Smart Solutions for Railways

Category: *Internet of Things*

PROJECT REPORT

SUBMITEED BY

Team ID: PNT2022TMID14986

NAME	REGISTER NUMBER
1. Nagarapu Sumadhar	111519106096
2. Muppala Kiran Sai	111519106095
3.Pola Nagamurali Prasad	111519106115
4. Pabbathi Pavan Kumar	111519106109

INPARTIAL FULFILLMENT FOR

THEAWARD OF THE DEGREE

Of

BACHELOROF ENGINEERING

in

ELECTRONICS AND

COMMUNICATIONENGINEERING

RMD ENGINEERING COLLEGE

R.S.M NAGAR – 601 206

Project Report Format

1. INTRODUCTION

a. Project Overview

b. Purpose

2. LITERATURESURVEY

a. Existingproblem

b. References

c. ProblemStatement Definition

3. IDEATION &PROPOSED SOLUTION

a. Empathy Map Canvas

b. Ideation& Brainstorming

c. ProposedSolution

d. Problem Solutionfit

4. REQUIREMENTANALYSIS

a. Functional requirement

b. Non-Functional requirements

5. PROJECT DESIGN

a. Data Flow Diagrams

1. INTRODUCTION

a. Project Overview

As trains are one of the most preferred modes of transportation amongmiddle 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 allsuch brutality we came up with a solution by providing an application which can beaccessed 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 passengerdata.

b. Purpose

The purpose of this projectis to report and get relived from the issuesrelated totrains.

2. LITERATURE SURVEY

a. Existing problem

A Web page is designedfor the public where they can book tickets by seeingthe available seats.

After booking the train, the person will get a QR code which has to be shown to theTicketCollector while boardingthe train.

The ticket collectors can scan the QR code to identifythe personal details.

A GPS moduleis present in the train to trackit. The live status of the journeyisupdated in the Web app continuously

All the bookingdetails of the customers will be storedin the database with a uniqueID and they can be retrieved back when the Ticket Collectorscans

the QR Code.

2.2 References

S. NO	TITLE	AUTHOR	YEAR	KEYTECHNOLOGY
1	Main geotechnical problems of railways and roads inkriolitozone and theirsolutions.	Kondratiev, Valentin G	2017	Main problems in railways
2	Construction andBuilding Materials	Sañudo, Roberto, Marina Miranda,Carlos García, and David García-Sanchez	2019	Drainage in railways
3	Problems of Indian Railways	Benjamin	2021	Common problems inIndian railways

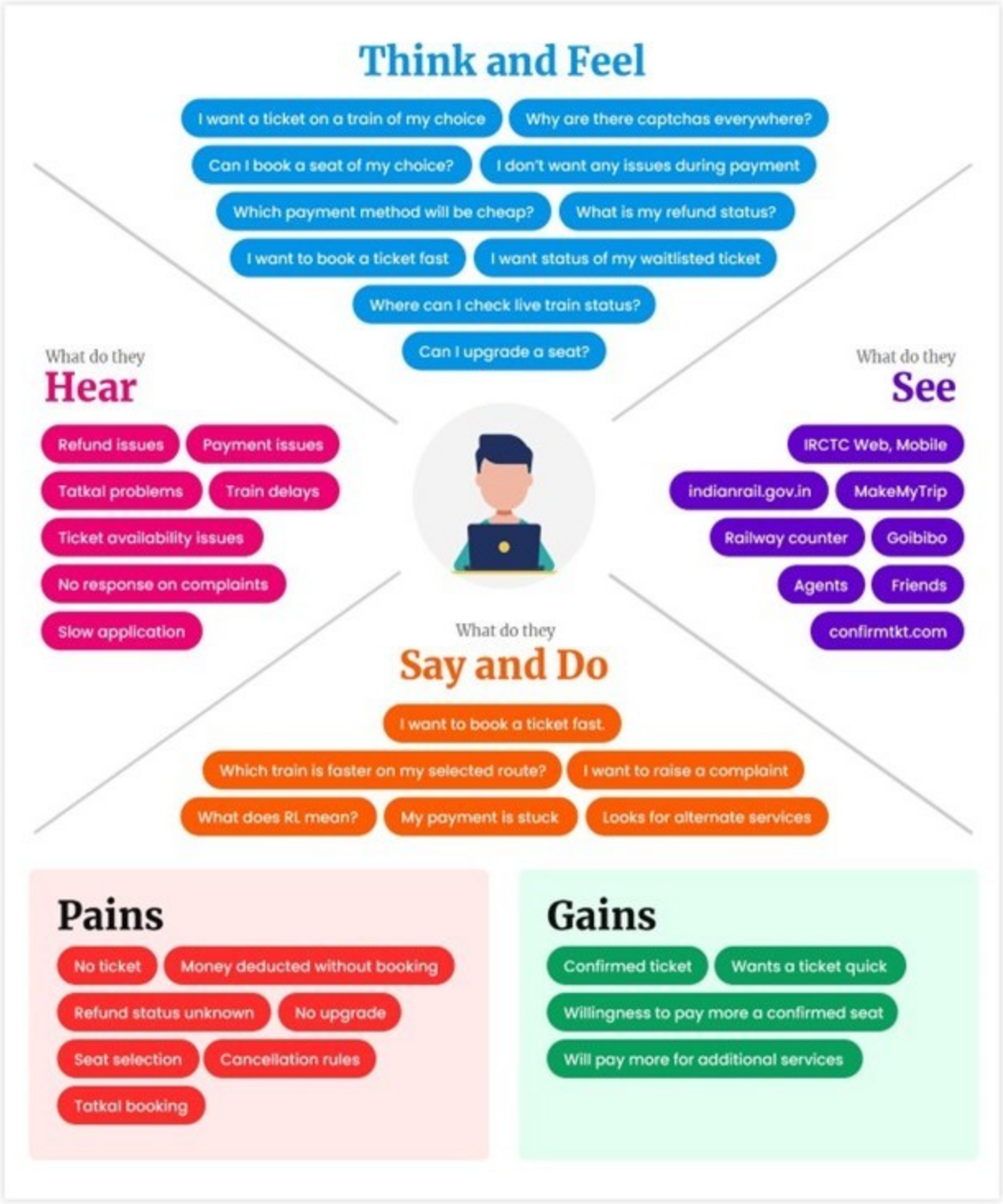
4	A comparative study of Indianand worldwide railways.	Sharma,Sunil Kumar, and AnilKumar	2014	Study of Indian railways
5	Ticketing solutions for Indian railways using RFID technology	Prasanth,Venugopal, and K.P. Soman	2009	Solution for ticketing using RFID

