SPECIMEN

SMART SOLUTIONS FOR RAILWAYS

A PROJECT REPORT

Submitted by

ALBIN B

BEAUTLIN PANISHA MALAR J D

JIJITHA A L

(TEAM ID : PNT2022TMID51722)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

MAR EPHRAEM COLLEGE OF ENGINEERING AND TECHNOLOGY, ELAVUVILAI

ANNA UNIVERSITY::CHENNAI 600 025

NOV 2022

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“SMART SOLUTIONS FOR RAILWAYS”**

is the bonafide work of

“ALBIN B (8903963582), BEAUTLIN PANISHA MALAR J D (7305800122),

JIJITHA A L (6380527490)”

“S REEBA REX” who carried out the project work under my supervision

SIGNATURE

Dr. V SURESH, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

Electronics and Communication Engineering

Mar Ephraem College of Engineering and Technology, Elavuvilai

SIGNATURE

Dr. S REEBA REX, M.E.,Ph.D.,

Assistant Professor

Electronics and Communication Engineering

Mar Ephraem College of Engineering and Technology,

TABLE OF CONTENTS

CHAPTER

NO. **TITLE**

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW:

Information Technology has elaborated to an immense level and is being utilized in the field of transportation services. Roaming in Train is the most under-value form of long-distance travel. In addition, travelling by train is time preserving as the trains usually runs on time. Furthermore, many train journeys over long distance need reservations, but the short distance train journey requires an ordinary railway ticket. For Indian railways, to buy ordinary tickets, the traveler has to abide in long queue. Standing in queues which is a protracted, vexatious and inconvenient process as time demand is more. With respect to Mumbai Suburban Railway, one of the major problems faced by the 7 million people, who travel by local trains every day, is standing in the long queues for an average of 10-15 minutes to buy a ticket.

Sometimes many citizens facing problems to reach the railway station on time due to one or more reasons. Sometime ticket issuing officer generates a ticket only if we provide change money. It is risky to carry money for purchasing tickets. In addition, a citizen who wants to issue season ticket, has to abide in the queues.

To overcome all these offline railway reservation issues, it is highly recommended to go with online ticket booking system. Because there is no serious drawback in online reservation system. It is very convenient for travelers as they can book reserved ticket from anywhere at any time as per their convenience. No cash is required, payment is made through net banking/cards/UPI etc. Refund also gets credited to the bank account within a week.

Smart Solutions for railways is designed to reduced 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 the available seats.
- After booking the train, the person will get a QR code which has to 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.
- 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.

1.2 PURPOSE:

An online railway reservation is a Software solution that makes it simple for users to book and pay for user’s trips and activities online. Some of the systems also include reporting software for tour operators and other user-friendly tools that helps the user to improve efficiencies and boost bookings.

The key purpose of an online railway reservation system is to help capture the growing demand, with the ultimate objective of bringing in more bookings for business. Of course, there are a few other reasons:

- Deliver an amazing user experience for the users.
- Improve the security of online payments for the users.
- Easily capture user data to help with other business areas, such as tour management and marketing.
- Offer promotions and special pricing to boost bookings in quiet times and off-seasons.
- Gain insights into how and why users book in online – and use this to drive new bookings.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM:

Facility for purchasing unreserved tickets through mobile phone is available at DELHI, Mumbai, Chennai, Kolkata and Secunderabad railway stations.

Android Suburban Railway (ASR) ticketing is to purchase the suburban tickets by 'M-ticket'. ASR ticket can be purchased with smart phone app through which travelers can travel with railway tickets in smart phone as QR (Quick Response) code. ASR utilizes smart phone's GPS (Global Positioning System) to authenticate and obliterate generated ticket itself following a definite time of period once traveler accomplish the journey. In addition, the railway ticket checker is supplied with a checker application to verify the traveler's ticket with the ticket number and generated QR code.

A Wi-Fi router is set up at ticket counter. In addition, router does not require internet connection. Furthermore, Wi-Fi router is connected to the railway server. When travelling with an android application in smart phone arrives near the region of ticket issuing counter area, traveler initially connect with Wi-Fi router. Later traveler will prompt with source railway station name automatically. Subsequently traveler has to enter his credential for authentication of traveler. If traveler enters correct credential then application allows travelers to select destination based on route that has to be travelled by traveler. Further traveler has to enter number of tickets required. Traveler information will be verified at a railway server via Wi-Fi router. If information entered by travelers are valid then based on number of ticket required payment will be done and later travelers receive message which is identical to the railway ordinary ticket.

Traveler who want to buy an online ticket has to download application for smart phone. Then the traveler will prompt to insert the train information like source and destination station, number of tickets required to generate including the category of passenger like an adult or child, and class of ordinary ticket like first or second class. Final cost of ticket/s will be calculated based on information inserted and train information stored in database. After successful deduction of payment via diverse payment methods, the ticket is generated with information similar to the paper-based tickets.

The Android Ticketing of Railways (ATR) is an online railway ticket system where the ticket can be bought easily from any place in the internet enable smart phone. The ticket will be generated in a smart phone in the form of Quick Response (QR) Code. In addition, the ticket checker is facilitate with a QR code scanner to check traveler tickets based on information store at database.

Problem in Passenger's view:

There were a lot of problems faced by the passengers during offline railway reservation. Like, the passengers have to wait in a queue for a while to buy tickets. It affects their daily routine. Also the passengers waiting in a queue feels tired. If all the available tickets get sold to the first half of the passengers in a queue, then the time spent is completely waste for the passengers who were not received the tickets. This kind of process is not possible to follow in an emergency situation. That's why, it is better to adopt online railway reservation methodology. It saves the waiting time. Ticket booking can be possible at the passenger's residence itself, if he/she has a very good net facility and a smart phone.

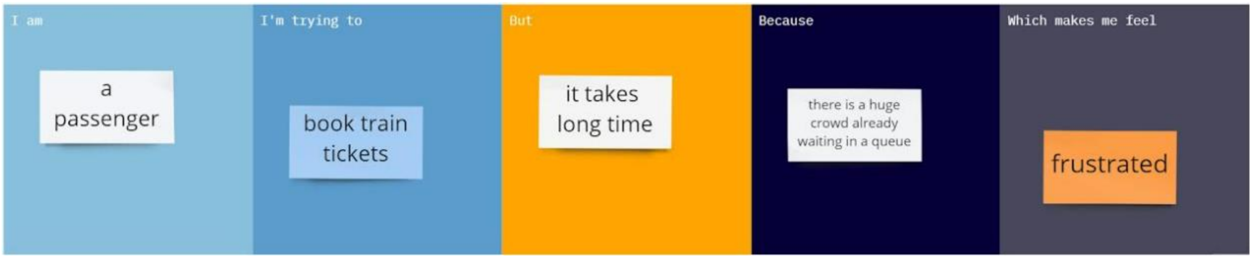
2.2 REFERENCES:

1. www.projectsgeek.com
2. www.researchgate.net
3. www.geeksforgeeks.org
4. S. Shaikh, G. Shinde, M. Potghan, T. Shaikh, and R. Suryawanshi, "Urban Railway Ticketing Application," vol. 4, no. 1, pp. 130–132, 2014.
5. International Journal of Engineering Development and Research (www.ijedr.org)

2.3 PROBLEM STATEMENT DEFINITION:

A problem statement is important to a process improvement project because it helps clearly identify the goals of the project and outline the scope of a project. It also helps guide the activities and decisions of the people who are working on the project. The problem statement can help a business or organization gain support and buy-in for a process improvement project. A good problem statement can be created by identifying and answering several questions related to the problem.

This process involves identifying what the problem is, why it is a problem, when and where the problem was identified, who the problem impacts, how they are impacted by the problem and how much of an impact the problem has. Creating a problem statement to understand customer's point of view. The below shown block diagram is a perfect example for our topic.



I am
a passenger

I'm trying to
book train tickets

But
it takes long time

Because
there is a huge crowd already waiting in a queue

Which makes me feel
frustrated

Fig.1 Problem Statement

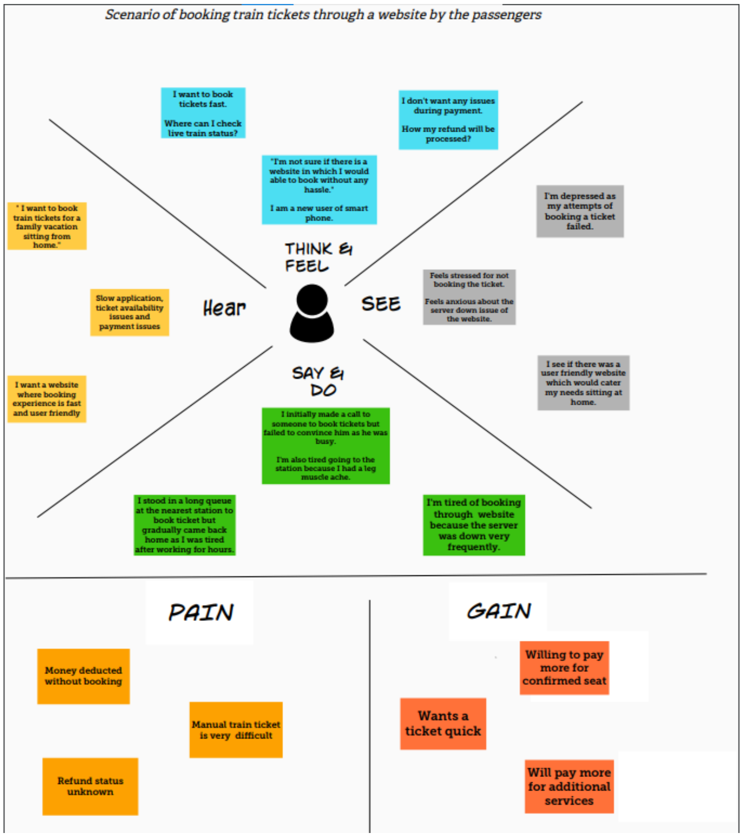
CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:

Empathy maps are an efficient tool used by designers to not only understand user behavior, but also visually communicate those findings to colleagues, uniting the team under one shared understanding of the user. Essentially, an empathy map is a square divided into four quadrants with the user or client in the middle. Each of the four comprises a category that helps us delve into the mind of the user. The four empathy map quadrants look at what the user says, thinks, feels and does.

With the user at the center and the categories in each of the four surrounding quadrants, an empathy map arranges all of your research about the user into an easy-to-read visual.



Scenario of booking train tickets through a website by the passengers

I want to book tickets fast.

Where can I check live train status?

I don't want any issues during payment.

How my refund will be processed?

"I'm not sure if there is a website in which I would able to book without any hassle."

I am a new user of smart phone.

" I want to book train tickets for a family vacation sitting from home."

Slow application, ticket availability issues and payment issues

I want a website where booking experience is fast and user friendly

I'm depressed as my attempts of booking a ticket failed.

Feels stressed for not booking the ticket.

Feels anxious about the server down issue of the website.

I see if there was a user friendly website which would cater my needs sitting at home.

I initially made a call to someone to book tickets but failed to convince him as he was busy.

I'm also tired going to the station because I had a leg muscle ache.

I stood in a long queue at the nearest station to book ticket but gradually came back home as I was tired after working for hours.

I'm tired of booking through website because the server was down very frequently.

