PROJECT REPORT

Project Name : SMART SOLUTION FOR RAILWAYS

Team ID : PNT2022TMID49411

TEAM LEADER - HARINI DEEPA.J

TEAM MEMBER1 - REENA.S

TEAM MEMBER2 - RITHIKA SRI.P

TEAM MEMBER3 - LATCHAGA.M

INTRODUCTION

1.1 Project Overview

- There will a Mobile app for the public through which they can book tickets by seeing the available seats.
- After booking the person will get a QR code which has to be shown to the Ticket Collector at boarding. He scans the QR code to identify the personal details.
- Through this app the traveler can order the food, the pantry section will get the notification of order.
- A GPS module is present in the train to track it. The live status of the journey is updated in the mobile app continuously.
- The user can set a notification for intimating the train live status for both boarding and destination stations.

.

1.2 PURPOSE

Smart solution for railways is used to Its application increases safety, efficiency and ease of use with train management systems. Control and surveillance systems reduce the risk of collisions and regulate speed. Advanced consumer technologies help maximize connectivity and allow passengers to continue their Activities on smart devices while travelling.

LITERATURE SURVEY

2.1. EXISITING PROBLEM

Paper pen works takes time and can be time consuming. People in this current fast world won't like to still standing in a queue and book tickets. And most importantly, nowadays most people have a smartphone on their own. So that implementing the solution proposed won't be a challenging task. In fact, compared to the traditional method of booking tickets standing in a queue, our method has a lot of advantages.

Our solution is to design a website where we can book ticket and receive QR Code which can be scanned during boarding. Passengers can also monitor the train status and as well as they are alerted through mobile before their destination arrives.

There won't be any loss in implementing this solution proposed by us. In fact, the truth is the Tamil Nadu government has been already involved in developing and promoting the solution proposed by us by providing 60%discounts to those passengers who book tickets through online in Chennai Metro Rail Service(CMRL) .

And Finally, Digitizing the booking and verification process & alert passenger before their destination arrives. Digitizing the works reduces manual paper pen work and it becomes easier and time saving. Because of the smart changes that ca n be introduced by implementing in the railway system, Our Indian Railways can be called as "SMARTRAILWAYS".

2.2.LITERATURE REVIEW

PAPERNAME	AUTHOR	YEAR	METHOLOGY	MERITS	DEMERITS
Passenger Monitoring Model for easily Accessible Public CityTrams/Trains.	RomanKhoebl, Teeravisi Laohapensaeg ,Roungsan Chsricharoen	2015	Passenger Monitoring,passenger control RFID distance reading, ticket control,RFID ticket inspection.	It is possible tot travel cross-country with single public ansportatio nc ard, using transport system sofseveral transport operators.	Applicable Only for passenger monitoring
Application of smart computing Indian Railway Systems.	ParagCje ,Asokeh	2014	By Interlinking Unique identification system with trainticket reservation system by usingvideo surveillance,rail sensors,biometrici input devices and multimedia displays.	Reduces manual effort in passenger dataentry. Provides Security Verification.	Significant estment is needed. Risk of databas e.
Android Suburban Railway Ticketing with GPS as Ticket Checker.	Sana Khoja,Maithi liK adam	2012	Android, SQ lite,Cloud Database,ASR ,Q RCode.	E-Ticket facility,enabg reuse andreplaceet of components.	QR Codes before the user enters or leaves the station, where the user can have access which is risk in

					ticketbookin g.
Sujith Kumar, K.M.Yatheendra Parvan, V.Sumathy, Thejeswari C.K		2017	Digitalization, SmartRailways, AadharCard, Smartphone,Identit y Verification.	Employ a mobile application through whichpassenger s canaccess various ticketing optionsin user friendlyand efficientmanner.	Biometric database is risk of hacking.
Sarvath Sab,Sharon Philip,Shriharsha, MukundNaik, SudeepSherry	2022			The System Is not fool- proof andr equires adramatic change in the existing system minterms of the people owed on platforms, etc.but baby stepsmatt er.	