2.3 Problem Statement Definition

Smart Solutions for railways are designed to reduce the work load of theuser and theuse of paper.

3 IDEATION & PROPOSEDSOLUTION

3.1Empathy Map CanvasOnline Ticket Booking



3.2

a. **Ideation & Brainstorming**

i. Creating an Application for passengers

ii. Digital Railway solution

iii. Digital Twin digitalplatform for Railwaysand Airways

iv. Role of sensorsin predictive maintanance

v. Predictive maintanace and CMMS

vi. The IOT connected trains

vii. Big Data analytics for smart Railways

Safety is a key area of connection

Idea prioritization:

i. To prect from:

ii. Ticket booking Jamming

iii. Fire accident

iv. Theft

v. Robbery

Include Features like:

vi. Tracking management

vii. QR code

3.3Proposed solutions

C	CC	A
S 1- CUSTOMER SEGMENT(S)	**6-CUSTOMER CONTRAINTS**	**S 5-AVAILABE SOLUTIONS**

Passengers are thecustomers.	Fewer Maintenance Delays Restructured and Optimized Passenger	AGPS tracking devicewill be placedin train whichis helpful to find the live statusof the train. Booking tickets is made
	Experience Advanced Analytics for Streamlined Operations	easier from a webpageand for each ticket a unique QR will be provided.

J &P	RC	BE
2- JOBS TO BE DONE /PROBLEMS2- **Ticket:** To provide a web page orweb appto the customers tobook their Railway tickets from anywhere at any time. **Tracking:** The livestatus of the trainmust be updated to the passengers.	**9. PROBLEM ROOTCAUSE** The Passengers it difficult to get the ticketby Standing in queue. Atthe sametime they cantable to know the information about the delay of train. To overcome this problem we provide a unique QR and GPS module was installed inthe train is used to trackit.	**7. BEHAVIOUR** According to the needsofthe passengers we should provide a genuineempathy for the problem regarded. Looking over the rating sectionwe can easilyfind out how the customer getsissue while usingthe application.

T R	SL	C H
3. TRIGGERS	**10.YOUR SOLUTION**	**8.CHANNELS of BEHAVIOUR** **ONLINE** Customers try to request for the problems through the application how they use and how it is favouring them using the rating option by which we can findthe behaviour of the customer and issues or problems they face. **OFFLINE** By direct booking of ticketthey need to be in a queue for receiving a ticket which seems to be a big deal for the customers.

Customer can be triggeredto the application by the usage of their neighbours. 4. EMOTIONS Before: They feel nervous because there is no option to proceed further and if they miss the train they can’ttrackit too. After: Now the passengers can track the live location ofthe train and will never lose their confidence.	A web page will be provided and the passenger can signin the page and they can book their train ticket using it. When a ticket is booked the passenger will get a unique QR code for further verifications by the railway department. The passenger can also trackthe live statusof the train in that web page.	

3.3 ProposedSolution

S.No	Parameter	Description
1	Problem Statement (Problem to be solved)	To provide a smart way for booking tickets in railway department through a webpage with auniqueQR for eachticket and to deliver the live status of the train to the passengers which is helpful in the critical situations (Stuck of traininforest areas)
2	Idea/ Solution description	Passengers can book their ticket using a web pageor web app. When the passenger is booking a ticketand successfully completed the payment for it, they will be provided with a unique QR code which contains the ticket details and passenger details. The passengers willget notified withthe train timings and train's live status.

3	Novelty/ Uniqueness	Efficient booking system by verifying and validatingthe ticket as only registered users can book the tickets. Each passengers willbe provided by a uniqueID tothem during first login so that their data will be stored and processed securely. GPS tracking facility is provided to track the currentlocation of the trainfrom any place. A chat box will be provided for the passengers topost theirqueries or theirneeds and thatwill be fulfilled as soonas possible
4	Social Impact/ Customer Satisfaction	User friendly environment Services willbe made for 24 x 7 Passenger data willbe more securely maintained Reservation of tickets madeeasier
5	Business Model (Revenue Model)	Using chat bot we can contact user's ticket booking. The chatbox can giveinstructions to theusers based on their location. It will storethe customer's details and ticketorders in the

		database. The chatbot will senda notification tothe passenger if the booking is confirmed. Chat bot can also helpin collecting passenger feedback.
6	Scalability of thesolution	This model is easily adopted among online usersand it can be easilydeployed. It can be usedand accessed by everyone and it can handle the requests from the passengers.