Money deducted without booking

Manual train ticket is very difficult

Refund status unknown

Willing to pay more for confirmed seat

Wants a ticket quick

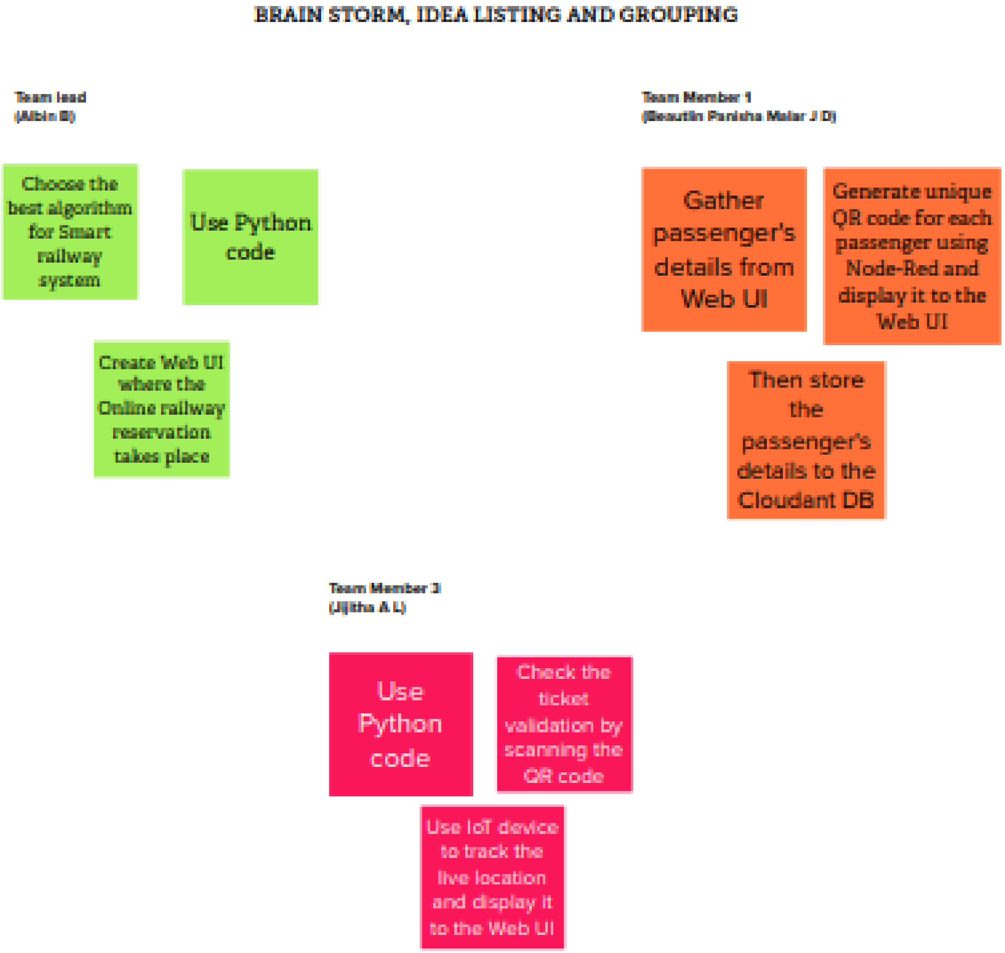
Will pay more for additional services

Fig.2 Empathy Map

3.2 IDEATION & BRAINSTORMING:

Brainstorming is a method design teams use to generate ideas to solve clearly defined design problems. Brainstorming is a method of generating ideas and sharing knowledge to solve a particular commercial or technical problem, in which participants are encouraged to think without interruption. Brainstorming is a group activity where each participant shares their ideas as soon as they come to mind. At the conclusion of the session, ideas are categorized and ranked for follow-on action.

When planning a brainstorming session it is important to define clearly the topic to be addressed. A topic which is too specific can constrict thinking, while an ill-defined topic will not generate enough directly applicable ideas. The composition of the brainstorming group is important too. It should include people linked directly with the subject as well as those who can contribute novel and unexpected ideas. It can comprise staff from inside or outside the organization.



BRAIN STORM, IDEA LISTING AND GROUPING
Team lead
(Albin B)
Choose the best algorithm for Smart railway system
Use Python code
Create Web UI where the Online railway reservation takes place
Team Member 1
(Beautin Panisha Malar J D)
Gather passenger's details from Web UI
Generate unique QR code for each passenger using Node-Red and display it to the Web UI
Then store the passenger's details to the Cloudant DB
Team Member 3
(Jijitha A L)
Use Python code
Check the ticket validation by scanning the QR code
Use IoT device to track the live location and display it to the Web UI

Fig.3 Brainstorming

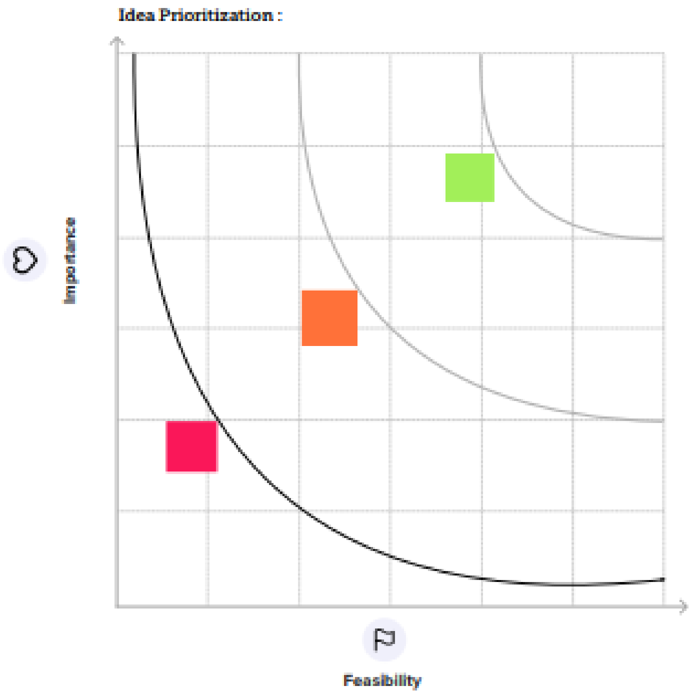


Fig.4 Idea Prioritization

3.3 PROPOSED SOLUTION:

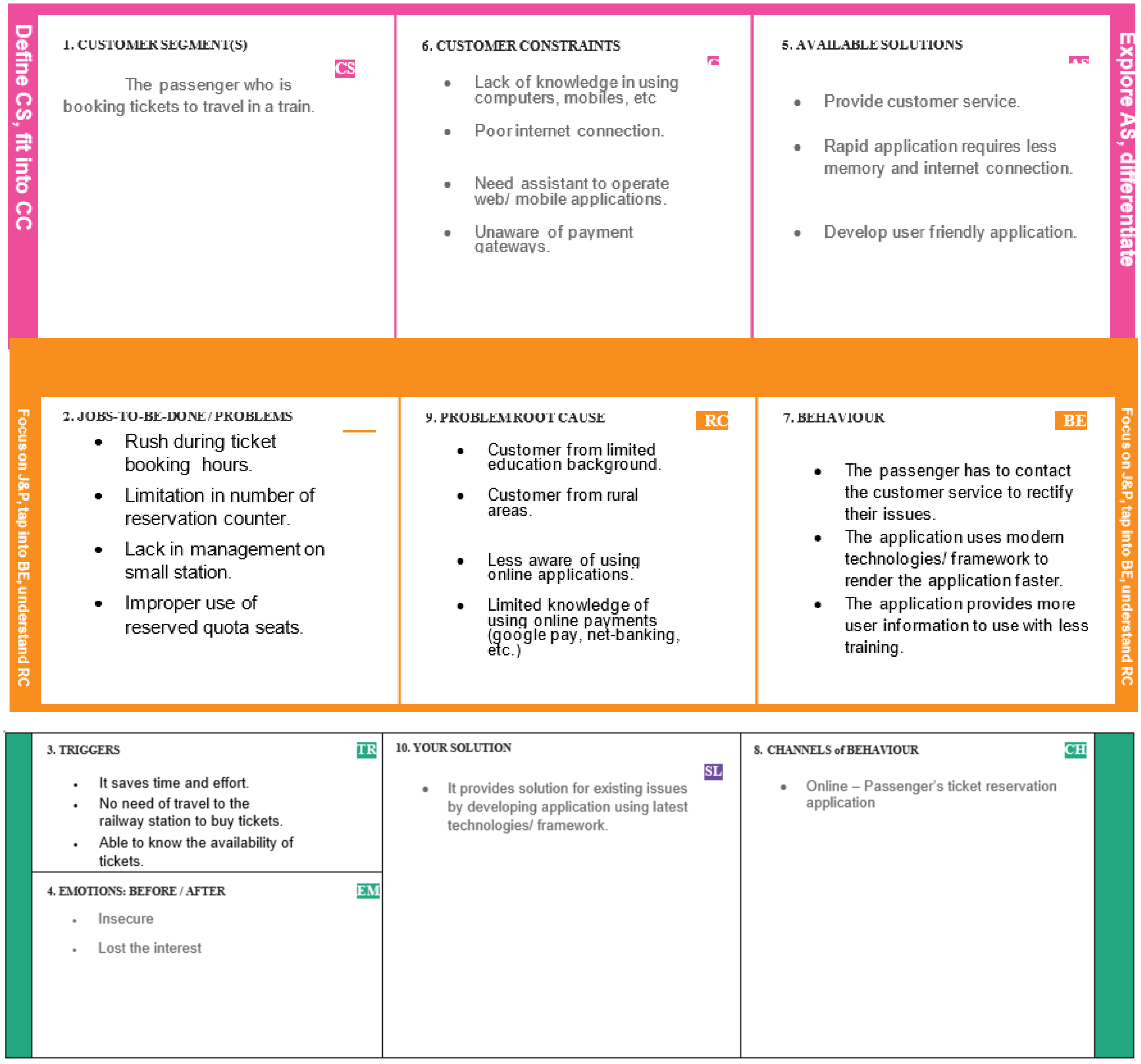
Proposed Solution means the technical solution to be provided by the Implementation agency in response to the requirements and the objectives of the Project.

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Standing in a queue for a while to book railway tickets without knowing about the availability of tickets.
2.	Idea / Solution description	IoT based smart railways in Online reservation is highly recommended to overcome the above mentioned problems. • Using Python Code, create a Web UI. • Register as a passenger by providing certain personal details in the Web UI. • Search trains between a pair of stations, check seat availability and get the fare details. • Make fresh reservations and book one or more seats.

3.	Novelty / Uniqueness	• The personal details of the passenger gets stored in Cloudant DB and a unique QR code gets generated with the help of Node-Red and displayed to the Web UI. • To track the live location of the passenger while travelling, IoT device is used. • Live location also gets displayed in the Web UI. • Ticket Collector can able to identify whether the passenger's ticket is valid or not by scanning the generated QR Code.
4.	Social Impact / Customer Satisfaction	• It doesn't affects the daily routine. • No need to do paper work. • Management of online database. • Increase payment speed and security. • Ticket cancelling & refunding process are occurred.
5.	Business Model (Revenue Model)	Key activities: • Networking • Problem solving Key resources: • Sensors • Cloud service • IoT dedicated network Cost structure: • Maintenance • Digital infrastructure Value preposition: • Convenience • Preference • Marketing the service Revenue Stream: • Reservation fees • Advertising
6.	Scalability of the Solution	• Effectiveness • Adjustment value needed to achieve the maximum capacity which depends on the reconfiguration of time and cost. • Increase in number of users

3.4 PROBLEM SOLUTION FIT:

Problem Solution fit – this occurs when you have evidence that customers care about certain jobs, pains and gains. At this stage you have to prove the existence of a problem and have to design a value proposition that addresses your customer's jobs, pains and gains.



Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC

1. CUSTOMER SEGMENT(S) CS

The passenger who is booking tickets to travel in a train.

6. CUSTOMER CONSTRAINTS CC

- Lack of knowledge in using computers, mobiles, etc
- Poor internet connection.
- Need assistant to operate web/ mobile applications.
- Unaware of payment gateways.

5. AVAILABLE SOLUTIONS AS

- Provide customer service.
- Rapid application requires less memory and internet connection.
- Develop user friendly application.

2. JOBS-TO-BE-DONE / PROBLEMS

- Rush during ticket booking hours.
- Limitation in number of reservation counter.
- Lack in management on small station.
- Improper use of reserved quota seats.

9. PROBLEM ROOT CAUSE RC

- Customer from limited education background.
- Customer from rural areas.
- Less aware of using online applications.
- Limited knowledge of using online payments (google pay, net-banking, etc.)

7. BEHAVIOUR BE

- The passenger has to contact the customer service to rectify their issues.
- The application uses modern technologies/ framework to render the application faster.
- The application provides more user information to use with less training.

3. TRIGGERS TR

- It saves time and effort.
- No need of travel to the railway station to buy tickets.
- Able to know the availability of tickets.

4. EMOTIONS: BEFORE / AFTER EM

- Insecure
- Lost the interest

10. YOUR SOLUTION SL

- It provides solution for existing issues by developing application using latest technologies/ framework.

8. CHANNELS of BEHAVIOUR CH

- Online – Passenger's ticket reservation application

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through entering password Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through Email Login into web by entering the given password
FR-4	User Dashboard	Learn to access the Web UI
	Ticket reservation	Enter the basic train information for verifying availability of trains and tickets. Fill up the basic travelling passenger’s details after choosing the no. of tickets.
	Payment	By completing the online payment, user will navigate to the confirmation page.
FR-5	User Identification	It generates a unique QR Code for each passenger that is to differentiate between passengers with their personal details.
FR-6	Ticket Validation	By scanning the QR Code, it shows whether the passenger’s ticket is valid or not
FR-7	User Data	User’s data get stored in a Cloudant DB
FR-8	Live Location Tracking	With the help of IoT based sensors, Live location of the passenger while traveling gets displayed in the Web UI

4.2 NON FUNCTIONAL REQUIREMENTS:

Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirements	Description
NFR-1	Usability	It is about effectiveness, efficiency and the overall satisfaction of the user.
NFR-2	Security	The Web UI will only require the registered users to login to it. It will resist the unauthorized users.
NFR-3	Reliability	The service will perform its intended function adequately for a specified period of time, or will operate in a defined environment without failure.

