, “Android Suburban Railway Ticketing with GPS as TicketChecker” (2012).

4. Sujith Kumar, K.M.Yatheendra Parvan, V.Sumathy, Thejeswari C.K, “Novel Approachfor Smart Indian Railways” (2017).

5. Sarvath Saba, Sharon Philip, Shri harsha, Mukund Naik, Sudeep Sherry, “A Reviewon IOT based automated seat allocation and verification using QR code”(2022)

2.3) Problem Statement:

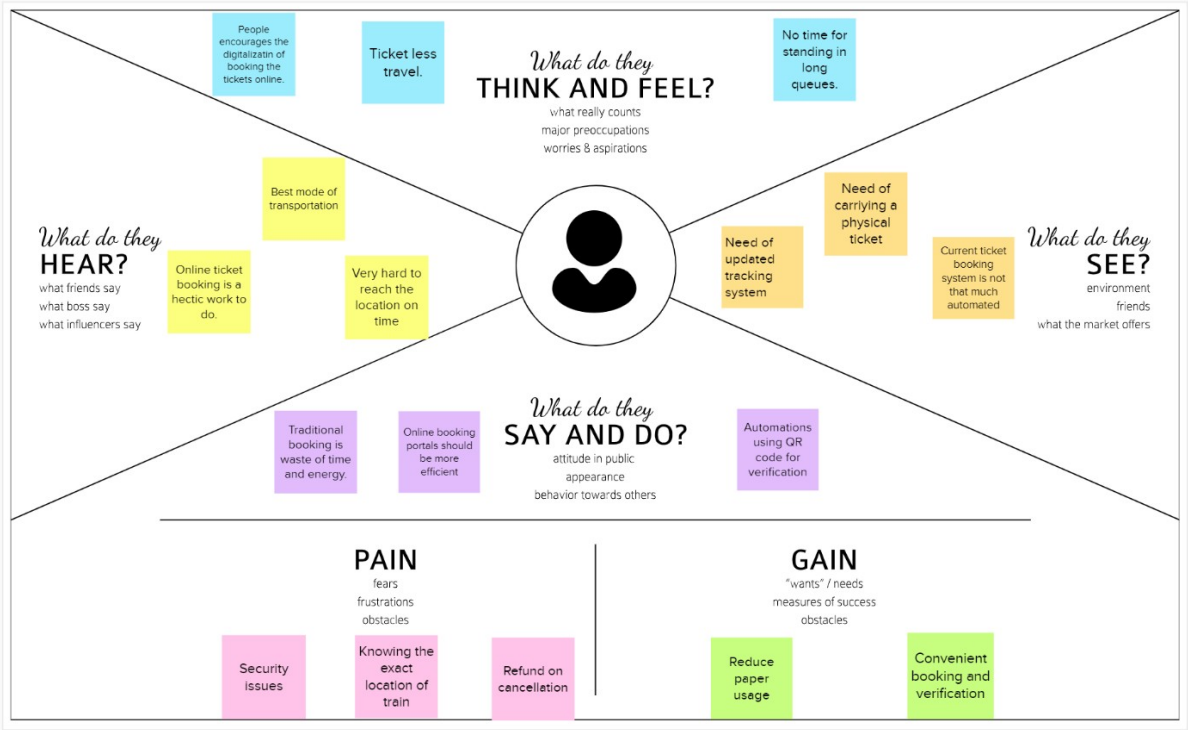
1. People wait for long duration at the ticketing queues to purchase a train ticket

2. Ticket collector goes through different documents to verify the ticket of anindividual, this could be a time-consuming process.

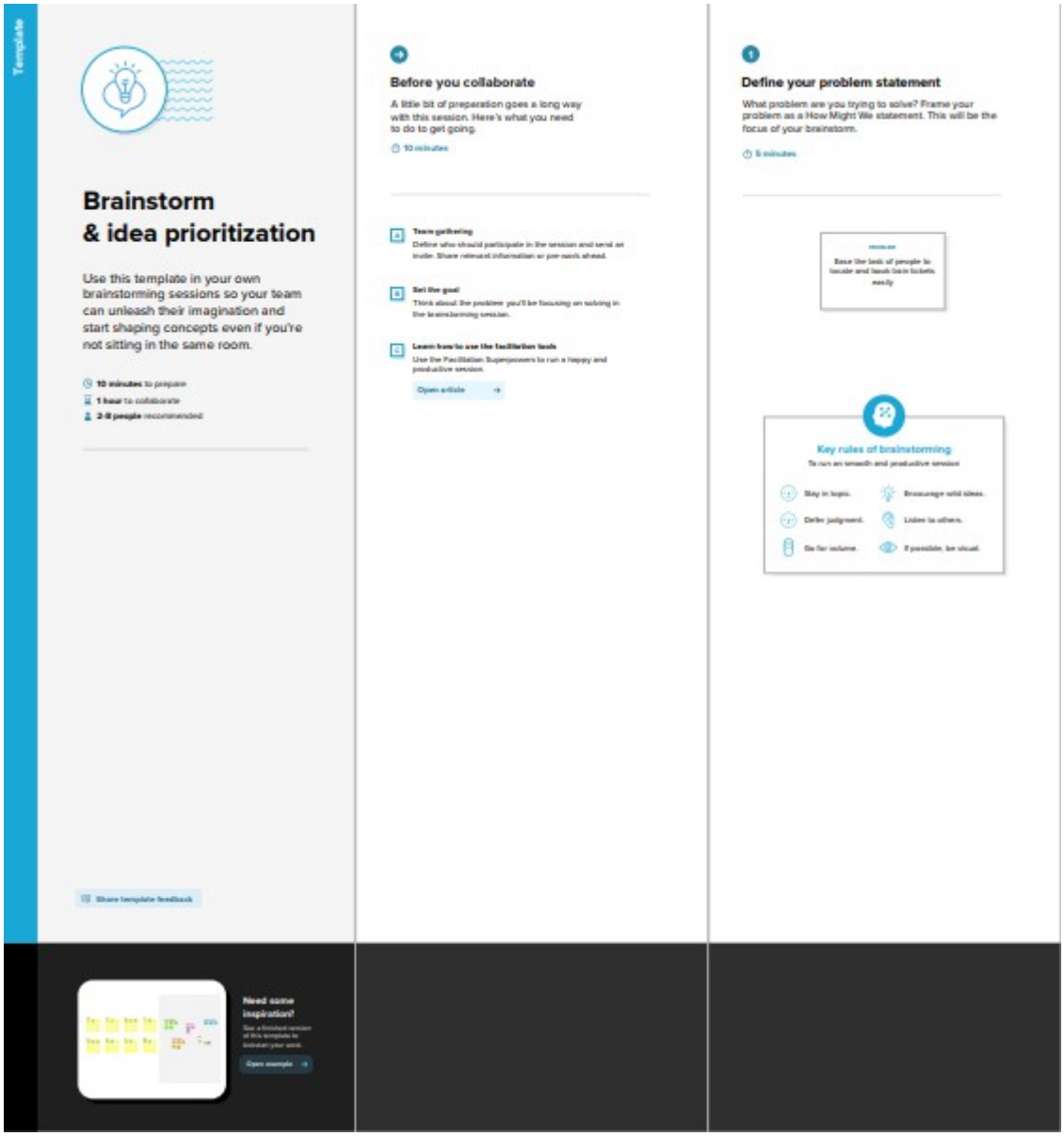
3. People are unaware of the train running status, if they were early or missed it.

IDEATION AND PROPOSED SOLUTION

3.1) EMPATHY MAP CANVAS



3.2) IDEATION AND BRAINSTORMING:



2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Harsha

Jithendra

Kaushik

Manikanta

3 Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

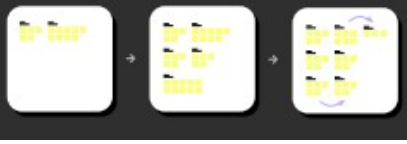
Add custom[sticky tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Affordable