4.REQUIREMENT ANALAYSIS

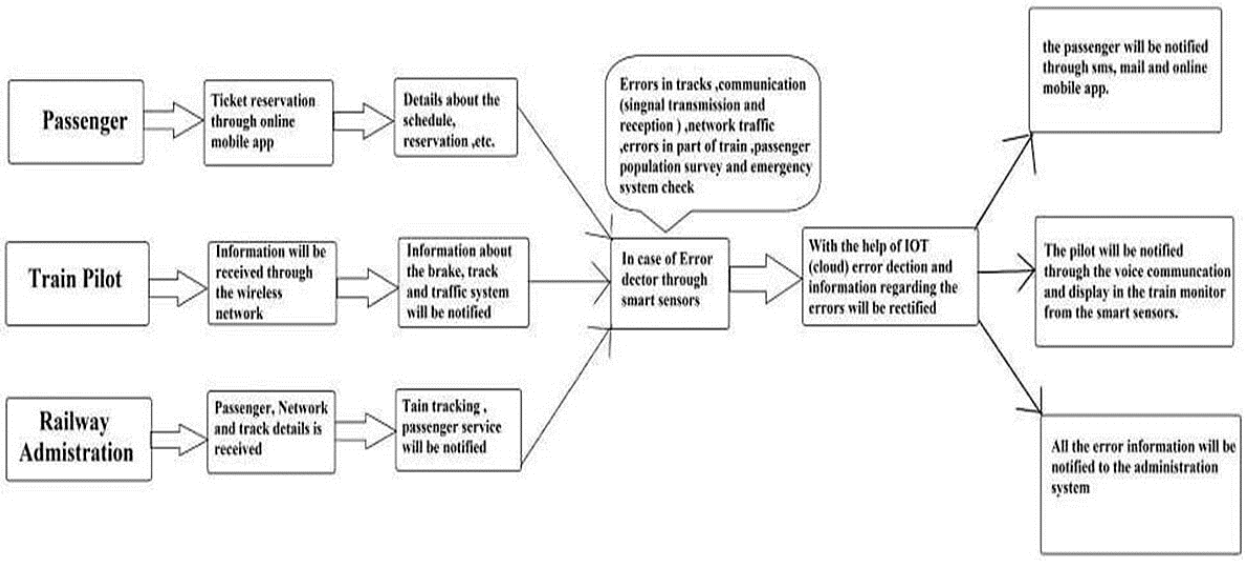
4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The app can be usedduring thetravelling time Easy and simple Efficiency is high
NFR-2	Security	By clicking on the icon, thealert will be given to the respective officials
NFR-3	Reliability	Highly reliable to use
NFR-4	Performance	Low error rate
NFR-5	Availability	Free source

NFR-6	Scalability	It is scalable enoughto support many users at the sametime

5. PROJECT DESIGN

a. Data Flow Diagrams



a. Solution Architecture

As trains are one of the most preferred modes of transportation amongmiddle 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 allsuch brutality we came up with a solution by providing an application which can beaccessed by the user after bookingtheir tickets. With a singleclick 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 cloudplatform to storepassenger data.

5.3 UserStories

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 forthe tickets by entering my email, and password, and confirming my password.	I can access my account/dash board	High	Sprint-1

		USN-2	As a user, I will receive a confirmation email once I have registered for the tickets.	I can receive a confirmation email &click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through the Railway	I can register & access the dashboard with a registration	Low	Sprint-1

			application.	login.		

		USN-4	As a user, I canregister for the application through Online websites		Medium	Sprint-2
	Login	USN-5	As a user, I can log into the application by entering my email & password		High	Sprint-1

Tra in pil ot	Dashboa rd	USN-6	To get information regarding the trainsystem, users check the system's status through mobile applications or the dashboard display.	I can access it throught hemobile app.		Sprint -1
		USN-7	While traveling the status of the trackwil display in the dashboard.		Medi um	Sprint -2
		USN-8	other information from the admin will be displayed with an alertin the dashboard		High	Sprint -2

			display			
Administrator		USN-9	The Railway network can be monitored fromthe basestationof the railway	Access through the wireless networkand computer system	High	Sprint-1
		USN-10	In the computer system, the railway network traffic can be analysed and easy paths can be chosen.		High	Sprint-1

		USN-11	In case of a communication signal error or problem, it willbedisplayed on the monitor so that the data canbe sent again.		High	Sprint-1

		USN-12	The error in the tracks will be informed to the train pilot'sadmin and received through the mobile app or computer system.	Can be accessed throught hedisplay system ie computer system in the train	High	Sprint-1

		USN-13	The passenger details will be automatically saved on the database of the admin computer system.		Medium	Sprint-1
CustomerCare Executive		USN-14	A portal is been arranged for the passenger help. the passenger can directly makea call to the respective numberand ask for help	Can be accessed through telephony itself	High	Sprint-1
		USN-15	Passengers can textthe respective number through the mobile app.		Medium	Sprint-2

Customer(Web User)	Passenger objection and feedback	USN-16	Passenger call togive their feedback to the railway website.		High	Sprint - 2
		USN-17	In case of any software error from therailway side, it can be reported to the inquiry desk through mailor message.	Accessed through mail or SMS	High	Sprint - 2

