

# **KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION  
ENGINEERING**

**HX 8001-PROFESSIONAL READINESS FOR INNOVATION,  
EMPLOYABILITY AND ENTREPRENEURSHIP**

## **SMART SOLUTION FOR RAILWAYS USING IOT**

**NALAIYA THIRAN PROJECT REPORT 2022**

*Submitted by*

<b>KAVIN P</b>	<b>621319106038</b>
<b>GOKULNATH S</b>	<b>621319106021</b>
<b>JAWAHAR M</b>	<b>621319106031</b>
<b>SUNDARAMOORTHY M</b>	<b>621319106316</b>

**Team ID: PNT2022TMID13493**

**NOVEMBER 2022**

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
1.	<b>INTRODUCTION</b>	4
	1.1 Project Overview	4
	1.2 Purpose	5
2.	<b>LITERATURE SURVEY</b>	6
	2.1 Existing problem	7
	2.2 References	8
	2.3 Problem Statement Definition	9
3.	<b>IDEATION &amp; PROPOSED SOLUTION</b>	10
	3.1 Empathy Map Canvas	10
	3.2 Ideation & Brainstorming	11
	3.3 Proposed Solution	14
	3.4 Problem Solution fit	15
4.	<b>REQUIREMENT ANALYSIS</b>	16
	4.1 Functional requirement	16
	4.2 Non-Functional requirements	17
5.	<b>PROJECT DESIGN</b>	18
	5.1 Data Flow Diagrams	18
	5.2 Solution & Technical Architecture	19
	5.3 User Stories	20

<b>6.</b>	<b>PROJECT PLANNING &amp; SCHEDULING</b>	<b>21</b>
6.1	Sprint Planning & Estimation	21
6.2	Sprint Delivery Schedule	22
6.3	Reports from JIRA	23
<b>7.</b>	<b>CODING &amp; SOLUTIONING</b>	<b>24</b>
7.1	Feature 1	25
7.2	Feature 2	26
<b>8.</b>	<b>TESTING</b>	<b>27</b>
8.1	Test Cases	27
8.2	User Acceptance Testing	30
<b>9.</b>	<b>RESULTS</b>	<b>31</b>
9.1	Performance Metrics	31
<b>10.</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>	<b>32</b>
<b>11.</b>	<b>CONCLUSION</b>	<b>33</b>
<b>12.</b>	<b>FUTURE SCOPE</b>	<b>33</b>
<b>13.</b>	<b>APPENDIX</b>	<b>34</b>
	Source Code	34
	GitHub & Project Demo Link	41

# **1. INTRODUCTION**

## **1.1 PROJECT OVERVIEW**

This project is used for booking the railway ticket from the web UI with customize features. By booking the ticket from the web UI user will get a special ID and also creates a QR code which contains all the reference detail about the user such as boarding and destination .The QR code is mainly for the checking purpose and this makes the work easier for the ticket checker for checking the originality of the ticket. After booking the ticket the user will get a unique ID and QR code. From the ticket checkers side they may get a special login from the web UI .The ticket checker uses the QR code reader to scan the QR code. By scanning the QR code the Ticket Checker will receive the booking details of the passenger directly from the cloud IOT.

## **1.2 PURPOSE**

The Internet is essential for computer to connect through network. However, as the world changes, its use is expanding beyond just email and web browsing. The creation of smart homes, smart rural communities, and e-health are all products of today's internet, which also deals with embedded sensors. The idea of IOT was introduced by care's etc. Without human-to-human or human-to-computer interaction, the Internet of Things refers to the connection or communication between two or more devices. The sensors or actuators, connected devices is used to sense their surrounding environment. Sensing the device will gain access to the device, processing the device's data, and offering applications and services make up the four main parts of IOT. Along with this, it also offers data security and privacy. All facets of our daily life have been impacted by automation. In order to save time and reduce human effort, more advancements are being made practically in every industry. The same is being considered while attempting to automate track testing. Railroad track is a crucial component of every company's asset base since it enables them to conduct business as usual. Problems brought on by issues with railroads must be solved. The Indian railroad's most recent technique involves following the train tracks, which takes a lot of time and labour.

## 2. LITERATURE SURVEY

TITLE	AUTHOR & YEAR	JOURNAL NAME	REMARKS
Application of smart computing in Indian railway System	Asokh Nath & 2017	International Journal of Scientific Research and Management Studies (IJSRMS)	The smart model approach for passenger reservation system depends on some pre-requisites, without which the benefits would not be fully enjoyed. This includes the comprehensive UID registration of all passengers who needs to travel.
Smart Railway solutions	Ekaterina KOZYREVA & 2021	Indonesia Journals of Innovative and Research in Science	To examine the theoretical relationship between sense of community, perceived value, consumer satisfaction, and future intentions in low-cost fitness clubs.

5G Key Technologies for Smart Railways	Markus Rupp & 2020	Institute of Electrical and Electronics Engineers (IEEE)	This paper explored a potential solution by leveraging emerging 5G technologies to provide a plethora of services in HSRs, both control and data services. More specifically, we first briefly described the current trend of wireless communications for smart railway.
Internet Of Things for Smart Railways	Arghya Biswas & 2019	Institute of Electrical and Electronics Engineers (IEEE)	The IOT is the key enabling solution to the CBM to enhance the efficiency of the maintenance. In some railway area already start to use the GSM-R technology for communication. But they are also faraway from IOT solution.