Convenience

Security

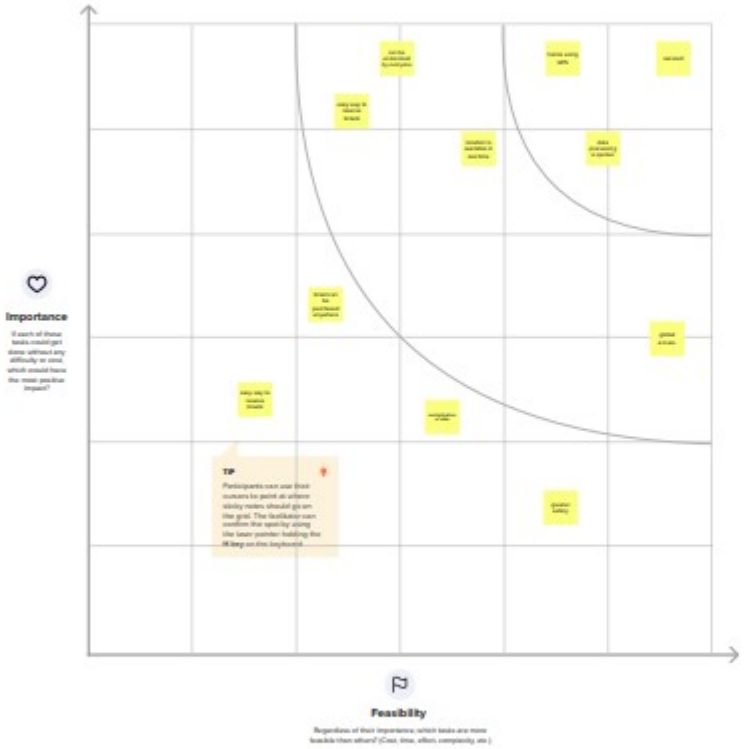
Data Processing



Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

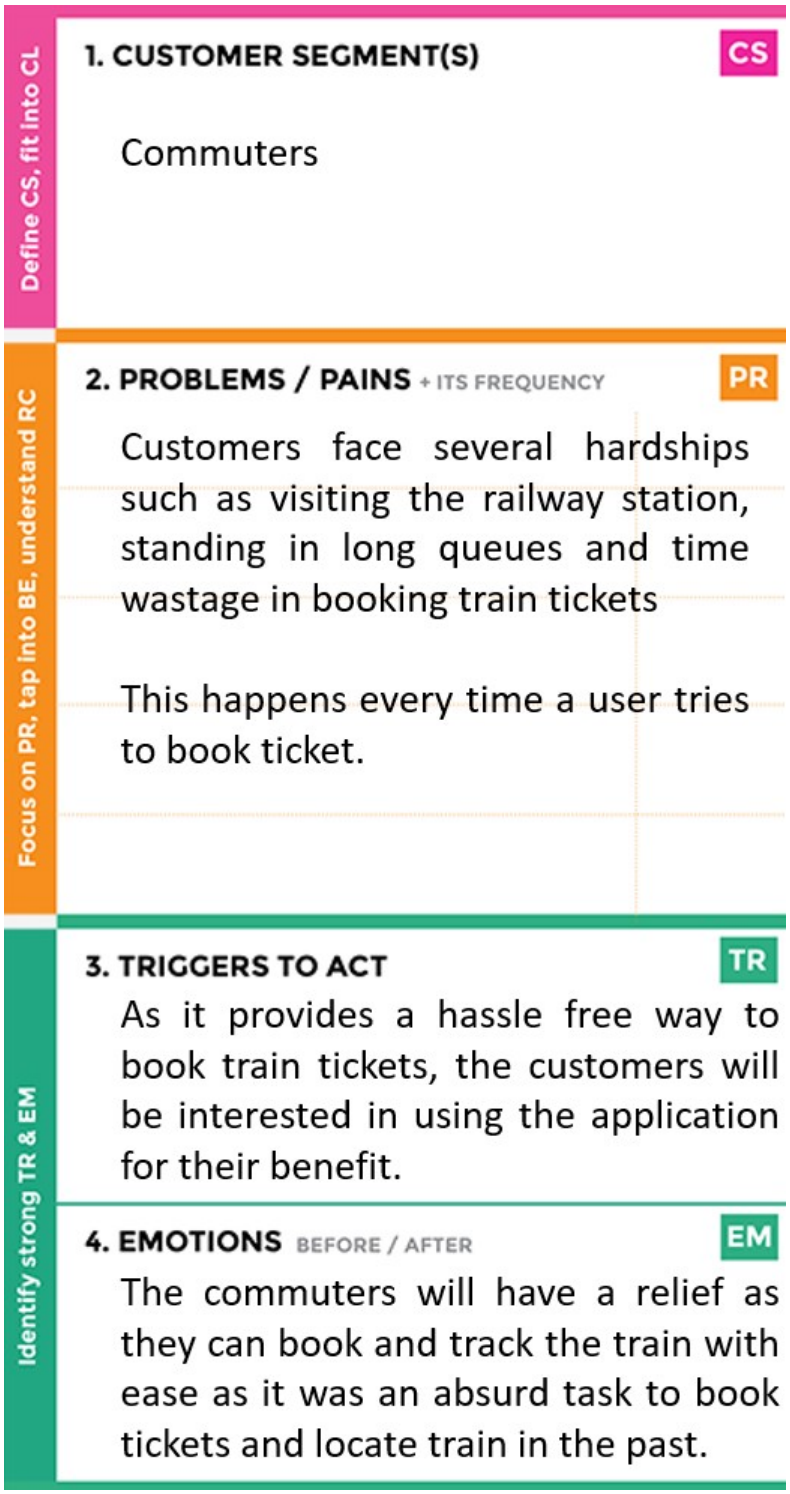
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3) PROPOSED SOLUTION

- A Web page is designed for the public where they can book tickets by seeing the available seats. After booking the train, the person will get a QR code which hasto be shown to the Ticket Collector while boarding the train.
- The ticket collectors can scan the QR code to identify the personal details.
- A GPS module is present in the train to track it. The live status of the journey is updated in the Web app continuously making it easier for people to locate it.

3.4) PROBLEM SOLUTION FIT



6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES — CL

Customer should have access to a smartphone when needed and the internet connectivity should be reliable to carry out processing without errors

5. AVAILABLE SOLUTIONS PROS & CONS — AS

Tickets are booked via ticket counter at railway stations, but they demand long free time of people.
People had to visit the station every time they need tickets.

Explore AS, differentiate

9. PROBLEM ROOT / CAUSE — RC

In the past, lack of technology gave rise to this problem.

Lack of awareness is still causing this problem.

7. BEHAVIOR + ITS INTENSITY — BE

Taking feedback on different aspects varying from 1 to 10

Focus on PR, tap into BE, understand RC

10. YOUR SOLUTION — SL

Web page is designed for the public where they can book tickets.
GPS module is present in the train to track it

8. CHANNELS of BEHAVIOR — CH

ONLINE

By providing feedback or filling an online form

OFFLINE

Visiting the office present at major railway stations or contacting customer care through call

Extract online & offline CH of BE

REQUIREMENT ANALYSIS

There are two types: i) Functional Requirements

ii) Non – Functional Requirements

4.1) FUNCTIONAL REQUIREMENTS

- Registration
  - Through Email
  - Through Phone Number
  - Through Google, Facebook
- User Confirmation
  - Email Verification
  - OTP
  - Confirmation via Call
- Journey Details
  - To and from station information
  - Class of Travel
  - Date of Travel
- Train Information
  - Available seats
  - Class availability
  - Quota (General, Senior citizen, Tatkaal)
- Passenger Information
  - Name
  - Gender
  - Date Of Birth
  - Citizen Proof
  - Contact Details
- Payment
  - Through Cards
  - Through Net Banking
  - Through UPI/Wallet
  - Through Voucher
- Information Storage
  - Databases to store passenger details
  - Databases to store train details
  - QR code generator
- Train Tracking
  - GPS module API
  - Real time location upload

4.2) NON - FUNCTIONAL REQUIREMENTS

- Usability

  - Server should be responsive

  - It should accommodate a greater number of users

  - Should be maintained regularly

- Reliability

  - User should have a secure transaction

  - User data should be protected from third party access

- Performance

  - Sever should be working 24x7

  - Server should not experience any crashed due to overload

  - It should be responsive

- Availability

  - Ticketing should be available through website or app

  - Offline ticketing must be present at railway stations

- Scalability

  - Users interacting with our service should feel secured and reliable

- Security

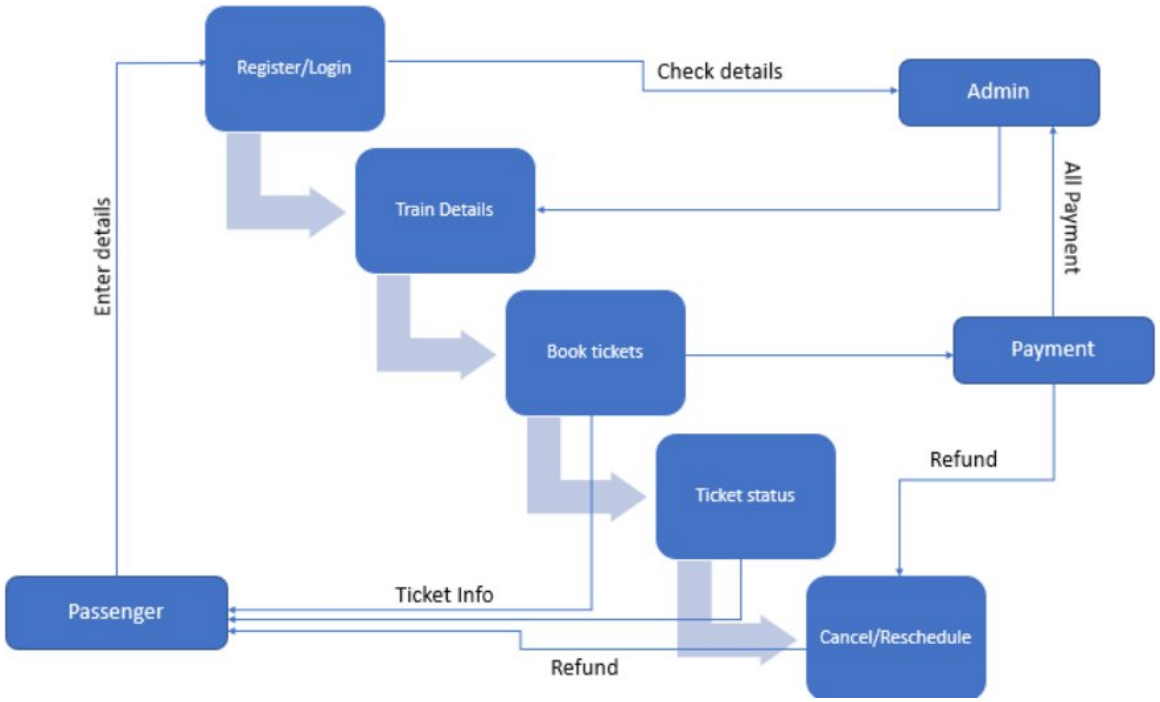
  - User data must be protected

  - Encryption of user data is needed to prevent third party from accessing it

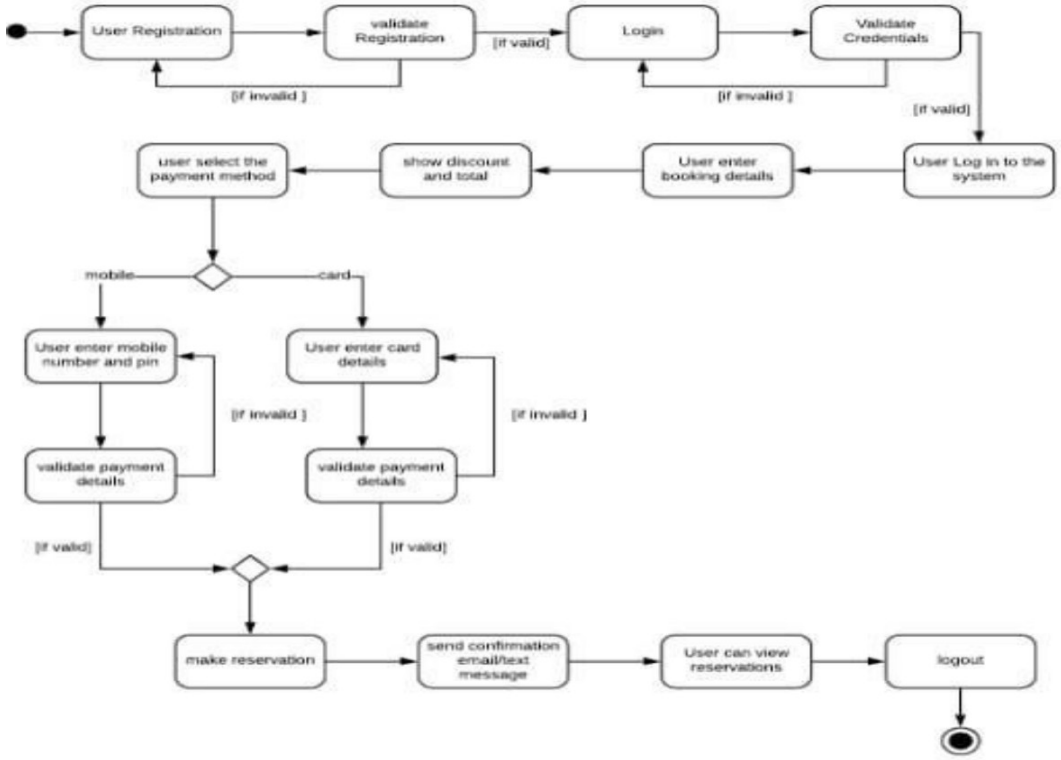
  - Payments should take place through a secure portal