1. PROJECT PLANNING & SCHEDULING

a. **Sprint Planning& Estimation**

STEP 1	Identify the problem
STEP 2	Prepare an abstract, problem statement
STEP 3	List required objects needed
STEP 4	Create a code andrun it
STEP 6	Make a prototype
STEP 7	Test with the created code and check the designedprototypeis working
STEP 8	Solution for the problemis found

a. **Re**

po

rts

fr

o

m

JI

RA

SP

RI

NT

1

```
#include <LiquidCrystal.h>

LiquidCrystal 1cd(5,6,8,9,10,11); int red1ed = 2; int
green1ed = 3;int buzzer = 4; int sensor = A0;
int
sensorThresh =
400;void
setup()
{

pinMode(red1ed, OUTPUT); pinMode(green1ed,OUTPUT);
pinMode(buzzer,OUTPUT); pinMode(sensor,INPUT); serial.begin(9600);
```

```
1cd.begin(16,2);
}
Void loop()
{
 int analogValue = analogRead(sensor); Serial.print(analogvalue);
if(analogValue>sensorThresh)
    {
      digitalWrite(red1ed,HIGH); digit1Weite(green1ed,LOW);
tone(buzzer,1000,10000);
      1cd.clear(
      );
      1cd.setCu
      rsor(0,1);
      1cd.print("RAILWAYS");
      delay(1000);1cd.clear();
      1cd.setCursor(0,1);
      1cd.print("SMARTSOLUTION"); delay(1000);
   }
```

```
    else
   {
        digitalWrite(greenlad,HIGH);
        digitalWrite(red1ed,LOW); noTone(buzzer);
        1cd.clear(); 1cd.setCursor(0,0);
        1cd.print("SAFE"); delay(1000);
          1cd.cle
          ar();
          1cd.set
          Cursor(
          0,1);
        1cd.print("ALL CLEAR"); delay(1000);
      }
 }
```

SPRINT 2

Main Program:

```
importwiotp.s
dk.device
importtime
```

```
importrandom
myConfig={
"iden
tity":{
"orgI
d":"g
agte
y",
"typeId":"GPS","d
eviceId":"12345"
},
"auth":{
"token":"12345
678"
}
}
defmyCommandcallback(cmd):
print("messagereceivedfromIBMIOTPlatform:%s"%cmd.data['command'])
m=cmd.data['command']
client=wiotp.sdk.device.deviceclient(config=myConfig,logHandlers=Non
e) client.connect()
defpub(data):
client.publishEvent(eventId="status",msgFormat="json",data=mydata,qos
=0, print("publishedatasuccessfully:%s",mydata)
whileTrue:
mydata={'name':'Train1','lat':17.6387448,'lon':78.47
54336) pub(myData)
```

```
time.sleep(3)
#mydata={'name':'Train2','lat':17.6387448,'lon':78.4
754336) #pub(myData)

#time.sleep(3)
mydata={'name':'Train1','lat':17.6341908,'lon':78.47
44722) pub(myData)
time.sleep(3)
mydata={'name':'Train1','lat':17.6340889,'lon':78.47
45052) pub(myData)
time.sleep(3)
mydata={'name':'Train1','lat':17.6248626,'lon':78.47
20259) pub(myData)
time.sleep(3)
mydata={'name':'Train1','lat':17.6188577,'lon':78.46
98726) pub(myData)
time.sleep(3)
mydata={'name':'Train1','lat':17.6132382,'lon':78.47
07318) pub(myData)
time.sleep(3)
client.commandCallback=mycommanC
allbak client.disconnect()
```

Code:

```
 importcv2
importnumpyas
np importtime
```

```
importpyzbar.pyzbaraspuzbar
fromibmcloudant.cloudant_v1importcloudantv1
fromibmcloudantimportcouchDbsessionAuthenticator
fromibm_cloud_sdk_core.AuthenticatorsimportBasicAu
htenticator authenticator=BasicAuthenticator('apikey-
v2-
16u3crmdpkghhxefdikvpssoh5fwezrmuup5fv5g3ubz','b0ab119f45d3e6255ea
bb978) service=cloudantv1(authenticator=authenticator)

service.set_service_url('https:/ apikey-v2-
16u3crmdpkghhxefdikvpssoh5fwezrmuup5fv5g3ubz:b0ab119f45d3e6255ea
bb978 cap=cv2.videoCapture(0)
font=cv2.FONT_HERSHEY_PLAIN
whileTrue:
_,frame=cap.read(0)
decodeObjects=pyzbar.decode(frame)
forobjindecodeObjects:
#print("Data",obj
.data)a=obj.dat
a.decode('UTF-
8')
cv2.putText(frame,"Ticket",(50,50),font,2,(255,
0,0),3)#print(a)
try:
responce=service.get_document(db='booking',doc_id=a
).get_result() print(response)
time.sleep(5)
```

```
exceptExceptio
nase:
print("NotvalidTicket"
)time.sleep(5)
cap.imshow("Frame",
frame)
ifcv2.waitKey{1}&0X
FF==ord('q'):
br
e
ak
ca
p.r
el
ea
se
()
cv2.destroyAllW
indows()
client.disconne
ct()
```

SPRINT 3

1. This project presents its first ever digital event dedicated to rail transport, the“Smart MobilityExperience” which will take placeon March 24th. This event

will be the occasion for clients and partners of the rail ecosystem, to discovernew products and major innovations, as well as to exchange about the digitalization and future of rail.

2. for improved service performance and energy efficiency, and to boosttheattractiveness for users.
3. It helps transporting passengers safely, and with best possibleexperience,supervises operations with accurate situation awareness, and optimizes transportservice efficiency.
4. Using digital technologies such as IoT, cloud and web IT, data analytics, it designs innovative solutions such as digitalsignalling, train autonomy, mobile ticketing, passenger flow analytics, data driven operation control, smart maintenance, which will drastically impact the way we all travel.
5. Provide real-time passenger densityinsights to publictransport operators
6. The solution helps alleviate crowding by reducing busy times, and consequently enhances overall passenger safety, comfort, and travelexperience.
7. The targeted performances of density accuracyare above 90%.