NFR-4	Performance	Provider systems must meet the agreed response time performance targets.
NFR-5	Availability	Provider systems shall meet the agreed availability targets i.e. service time and/or hours and planned downtime.
NFR-6	Scalability	Provider systems shall be designed to accommodate increased volumes, workloads and users.

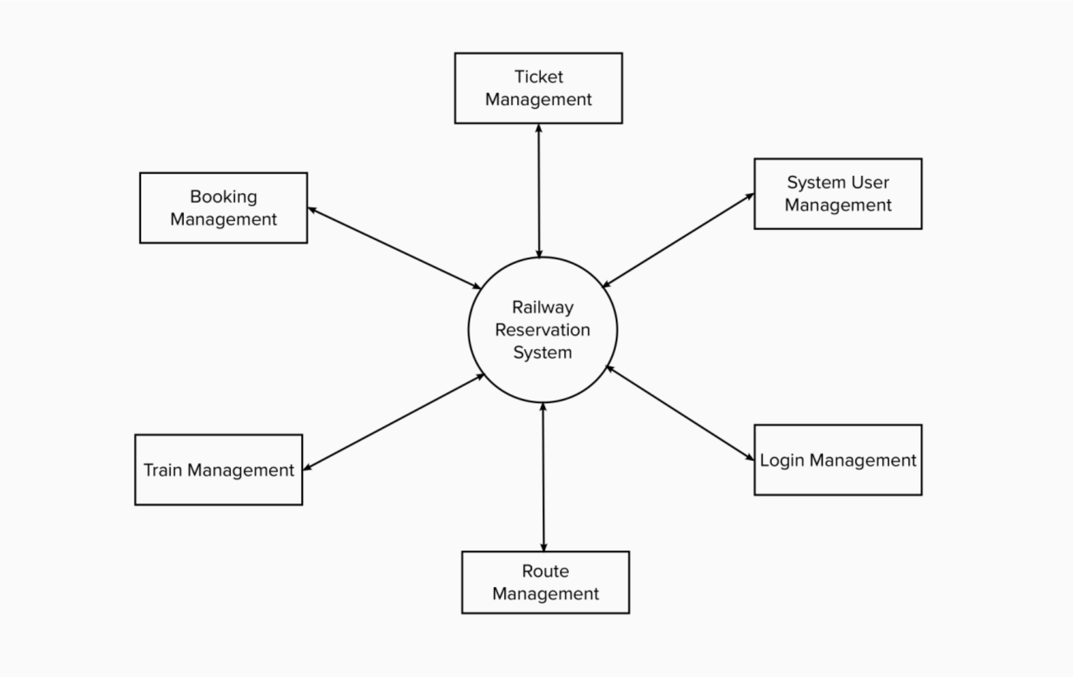
CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM:

A data flow diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enter and leaves the system, what changes the information and where the data is stored.

Zero level DFD:

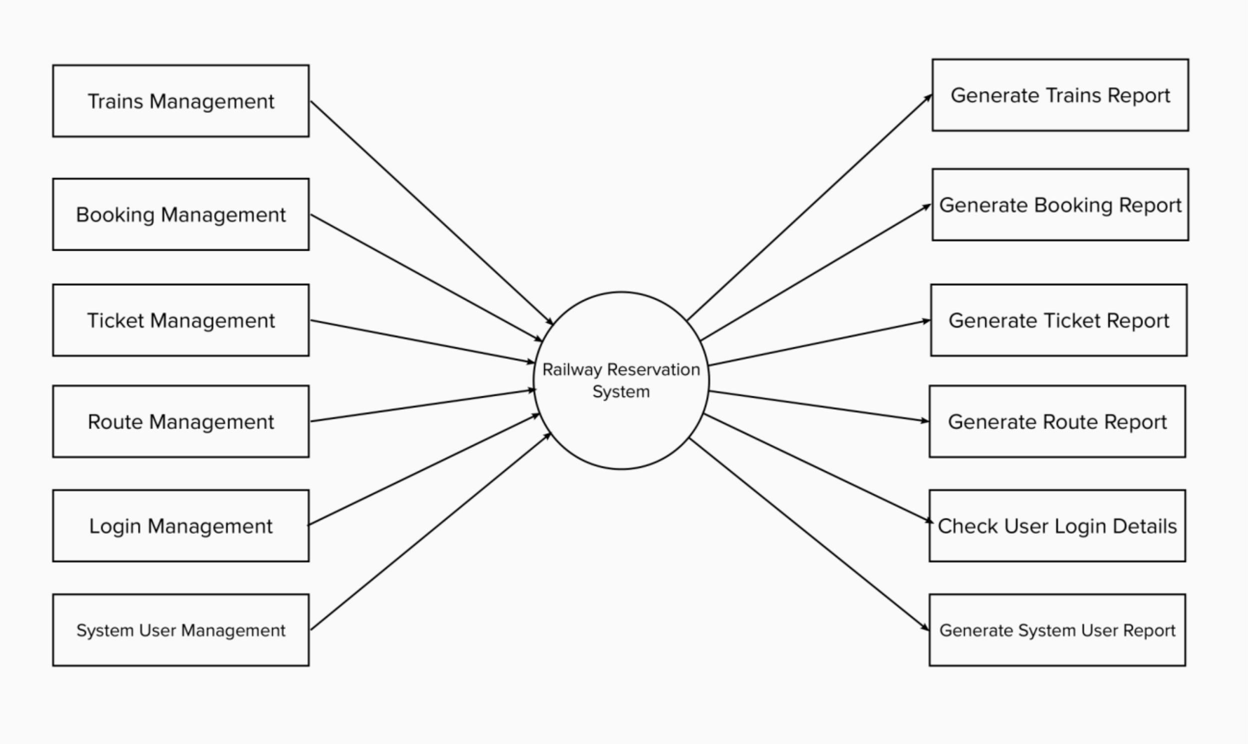


Working Process:

- Managing all the trains

- Managing all the booking
- Managing all the ticket
- Managing all the train schedule
- Managing all the train route
- Managing all the customer
- Managing all the payment

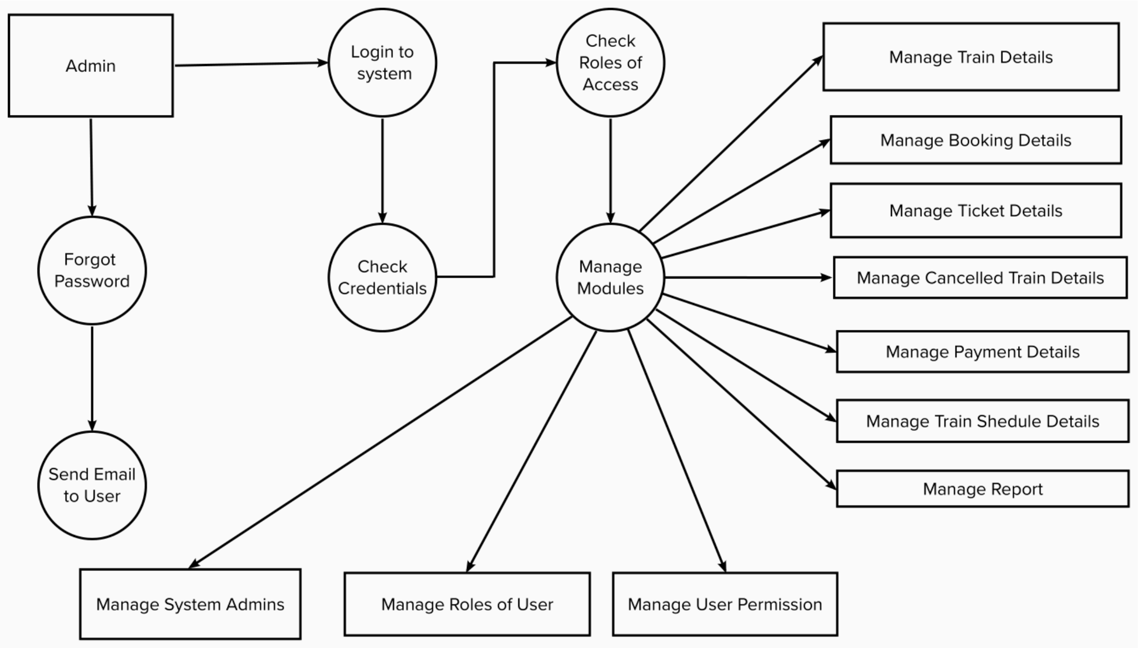
First level DFD:



Working Process:

- Processing trains records and generate report of all trains
- Processing booking records and generate report of all booking
- Processing ticket records and generate report of all ticket
- Processing train schedule records and generate report of all train schedule
- Processing train route records and generate report of all train route
- Processing customer records and generate report of all customer
- Processing payment records and generate report of all payment

Second Level DFD:



Working Process:

- Admin logins to the system and manage all the functionalities of railway reservation system.
- Admin can add, edit, delete and view the records of trains, ticket, train route, payment.
- Admin can manage all the details of booking, train schedule, customer.
- Admin can also generate reports of trains, booking, ticket, train schedule, train route, customer.
- Admin can search the details of booking, train route, customer.
- Admin can apply different level of filters on report of trains, train schedule, train route.
- Admin can tracks the detailed information of booking, ticket, train schedule, train route.

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Solution Architecture:

Solution Architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. It’s goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior and other aspects of the software to project stakeholders.

- Define features, development phases and solution requirements.
- Provide specifications according to which the solution is defined, managed and delivered.

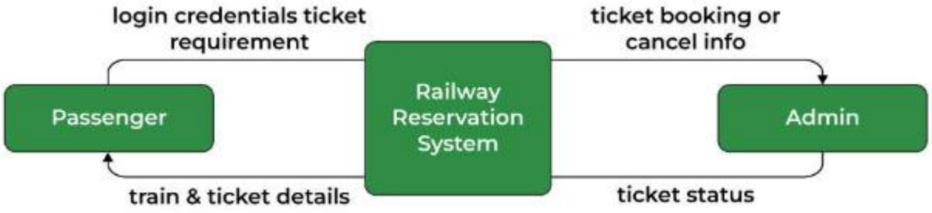

login credentials ticket requirement
ticket booking or cancel info
Passenger
Railway Reservation System
Admin
train & ticket details
ticket status

Fig.5 Solution Architecture

Technical Architecture:

Technical Architecture – which is often referred to as application architecture, IT architecture, business architecture, etc. – refers to create a structured software solution that will meet the business needs and expectations while providing a strong technical plan for the growth of the software application through lifetime. IT architecture is equally important to the business team and the information technology team.

Technical architecture includes the major components of the system, their relationships and the contracts that define the interactions between the components. The goal of technical architects is to achieve all the business needs with an application that is optimized for both performance and security. IT architects plan for things they know are coming in the future and for things they don't yet envision or dream. Taking the time to design the architecture at the start will prevent major design changes, code refactoring and expensive rework later in the project.

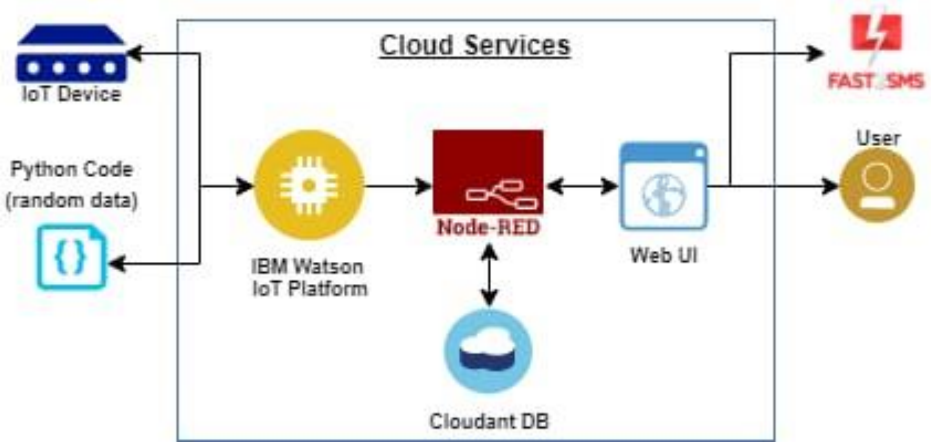


Fig.6 Technical Architecture

5.3 USER STORIES:

A user story is an informal, general explanation of a software feature written from the perspective of an end user. Its purpose is to articulate how a software feature will provide value to the customer. It’s tempting to think that user stories are simply put software system requirements, but they are not.