A Novel Design of Smart Train	Abishek Gupta & 2018	Institute of Electrical and Electronics Engineers (IEEE)	This includes the comprehensive UID registration of all passengers who needs to travel. IoT data in the aspects of power consumption.
Internet Of Things(IOT) and Indian Railways	Rajnish Kumar & 2016	International Journal of Scientific Research & Management Studies	The role of purchase department can be limited just to give the purchase order, the balance work can be handled by intelligent systems. When the network has information on consignments, stock position etc.

## 2.2 REFERENCES

1. 1. Shaofu Lin. “Research and Analysis on the Top Design of Smart Railway” International Journals of Electrical and Computer Engineering(IJECE),2017.
2. Dr. A. Benjamin Joseph. “Smart railway automation system using IOT.” International journal of current engineering and scientific research (IJCESR),2018.
3. Yong-Kyu Kim . “Internet of Things for Smart Railway: Feasibility and Applications.” Institute of Electrical and Electronics Engineers(IEEE),2018.
4. Asokh Nath. “Application of smart computing in Indian railway system.” International Journal of Scientific Research and Management Studies(IJSRMS),2017.
5. Rajnish Kumar. “Internet of Things(IOT) and Indian Railway.” International Journals of Scientific Research and Management Studies,2016.
6. Ekaterina KOZYREVA. “Smart Railway Solutions.” Indonesia Journals of Innovative and Research in Science,2021.
7. Markus Rupp. “5G key technologies for Smart Railways.” Institute of Electrical and Electronics Engineers(IEEE),2020.
8. Arghya Biswas. “ Internet Of Things for Smart Railways.” Institute of Electrical and Electronics Engineers(IEEE),2019.
9. Abishek Gupta. “A Novel design of Smart Train.” Institute of Electrical and Electronics Engineers(IEEE),2018.
10. Marilia Curado “Smart Railway Maintenance – Challenges and Research Directions.” International Journals of Electrical and Computer Engineering(IJECE),2020.



## 2.3 PROBLEM STATEMENT

Information about  
route , cancellation  
 , arrival time ,  
departure time

Store and retrieve  
information about  
the various  
transactions related  
to rail travel

User friendly  
interface to  
administrator and  
customer

Confirmation  
of the  
track

Fewer  
maintenance  
delays

Great  
reliability  
and safety

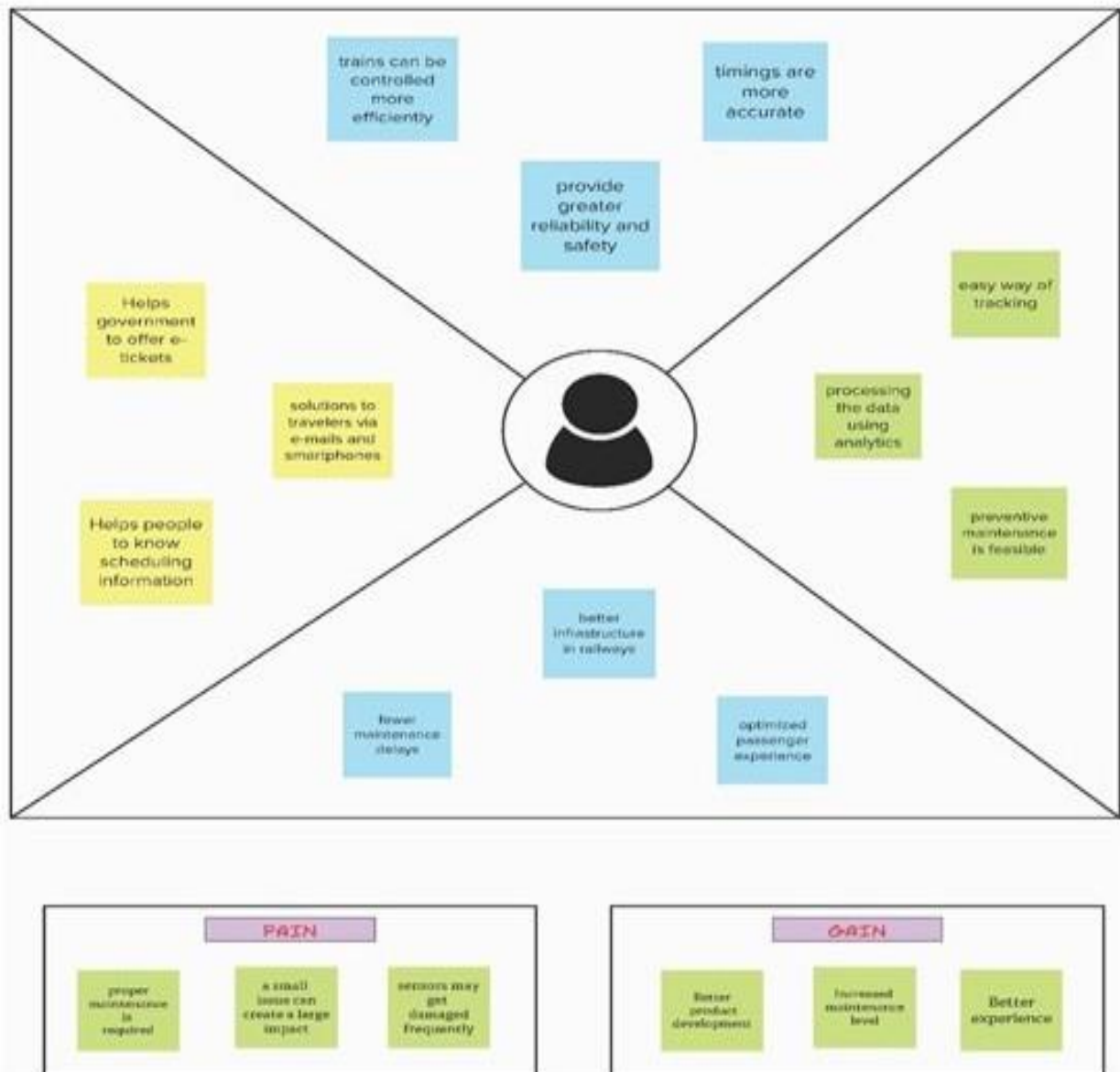
Better  
product  
development  
in the industry