In Hand'sConnectivity Solution for Rail

Transit:MAIN:

importwiotp.s

```
dk.device
importtime
importrandom
myConfig={
"iden
tity":{
"orgI
d":"g
agte
y",

"typeId":"GPS","d
eviceId":"12345"
},
"auth":{
"token":"12345
678"
}
}
defmyCommandcallback(cmd):
print("messagereceivedfromIBMIOTPlatform:%s"%cmd.data['command'])
m=cmd.data['command']
client=wiotp.sdk.device.deviceclient(config=myConfig,logHandlers=Non
e) client.connect()
defpub(data):
client.publishEvent(eventId="status",msgFormat="json",data=mydata,qos
=0, print("publishedatasuccessfully:%s",mydata)
```

```
whileTrue:
mydata={'name':'Train1','lat':17.6387448,'lon':78.47
54336) pub(myData)
time.sleep(3)
#mydata={'name':'Train2','lat':17.6387448,'lon':78.4
754336) #pub(myData)
#time.sleep(3)
mydata={'name':'Train1','lat':17.6341908,'lon':78.47
44722) pub(myData)
time.sleep(3)
mydata={'name':'Train1','lat':17.6340889,'lon':78.47
45052) pub(myData)
time.sleep(3)
mydata={'name':'Train1','lat':17.6248626,'lon':78.47
20259) pub(myData)

time.sleep(3)
mydata={'name':'Train1','lat':17.6188577,'lon':78.46
98726) pub(myData)
time.sleep(3)
mydata={'name':'Train1','lat':17.6132382,'lon':78.47
07318) pub(myData)
time.sleep(3)
client.commandCallback=mycommanC
allbak client.disconnect()
```

PROGRAM:

```
importcv2
importnumpyas
np importtime
importpyzbar.pyzbaraspuzbar
fromibmcloudant.cloudant_v1importcloudantv1
fromibmcloudantimportcouchDbsessionAuthenticator
fromibm_cloud_sdk_core.AuthenticatorsimportBasicAu
htenticator authenticator=BasicAuthenticator('apikey-
v2-
16u3crmdpkghhxefdikvpssoh5fwezrmuup5fv5g3ubz','b0ab119f45d3e6255ea
bb978) service=cloudantv1(authenticator=authenticator)
service.set_service_url('https:/ apikey-v2-
16u3crmdpkghhxefdikvpssoh5fwezrmuup5fv5g3ubz:b0ab119f45d3e6255ea
bb978 cap=cv2.videoCapture(0)
font=cv2.FONT_HERSHEY_PLAIN
whileTrue:
_,frame=cap.read(0)
decodeObjects=pyzbar.decode(frame)

forobjindecodeObjects:
#print("Data",obj
.data)a=obj.dat
a.decode('UTF-
8')
cv2.putText(frame,"Ticket",(50,50),font,2,(255,
```

0,0),3)#print(a)

try:

responce=service.get_document(db='booking',doc_id=a

).get_result() print(response)

time.sleep(5)

exceptExceptionase:

print("NotvalidTicket

") time.sleep(5)

cap.imshow("Frame",

frame)

ifcv2.waitKey{1}&0X

FF==ord('q'): break

cap.release()

cv2.destroyAllW

indows()

client.disconne

ct()

SPRINT 4

Main:

importwiotp.s

dk.device

importtime

importrandom

myConfig={

"iden

tity":{

```
"orgI
d":"g
agte
y",
"typeId":"GPS","d
eviceId":"12345"
},

"auth":{
"token":"12345
678"
}
}
defmyCommandcallback(cmd):
print("messagereceivedfromIBMIOTPlatform:%s"%cmd.data['command'])
m=cmd.data['command']
client=wiotp.sdk.device.deviceclient(config=myConfig,logHandlers=Non
e) client.connect()
defpub(data):
client.publishEvent(eventId="status",msgFormat="json",data=mydata,qos
=0, print("publishedatasuccessfully:%s",mydata)
whileTrue:
mydata={'name':'Train1','lat':17.6387448,'lon':78.47
54336) pub(myData)
time.sleep(3)
#mydata={'name':'Train2','lat':17.6387448,'lon':78.4
754336) #pub(myData)
```

```
#time.sleep(3)
mydata={'name':'Train1','lat':17.6341908,'lon':78.47
44722) pub(myData)
time.sleep(3)
mydata={'name':'Train1','lat':17.6340889,'lon':78.47
45052) pub(myData)
time.sleep(3)
mydata={'name':'Train1','lat':17.6248626,'lon':78.47
20259) pub(myData)
time.sleep(3)
mydata={'name':'Train1','lat':17.6188577,'lon':78.46
98726) pub(myData)

time.sleep(3)
mydata={'name':'Train1','lat':17.6132382,'lon':78.47
07318) pub(myData)
time.sleep(3)
client.commandCallback=mycommanC
allbak client.disconnect()
```

Program:

```
importcv2
importnumpyas
np importtime
importpyzbar.pyzbaraspuzbar
fromibmcloudant.cloudant_v1importcloudantv1
```

```
fromibmcloudantimportcouchDbsessionAuthenticator
fromibm_cloud_sdk_core.AuthenticatorsimportBasicAu
htenticator authenticator=BasicAuthenticator('apikey-
v2-
16u3crmdpkghhxefdikvpssoh5fwezrmuup5fv5g3ubz','b0ab119f45d3e6255ea
bb978) service=cloudantv1(authenticator=authenticator)
service.set_service_url('https:/ apikey-v2-
16u3crmdpkghhxefdikvpssoh5fwezrmuup5fv5g3ubz:b0ab119f45d3e6255ea
bb978 cap=cv2.videoCapture(0)
font=cv2.FONT_HERSHEY_PLAIN
whileTrue:
_,frame=cap.read(0)
decodeObjects=pyzbar.dec
ode(frame)
forobjindecodeObjects:
#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:
responce=service.get_document(db='booking',doc_id=a
).get_result() print(response)
time.sleep(5)
exceptExceptionase:
print("NotvalidTicket
") time.sleep(5)
```

```
cap.imshow("Frame",
frame)
ifcv2.waitKey{1}&0X
FF==ord('q'): break
cap.release()
cv2.destroyAllW
indows()
client.disconne
ct()
```