A key component of agile software development is putting people first and a user story puts end users at the center of the conversation. These stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building, what they are building and what value it creates. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work – which drives collaboration, creativity and a better product overall.

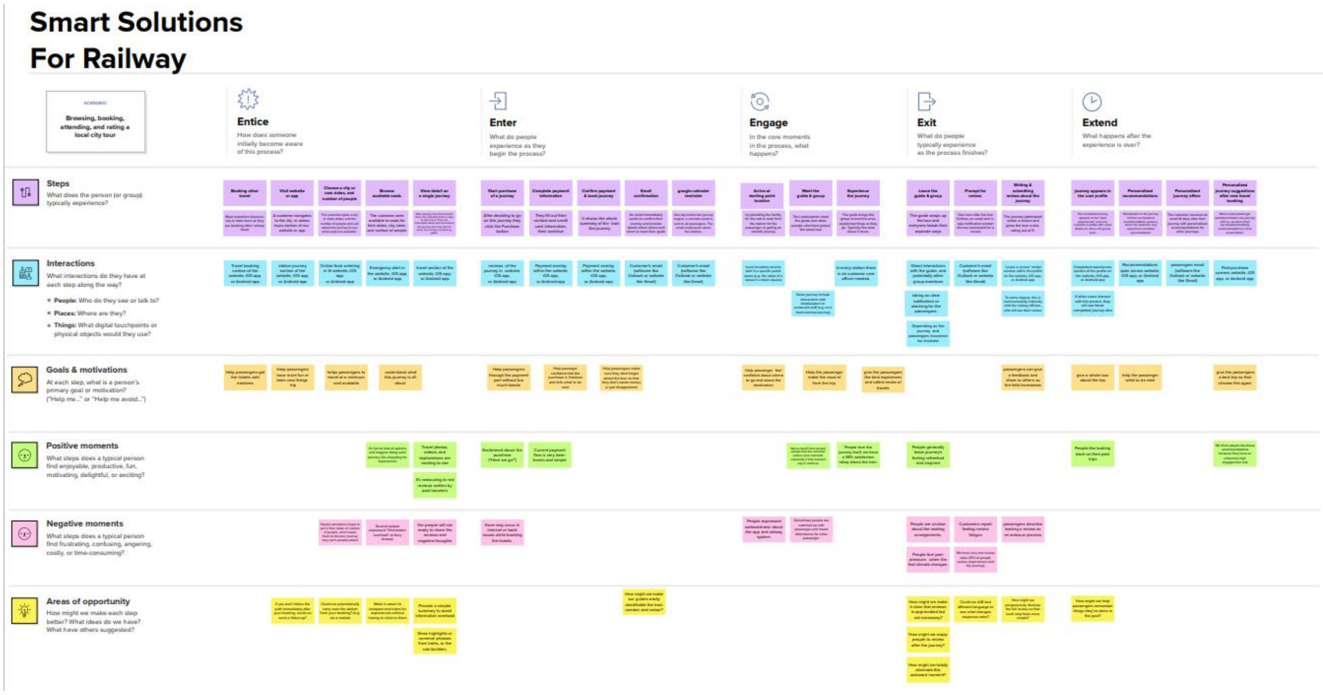

Smart Solutions
For Railway
SCENARIO
Browsing, booking, attending, and rating a local city tour
Entice
How does someone initially become aware of this process?
Enter
What do people experience as they begin the process?
Engage
In the core moments in the process, what happens?
Exit
What do people typically experience as the process finishes?
Extend
What happens after the experience is over?
Steps
What does the person (or group) typically experience?
Interactions
What interactions do they have at each step along the way?
People: Who do they see or talk to?
Places: Where are they?
Things: What digital touchpoints or physical objects would they use?
Goals & motivations
At each step, what is a person's primary goal or motivation? ("Help me..." or "Help me avoid...")
Positive moments
What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?
Negative moments
What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?
Areas of opportunity
How might we make each step better? What ideas do we have? What have others suggested?

Fig.7 Customer Journey

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION:

User type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer (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
Customer (Mobile user)	Registration	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
Customer (Mobile user)	Registration	USN-3	As a user, I can register for the application through Gmail	I can receive regular updates if wanted and save time to registration and get a QR code for reservation tickets	Medium	Sprint-1
Customer (Mobile user)	Login	USN-4	As a user, I can log into the application by entering email & password	I can access my profile and dashboard	High	Sprint-1
Customer (Mobile user)	Registration	USN-5	As a user I can search available train	I can access trains available	High	Sprint-2

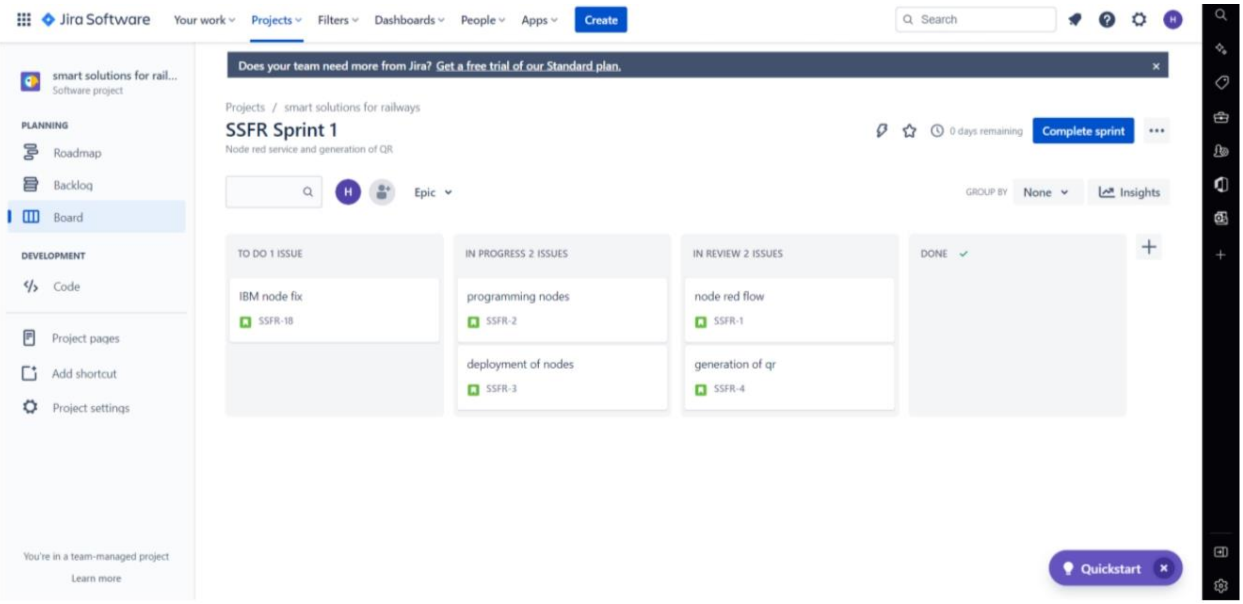
			by entering a location and can choose train to book tickets	seat or berth reservation		
Customer (Mobile user)	Dashboard	USN-6	As a user I can see my dashboard once logged into application	I can see recent activities which I have done and access the generated QR code for reserved tickets	High	Sprint-2
Customer (Web user)	Tracking	USN-7	As a passenger, I can know where the train is by using the application.	I can instantly know when will reach the destination through GPS tracking	Medium	Sprint-3
Customer Care Executive	Help Users to solve issues	USN-8	As a customer care executive, I have to take action for the customer complaints, request and query.	I can navigate the customers to find where the issue is	Medium	Sprint-4
Administration	Management	USN-9	As an Administrator I can manage the cloud and database.	I can report the problem to customer directly through server.	High	Sprint-3

6.2 SPRINT DELIVERY SCHEDULE:

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 REPORTS FROM JIRA:



Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / smart solutions for railways

SSFR Sprint 1

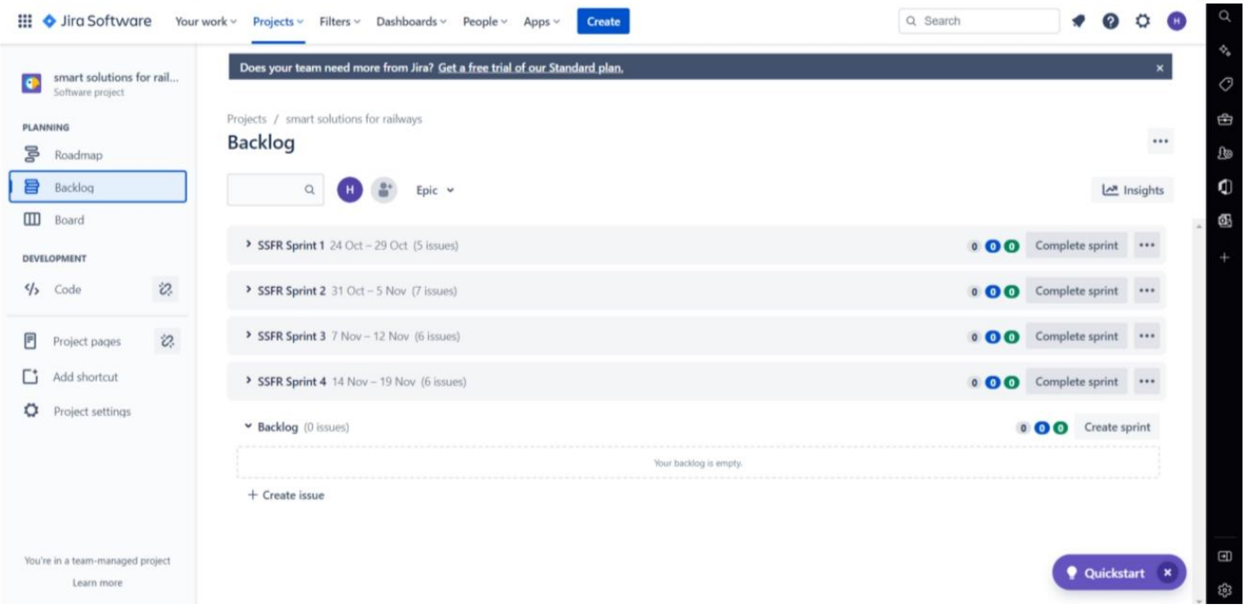
Node red service and generation of QR

0 days remaining

TO DO 1 ISSUE	IN PROGRESS 2 ISSUES	IN REVIEW 2 ISSUES	DONE ✓
IBM node fix SSFR-18	programming nodes SSFR-2	node red flow SSFR-1	
	deployment of nodes SSFR-3	generation of qr SSFR-4	

You're in a team-managed project

Learn more



Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / smart solutions for railways

Backlog

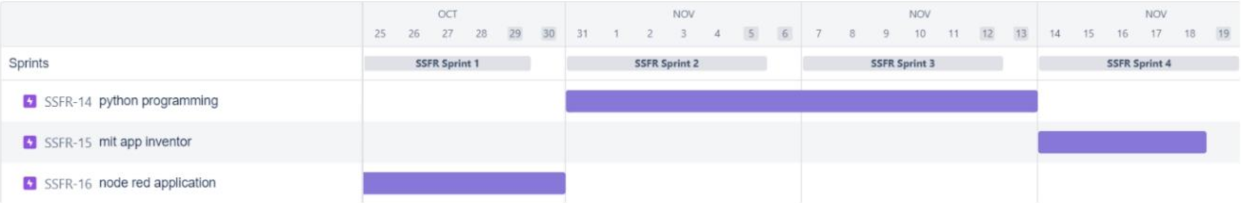
SSFR Sprint 1 24 Oct – 29 Oct (5 issues)	0 0 0	Complete sprint
SSFR Sprint 2 31 Oct – 5 Nov (7 issues)	0 0 0	Complete sprint
SSFR Sprint 3 7 Nov – 12 Nov (6 issues)	0 0 0	Complete sprint
SSFR Sprint 4 14 Nov – 19 Nov (6 issues)	0 0 0	Complete sprint
Backlog (0 issues)	0 0 0	Create sprint

Your backlog is empty.