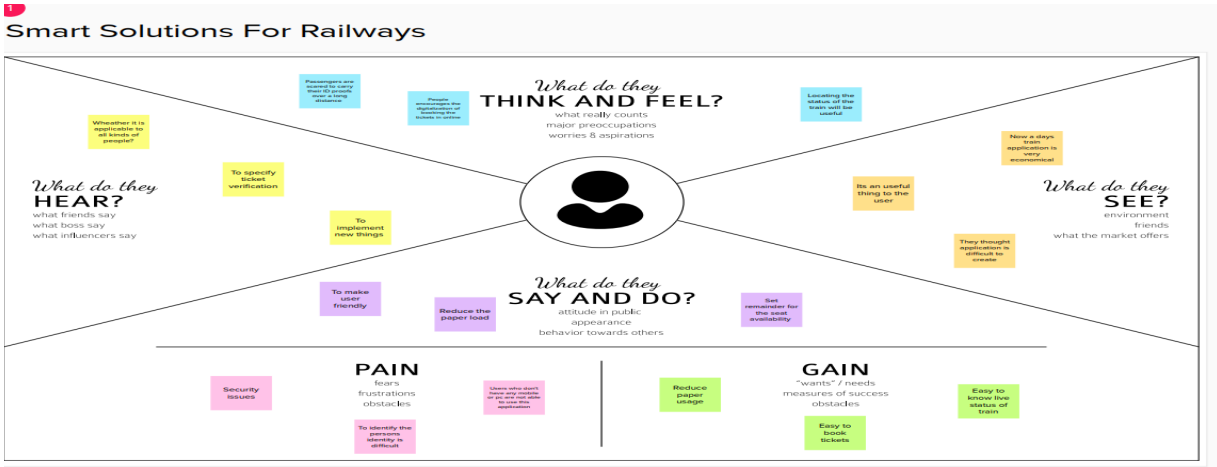
2.3.Reference:

1. Roman Khoeblal, Teeravisit Laohapensaeng, Roungsan Chaisricharoen, “Passenger Monitoring Model foreasily Accessible Public City Trams/Trains”(2015).
2. Parag Chatterjee, Asoke Nath, “Application of smart computing in Indian Railway Systems”(2014).
3. Sana Khoja, Maithili Kadam, “Android Suburban Railway Ticketing with GPS as Ticket Checker”(2012).
4. SujithKumar,K.M.YatheendraParvan,V.Sumathy,ThejeswariC.K,“Novel