PROJECT DESIGN

5.1) DATA FLOW DIAGRAMS



5.2) SOLUTION AND TECHNICAL ARCHITECTURE



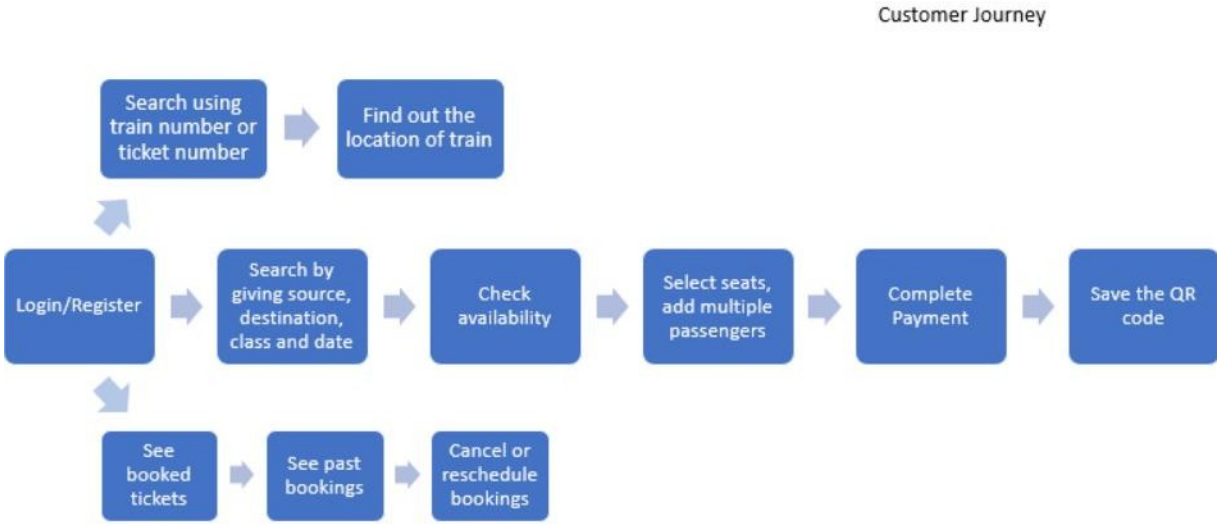
COMPONENTS AND TECHNOLOGIES

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g.Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js /React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other StorageService or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / CloudLocal Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

APPLICATION CHARACTERISTICS

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented,use of firewalls etc.	e.g. SHA-256, Encryptions, IAMControls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of theapplication (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

5.3) CUSTOMER JOURNEY / USER STORIES



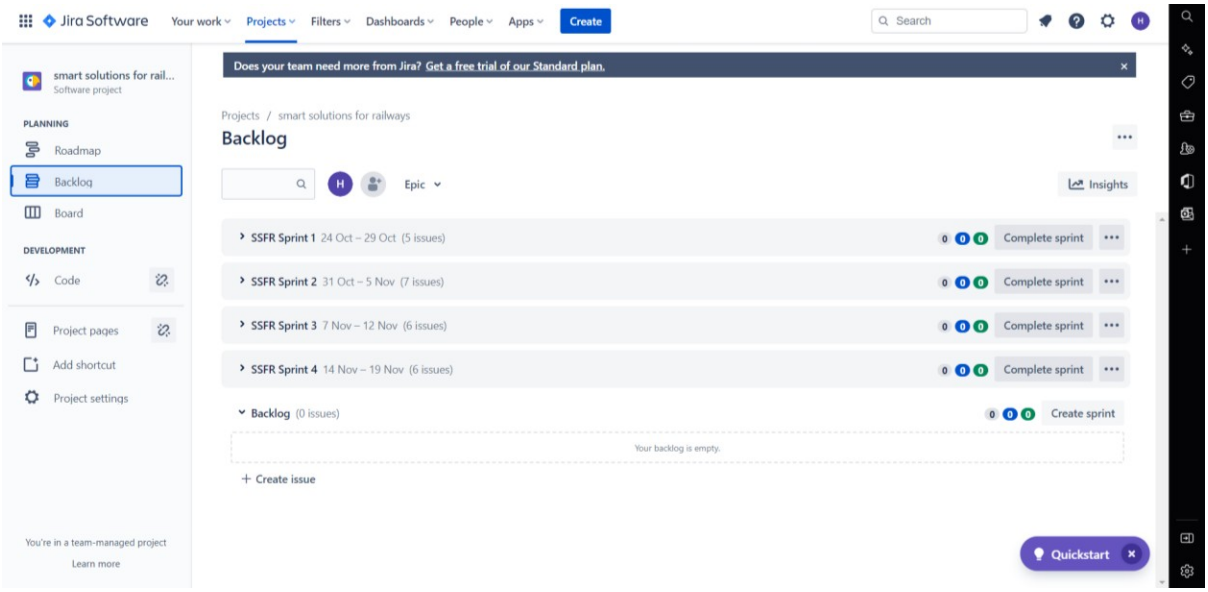
PROJECT PLANNING AND SCHEDULING

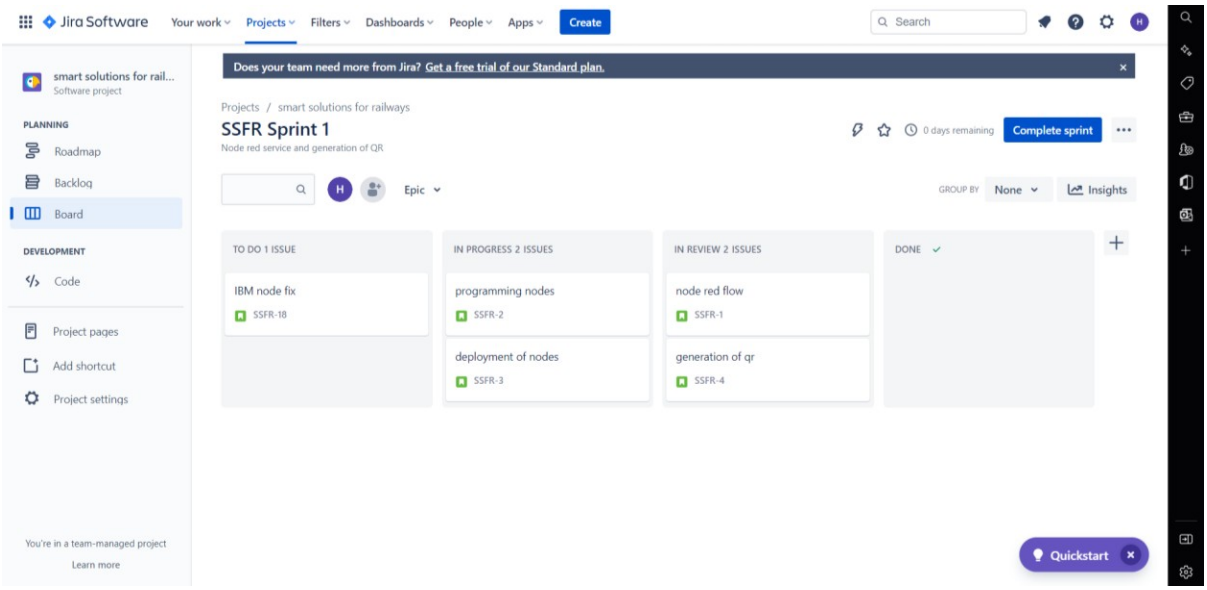
6.1) SPRINT DELIVERY PLANNING AND ESTIMATION

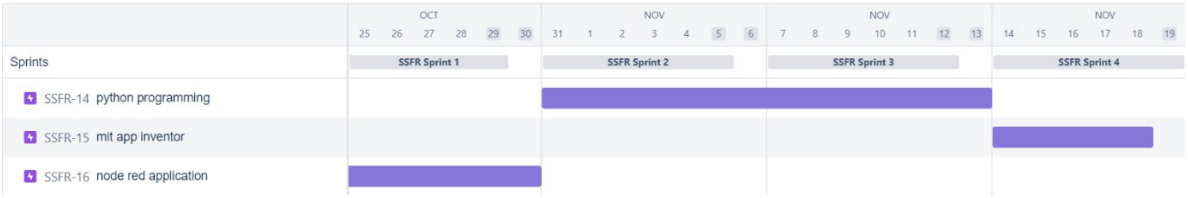
Sprint	Total Story Points	Duration	Sprint Start Date
Sprint-1	20	6 Days	24 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022

Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
29 Oct 2022	20	29 Oct 2022
05 Nov 2022	20	05 Nov 2022
12 Nov 2022	20	12 Nov 2022
19 Nov 2022	20	19 Nov 2022