7. CODING & SOLUTIONING

Feature 1

1. IoT device
2. IBM Watson Platform
3. Node red
4. Cloudant DB
5. Web UI
6. MIT App Inventor
7. Python code

Feature 2

1. Login
2. Verification
3. Ticket Booking

4. Adding rating

8.TESTING AND RESULTS

TestCase 1:

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
1	Functional	Registration	Registration through the form by Filling in my details	1.Click on register 2.Fill the registration form 3.click Register		Registration form to be filled is to be displayed	Working as expected	PASS	VAISHNAVI
2	UI	Generating OTP	Generating the otp for further process	1.Generating of OTP number		user can register through phone numbers and to get otp number	Working as expected	PASS	MRITHULLA
3	Functional	OTP verification	Verify user otp using mail	1.Enter gmail id and enter password 2.click submit	Username: railways password: admin	OTP verifed is to be displayed	Working as expected	FAIL	JESLENE
4	Functional	Login page	Verify user is able to log into application with InValid credentials	1.Enter into log in page 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box	Username: railways password: admin	Application should show 'Incorrect email or password ' validation message.	Working as expected	FAIL	ABINAYA
5	Functional	Display Train details	The user can view about the available train details	1.As a user, I can enter the start and destination to get the list of trains available connecting the above	Username: railways password: admin	A user can view about the available trains to enter start and destination details	Working as expected	PASS	VAISHNAVI

TestCase2:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Expected Result	Actual Result	Status	Executed By
1	Functional	Booking	user can provide the basic details such as a name, number, etc		1. Enter the member's details like name, number.	Tickets booked to be displayed	Working as expected	Pass	Abinaya
2	UI	Booking seats	User can choose the train, starting and ending destination, date of travel.		1. Known to which train is available	known to which the seats are available	Working as expected	fail	Jeslene
3	Functional	Payment	user, I can choose to pay through credit Card/debit card/UPI.		1.user can choose payment method 2.payment method	payment for the booked tickets to be done using payment method through either the following methods credit Card/debit	Working as expected	Fail	Mrithulla
4	Functional	Redirection	user can be redirected to the selected		1.After payment the user will be redirected to the previous page	After payment the user will be redirected to the previous page	Working as expected	pass	Vaishnavi

Test Case3:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Expected Result	Actual Result	Status	Executed By
1	Functional	Ticket generation	a user can download the generated e ticket for my journey along with the QR code which is used for authentication during my journey.		1.Enter method of reservation 2.Enter name,age,gender 3.Enter how many tickets wants to be booked 4.Also enter the number member's details like name,age,gender	Tickets booked to be displayed	Working as expected	Pass	Abinaya
2	UI	Ticket status	a usercan see the status of my ticket Whether it's confirmed/waiting/RAC		1.known to the status of the tickets booked	known to the status of the tivkets booked	Working as expected	Fail	Mrithulla
3	Functional	Reporting issues	user can access the reporting portal once the jouney begins		1. reporting	Issues have been reported	Working as expected	pass	Vaishnavi

TestCase 4

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Expected Result	Actual Result	Status	Executed By
1	Functional	Ticket cancellatio	user can cancel my tickets there's any Change of plan		1.tickets to be cancelled	Tickets booked to be cancelled	Working as expected	Fail	Jeslene
2	Functional	Rate	a user will feed rating about the train journey		1.information feeding on trains	information feeding on trains	Working as expected	pass	Vaishnavi

1. **ADVANTAGES**

1. The passengers can use this application, while they are travellingalone to ensure their safety.
2. It is easy to use.
3. It has minimized errorrate.

DISADVANTAGES

1. Network issuesmay arise.

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 industrystruggles to meet the increasing demand for freightand passenger transportation due to lack of optimizeduse 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 smarterand 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 passengerdata, 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 intelligentprocessing and timely retrieval would help backingup the security breaches. Here

we've explored different issues of implementing smart computing in railway systems pertaining to reservation modelsbesides pointing out some futurescopes 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 purchasesand timetable information implemented at stations and stops. With the rise of Industry, railway companies can now ensure that they are preparedto avoid the surprise of equipment downtime. Like above mentioned, the developed application of our project can lead the passenger who travel can travel safelywithout any fear.

FUTURE SCOPE :

This application is ensured for safety for the passengers while they are travellingalone as well as they travel with their familyor 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 featuresregarding their safety.