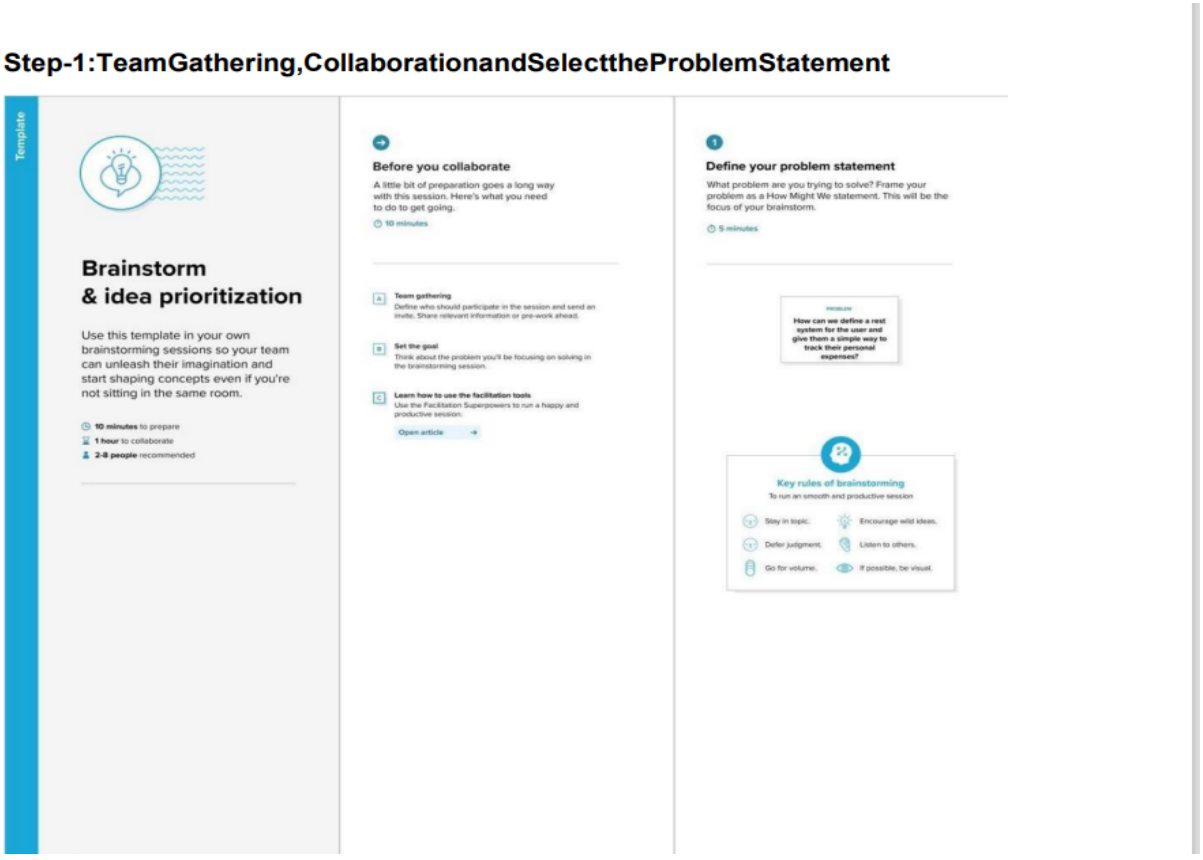
   Approach for Smart Indian Railways”(2017).
5. Srarvath saba, Sharon Philip, Mukund Naik,Sudeep as Sherry,A Review on IoT Based Automated seat allocation and verification using QR code (2022)