Advanced  
analytics for  
streamlined  
operations

Restricted and  
optimized  
passenger  
experience

### 3.IDEATION AND PROPOSED SOLUTION

#### 3.1 EMPATHY MAP CANVAS



## 3.2 IDEATION AND BRAINSTORMING

### Brainstorm

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

10 minutes

---

#### Kavin P

Track  
notification  
can be find  
earlier

Controls  
are more  
efficient

Any repairs  
are identified  
quickly

Preventive  
maintenance

More  
effective in  
IOT era

#### Gokulnath S

security

Tracking  
are more  
easier

Processing  
data using  
analytics

Better  
operations

#### Jawahar M

Easy  
Accessibility

Offer  
e-tickets

Scheduling  
information

Solution  
via  
smartphones

#### Sundara Moorthi M

Solutions  
via  
e-mails

Good  
infrastructure

Better  
experience

Better  
development



### Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

#### Integration

Can integrate with UPI	integrate multiple bank accounts
Can integrate with crypto hardware	Integrate Any wallets like Paytm, Amazon

#### Experience

Better service for users	monitor transactions	Various themes are available
Easy Accessibility	Reduces booking time	

#### Alerts

Booking notifications are sent to the user
--

#### Categorization

Tickets have been categorized	Price range will be based on category
Well Category the Expenses	Allocate budget based on each

#### Awareness

Don't use third party apps to register	Strictly recommended to use approved sites
Check whether you received notification	All the fields are mandatory