+ Create issue

You're in a team-managed project

Learn more



	OCT 25	26	27	28	29	30	NOV 31	1	2	3	4	5	6	NOV 7	8	9	10	11	12	13	NOV 14	15	16	17	18	19
Sprints	SSFR Sprint 1						SSFR Sprint 2							SSFR Sprint 3							SSFR Sprint 4					

SSFR-14 python programming

SSFR-15 mit app inventor

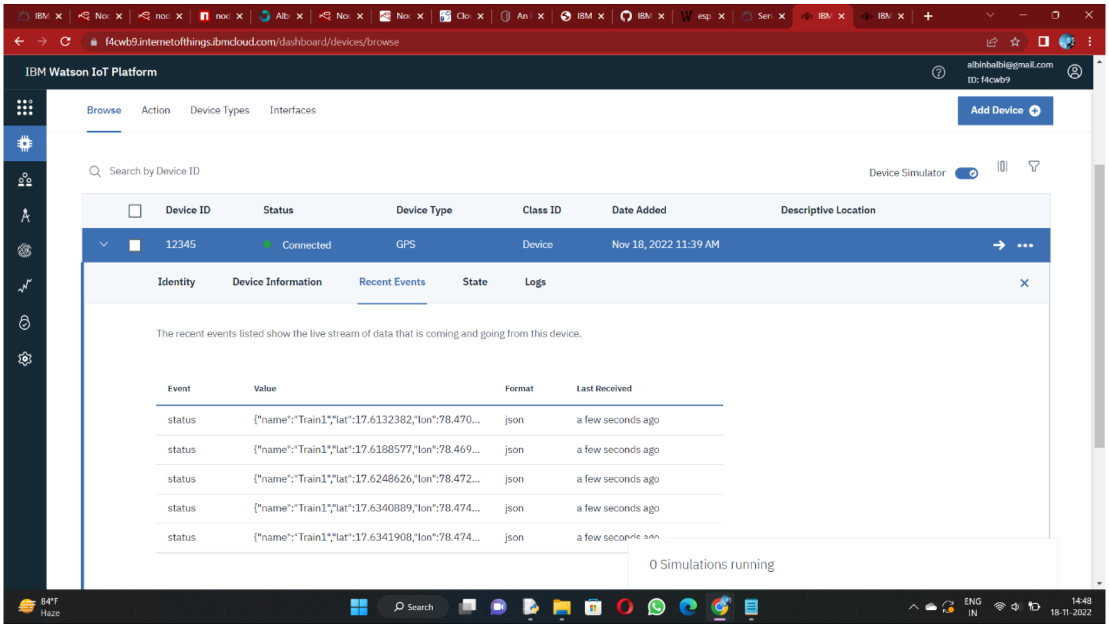
SSFR-16 node red application

CHAPTER 7

CODING & SOLUTIONING

7.1 FEATURE-1

- IOT device
- **IBM Watson platform**



f4cwb9.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform

albinbalbi@gmail.com
ID: f4cwb9

Browse Action Device Types Interfaces

Add Device

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	GPS	Device	Nov 18, 2022 11:39 AM	

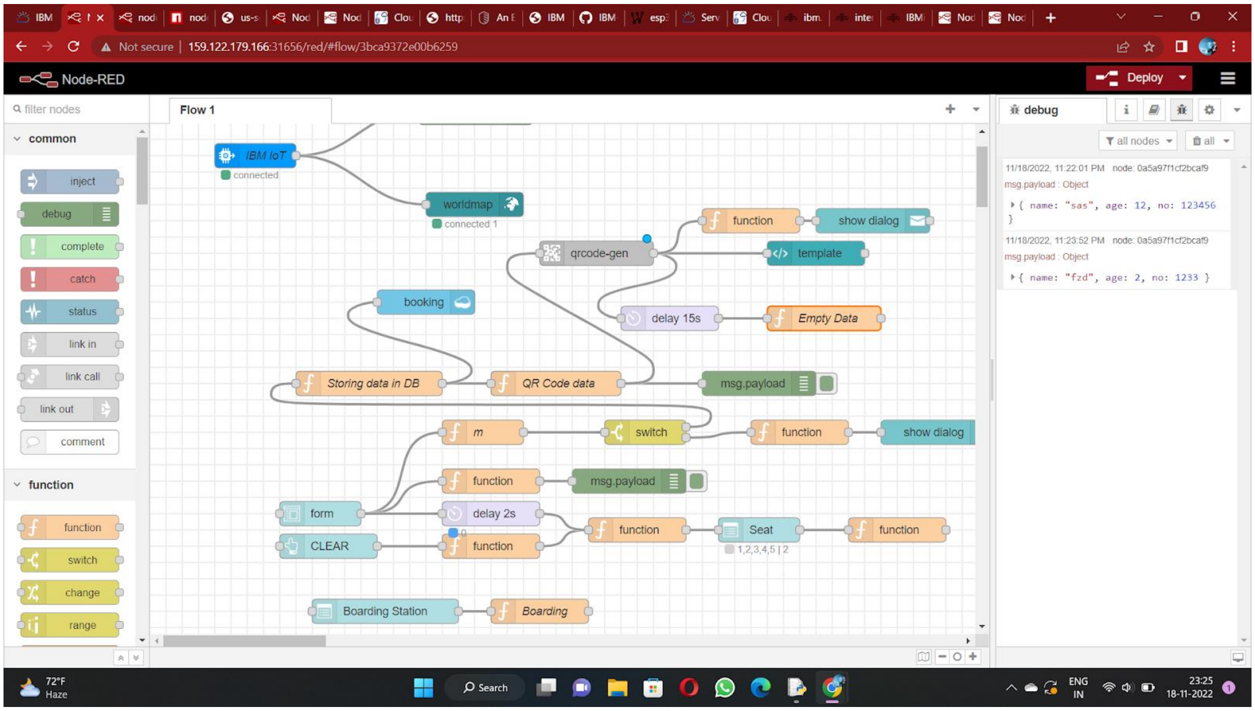
Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
status	{"name":"Train1","lat":17.6132382,"lon":78.470...	json	a few seconds ago
status	{"name":"Train1","lat":17.6188577,"lon":78.469...	json	a few seconds ago
status	{"name":"Train1","lat":17.6248626,"lon":78.472...	json	a few seconds ago
status	{"name":"Train1","lat":17.6340889,"lon":78.474...	json	a few seconds ago
status	{"name":"Train1","lat":17.6341908,"lon":78.474...	json	a few seconds ago

0 Simulations running

- **Node red**

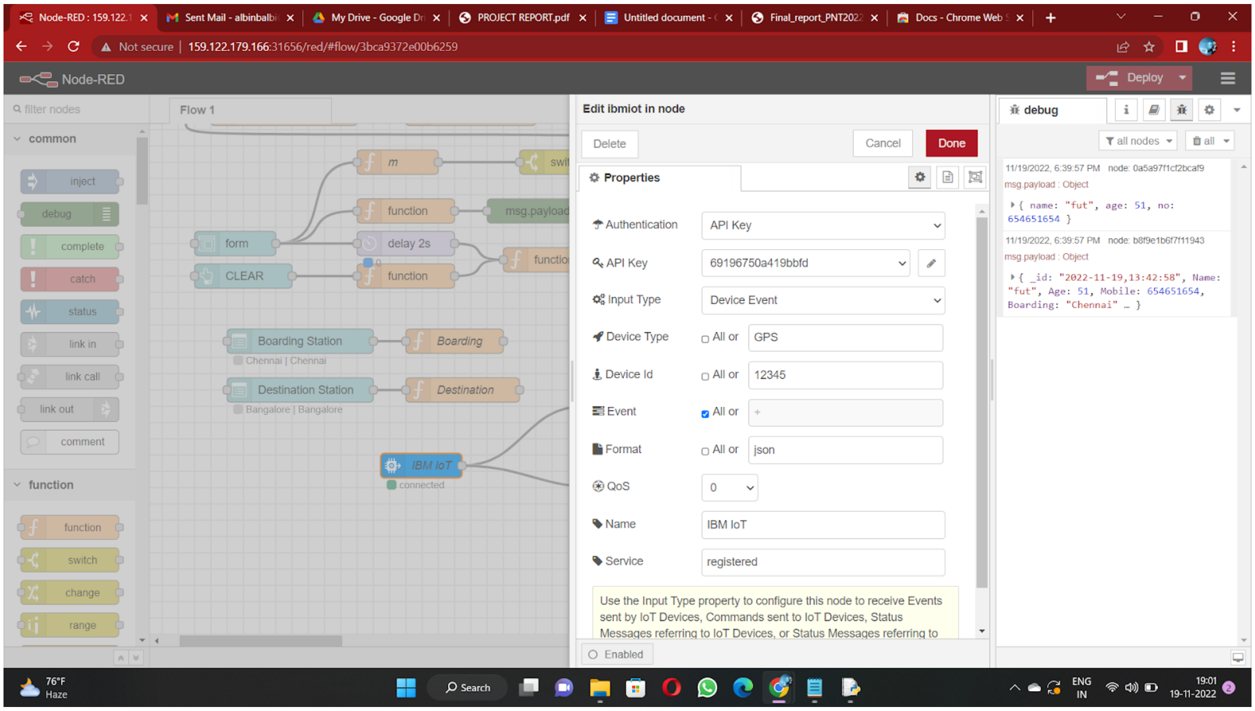


159.122.179.166:31656/red/#flow/3bca9372e00b6259

```
11/18/2022, 11:22:01 PM   node: 0a5a97f1cf2bcaf9
msg.payload : Object
▸{ name: "sas", age: 12, no: 123456
}
11/18/2022, 11:23:52 PM   node: 0a5a97f1cf2bcaf9
msg.payload : Object
▸{ name: "fzd", age: 2, no: 1233 }
```

- Cloudant DB

- **Web UI**



159.122.179.166:31656/red/#flow/3bca9372e00b6259

Edit ibmiot in node

Delete   Cancel   Done

Properties

Authentication	API Key	
API Key	69196750a419bbfd	
Input Type	Device Event	
Device Type	☐ All or	GPS
Device Id	☐ All or	12345
Event	☑ All or	+
Format	☐ All or	json
QoS	0	
Name	IBM IoT	
Service	registered	

Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to

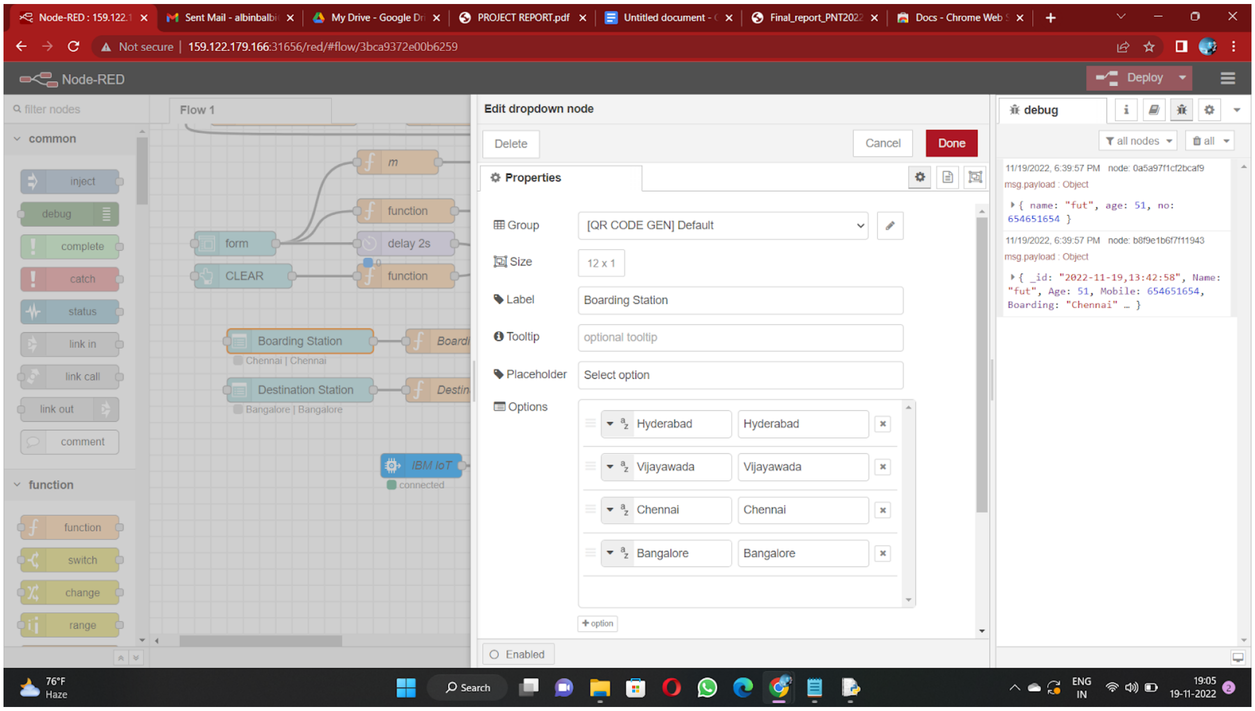
○ Enabled

```
11/19/2022, 6:39:57 PM   node: 0a5a97f1cf2bcaf9
msg.payload : Object
▸{ name: "fut", age: 51, no:
654651654 }
11/19/2022, 6:39:57 PM   node: b8f9e1b6f7f11943
msg.payload : Object
▸{ _id: "2022-11-19,13:42:58", Name:
"fut", Age: 51, Mobile: 654651654,
Boarding: "Chennai" … }
```