6.3) JIRA REPORTS







CODING AND SOLUTIONING

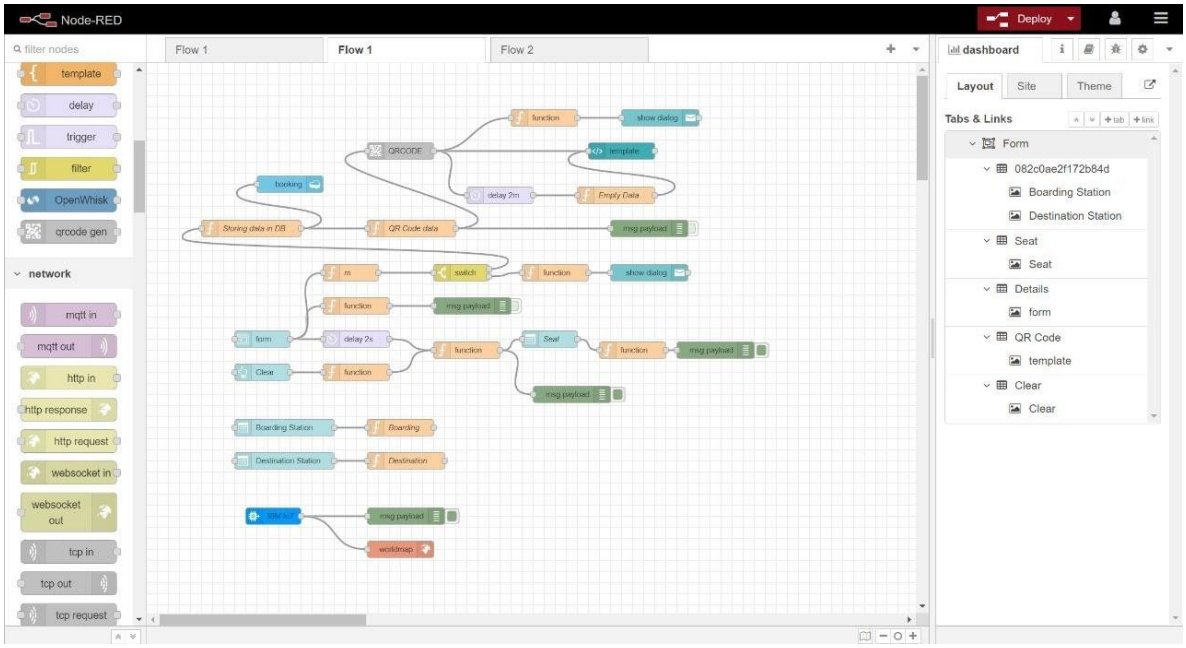
7.1) FEATURE 1

NODE RED SERVICE

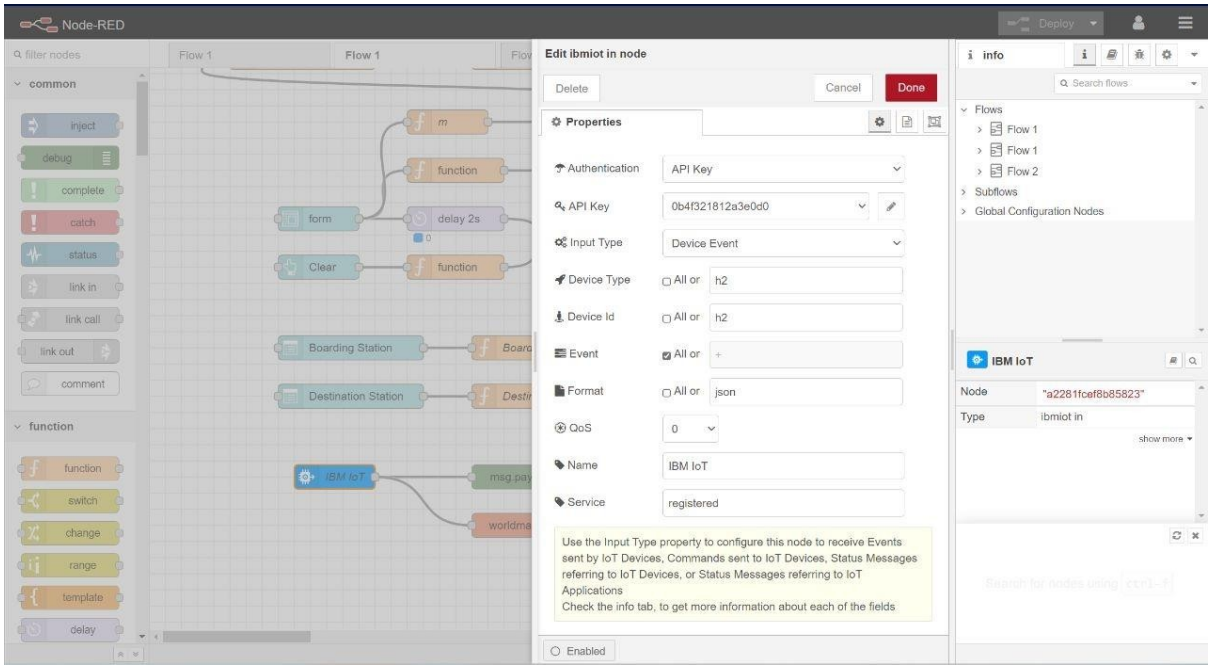
Main application

- Connect to cloud
- View map
- Searching for trains
- Storing database
- Booking
- QR code generation