3.IDEATION & PROPOSED SOLUTION

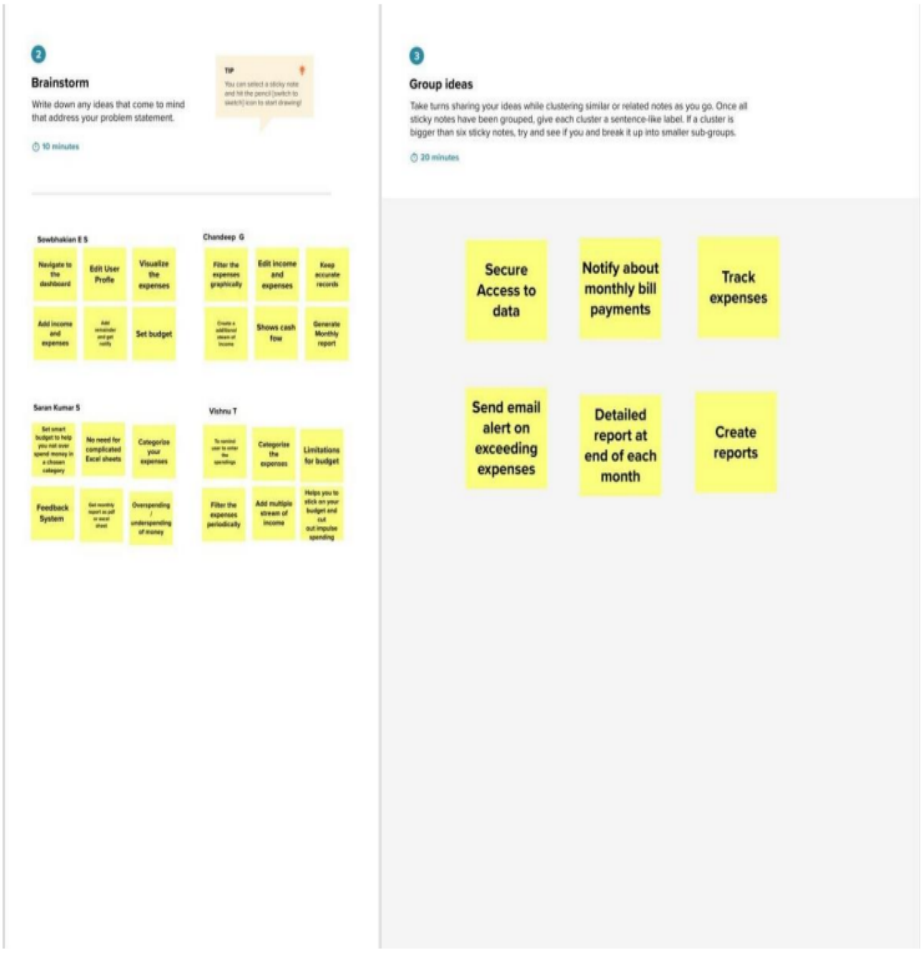
3.1.EMPATHY MAP



3.2.Ideation and Brainstorming



Step-2: Brainstorm,IdeaListingandGrouping



Step-3: IdeaPrioritization

3.3.Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	In order to satisfy the passengers, the Railways provides various services to its passengers But,the passengers can face some problems.
2.	Idea/Solution description	The idea is to minimize the ticket booking problems among the passengers by providing Online mode of booking rather than papers. . In queues in front of the ticket counters in railway stations have been drastically increased over the time.
3.	Novelty/Uniqueness	Online mode of booking is most common and so ease of access to everyone that makes more efficient uniqueness of utilizing The technique. People can book their ticket through online and they geta QR code through SMS
4.	Social Impact/ Customer Satisfaction	Customers for sure they get satisfied as they are in the fast roaming world this technique makes more easier for travelling passengers. A web page is designed in which the user can book tickets and will be provided with the QR code, which will be shown to the ticket collector and by scanning the QR code the ticket collect or will get the passenger details.
5.	Business Model (Revenue Model)	A web page is designed in which the user can book tickets and will be provided with the QR code ,which will be shown to the ticket collector and by scanning the QR code the ticket collector will get the passenger details. The booking details of the user will be stored in the database, which can be -retrieved anytime.

6.	Scalability of the Solution	The scalability of this solution is most feasible among the passengers who are willing to travel. No need of taking printout Counter ticket has to be handled with care, but SMS on mobile is enough. No need to taking out wallet and showing your ticket to TTR just tell your name to TTR that you area passenger with valid proof.