1. APPENDIX

Source : codelogin

a.

```
 from
tkinter
import*
import
sqlite3


    root = Tk()
    root.title("Python: Simple Login Application") width = 400 height =
    280 screen_width = root.winfo_screenwidth() screen_height =
    root.winfo_screenheight() x = (screen_width/2) - (width/2) y =
    (screen_height/2) - (height/2)
    root.geometry("%dx%d+%d+%d" % (width,
  height,x, y))root.resizable(0, 0)


    #==============================VARIABLES==================
  =======
    =============
    USERNAME
    =
    StringVar()
    PASSWORD
```

```
=
StringVar()

#==============================FRAMES==================
========
=============
Top = Frame(root, bd=2,
relief=RIDGE)
Top.pack(side=TOP,
fill=X)
Form = Frame(root, height=200)
Form.pack(side=TOP, pady=20)

#==============================LABELS==================
=======
=============
  lbl_title = Label(Top, text = "Python:
Simple Login Application",font=('arial', 15))
lbl_title.pack(fill=X)
lbl_username = Label(Form, text = "Username:", font=('arial', 14), bd=15)
lbl_username.grid(row=0, sticky="e")
lbl_password = Label(Form, text = "Password:", font=('arial', 14), bd=15)
lbl_password.grid(row=1, sticky="e") lbl_text= Label(Form)
lbl_text.grid(row=2, columnspan=2)
```

```
  #============================ENTRY

  WIDGETS==================================
  username    =    Entry(Form,
textvariable=USERNAME,                    font=(14))
username.grid(row=0, column=1)
  password = Entry(Form, textvariable=PASSWORD, show="*", font=(14))
password.grid(row=1, column=1)




  #============================METHODS==================
========
  =============
     def Database():
     global conn,
     cursor
                              co
nn
=
sqlite3.connect("python
tut.db") cursor=
conn.cursor()
        cursor.execute("CREATE TABLE IF NOT EXISTS
`member`(mem_id INTEGERNOT NULL PRIMARYKEY
```

```
AUTOINCREMENT, usernameTEXT, password TEXT)")
cursor.execute("SELECT * FROM `member` WHERE `username` = 'admin' AND
`passwor
        d` =
'admin'") if
cursor.fe
tchone()
isNone:
                    cursor.execute("INSERT   INTO
                    `member`                    (username,
                    password)
VALUES('admin',
  'admin')")            conn.commit()  def  Login(event=None):
Database()                      ifUSERNAME.get() == "" or
PASSWORD.get() == "":
              lbl_text.config(text="Please
completethe requiredfield!", fg="red")
else:

          cursor.execute("SELECT * FROM `member`WHERE `username` = ?
AND `password`
  =?", (USERNAME.get(), PASSWORD.get()))        if cursor.fetchone()
           is not None:HomeWindow()
           USERNAME.set("")
PASSWORD.set("")
lbl_text.config(text="")        else:
           lbl_text.config(text="Invalid  usernameor
```

password", fg="red") USERNAME.set("")

PASSWORD.set("")

c

u

r

s

o

r.

c

l

o

s

e

(

)

c

o

n

n

.

c

l

o

s

e

(

)

```
#==============================BUTTON
WIDGETS==============================
==
btn_login = Button(Form, text="Login", width=45,
command=Login) btn_login.grid(pady=25, row=3,columnspan=2)
btn_login.bind('<Return>', Login)



def HomeWindow(): global Home root.withdraw()
Home = Toplevel()
    Home.title("Python: Simple Login Application")
width = 600                                    height = 500
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()  x = (screen_width/2) -
(width/2) y = (screen_height/2) - (height/2)
    root.resizable(0,0)
    Home.geometry("%dx%d+%d+%d" % (width, height,x, y))
    lbl_home = Label(Home, text="Successfully Login!", font=('times new
  roman',20)).pack()

   btn_back = Button(Home, text='Back',
   command=Back).pack(pady=20, fill=X)


          def Back(): Home.destroy()                root.deiconify()
```

REGISTRATION :

```
from tkinter import* base = Tk()
base.geometry("500x500") base.title("registration form")

labl_0 = Label(base, text="Registration
form",width=20,font=("bold",20)) labl_0.place(x=90,y=53)

lb1= Label(base, text="Enter Name",
width=10, font=("arial",12)) lb1.place(x=20, y=120)en1=
Entry(base)
en1.place(x=200, y=120)

lb3= Label(base, text="Enter Email", width=10, font=("arial",12))
lb3.place(x=19, y=160) en3= Entry(base)
en3.place(x=200, y=160)

lb4= Label(base, text="Contact Number",
width=13,font=("arial",12)) lb4.place(x=19, y=200)en4= Entry(base)
en4.place(x=200, y=200)

lb5= Label(base, text="Select Gender", width=15, font=("arial",12))
lb5.place(x=5,y=240) var = IntVar()
Radiobutton(base,
text="Male", padx=5,variable=var,
value=1).place(x=180, y=240)
```

```
Radiobutton(base, text="Female", padx
=10,variable=var,value=2).place(x=240,y=240)
Radiobutton(base, text="others", padx=15,

variable=var, value=3).place(x=310,y=240)


list_of_cntry = ("United States", "India", "Nepal", "Germany") cv =
StringVar()drplist= OptionMenu(base, cv, *list_of_cntry)
drplist.config(width=15) cv.set("United States")
lb2= Label(base, text="Select Country",
width=13,font=("arial",12)) lb2.place(x=14,y=280)
drplist.place(x=200, y=275)


lb6= Label(base, text="Enter Password",
width=13,font=("arial",12))
lb6.place(x=19, y=320) en6= Entry(base,
show='*') en6.place(x=200, y=320)


lb7= Label(base, text="Re-Enter Password",
width=15,font=("arial",12)) lb7.place(x=21, y=360)en7 =Entry(base,
show='*')en7.place(x=200, y=360)


Button(base, text="Register",
width=10).place(x=200,y=400)base.mainloop()
```

START AND DESTINATION :

```
# import
moduleimport
requestsfrom bs4
import
BeautifulSoup


# user define function # Scrape the data def getdata(url):
r = requests.get(url)returnr.text



# input by geek from_Station_code = "GAYA"
from_Station_name = "GAYA"
To_station_co
de =
"PNBE"To_stati
on_name =
"PATNA"# url
 url                    = "https:/ www.railyatri.in/booking/trains-
between-
stations?from_code="+from_Station_code+"&from_name="+from_
Station_name+
"+JN+&j ourney_date=+Wed&src=tbs&to_code=" + \
    To_station_code+"&to_name="+To_stat
    ion_name + \ "+JN+&user_id=-
1603228437&user_token=355740&utm_source=dwebsearch_tbs_search_
trains"
```

```
# pass the url # into getdatafunction htmldata =
getdata(url)soup = BeautifulSoup(htmldata,
'html.parser')

# find the Html tag
# with find() # and convert into string data_str = "" for item in soup.find_all("div", class_="col-xs-12 TrainSearchSection"): data_str = data_str + item.get_text() result = data_str.split("\n")

print("Train between "+from_Station_name+" and "+To_station_name)
print("")

# Display the result for item in result:    if item != "":       print(item)
```