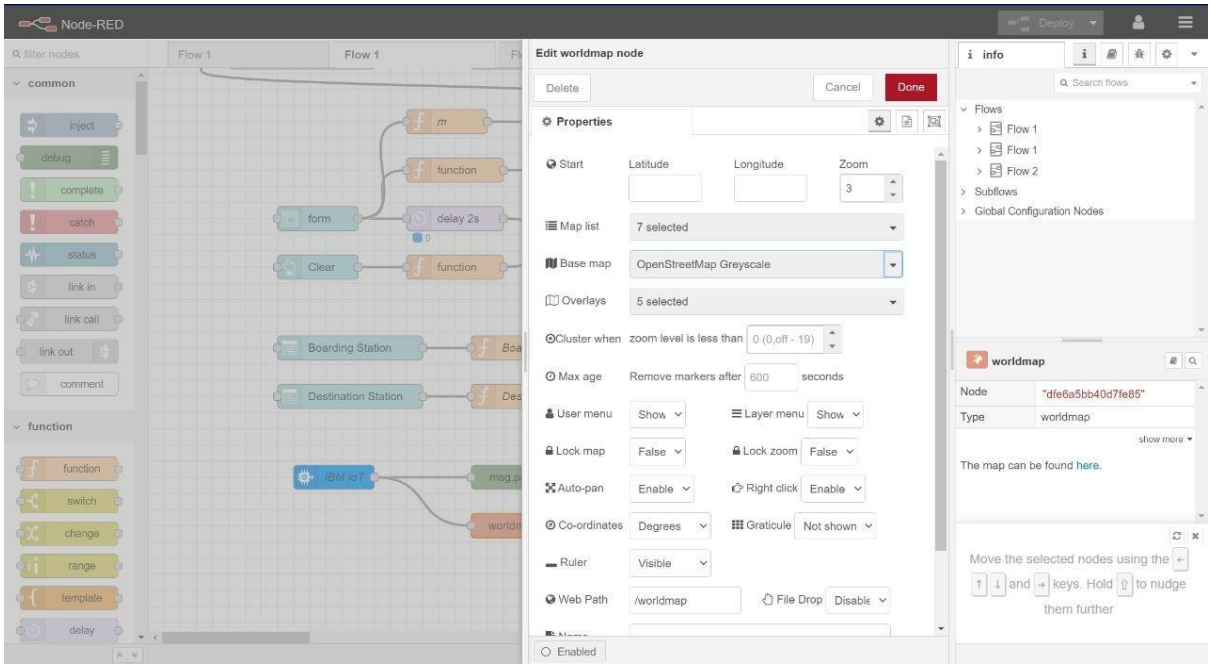
FLOWCHART



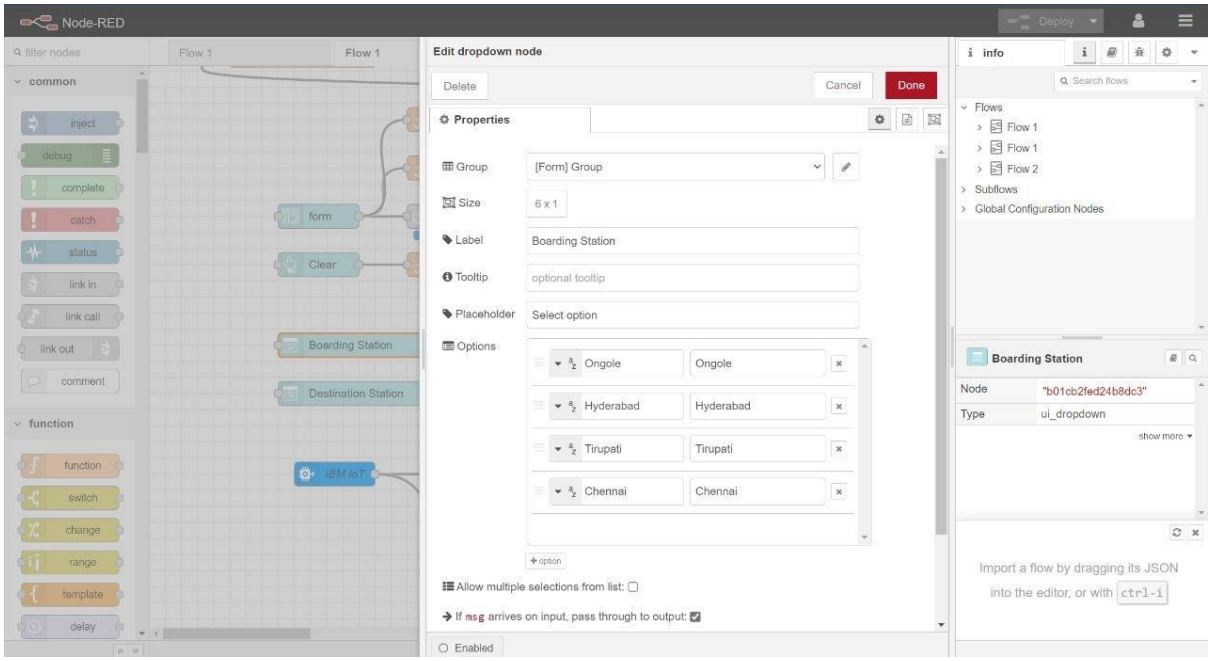
IBM IOT



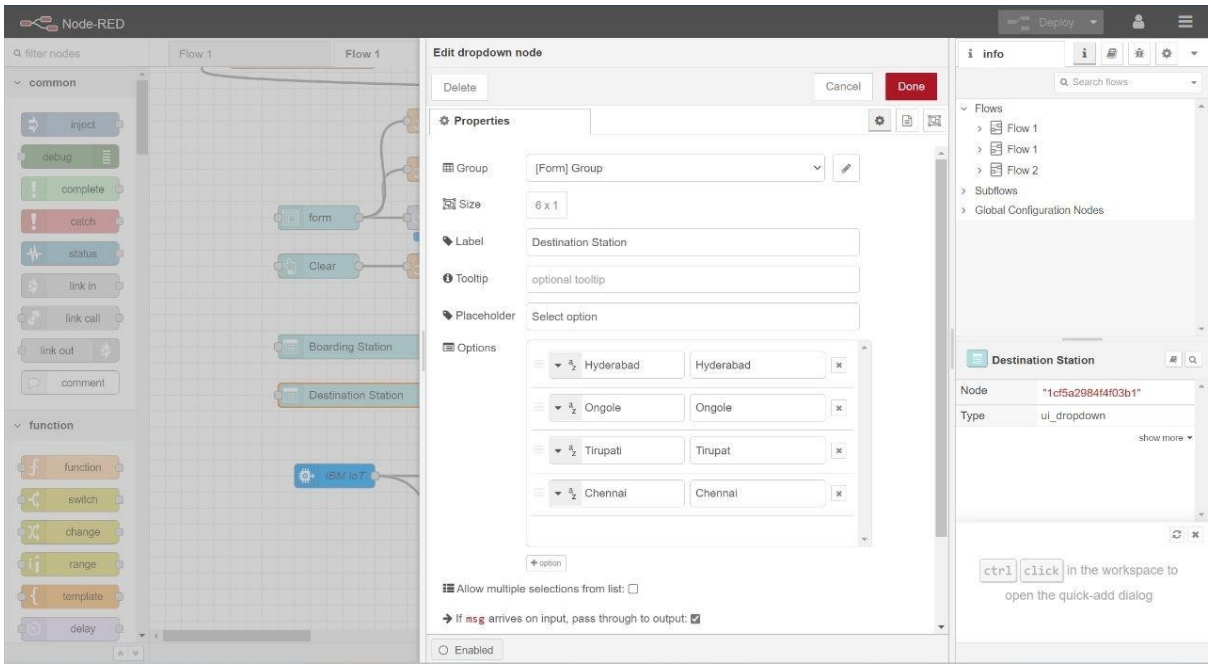
FOR WORLD MAP



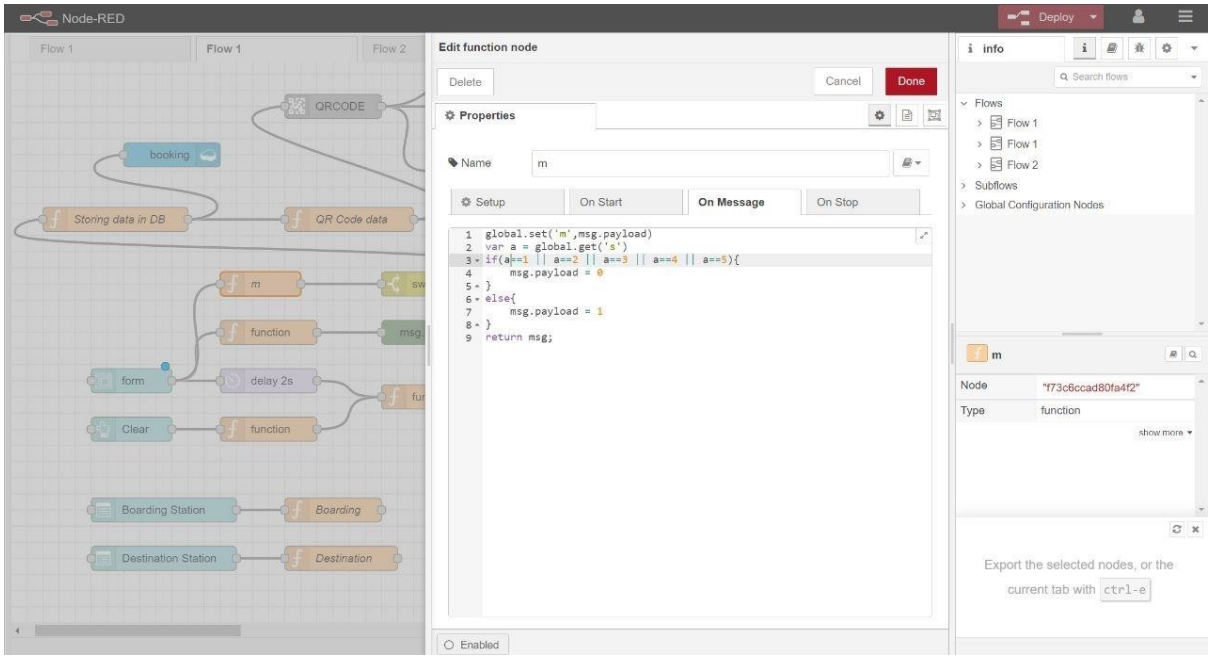
FOR BOARDING STATION



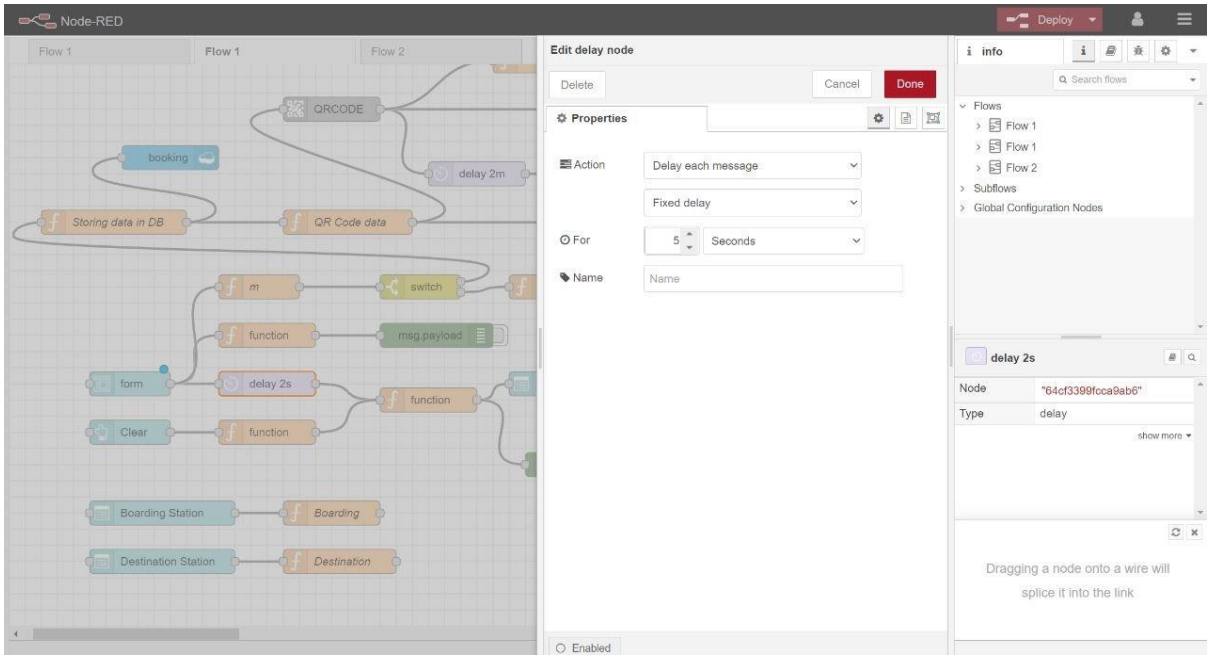