- Geo-fence MIT App

- Python code

BOARD STATION AND DESTINATION:



Edit dropdown node

Enabled

Group	[QR CODE GEN] Default	
Size	12 x 1	
Label	Boarding Station	
Tooltip	optional tooltip	
Placeholder	Select option	
Options	Hyderabad	Hyderabad
	Vijayawada	Vijayawada
	Chennai	Chennai
	Bangalore	Bangalore

debug

```
11/19/2022, 6:39:57 PM   node: 0a5a97f1cf2bcaf9
msg.payload : Object
{ name: "fut", age: 51, no:
654651654 }
11/19/2022, 6:39:57 PM   node: b8f9e1b6f7f11943
msg.payload : Object
{ _id: "2022-11-19,13:42:58", Name:
"fut", Age: 51, Mobile: 654651654,
Boarding: "Chennai" … }
```

Function Mode:

m:

```
global.set('m', msg.payload)
var a = global.get('s')
if(a==1 || a == 2 || a==3 || a ==4 || a==5){
  msg.payload = 0
}
else{
  msg.payload = 1
}
return msg;
```

Function:

```
var s = global.get('s')
var a = global.get('a')
function rem(x){
  for(let i=0; i<a.length; i++){
    if(a[i]==x){
      a.splice(i,1)
    }
  }
}
```

```
if(s == 1){
   global.set('s1',s)
   rem(s)
}
else if(s == 2){
   global.set('s2',s)
   rem(s)
}
else if(s == 3){
   global.set('s3',s)
   rem(s)
}
else if(s == 4){
   global.set('s4',s)
   rem(s)
}
else if(s == 5){
   global.set('s5',s)
   rem(s)
}
return msg;
```

Function:

```
global.set('s1',0)
global.set('s2',0)
global.set('s3',0)
global.set('s4',0)
global.set('s5',0)
var a1 = [1,2,3,4,5]
global.set('a',a1)
msg.payload = global.get('a')
return msg;
```

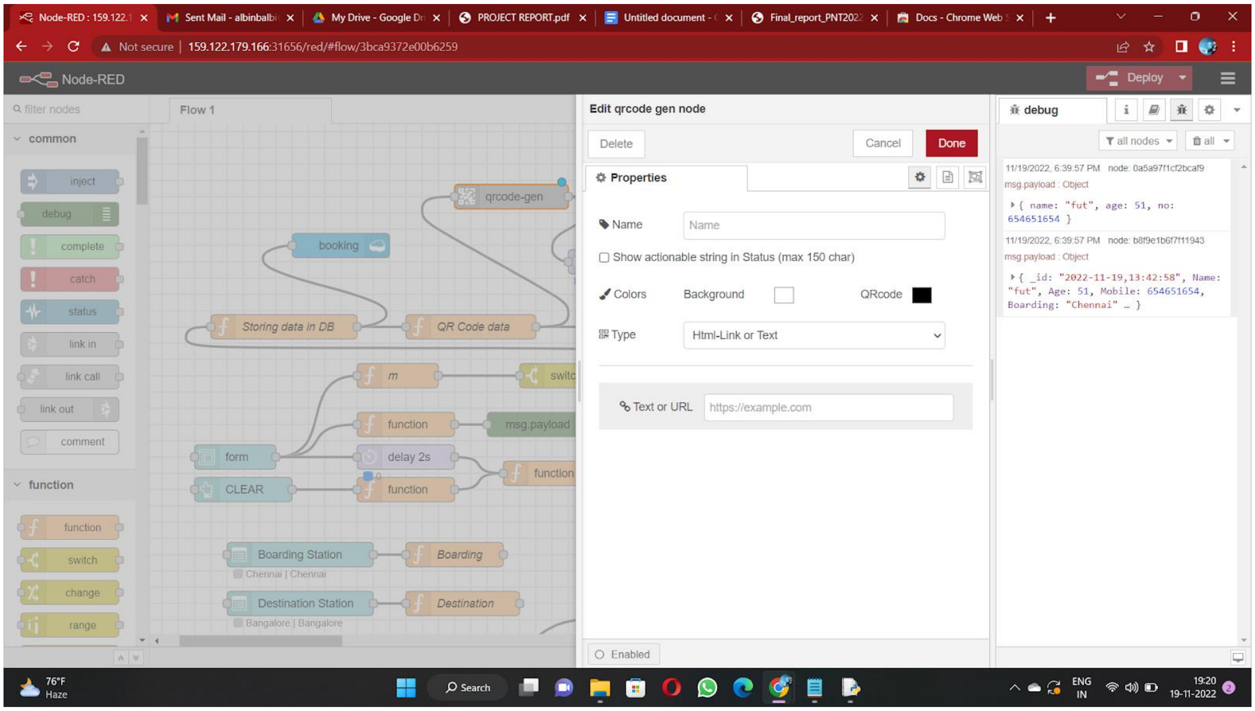
Function:

```
var a = global.get('a')
var s = []
for(let i=0; i<a.length; i++){
   s.push(a[i])
}
if(s.length==0){
   msg.options = [{"No seats Available":0}]
}
else{
   msg.options = s
}
msg.payload = s
return msg;
```

Storing data in DB:

```
var m = global.get('m')
var d = new Date();
var utc = d.getTime() + (d.getTimezoneOffset() * 60000);
var offset = 5.5;
newDate = new Date(utc + (360000*offset));
var n=newDate.toISOString()
var date = n.slice(0,10)
var time = n.slice(11,19)
var d1 = date+','+time
msg.payload = {
   "_id" : d1,
   "Name" : m.name,
   "Age" : m.age,
   "Mobile" : m.no,
   "Boarding" : global.get('b'),
   "Destination" : global.get('d'),
   "Seat" : global.get('s')
}
return msg;
```

QR CODE GENERATION:



159.122.179.166:31656/red/#flow/3bca9372e00b6259

Edit qrcode gen node

Delete | Cancel | Done

Properties

Name: Name

☐ Show actionable string in Status (max 150 char)

Colors: Background ☐ QRcode ■

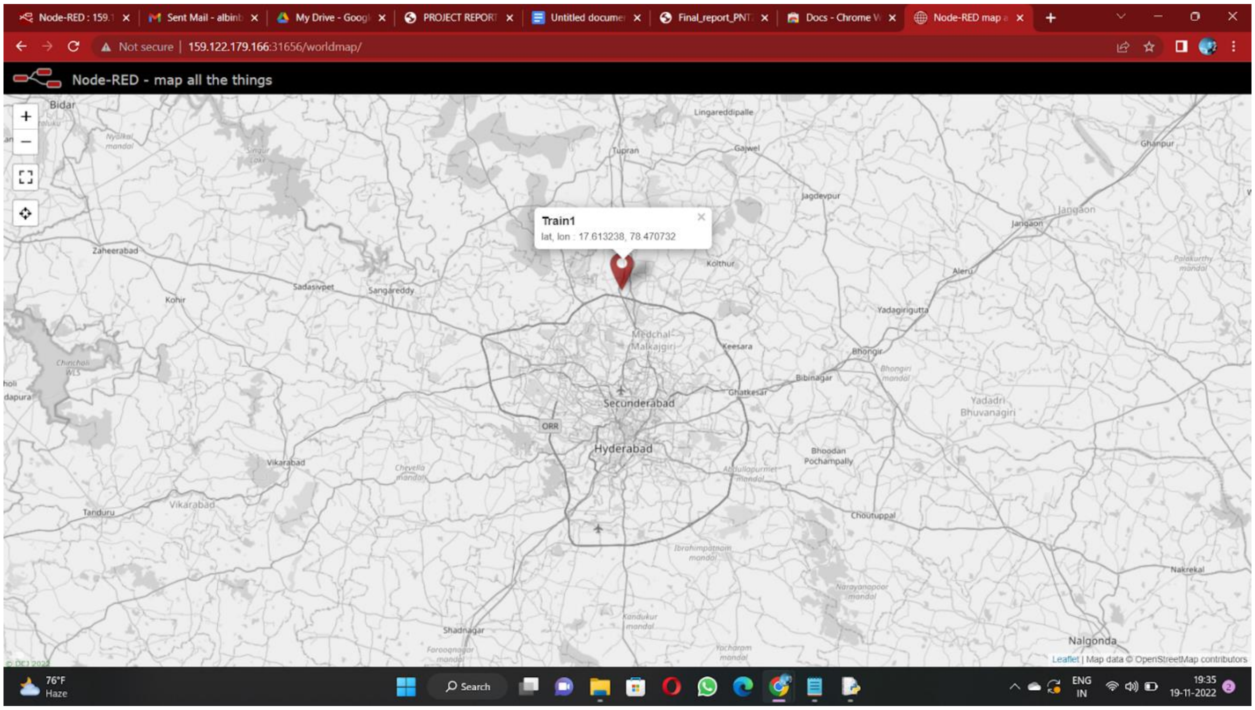
Type: Html-Link or Text

Text or URL: https://example.com

○ Enabled

```
11/19/2022, 6:39:57 PM   node: 0a5a97f1cf2bcaf9
msg.payload : Object
▸{ name: "fut", age: 51, no:
654651654 }
11/19/2022, 6:39:57 PM   node: b8f9e1b6f7f11943
msg.payload : Object
▸{ _id: "2022-11-19,13:42:58", Name:
"fut", Age: 51, Mobile: 654651654,
Boarding: "Chennai" … }
```

NODE RED WORLD MAP:


159.122.179.166:31656/worldmap/
Node-RED - map all the things
Train1
lat, lon : 17.613238, 78.470732
Secunderabad
Hyderabad
Leaflet | Map data © OpenStreetMap contributors

7.2 FEATURE-2

- Registration
- Login
- Verification
- Ticket Booking
- Payment
- Ticket Cancellation
- Adding Queries

7.3 DATABASE SCHEMA:

LOCATION DATA:

```
import wiotp.sdk.device

import time

import random

myConfig={

"identity": (
 "orgId": "f4cwb9",
 "typeId": "GPS",
 "deviceId":"12345"},
 "auth": {
 "token": "S1Dl?V9*ZIB)3RN)Xb"
 }}
 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()
def pub (data):
 client.publishEvent (eventId="status", msgFormat="json",
 data=myData, qos=0, print("Published data Successfully: %s",
 myData)
 while True:
 myData={'name': 'Train1', 'lat': 17.6387448, 'lon':
 78.4754336)
 pub (myData)
 time.sleep (3)
 #myData('name': 'Train2', 'lat': 17.6387448, 'lon':
 78.4754336)
 #pub (myData)
 #time.sleep (3)
 myData={'name': 'Train1', 'lat': 17.6341908, 'lon':
78.4744722)
 pub (myData)
```

```
time.sleep(3)
myData={'name': 'Trainl', 'lat': 17.6340889, lon': 78.4745052)
pub (myData)
time.sleep(3)
myData={'name': 'Trainl', 'lat': 17.6248626, 'lon': 78.4720259)
pub (myData)
time.sleep (3)
myData={'name': 'Trainl', 'lat': 17.6188577, 'lon': 78.4698726)
pub (myData)
time.sleep (3)
myData={'name': 'Train1', 'lat': 17.6132382, 'lon':
78.4707318)
pub (myData)
time.sleep (3)
client.commandCallback = myCommandCallback
client.disconnect()
```