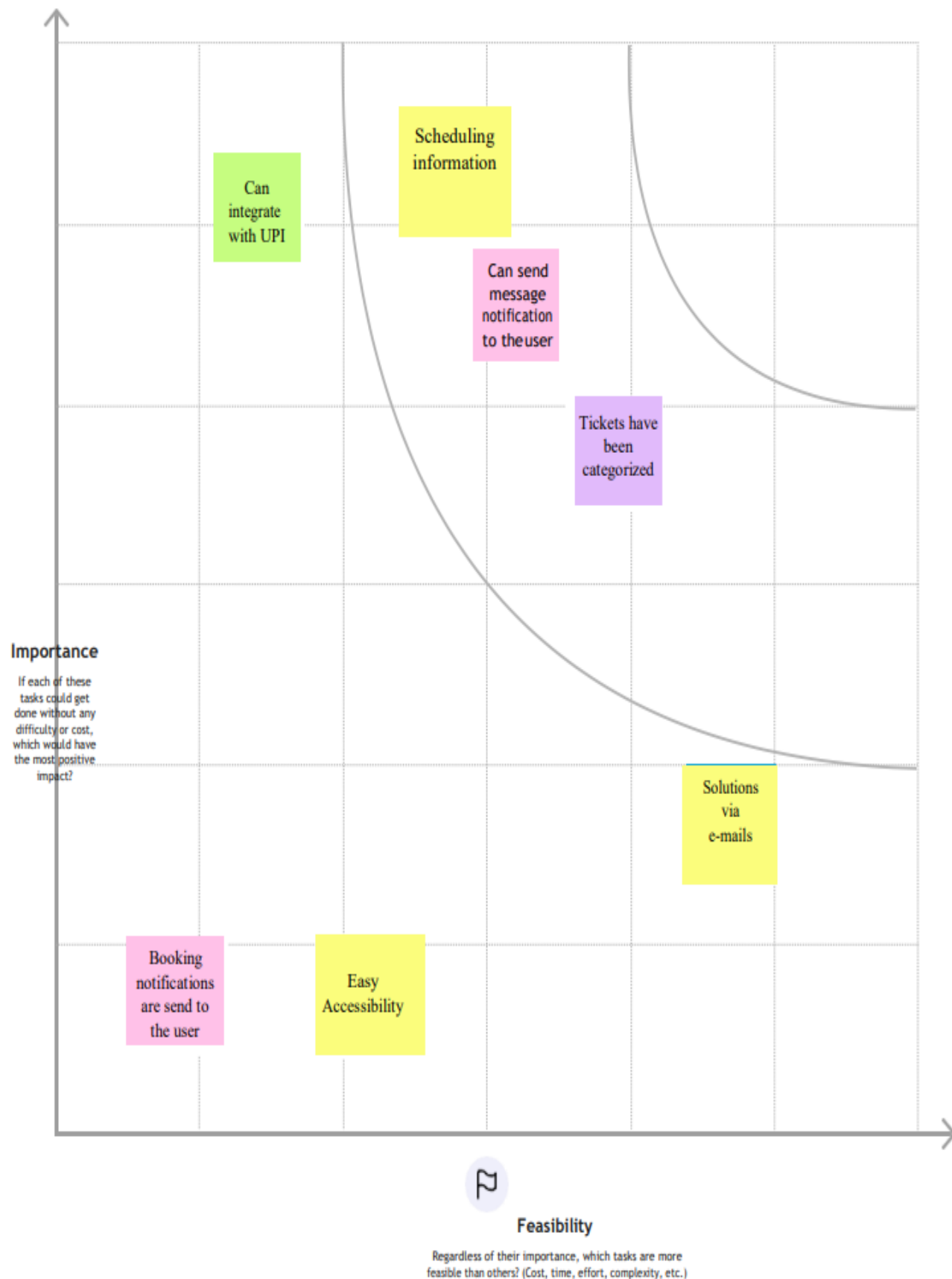
#### Customization

Auto adaptable to locations
-----------------------------

## 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



### 3.3 PROPOSED SOLUTION

#### Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Smart solution for Railways will provide will provide information about tracks, e-tickets and also the arriving time of the train
2.	Idea / Solution description	We are using various sensors and internet connection to send and receive the notifications and alerts immediately to the railway department and people.
3.	Novelty / Uniqueness	The uniqueness of this project is we can easily identify the track information within short period of time with less manpower.
4.	Social Impact / Customer Satisfaction	It will helps people to book their tickets more easier and more quicker and save their time of booking.
5.	Business Model (Revenue Model)	This project requires less manpower and and have a great life and more accuracy in the system.
6.	Scalability of the Solution	This project can withstand for huge years and technology updation can also applicable to it.

### 3.4 PROBLEM SOLUTION FIT

Project Title: Smart Solution for Railways

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID13493

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> <p>People who travel from long distance through trains need to prebook train tickets for their for their travel and know the live status of the journey. This project mainly focus on making passengers more comfort.</p>	<b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span> <p>Network availability and server jamming are the available issues face by the passengers and it may difficult to understand by the fresh users</p>	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> <p>Nowadays, ticket booking are available on online but it doesn't provide any additional information about the train to the passengers, this project help the passenger to get a live update and live status of train they travel.</p>	Explore AS, differentiate	
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span> <p>The passengers had to know the arrival, departure of the train, but in conventional method it contain only prebooking through offline and it is difficult to track the live location of the train.</p>	<b>9. PROBLEM ROOT CAUSE</b> <span>RC</span> <p>The reason for the arrival of this project is to provide the complete information about the train for the passengers. This project may bring a biggest change in railway system and more helpful to passengers.</p>	<b>7. BEHAVIOUR</b> <span>BE</span> <p>Directly: find better network availability and perfect device for getting live update. Indirectly: Passengers may suggest this service to their surroundings.</p>		Focus on J&P, map into BE, understand RC
	<b>3. TRIGGERS</b> <span>TR</span> <p>By installing this project we can trigger people by seeing their neighbouruse the utilization of arrival of newtrending in railways or through advertisement we can trigger people.</p>	<b>10. YOUR SOLUTION</b> <span>SL</span> <p>Through this project we provide a better solution to passengers for their problems.</p>	<b>8. CHANNELS OF BEHAVIOUR</b> <span>CH</span> <p>ONLINE: Passengers may provide suggestions to improve the service. OFFLINE: Passengers may provide or rise funds to develop the service in future.</p>		
<b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> <p>People felt inconvenient during booking of tickets, now they can easily know the information about their travel in sitting place.</p>					