FOR DESTINATION



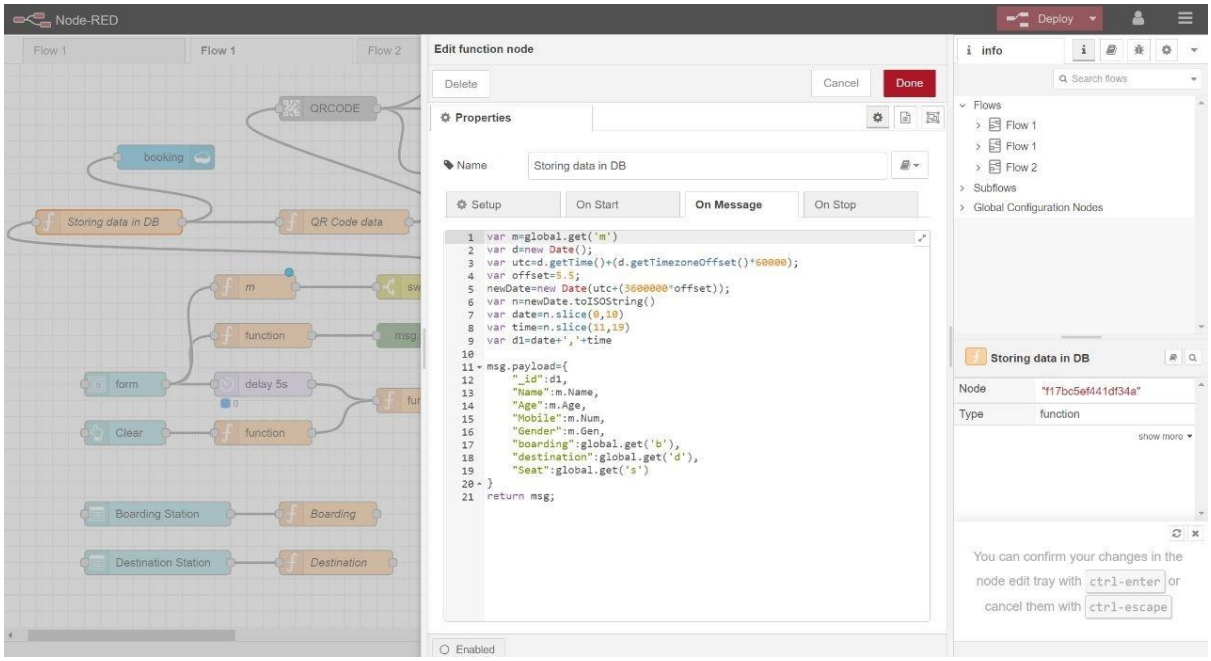
FUNCTION MODE



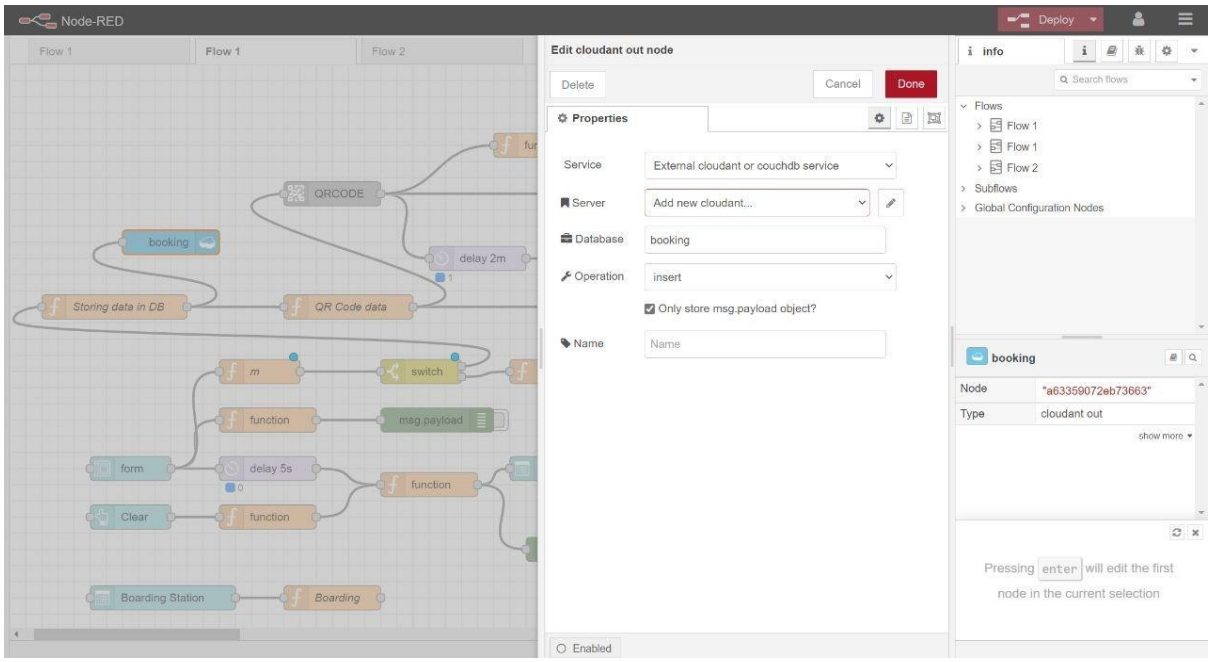
DELAY MODE



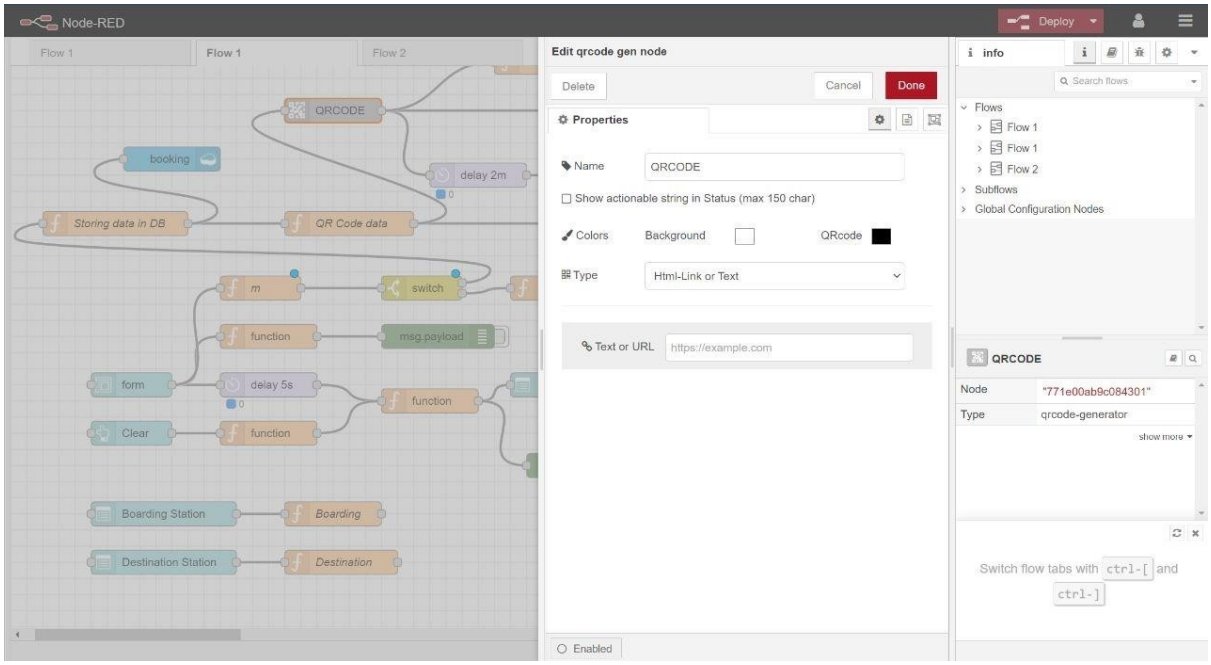
FOR STORING DATABASE



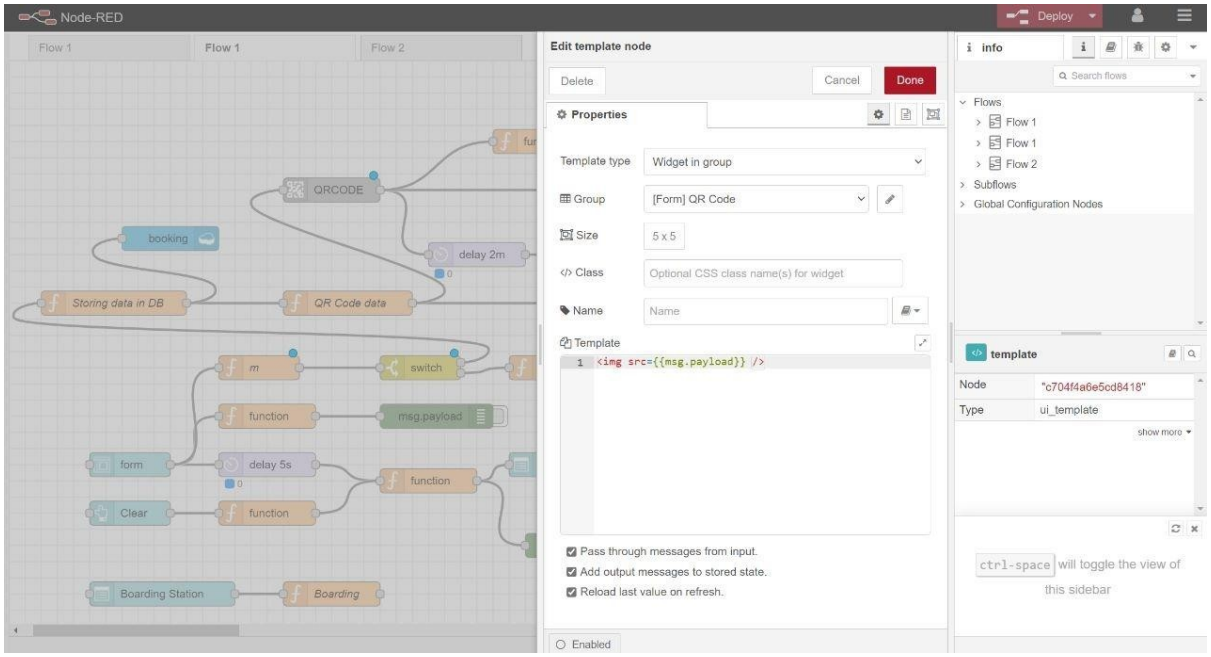
FOR BOOKING (OUTPUT)