4. REQUIREMENT ANALYSIS

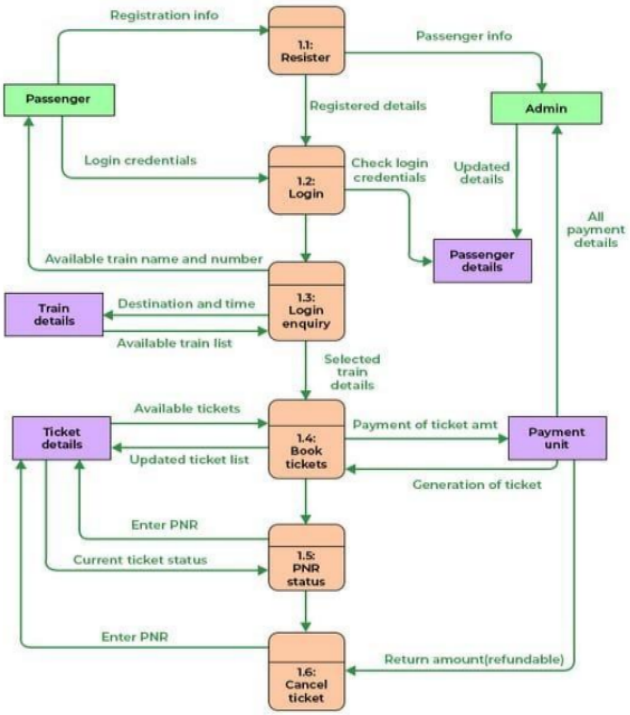
4.1.Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Before the user registration there will be language selector .All the language is applicable. When user enter in to the application they can see the page of showing enter the email ,mobile number and name. After that in screen it shows the verification code is sended through the email id.
FR-2	User verification	The verification code is sended to the registered email id
FR-3	User confirmation	The verification code is entered in to the app application. After finishing that home page is opened.
FR-4	Process of booking	When the home page is opened there will be a from and to option. We must enter the details then after that we can able to see the number of trains availability and seats availability. We can select the particular train and particular seats which we need and click the confirm option.
FR-5	Payment process	After entering all the details select the payment option like google payp hone pay,pay tym,etc…….. When we select that method it process through selected payment option then payment should be done carefully,then the ticket is confirmed.After confirmation it will return to the page and we can see the details of booking.
FR-6	Confirmation message	After all the QR code will be send through the sms and email id. QR code will be shown to the ticket collector when the QR code is scanned booking details will be shown .

5. PRODUCT 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 enters and leaves the system, what changes the information, and where data is stored.

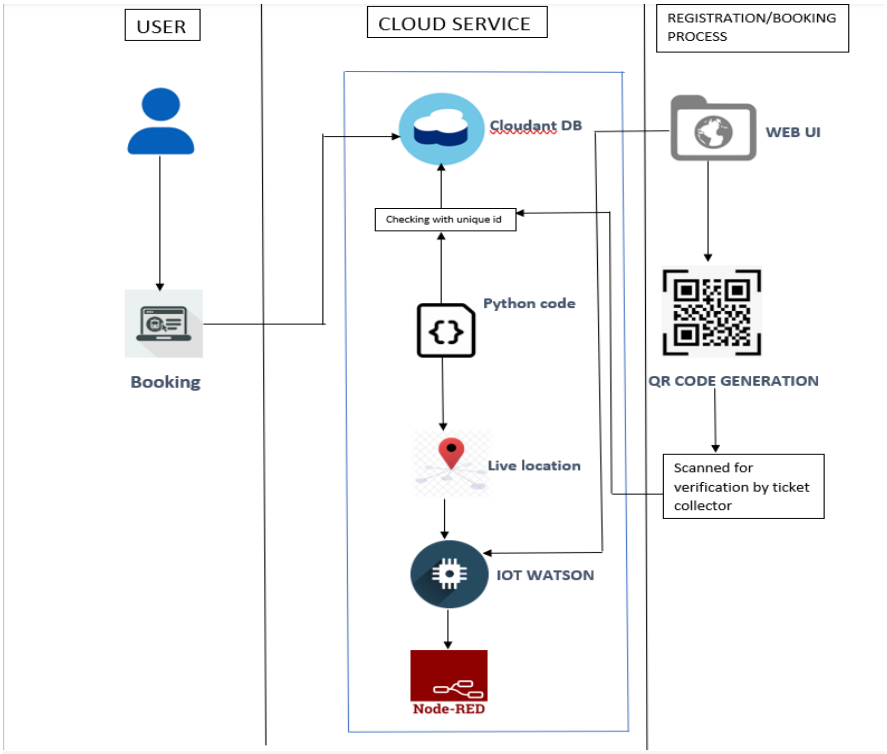


5.2. SOLUTION AND TECHNICAL ARCHITECTURE

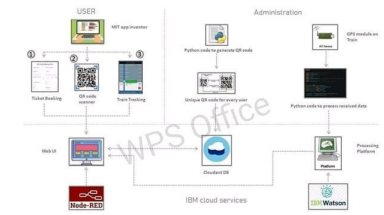
Solution Requirements:

- IBM Watson IOT platform
- Node-Red
- MIT app
- IOT device
- Cloudant DB

Solution Architecture Diagram And Data Flow Of Smart Solutions For Railways:



5.3 User Stories:



5.3 User Stories

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Web UI	User can login and book their ticket through the website based on the availability of the seats.	HTML, CSS, JavaScript
2.	Cloud Services	Requirements filled by the passenger is stored in the cloud database.	Python
3.	GPS Tracking	Live Location details shared through the code to share the location in the website	IBM Watson Service
4.	External API-1	Used for rail schedule, ticketing and travel documents generation, cancellation.	Sabre API
5.	External API-2	Used for combining carriers and ticket types, Multilanguage & currency support.	Trainline B2B API
6.	Data Processing	Ticket is verified with the unique ID generated with the cloudland DB	Python, IBM cloud

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	CSS, Backend framework,	Python, IBM cloudant DB
2.	Security Implementations	Data entered are encrypted, Continuous Location Tracking	Python, Cloud service
3.	Scalable Architecture	The scanner and the codes written are highly scalable where any implementation can be done anytime needed	Python
4.	Availability	Any time available system. The ticket can be verified by the ticket collector from anywhere.	IBM Load Balancer
5.	Performance	Though the details are get stored in the cloud the system crash will not affect the data. The data can be retrieved from anywhere with a scanner. And the GPS states the exact location of the train.	Distributed Services, GPS Tracker

6. **PROJECT PLANNING AND SCHEDULING:**

ProjectPlanningPhase

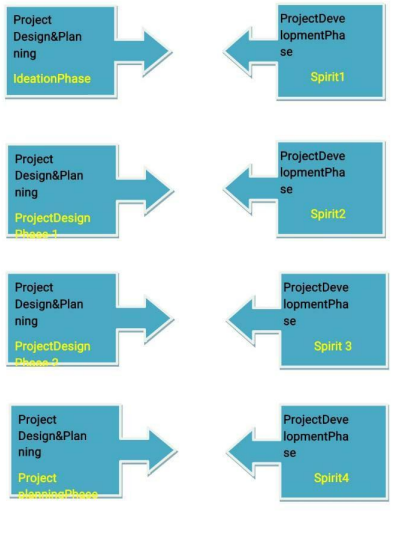
SprintDeliveryPlan

TeamID:PNT2022TMID49411

ProjectName: SmartSolutionForRailways

SPRINTPLAN
1.IdentifytheProblem
2.PrepareaAbstract,ProblemStatements
3.ListaRequire Needed
4.CreateaCodeandRunit
5.MakeaPrototype
6.TestwiththeCreatedcodeandcheckthedeignedprototype
7.SolutionfortheProblemisFound!!!

Team:PNT2022TMID49411 **ProjetTitle:**Smart SolutionForRailways



7.CODING:

Login app:

```
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
    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()


#==============================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()
```

VERIFY:

```
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()
```

SENSOR SIM:

```
#include"DHT.h"
#defineDHTPIN2
#defineDHTTYPEDHT22 //DHT22(AM2302),AM2321
DHTdht(DHTPIN,DHTTYPE);
voidsetup(){
Serial.begin(115200);
Serial.println(F("DHT22example!"));
dht.begin();
}
Void loop(){
Float temperature=dht.read Temperature();
Float humidity=dht.read Humidity();
//Checkifanyreadsfailedandexitearly(totryagain).
if(isnan(temperature)||isnan(humidity)){
Serial.println(F("FailedtoreadfromDHTsensor!"));
return;
}
Serial.print(F("Humidity:"));
Serial.print(humidity);
Serial.print(F("% Temperature:"));
Serial.print(temperature);
Serial.println(F("°C"));
//Waitafewsecondsbetweenmeasurements.
```

```
delay(2000);
```