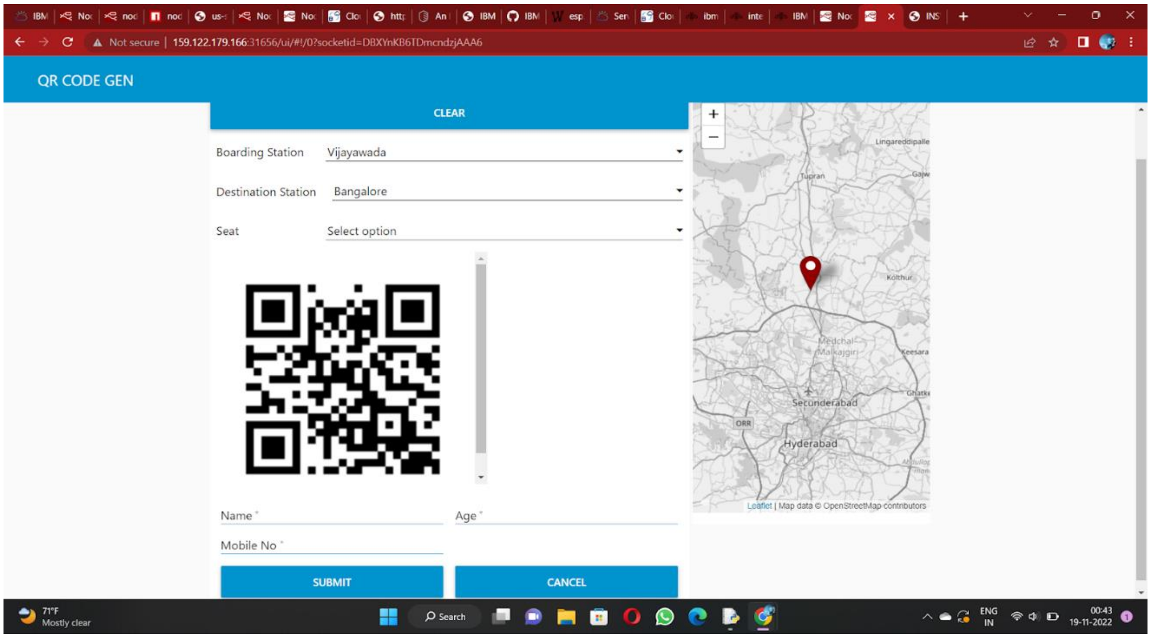
QR SCANNER CODE:

```
import cv2
import numpy as np
import time
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-
16u3crmdpkghhxefdikvpssoh5fw
ezrmuup5fv5g3ubz',
'b0ab119f45d3e6255eabb978')
service=Cloudant V1
(authenticator=authenticator)
service.set_service_url('https://api
key-v2-
16u3crmdpkghhxefdikvpssoh5fw
ezrmuup5fv5g3ubz:b0ab119f45d3
e6255eabb978e7e2f0')
cap= cv2.VideoCapture (0)
font cv2.FONT HERSHEY
PLAIN
while True:
      frame=cap.read(0)
     decodedobjects pyzbar.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='booking', doc_id = a
).get_result()
   print (response)
   time.sleep(5)
    except Exception as e:
        print ("Not a Valid
Ticket")
        time.sleep (5)
cap.inshow("Frame", frame)
if cv2.waitKey(1) &
0xFF==ord('q'):
    Break
cap.release()
cv2.destroyAllWindows ()
client.disconnect()
```

CHAPTER 8
TESTING

8.1 TEST CASES:



QR CODE GEN

Boarding Station: Vijayawada

Destination Station: Bangalore

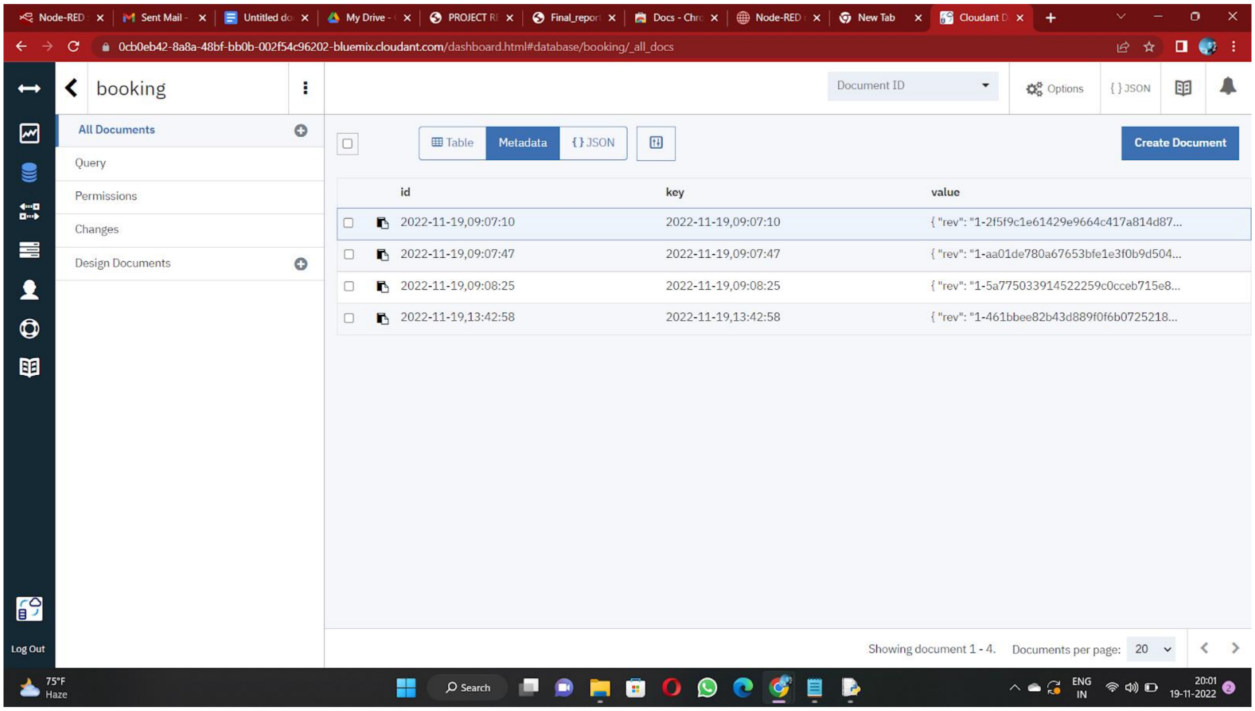
Seat: Select option

Name *

Age *

Mobile No *

DATABASE:



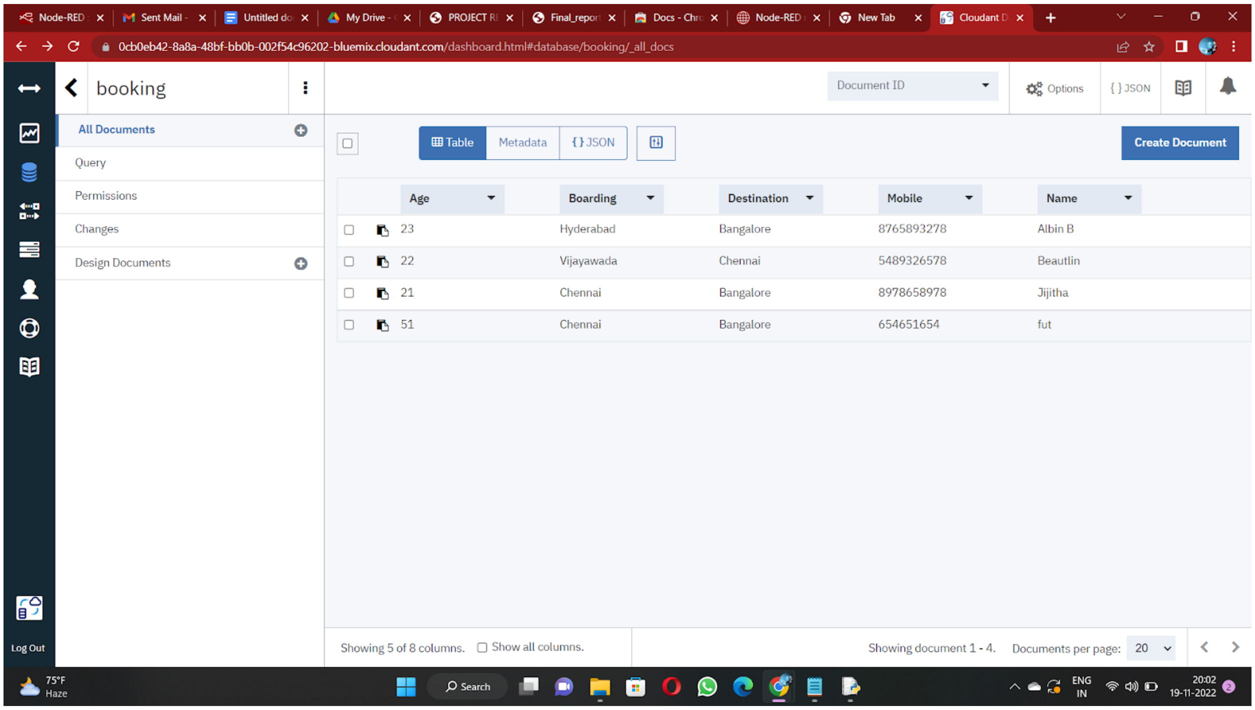
0cb0eb42-8a8a-48bf-bb0b-002f54c96202-bluemix.cloudant.com/dashboard.html#database/booking/_all_docs

booking

id	key	value
2022-11-19,09:07:10	2022-11-19,09:07:10	{ "rev": "1-2f5f9c1e61429e9664c417a814d87...
2022-11-19,09:07:47	2022-11-19,09:07:47	{ "rev": "1-aa01de780a67653bfe1e3f0b9d504...
2022-11-19,09:08:25	2022-11-19,09:08:25	{ "rev": "1-5a775033914522259c0cceb715e8...
2022-11-19,13:42:58	2022-11-19,13:42:58	{ "rev": "1-461bbee82b43d889f0f6b0725218...

Showing document 1 - 4. Documents per page: 20

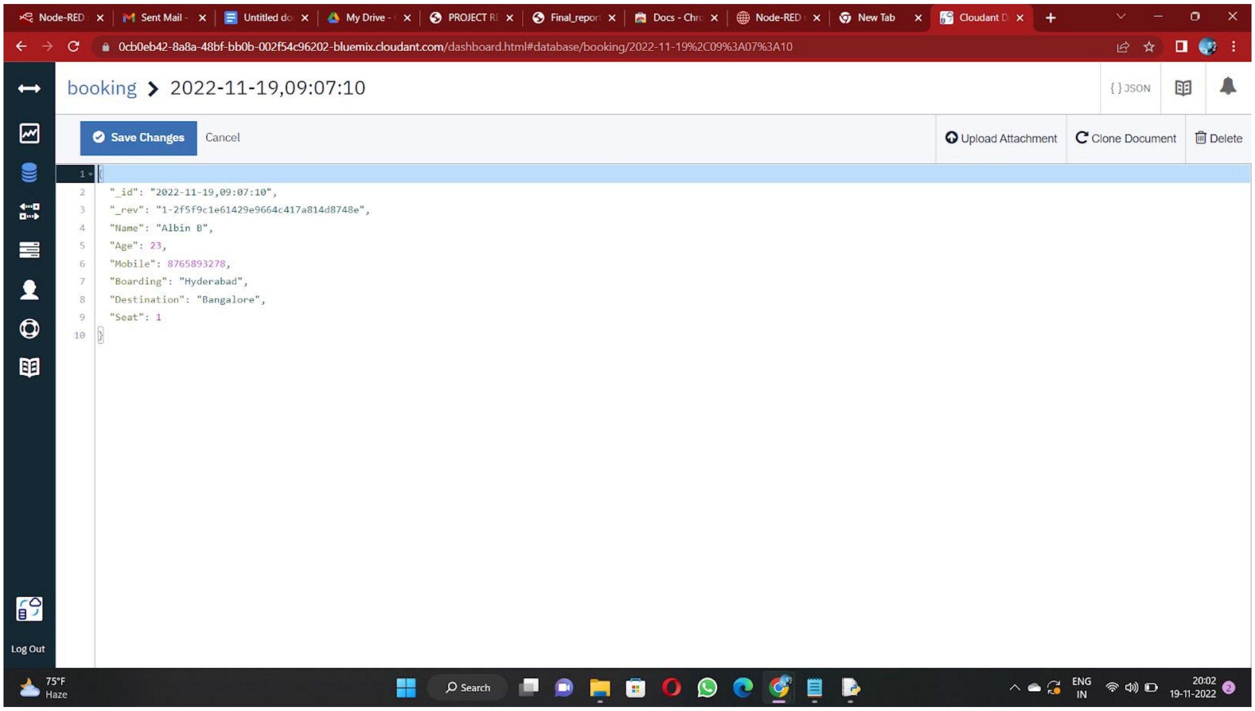
BOOKING DETAILS:



Age	Boarding	Destination	Mobile	Name
23	Hyderabad	Bangalore	8765893278	Albin B
22	Vijayawada	Chennai	5489326578	Beautlin
21	Chennai	Bangalore	8978658978	Jijitha
51	Chennai	Bangalore	654651654	fut

Showing 5 of 8 columns. □ Show all columns. Showing document 1 - 4. Documents per page: 20