## 4.REQUIREMENT ANALYSIS

### 4.1 FUNCTIONAL REQUIREMENTS

#### 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 Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3		
FR-4		



## 4.2 NON – FUNCTIONAL REQUIREMENTS

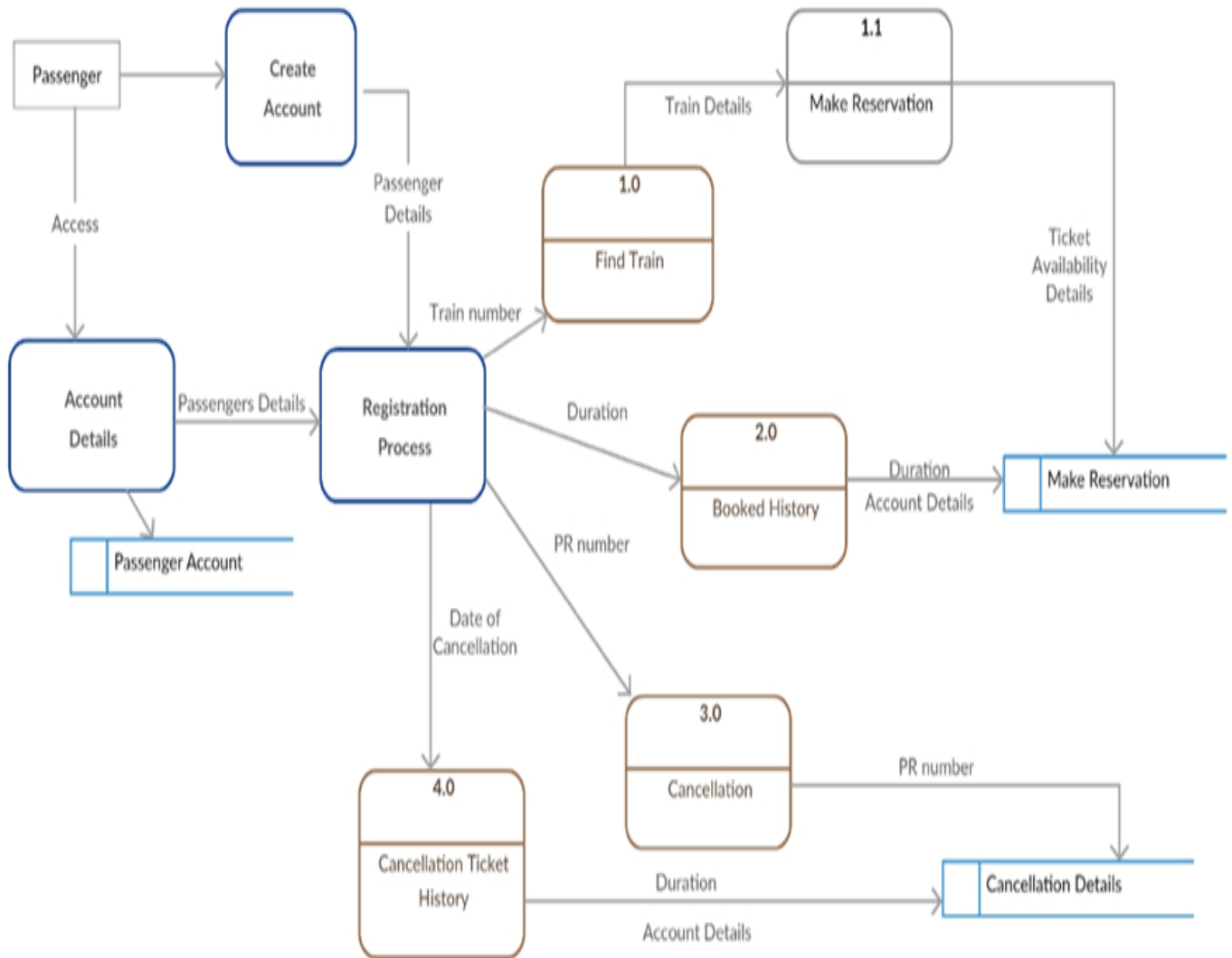
### Non-functional Requirements:

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

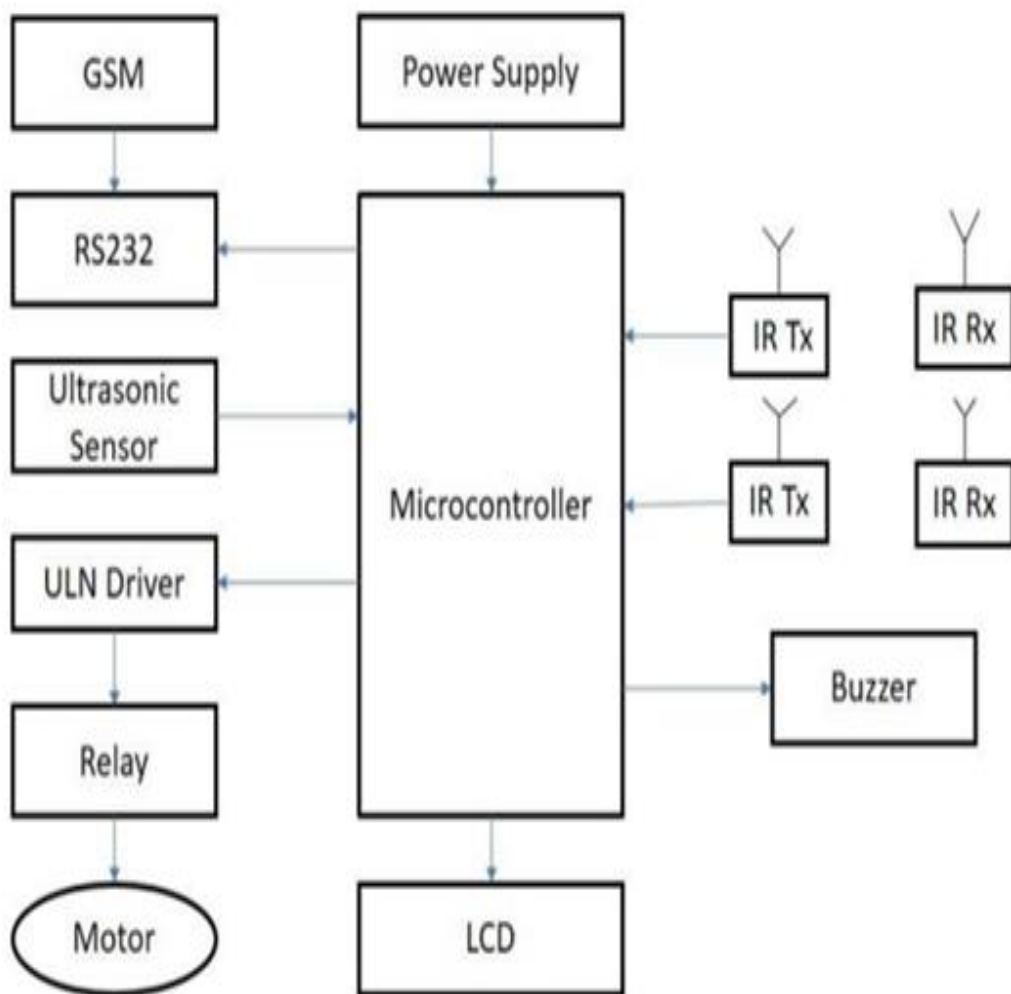
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	We need to register our tickets through online and if it confirmed it will be notified through mail.
NFR-2	Security	The customer details will be kept safe and it will not be shared like any other database.
NFR-3	Reliability	It's like a public oriented project and all the details of a common people have been stored in it. So high security and higher performance is mandatory. Hence reliability will also be more.
NFR-4	Performance	It can have a better performance and can withstand with large number of users without having any lagging issue.
NFR-5	Availability	Since it is a government oriented project and it will be available for all the time 24/7 and people can book at anytime and anywhere.
NFR-6	Scalability	This project can withstand for huge years and technology updation can also applicable to it.