TICKET BOOKING:

```
print("\n\nTicket
BookingSystem\n")
restart = ('Y')
while restart != ('N','NO','n','no'): print("1.Check PNR status")
print("2.TicketReservation")
option = int(input("\nEnter your option : "))

if option == 1: print("Your PNR
status is t3")exit(0)
elif option == 2: people = int(input("\nEnter no. of Ticket you
want : ")) name_l = [] age_l = [] sex_l = [] for p in
```

```
range(people): name = str(input("\nName : "))
name_l.append(name) age = int(input("\nAge : "))
age_l.append(age)
sex = str(input("\nMale or Female : "))
sex_l.append(sex)
restart = str(input("\nDid you forgot someone? y/n: ")) if restart in
('y','YES','yes','Yes'): restart = ('Y') else : x = 0
print("\nTotal Ticket : ",people) forp in range(1,people+1):
print("Ticket : ",p) print("Name : ", name_l[x])
print("Age
: ", age_l[x]) print("Sex : ",sex_l[x]) x += 1
```

SEATS BOOKING:

```
berth_type(s):


    if s>0and s<73: ifs % 8 == 1 or s % 8 == 4: print (s),
"is lowerberth" elif s % 8 == 2 or s % 8 == 5:
print (s), "is middleberth"
elif s % 8 == 3 or s % 8 == 6: print (s), "is upper berth"
elif s % 8 == 7:print (s), "is side lower berth" else:
print (s), "isside upper
berth"
else:print (s), "invalid
seat number"


# Drivercode s = 10
berth_type(s) # fxn call for berth type
```

```
s = 7
berth_type(s)     # fxn call for berth type

s = 0
berth_type(s)      # fxn call for berth type
```

CONFIRMATION:

```
  # import moduleimport requests from bs4 importBeautifulSoup
importpandasas pd


  # user define function # Scrape the data def
getdata(url): r =requests.get(url)
  return r.text


  # input by geek
  train_name  =  "03391-rajgir-new-delhi-clone-special-rgd-to-ndls"


  # url
  url = "https:/ www.railyatri.in/live-train-status/"+train_name


  # pass the url # into getdatafunction htmldata =
  getdata(url)soup = BeautifulSoup(htmldata,
  'html.parser')
```

```
  # traverse the live status from # this Html code data = [] for item in
soup.find_all('script', type="application/ld+json"):
  data.append(item.get_text())


  # convert
  into
  dataframe df
  =
  pd.read_json
  (data[2])


  # display this column of
  #
  dataframeprint(df["main
  Entity"][0]['name'])

  print(df["mainEntity"][0]['acceptedAnswer']['text'])
```

TICKET GENERATION:

```
   class Ticket:      counter=0
  def__init_
(self,passenger_name,source,destination):
self.passenger_name=passenger_name
  self._____source=source              self._
destination=destination
self.Counter=Ticket.counter
Ticket.counter+=1                          def
```

```
validate_source_destination(self):
  if (self._source=="Delhi" and (self.__
destination=="Pune" or self.destination=="Mumbai"
or self.destination=="Chennai" or
self.destination=="Kolkata")):              return True
else:
                                ret
urn Falsedef
generate_ticket(se
lf ):
 if True:
__ticket_id=self._source[0]+self._
destination[0]+"0"+str(self. Counter)              print(
"Ticket id will be:",__ticket_id)
else: returnFalse                  def get_ticket_id(self):
return self.ticket_id    def
get_passenger_name(self):
return self.__passenger_name        defget_source(self):
if self.source=="Delhi":
return
self._
source
else:
print("you have written invalid soure option")
return None                                                                    def
get_destination(self):         if self.__destination=="Pune":
returnself.__destination                    elif self._
```

```
destination=="Mumbai":
return self.__destination
elif self.destination=="Chennai":return self.__
destination                                    elif self._
destination=="Kolkata":returnself._destination
else:

        return  None
```

OTP GENERATION:

```
import os importmath import
randomimport smtplib


digits=
"012345678
9"OTP = ""


for i in range (6):
    OTP += digits[math.floor(random.random()*10)]


otp  =  OTP  +  "is  your  OTP"
message    =    otp    s    =
smtplib.SMTP('smtp.gmail.c
om', 587)s.starttls()


emailid = input("Enter your email: ")
s.login("YOURGmail ID", "YOUR APP PASSWORD")
```

```
s.sendmail('&&&&&&',emailid,message)


a = input("Enter your OTP >>:") if a ==
    OTP:print("Verified") else:
    print("Please Check your OTP again")
```

OTP VERIFICATION:

```
import os importmath import
randomimport smtplib

digits  =
  "01234567
  89"OTP =
  ""
  for i in range (6):
     OTP +=
     digits[math.floor(random.random()*
     10)] otp = OTP + " is your
     OTP"message = otp
  s  =
    smtplib.SMTP('smtp.gmail.com',
    587)s.starttls()


    emailid = input("Enter your email: ")
    s.login("YOURGmail ID", "YOUR APP PASSWORD")
    s.sendmail('&&&&&&',emailid,message)
```

```
a = input("Enter your OTP >>:") if a == OTP:
    print("Verified") else:
    print("Please Check your OTP again")
```

GitHub:

GitHub link: https://github.com/IBM-EPBL/IBM-Project-9633-1659031004

Demo Video Link

https://drive.google.com/file/d/1ZRBCOliSVHUWebJwxZP6Op7rv8aTeEtv/view?usp=share_link