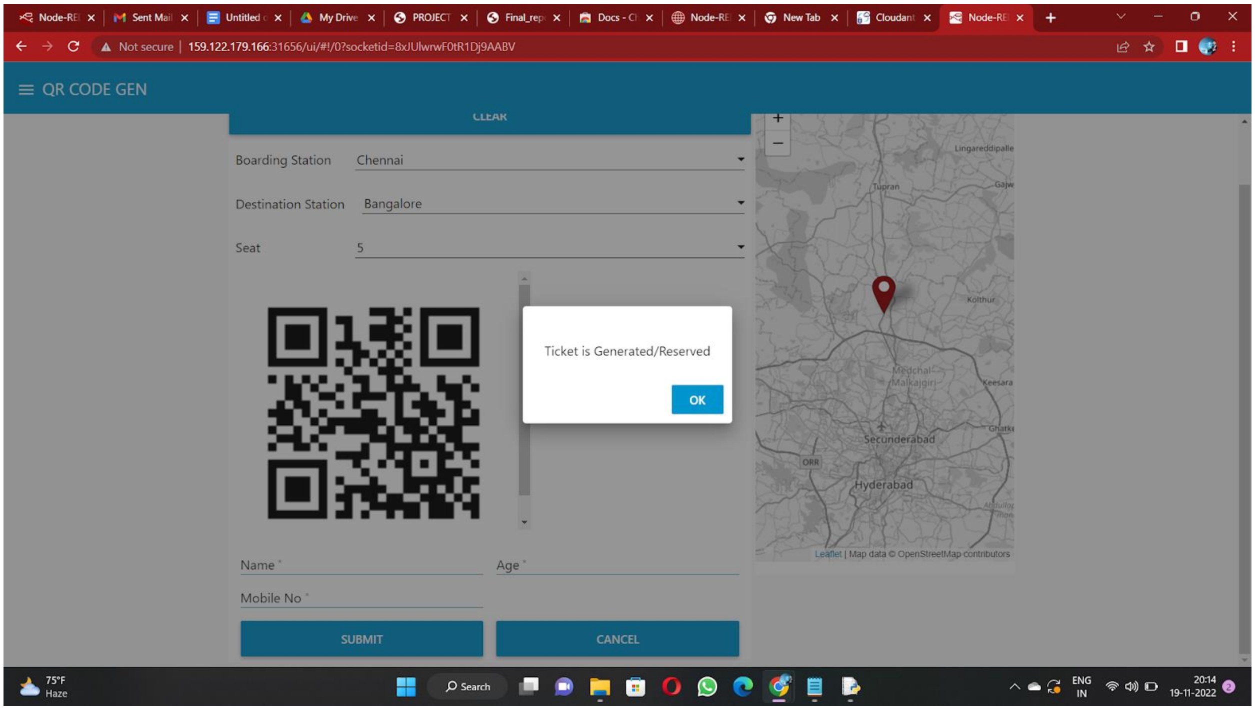
PERSONAL DETAIL AND ID INFORMATION:



0cb0eb42-8a8a-48bf-bb0b-002f54c96202-bluemix.cloudant.com/dashboard.html#database/booking/2022-11-19%2C09%3A07%3A10
booking > 2022-11-19,09:07:10
{ } JSON
Save Changes
Cancel
Upload Attachment
Clone Document
Delete
Log Out

```
{
  "_id": "2022-11-19,09:07:10",
  "_rev": "1-2f5f9c1e61429e9664c417a814d8748e",
  "Name": "Albin B",
  "Age": 23,
  "Mobile": 8765893278,
  "Boarding": "Hyderabad",
  "Destination": "Bangalore",
  "Seat": 1
}
```

8.2 USER ACCEPTANCE TESTING:


Not secure | 159.122.179.166:31656/ui/#!/0?socketid=8xJUlwrwF0tR1Dj9AABV
QR CODE GEN
CLEAR
OK
Name *
Age *
Mobile No *
SUBMIT
CANCEL
Secunderabad
Hyderabad
Medchal-Malkajgiri
Leaflet | Map data © OpenStreetMap contributors

Boarding Station: Chennai

Destination Station: Bangalore

Seat: 5

Ticket is Generated/Reserved

- Ticket is generated.
- Ticket Confirmation message the passenger are recived.
- GPS map location.

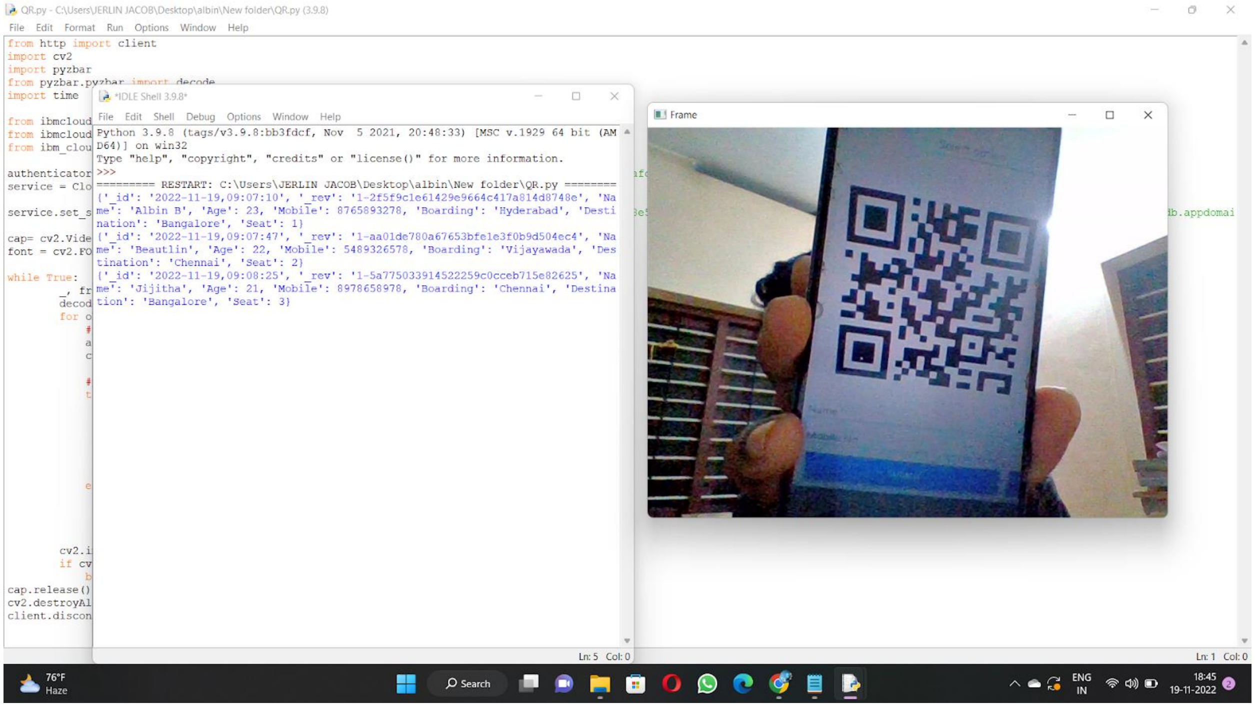
CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS:

The performance and the working of the code is very quick and the results appear in quick succession. Our code is linked with the most used IBM Watson IoT Platform which works with much perfection. This cloud platform is very secure to use and configure easily. As the code is simulated within seconds, the result appears. We have done a lot of works using this IoT platform which is very simple and good user friendly platform. Below we displayed our connected IoT platform which delivers the results as the code run.

QR CODE SCANNER:


QR.py - C:\Users\JERLIN JACOB\Desktop\albin\New folder\QR.py (3.9.8)
File Edit Format Run Options Window Help
IDLE Shell 3.9.8
File Edit Shell Debug Options Window Help
Frame
Ln: 5 Col: 0
Ln: 1 Col: 0
76°F
Haze
Search
ENG
IN
18:45
19-11-2022

```
from http import client
import cv2
import pyzbar
from pyzbar.pyzbar import decode
import time

from ibmcloud
from ibmcloud
from ibm_clou

authenticator
service = Clo

service.set_s

cap= cv2.Vide
font = cv2.FO

while True:
        _, fr
        decod
        for o
            #
            a
            c

            #
            t



            e




        cv2.i
        if cv
            b
cap.release()
cv2.destroyAl
client.discon
```

```
Python 3.9.8 (tags/v3.9.8:bb3fdcf, Nov  5 2021, 20:48:33) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:\Users\JERLIN JACOB\Desktop\albin\New folder\QR.py ========
{'_id': '2022-11-19,09:07:10', '_rev': '1-2f5f9c1e61429e9664c417a814d8748e', 'Na
me': 'Albin B', 'Age': 23, 'Mobile': 8765893278, 'Boarding': 'Hyderabad', 'Desti
nation': 'Bangalore', 'Seat': 1}
{'_id': '2022-11-19,09:07:47', '_rev': '1-aa01de780a67653bfe1e3f0b9d504ec4', 'Na
me': 'Beautlin', 'Age': 22, 'Mobile': 5489326578, 'Boarding': 'Vijayawada', 'Des
tination': 'Chennai', 'Seat': 2}
{'_id': '2022-11-19,09:08:25', '_rev': '1-5a775033914522259c0cceb715e82625', 'Na
me': 'Jijitha', 'Age': 21, 'Mobile': 8978658978, 'Boarding': 'Chennai', 'Destina
tion': 'Bangalore', 'Seat': 3}
```

- The python code to run an open camera.
- Received QR code to scan.

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- Openness – compatibility between different system modules, potentially from different vendors;
- Orchestration – ability to manage large numbers of devices, with full visibility over them;
- Dynamic scaling – ability to scale the system according to the application needs, through resource virtualization and cloud operation;
- Automation – ability to automate parts of the system monitoring application, leading to better performance and lower operation costs.

DISADVANTAGES:

- Approaches to flexible, effective, efficient, and low-cost data collection for both railway vehicles and infrastructure monitoring, using regular trains;
- Data processing, reduction, and analysis in local controllers, and subsequent sending of that data to the cloud, for further processing;
- Online data processing systems, for real-time monitoring, using emerging communication technologies;
- Integrated, interoperable, and scalable solutions for railway systems preventive maintenance.

CHAPTER 11

CONCLUSION

Using the application, user can book train tickets based on availability of seats in particular train. Once tickets are available, they can book them by inputting their general information. Upon completion of payment, the data gets stored in Cloudant DB with unique ID for every transaction and a QR code is generated for every 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.

CHAPTER 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 can be 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.

CHAPTER 13

APPENDIX

SOURCE CODE:

As we successfully developed and programmed or python code, this will be the final code of execution.

LOCATION CODE

```
import wiotp.sdk.device

import time

import random
```

```
 myConfig={
 "identity": (
 "orgId": "f4cwb9",
 "typeId": "GPS",
 "deviceId":"12345"},
 "auth": {
 "token": "S1Dl?V9*ZIB)3RN)Xb"
 }}
 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()
def pub (data):
 client.publishEvent (eventId="status", msgFormat="json",
 data=myData, qos=0, print("Published data Successfully: %s",
 myData)
 while True:
 myData={'name': 'Train1', 'lat': 17.6387448, 'lon':
 78.4754336)
 pub (myData)
 time.sleep (3)
 #myData('name': 'Train2', 'lat': 17.6387448, 'lon':
 78.4754336)
 #pub (myData)
 #time.sleep (3)
 myData={'name': 'Train1', 'lat': 17.6341908, 'lon':
```

```
78.4744722)
pub (myData)
time.sleep(3)
myData={'name': 'Trainl', 'lat': 17.6340889, lon': 78.4745052)
pub (myData)
time.sleep(3)
myData={'name': 'Trainl', 'lat': 17.6248626, 'lon': 78.4720259)
pub (myData)
time.sleep (3)
myData={'name': 'Trainl', 'lat': 17.6188577, 'lon': 78.4698726)
pub (myData)
time.sleep (3)
myData={'name': 'Train1', 'lat': 17.6132382, 'lon':
78.4707318)
pub (myData)
time.sleep (3)
client.commandCallback = myCommandCallback
client.disconnect()
```

QR SCANNER CODE:

```
import cv2
import numpy as np
import time
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-
16u3crmdpkghhxefdikvpssoh5fw
ezrmuup5fv5g3ubz',
'b0ab119f45d3e6255eabb978')
service=Cloudant V1
(authenticator=authenticator)
service.set_service_url('https://api
key-v2-
16u3crmdpkghhxefdikvpssoh5fw
ezrmuup5fv5g3ubz:b0ab119f45d3
e6255eabb978e7e2f0')
cap= cv2.VideoCapture (0)
font cv2.FONT HERSHEY
PLAIN
while True:
      frame=cap.read(0)
     decodedobjects pyzbar.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='booking', doc_id = a
).get_result()
    print (response)
    time.sleep(5)
     except Exception as e:
           print ("Not a Valid
Ticket")
           time.sleep (5)
cap.inshow("Frame", frame)
if cv2.waitKey(1) &
0xFF==ord('q'):
     Break
cap.release()
cv2.destroyAllWindows ()
client.disconnect()
```

PROJECT DEMONSTRATION VIDEO UPLOADED HERE

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-43577-1660718167

PROJECT DEMO LINK:

https://youtu.be/T043AAH0wDQ