## 5.PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAM



## 5.2 SOLUTION AND TECHNICAL ARCHITECTURE



### 5.3 USER STORIES

User Story Number	User Story / Task	Story Points	Priority	Team Members
USN-1	Getting into IBM Watson and create a device with device ID, device type with separate organization ID, authentication token in it.	1	High	Gokulnath S
USN-2	Getting into Cloudant DB to store our data in it and can be retrieved when the database is called. It will show the information about the tickets booked.	1	Medium	Gokulnath S
USN-3	Getting into node red and creating a design flow how the process will be working and connecting it with world map and ibm Watson and cloudant DB.	1	High	Jawahar M
USN-4	Creating a python code to locate the train by using its latitude and longitude and connect it with IBM Watson by organization ID, device ID, device type, token	1	High	Kavin P
USN-5	Creating a python code to generate a qr-code generator and reader. Data entered will be stored in DB and while scanning the code ticket details will be published.	1	High	Kavin P
USN-6	In MIT app design layout will be created and project will be deployed in it.	1	Medium	Sundaramoorthi S
USN -7	Every sprint will be merged with each other and testing with the required inputs.	1	High	Sundaramoorthi S

## 6.PROJECT PLANNING AND SCHEDULING

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	IBM Watson IOT platform,	USN-1	Getting into IBM Watson and create a device with device ID, device type with separate organization ID, authentication token in it.	1	High	Gokulnath S
	Cloudant DB	USN-2	Getting into Cloudant DB to store our data in it and can be retrieved when the database is called. It will show the information about the tickets booked.	1	Medium	Gokulnath S
Sprint-2	Node RED	USN-3	Getting into node red and creating a design flow how the process will be working and connecting it with world map and ibm Watson and cloudant DB.	1	High	Jawahar M
Sprint-3	Tracking	USN-4	Creating a python code to locate the train by using its latitude and longitude and connect it with IBM Watson by organization ID, device ID, device type, token	1	High	Kavin P
	QR-code	USN-5	Creating a python code to generate a qr-code generator and reader. Data entered will be stored in DB and while scanning the code ticket details will be published.	1	High	Kavin P
Sprint-4	MIT app inventor	USN-6	In MIT app design layout will be created and project will be deployed in it.	1	Medium	Sundaramoorthi S
	Testing	USN -7	Every sprint will be merged with each other and testing with the required inputs.	1	High	Sundaramoorthi S

## 6.2 SPRINT DELIVERY SCHEDULE

### Project Tracker, Velocity & Burndown Chart: (4 Marks)

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

# JIRA:-

Smart Solution for Railways by using IOT

Smart Solution for Railways by using IOT

Software project

PLANNING

Roadmap

Backlog

Board

Reports

DEVELOPMENT

Code

Project pages

Add shortcut

Project settings

You're in a team-managed project

Learn more

Projects / Smart Solution for Railways by using IOT

Roadmap

Q

K S J G

+

Status category

Epic

Give feedback

Share

Export

View settings

	OCT	NOV	NOV	NOV																											
	3	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Sprints																															
SSFRBUI-3 IBM IOT WATSON																															
SSFRBUI-6 Cloudant DB																															
SSFRBUI-7 Node-Red																															
SSFRBUI-8 Tracking																															
SSFRBUI-9 QR-code																															
SSFRBUI-10 MIT Inventor																															
SSFRBUI-11 Testing																															
+ Create Epic																															

Today

Weeks

Months

Quarters

23

## **7.CODING AND SOLUTIONING**

### **7.1 FEATURE 1**

- IoT Device
- IBM Watson platform
- Node Red
- Cloudant DB
- Web UI
- Geofence
- MIT App
- Python Code

### **7.2 FEATURE 2**

- Registration
- Seats
- Name
- Age
- Mobile Number
- Boarding Station
- Destination Station



## IBM code:-

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "625xj1",
        "typeId": "GPS",
        "deviceId": "12345"
    },
    "auth": {
        "token": "wOU&i?aL*2Le008hJ&"
    }
}