QR CODE GENERATOR



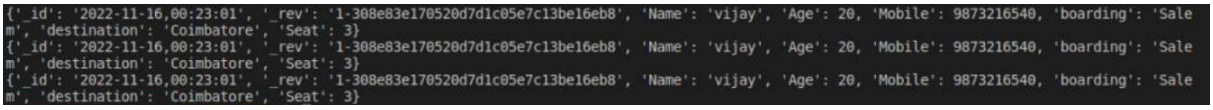
FUNCTION NODE TEMPLATE



7.2) FEATURE 2

QR CODE SCANNER

- **SCAN QR CODE AND VIEW DETAILS OF THE PASSENGER**



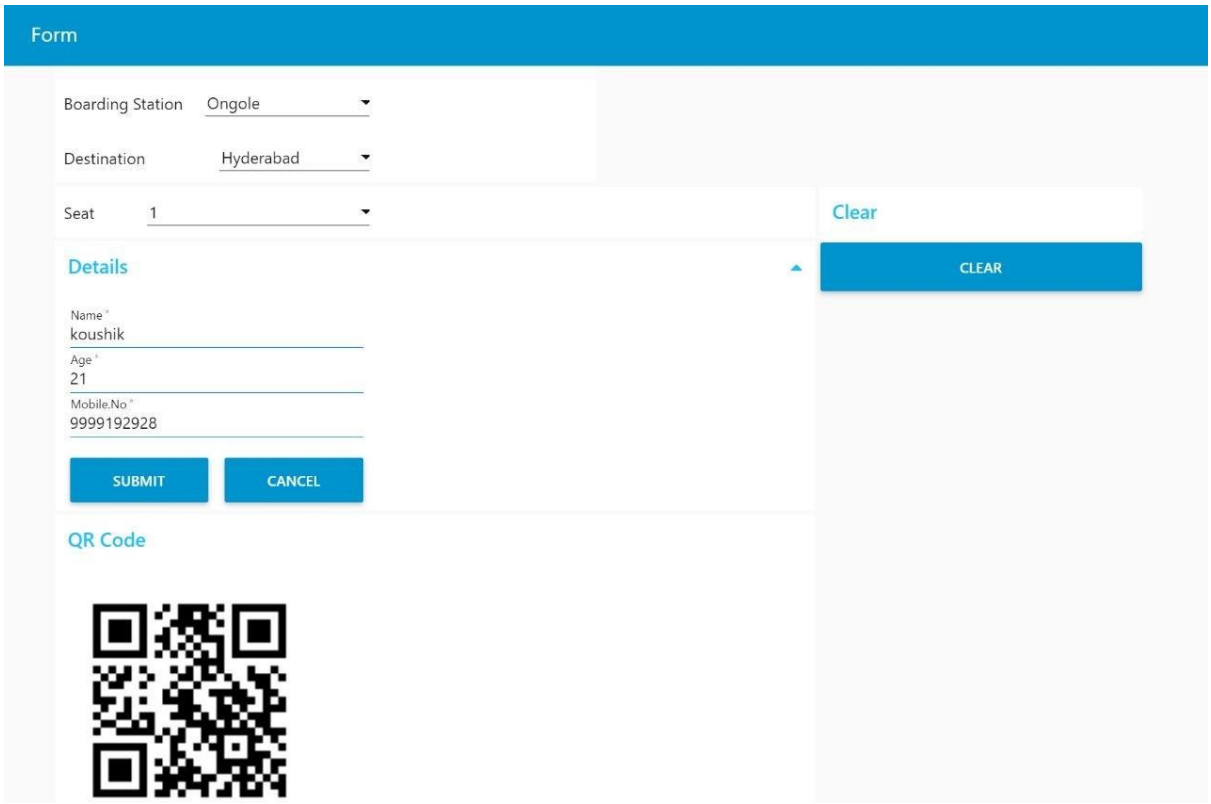
VIEW LOCATION OF TRAINS

- **GET THE COORDINATES OF THE TRAIN**
- **VIEW TRAIN LOCATION IN MOBILE APPLICATION**

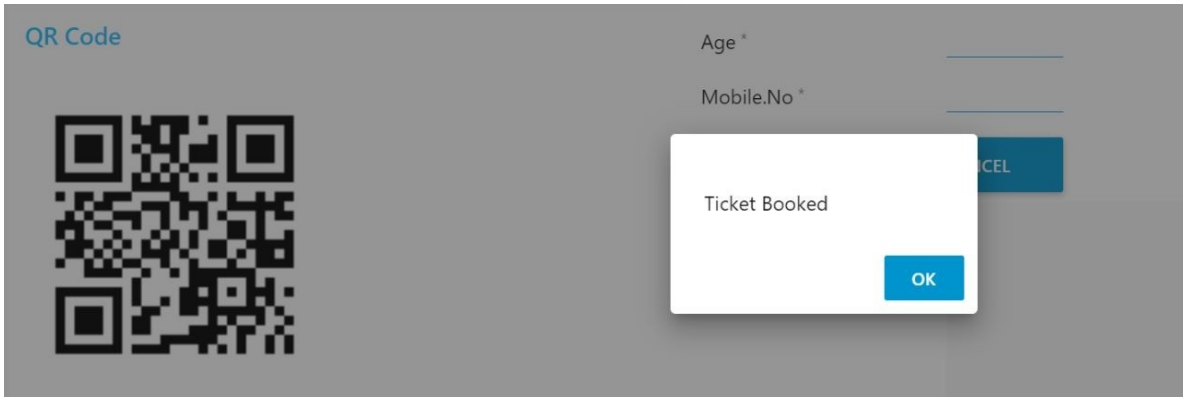
TESTING

8.1) TEST CASES

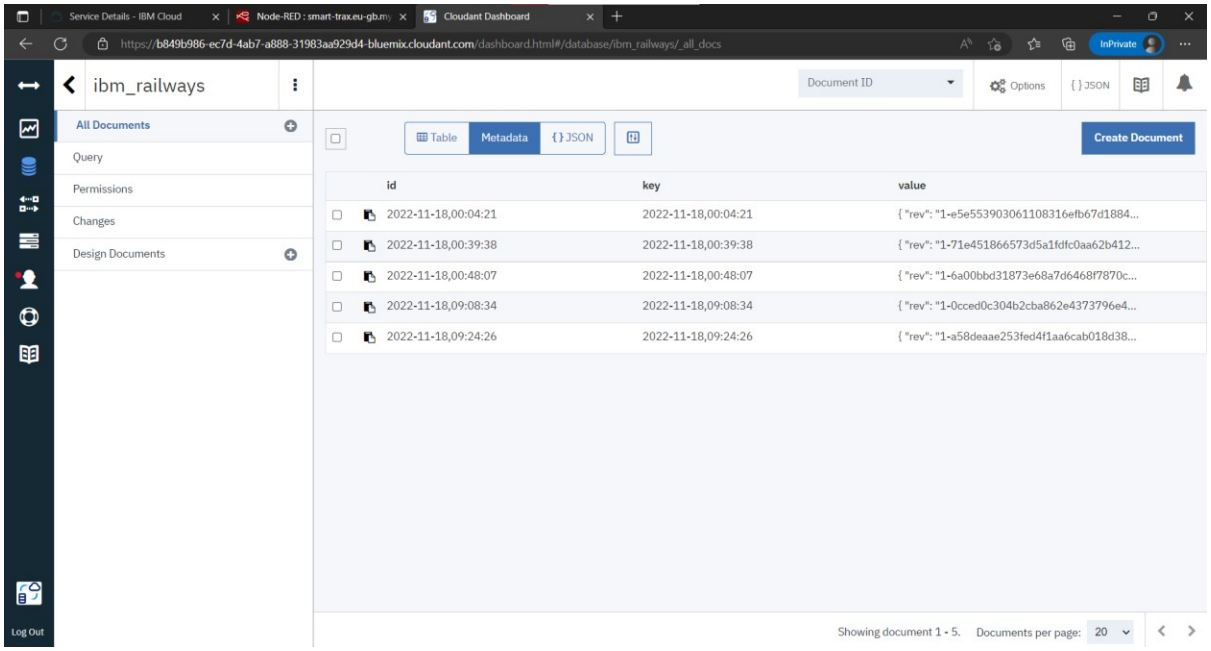
Test case 1

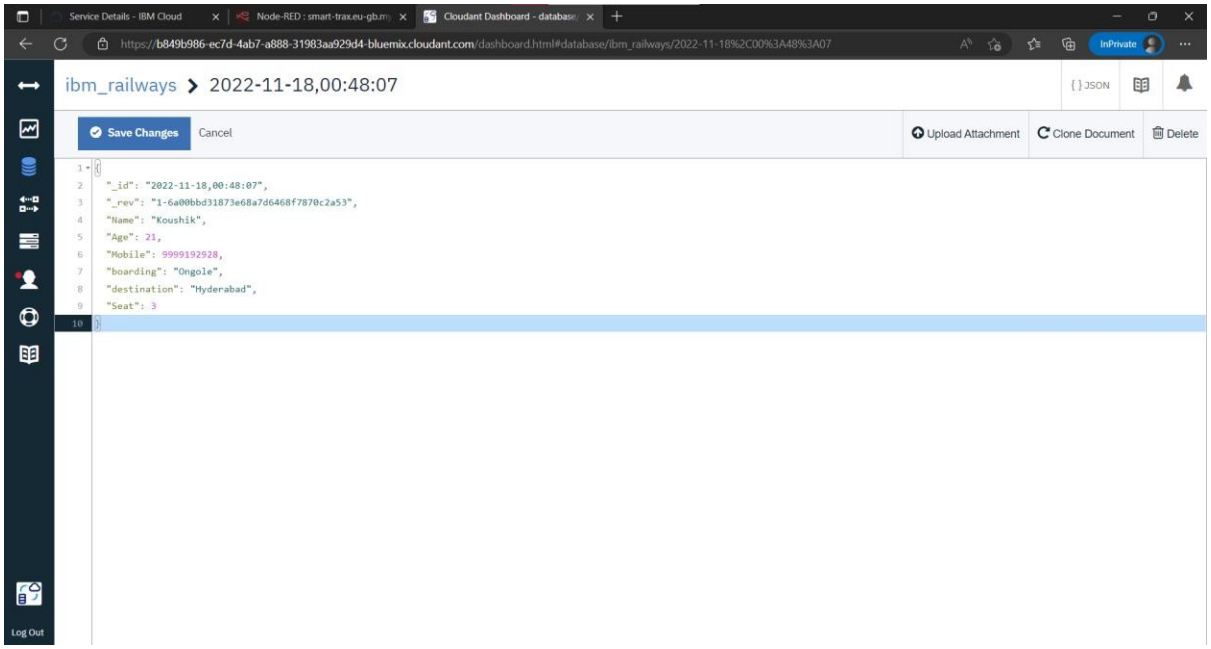


TICKET RESERVATION

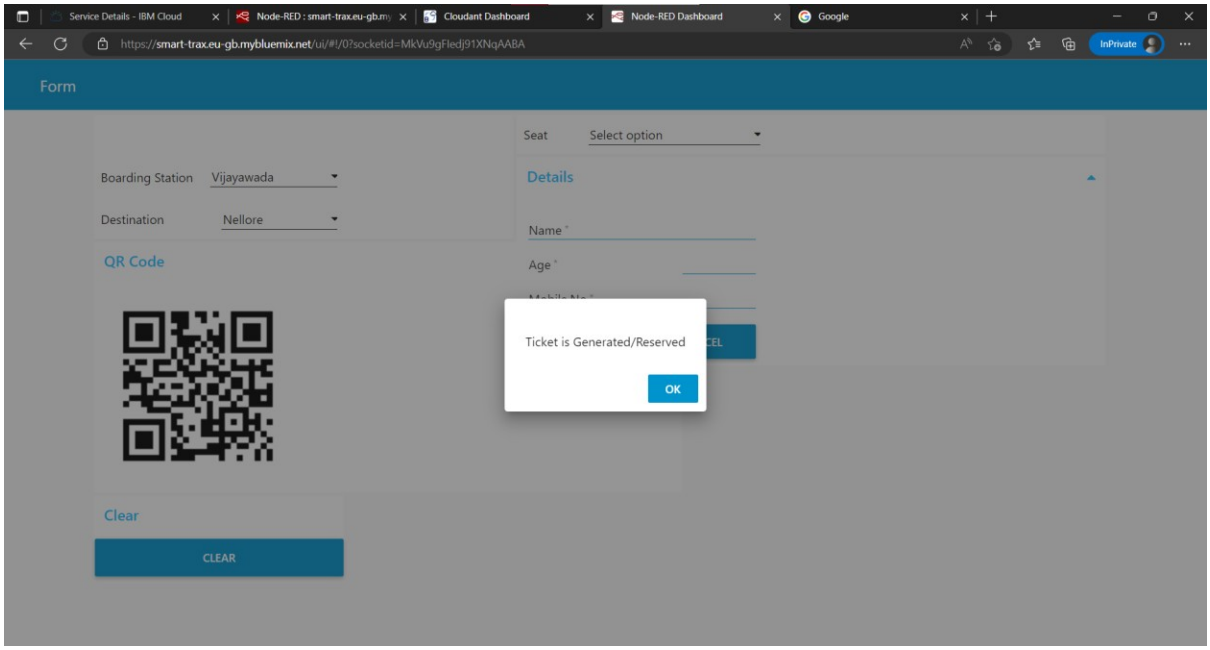


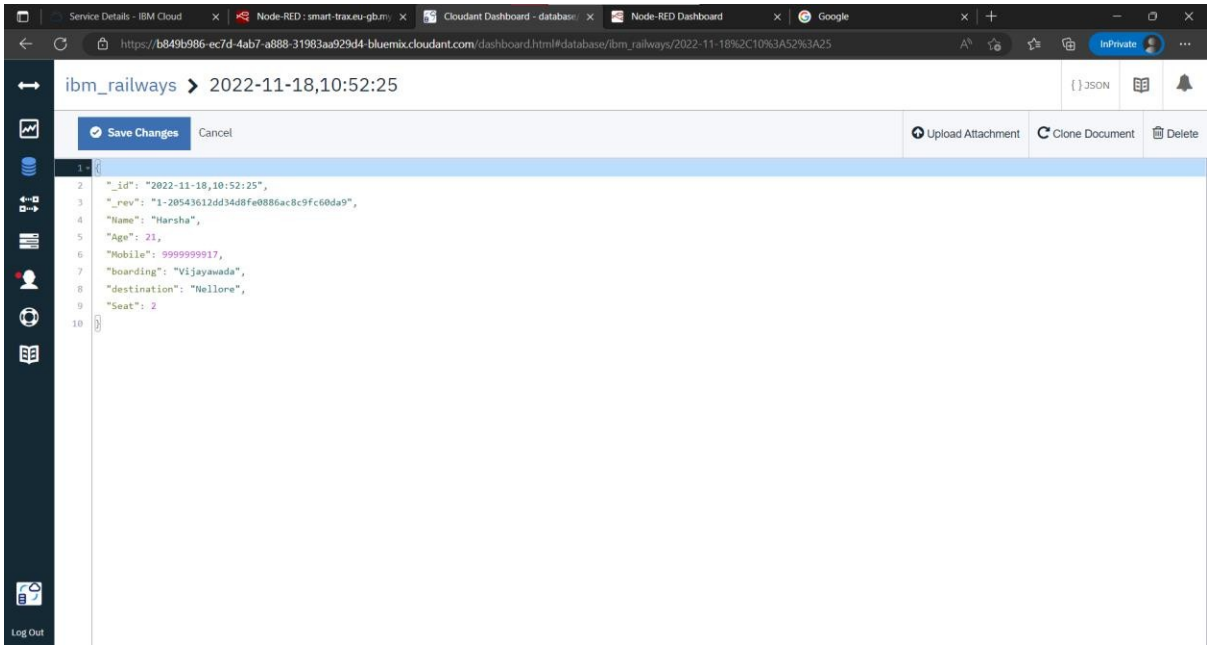
DATABASE


Service Details - IBM Cloud
Node-RED : smart-trax.eu-gb.m
Cloudant Dashboard
https://b849b986-ec7d-4ab7-a888-31983aa929d4-bluemix.cloudant.com/dashboard.html#/database/ibm_railways/_all_docs
InPrivate
ibm_railways
Document ID
Options
{ } JSON
All Documents
Query
Permissions
Changes
Design Documents
Table
Metadata
{ } JSON
Create Document
id
key
value
2022-11-18,00:04:21
2022-11-18,00:04:21
{ "rev": "1-e5e553903061108316efb67d1884...
2022-11-18,00:39:38
2022-11-18,00:39:38
{ "rev": "1-71e451866573d5a1fdfc0aa62b412...
2022-11-18,00:48:07
2022-11-18,00:48:07
{ "rev": "1-6a00bbd31873e68a7d6468f7870c...
2022-11-18,09:08:34
2022-11-18,09:08:34
{ "rev": "1-0cced0c304b2cba862e4373796e4...
2022-11-18,09:24:26
2022-11-18,09:24:26
{ "rev": "1-a58deaae253fed4f1aa6cab018d38...
Showing document 1 - 5.
Documents per page:
20
Log Out