8.TESTING:

1	19-Nov-22							
2	PNT2022TMID49411							
3	smart solutions for railways							
4	4 marks							
5	**Steps To Execute**	**Expected Result**	**Actual Result**	**Status**	**Commnets**	**TC for Automation(Y/N)**	**BUG ID**	**Executed By**
6	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				Harini deepa
7	1.Generating of OTP number	user can register through phone numbers, Gmail, Facebook or other social sites and to get oto number	Working as expected	pass				Reena
8	1.Enter gmail id and enter password 2.click submit	OTP verifed is to be displayed	Working as expected	pass				Rithika sri
9	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 5.Click on login button	Application should show 'Incorrect email or password ' validation message.	Working as expected	pass				latchaga
10	1.As a user, I can enter the start and destination to get the list of trains available connecting the above	A user can view about the available trains to enter start and destination details	Working as expected	fail				Harini deepa
11								
12								

9.RESULTS

9.1.PERFORMANCE METRICS:

10.ADVANTAGES:

1. Highly Time efficient
2. More organized and updated
3. Allows passenger to have full control over the information availability
4. Significantly more convenient
5. There is more flexibility with respect to initiating and processing changes
6. Immediate response and action occurrence
7. The passenger is always fully notified of any and all changes regarding scheduling and availability
8. A very convenient platform for rising and rectifying queries

DISADVANTAGES:

1. Requires heavy work to maintain and ensure efficient functioning of the platform , from the developer's side.
2. Requires constant and instantaneous traffic maintenance and rerouting
3. Probability of crashes and server issues
4. Payment issue occurrences
5. Minor glitches regarding precise location mapping
6. Requirement of a strong internet access for smooth functioning
7. Traffic related further delays
8. Requirement to constantly keep platform updated and ridding of all reported bugs

11.CONCLUSION:

The project that we've managed to implement is primarily to enable an user to book train tickets based on the current availability, meanwhile providing with all necessary information and also provide a platform for new user registration, and existing user login. We have also managed to enable a live train tracking system using GPS and making sure that the user is able to avail the accurate data at any time on user demand. We have provided a well-functioning booking system with a stable and fail safe payment procedural, which the user can utilize with ease and efficiency and the user can also confirm the booking using the QR-code generated. The user will also be receiving time to time notification on the necessary details like train chart, coach position, booking status and delays if any. Ticket collector can scan the QR code and extract the information from the QR Code i.e., Unique ID. With that Unique ID, data is fetched from the Cloudant DB, if it is not found, then it displays Not a Valid Ticket. The user can also cancel the booked ticket, making the whole process of ticket reservation and travel significantly hassle free and time efficient.

12.FUTURE SCOPE FUTURE WORK:

Our future renditions primarily aim at improving and increasing customer experience, vehicle tracking system. IOT is used along with AI which Provides enhanced features in finding out delays. Predicting delay and detecting the train arrival time so that help the passenger to act accordingly and keep tracking the location of the train and travel in easy and modern way.

To maintain the tracks, repairs and services to avoid accident, safeguard of things, track the running status of the train in smart railway system and reach the destination place on time due to train delay. Moreover we would also make sure that the payment system is much more secure and make sure that there is no room for error or any other midway glitches or lags of any sort.

We would also make sure to create a more inventive and minimal UI with maximum user ease and comfort with more snappy actions and faster response times and minimal UI stutters or glitches of any sort.

GitHub link:

https://github.com/IBM-EPBL/IBM-Project-9601-1659025359

Demo Link : https://photos.app.goo.gl/hPqnr31c7g9Fm1pv9