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, onPublish=None)
    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': 'Train1', 'lat': 17.6340889, 'lon': 78.4745052}
    pub (myData)
    time.sleep (3)
    myData={'name': 'Train1', 'lat': 17.6248626, 'lon': 78.4720259}
    pub (myData)
    time.sleep (3)
    myData={'name': 'Train1', '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 code:-

```
import cv2
import numpy as np
import time
import pyzbar
from ibmcloudant import cloudant_v1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator

authenticator = BasicAuthenticator('apikey-v2-
acv8gh5fnu0u4mh2f8c5x975ae5rnphr3jxkr5d9ril','c1dd4db6e976d915751882f688e410ec')
service = cloudant_v1(authenticator=authenticator)

service.set_service_url('https://apikey-v2-
acv8gh5fnu0u4mh2f8c5x975ae5rnphr3jxkr5d9ril:c1dd4db6e976d915751882f688e410ec@adad2
af9-59c4-41bb-b4b4-806f0d6962b2-bluemix.cloudantnosqldb.appdomain.cloud')

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

while True:
    __, frame = cap.read()
    decodedObjects = pyzbar.decode (frame)
    for obj in decodedObjects:
        #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)

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

## 8.TESTING

### 8.1 TEST CASES

1				TEAM ID : PNT2022TMD13493
2				PROJECT : SMART SOLUTION FOR RAILWAYS
3				DATE : 17 NOVEMBER 2022
4	TESTCASE ID	TESTCASE	TEST SCENARIO	TEST STEPS
5	1	IBM WATSON IOT PLATFORM	To check whether the ibm watson is get connected	login to ibm watson iot platform
6				id , password
7				check whether it has the separate organization id
8				team mates id
9				check whether separate device name , id , authentication token generated
				device name , type
				to check whether it is showing output
				device code and inputs

4	EXPECTED OUTPUT	ACTUAL OUTPUT	TEST RESULT	TEST COMMENTS	BUG ID	TESTED BY
5	it should get login to the watson page	it has been logged in to the login page	PASS	GOOD		GOKULNATH.S
6	it should shows the organization id	separate organization id has been shown	PASS	GOOD		GOKULNATH.S
7	it should shows the all the team members name / id	it is showing all the team members	PASS	GOOD		GOKULNATH.S
8	new device should be created	new device has been created	PASS	GOOD		GOKULNATH.S
9	it should shows device gets connected and should show the output	its showing that device gets connected and output are verified	PASS	GOOD		GOKULNATH.S

10	2	CLOUDANT DB	to check whether db is connected	login to cloudant db	id , password
11				check whether separate db is created	db name and type
12	3	NODE-RED	to check whether node-red is connected and shows the output	login in to node-red	id , password
13				check whether all the necessities are imported and connected	nodes
14				check whether cloudant is connected	cloudant db link
15				check whether ibm watson is connectd	watson device details
16				check whether map is connected	latitude , longitude

10	it should get login to the cloudant page	it has been logged in to the login page	PASS	GOOD		GOKULNATH.S
11	it should show separate db with given name	it shows separate db with the given name	PASS	GOOD		GOKULNATH.S
12	it should get login to the node-red page	its get entered into the login page	PASS	GOOD		JAWAHAR.M
13	it should not show any error on nodes	it is not showing any errors	PASS	GOOD		JAWAHAR.M
14	cloudant should gets connected	cloudant has been connected	PASS	GOOD		JAWAHAR.M
15	watson should gets connected	watson has been connected	PASS	GOOD		JAWAHAR.M
16	world map should gets connected	worldmap has been connected and shows the output	PASS	GOOD		JAWAHAR.M

17	4	TRACKING	check whether it locates the latitude and longitude	check whether python installed with all import files	import files
18				check whether the code shows any error	code
19				check whether it is running	code
20				check whether it is showing correct location	latitude , longitude
21				check whether it is connected with map	latitude , longitude
22	5	QR CODE	check whether qr code is generated	check whether the code shows any error	code
23				check whether UI page is created	node-red
24				check whether user able to select all criteria	ui
25				check whether db is connected	cloudant db link
26				check whether qr-code has been generated	user details

17	python should get installed with import files	python has been installed with import files	PASS	GOOD		KAVIN.P
18	it should not show any error on codes	it is not showing any errors	PASS	GOOD		KAVIN.P
19	it should gets run	it is running successfully	PASS	GOOD		KAVIN.P
20	it should shows the exact location	it is showing the exact location	PASS	GOOD		KAVIN.P
21	it should get connected with map	it has been connected with the map	PASS	GOOD		KAVIN.P
22	it should not shows any error	it is not showing any errors	PASS	GOOD		KAVIN.P
23	ui page should gets opened	ui page has been opened	PASS	GOOD		KAVIN.P
24	user should be able to access all	user has been able to access all	PASS	GOOD		KAVIN.P
25	cloudant should gets connected	cloudant has been connected	PASS	GOOD		KAVIN.P
26	qrcode should be generated	qrcode has been generated	PASS	GOOD		KAVIN.P

27			check whether qr code is reading	check whether it turns on the scanner/camera	camera/scanner
28			check whether qr-code scanned	check whether the qrcode is scanning	camera/scanner
29				check whether it showing all the details in db	db
30				check whether qrcode is disabled	qrcode
31	6	TESTING	check entire process	check watson is connected	watson
32				check node-red is connected	node-red
33				check whether db is connected	db
34				check whether details are shown	db

27	it should gets turned on camera/scanner	it has been turned on camera/scanner	PASS	GOOD		KAVIN.P
28	it should read the qrcode	qrcode readed successfully	PASS	GOOD		KAVIN.P
29	it should shows all the details about the ticket confirmation	it has showed all the details of the confirmation	PASS	GOOD		KAVIN.P
30	qrcode should gets disabled in few seconds	qrcode has been successfully disabled	PASS	GOOD		KAVIN.P
31	iot watson should produce its output	iot watson has producing its output	PASS	GOOD		SUNDARAMOORTHY
32	node-red should produce its output	node-red has been producing its output	PASS	GOOD		SUNDARAMOORTHY
33	cloudant should gets connected	cloudant has been connected	PASS	GOOD		SUNDARAMOORTHY
34	details in db should be shown	details in db should be shown	PASS	GOOD		SUNDARAMOORTHY

# 9.RESULTS

## 9.1 PERFORMANCE METRICES



## **10.ADVANTAGES AND DISADVANTAGES**

### **10.1 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.

### **10.2 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.



## **11.CONCLUSION**

A significant number of lives are lost as a result of accidents in the rail transportation system. Thus, this system aids in the prevention of accidents by informing the railroad authorities in advance of any faults or cracks. so that they can be fixed and the number of accidents decreases. This undertaking is economical. They can be improved and enhanced in accordance with their applications by utilising more strategies. By preventing accidents, this technology can save many lives. Long-term large-scale implementation of the concept is possible to support improved rail track safety requirements and offer efficient testing infrastructure for improved outcomes in the future.

## **12.FUTURE SCOPES**

In future CCTV systems with IP based camera can be used for monitoring the visual videos captured from the track. It will also increase security for both passengers and railways. GPS can also be used to detect exact location of track fault area, IP cameras canalso be used to show fault with the help of video. Locations on Google maps with the helpof sensors can be used to detect in which area track is broken.

## 13.APPENDIX

### 13.1 SOURCE PROGRAM