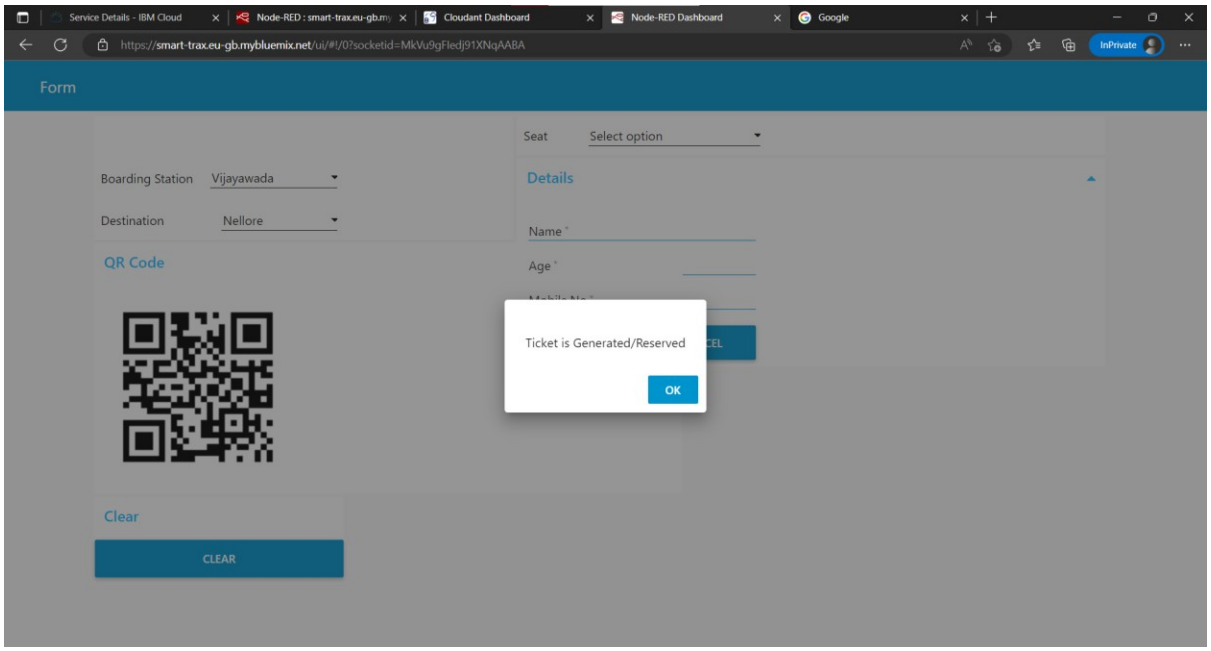

Service Details - IBM Cloud
Node-RED : smart-trax.eu-gb.m
Cloudant Dashboard - database
https://b849b986-ec7d-4ab7-a888-31983aa929d4-bluemix.cloudant.com/dashboard.html#database/ibm_railways/2022-11-18%2C00%3A48%3A07
InPrivate
ibm_railways > 2022-11-18,00:48:07
{ } JSON
Save Changes
Cancel
Upload Attachment
Clone Document
Delete
{
"_id": "2022-11-18,00:48:07",
"_rev": "1-6a00bbd31873e68a7d6468f7870c2a53",
"Name": "Koushik",
"Age": 21,
"Mobile": 9999192928,
"boarding": "Ongole",
"destination": "Hyderabad",
"Seat": 3
}
Log Out

Test case 2

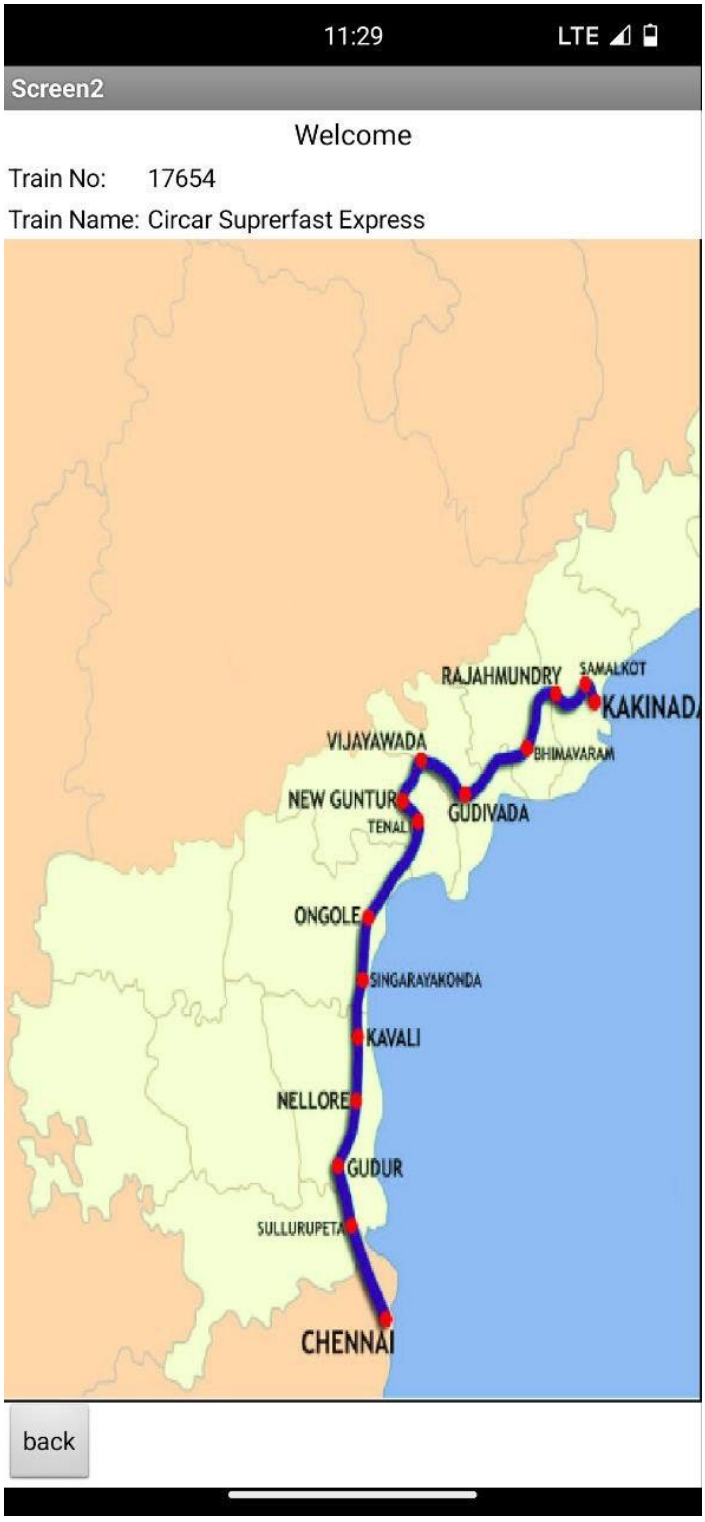

Service Details - IBM Cloud
Node-RED : smart-trax.eu-gb.my
Cloudant Dashboard
Node-RED Dashboard
Google
https://smart-trax.eu-gb.mybluemix.net/ui/#!/0?socketid=MkVu9gFledj91XNqAABA
InPrivate
Form
Seat
Select option
Boarding Station
Vijayawada
Details
Destination
Nellore
Name *
QR Code
Age *
Ticket is Generated/Reserved
OK
Clear
CLEAR


Service Details - IBM Cloud
Node-RED : smart-trax.eu-gb.my
Cloudant Dashboard - database
Node-RED Dashboard
Google
https://b849b986-ec7d-4ab7-a888-31983aa929d4-bluemix.cloudant.com/dashboard.html#database/ibm_railways/2022-11-18%2C10%3A52%3A25
InPrivate
ibm_railways > 2022-11-18,10:52:25
{ } JSON
Save Changes
Cancel
Upload Attachment
Clone Document
Delete
{
"_id": "2022-11-18,10:52:25",
"_rev": "1-20543612dd34d8fe0886ac8c9fc60da9",
"Name": "Harsha",
"Age": 21,
"Mobile": 9999999917,
"boarding": "Vijayawada",
"destination": "Nellore",
"Seat": 2
}
Log Out

RESULTS


Service Details - IBM Cloud
Node-RED : smart-trax.eu-gb.my
Cloudant Dashboard
Node-RED Dashboard
Google
https://smart-trax.eu-gb.mybluemix.net/ui/#!/0?socketid=MkVu9gFledj91XNqAABA
InPrivate
Form
Seat
Select option
Boarding Station
Vijayawada
Details
Destination
Nellore
Name *
QR Code
Age *
Ticket is Generated/Reserved
OK
Clear
CLEAR

Location in mobile application



11) CONCLUSION

Using the application, user can book train tickets based on availability of seats inparticular train. Once tickets are available, they can book them by inputting theirgeneral information. Upon completion of payment, the data gets stored in Cloudant DB with unique ID for every transaction and a QR code is generated forevery ticket. The ticket collector can scan the QR code to get information of the passenger, if the QR is correct, the details of the user are displayed, if the QR is
invalid, it displays “Not a Valid ticket”. Apart from ticketing, our application also allows the users to find out the live location and running status of the train.

12) FUTURE SCOPE

Cloud computing and IOT are integrated now to ease the ticketing system and tracking in railways. In near future, Internet of Things and Artificial Intelligence canbe combined to make railways safer and faster. Artificial Intelligence can be used to determine delay and arrival time so that the passenger can act accordingly. By the use of Internet of Things, things such as maintenance of tracks, repairs and services can be carried out with ease.

13) APPENDIX

13.1) SOURCE CODE

QR SCAN CODE:

```
from http import client

import cv2

import pyzbar

from pyzbar.pyzbar import decode

import time


from ibmcloudant.cloudant_v1 import CloudantV1

from ibmcloudant import CouchDbSessionAuthenticator

from ibm_cloud_sdk_core.authenticators import BasicAuthenticator


authenticator = BasicAuthenticator('apikey-v2-x5kge2ue09set5w3y7zqwppcsu8yssimj438ufbetnx',
'75782580a79baa3efdec4c2191edd956')

service = CloudantV1(authenticator=authenticator)


service.set_service_url('https://apikey-v2-
x5kge2ue09set5w3y7zqwppcsu8yssimj438ufbetnx:75782580a79baa3efdec4c2191edd956@b849b9
86-ec7d-4ab7-a888-31983aa929d4-bluemix.cloudantnosqldb.appdomain.cloud')

cap= cv2.VideoCapture(0)

font = cv2.FONT_HERSHEY_PLAIN


while True:

    _, frame = cap.read()
```

```
        decodedObjects = decode(frame)
        for obj in decodedObjects:
            #print ("Data", obj.data)
            a=obj.data.decode('UTF-8')
            cv2.putText(frame, "Ticket", (50, 50), font, 2, (255, 0, 0), 3)

            #print (a)
            try:
                response =
                service.get_document(db='IBM_r
                ailways',
                doc_id = a
                ).get_result()
                print (response)
                time.sleep(5)
            except Exception as e:
                print(a)
                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()
```

LOCATION CODE

```
import wiotp.sdk.device
import time
import random

myConfig =
 { "identity":
 { "orgId": "ekscpp",
   "typeId": "h1",
   "deviceId": "h1"
 },
 "auth": {
   "token": "12345678"
 }
}

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

def pub(data):
 client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
 print("Data is published Successfully:%s",myData)
```

```
client = wiotp.sdk.device.DeviceClient(config=myConfig)
client.connect()

while True:
  myData={'name':'Train1','lat':10.184363,'lon': 77.922702}
  pub(myData)
  time.sleep(3)
  myData={'name':'Train1','lat':10.213225,'lon': 77.898765}
  pub(myData)
  time.sleep(3)
  myData={'name':'Train1','lat':10.285035,'lon': 77.921569}
  pub(myData)
  time.sleep(3)
  myData={'name':'Train1','lat':10.343369,'lon': 77.958056}
  pub(myData)
  time.sleep(3)
  myData={'name':'Train1','lat':10.356829,'lon': 77.980861}
  pub(myData)
  time.sleep(3)
  client.commandCallback = myCommandCallback
client.disconnect()
```