```
import math, random import os
import smtplib import sqlite3 import requests
from bs4 import BeautifulSoup
from django.contrib.auth.base_user import AbstractBaseUser from django.db
import models
import logging import pandas as pd import pytsx3
from plyer import notification import time
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image, ImageDraw from pickle import load,dump import
smtplib, ssl
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart import email
from email import encoders
from email.mime.base import MIMEBase
import attr
from flask import Blueprint, flash, redirect, request, url_for from flask.views
import MethodView
from flask_babelplus import gettext as _
from flask_login import current_user, login_required
from pluggy import HookimplMarker
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()
def generateOTP() :
# Declare a digits variable # which stores all digits digits = "0123456789" OTP
= ""
# length of password can be changed # by changing value in range
for i in range(4) :
OTP += digits[math.floor(random.random() * 10)] return OTP
# Driver code
If name== "main" :
print("OTP of 4 digits:", generateOTP()) digits="0123456789"
OTP=""
for i in range(6): OTP+=digits[math.floor(random.random()*10)]
otp = OTP + " is your OTP" msg= otp
s = smtplib.SMTP('smtp.gmail.com', 587) s.starttls()
s.login("Your Gmail Account", "Your app password") emailid = input("Enter
your email: ") s.sendmail('&&&&&&&&&',emailid,msg)
a = input("Enter Your OTP >>: ")
if a == OTP: print("Verified")
else:
print("Please Check your OTP again") 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)
USERNAME = StringVar() PASSWORD = StringVar()
Top = Frame(root, bd=2, relief=RIDGE) Top.pack(side=TOP, fill=X)
Form = Frame(root, height=200) Form.pack(side=TOP, pady=20)

```

```

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)

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) def Database():

global conn, cursor

conn = sqlite3.connect("pythontut.db") cursor = conn.cursor()

cursor.execute("CREATE TABLE IF NOT EXISTS `member` (mem_id
INTEGER NOT NULL PRIMARY KEY

AUTOINCREMENT, username TEXT, password TEXT)")
cursor.execute("SELECT * FROM `member` WHERE `username` =
'admin' AND `password` = 'admin'")

if cursor.fetchone() is None:

cursor.execute("INSERT INTO `member` (username, password)
VALUES('admin', 'admin')")

conn.commit()

def Login(event=None): Database()

if USERNAME.get() == "" or PASSWORD.get() == "":

lbl_text.config(text="Please complete the required field!", 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 username or password", fg="red")
    USERNAME.set("")
    PASSWORD.set("")
    cursor.close() conn.close()

    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()

def getdata(url):
    r = requests.get(url) return r.text

# input by geek from_Station_code = "GAYA" from_Station_name = "GAYA"
To_station_code = "PNBE" To_station_name = "PATNA" # url
url = "https://www.railatri.in/booking/trains-between-
stations?from_code="+from_Station_code+"&from_name="+from_Stat
ion_name+"+JN+&journey_date="+Wed&src=tbs&to_code=" + \

```

```

To_station_code+"&to_name="+To_station_name + \ "+JN+"&user_id=-
1603228437&user_token=355740&utm_source=dwebsearch_tbs_search_
trains"
# pass the url
# into getdata function 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)
print("\n\nTicket Booking System\n") restart = ('Y')
while restart != ('N','NO','n','no'):
    print("1.Check PNR status") print("2.Ticket Reservation")
    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_1 = [] age_1 = [] sex_1 = []
        for p in range(people):
            name = str(input("\nName : ")) name_1.append(name)
            age = int(input("\nAge : ")) age_1.append(age)
            sex = str(input("\nMale or Female : ")) sex_1.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) for p 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
```



GITHUB LINK :

<https://github.com/IBM-EPBL/IBM-Project-2809-1658483443>

VIDEO LINK :

<https://youtu.be/IOOuhjmeF6w>