

## Project Report

Project title	Smart Solutions for Railways
project ID	PNT2022TMID46747
Name	PAVITRA-TEAM LEAD NITHITHA.U RUTRA.R SHATHANA.B
Date	19 November 2022

## I. INTRODUCTION

### Project Overview

As trains are one of the most preferred modes of transportation among middle 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 all such brutality we came up with a solution by providing an application which can be accessed 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 passenger data.

### Purpose

The purpose of this project is to report and get relived from the issues related to trains.

## 2. LITERATURE SURVEY

### Existing problem

- A Web page is designed for the public where they can book tickets by seeing the available seats.
- After booking the train, the person will get a QR code which has to be shown to the Ticket Collector while boarding the train.
- The ticket collectors can scan the QR code to identify the personal details.
- A GPS module is present in the train to track it. The live status of the journey is updated in the Web app continuously
- All the booking details of the customers will be stored in the database with a unique ID and they can be retrieved back when the Ticket Collector scans the QR Code.

## References

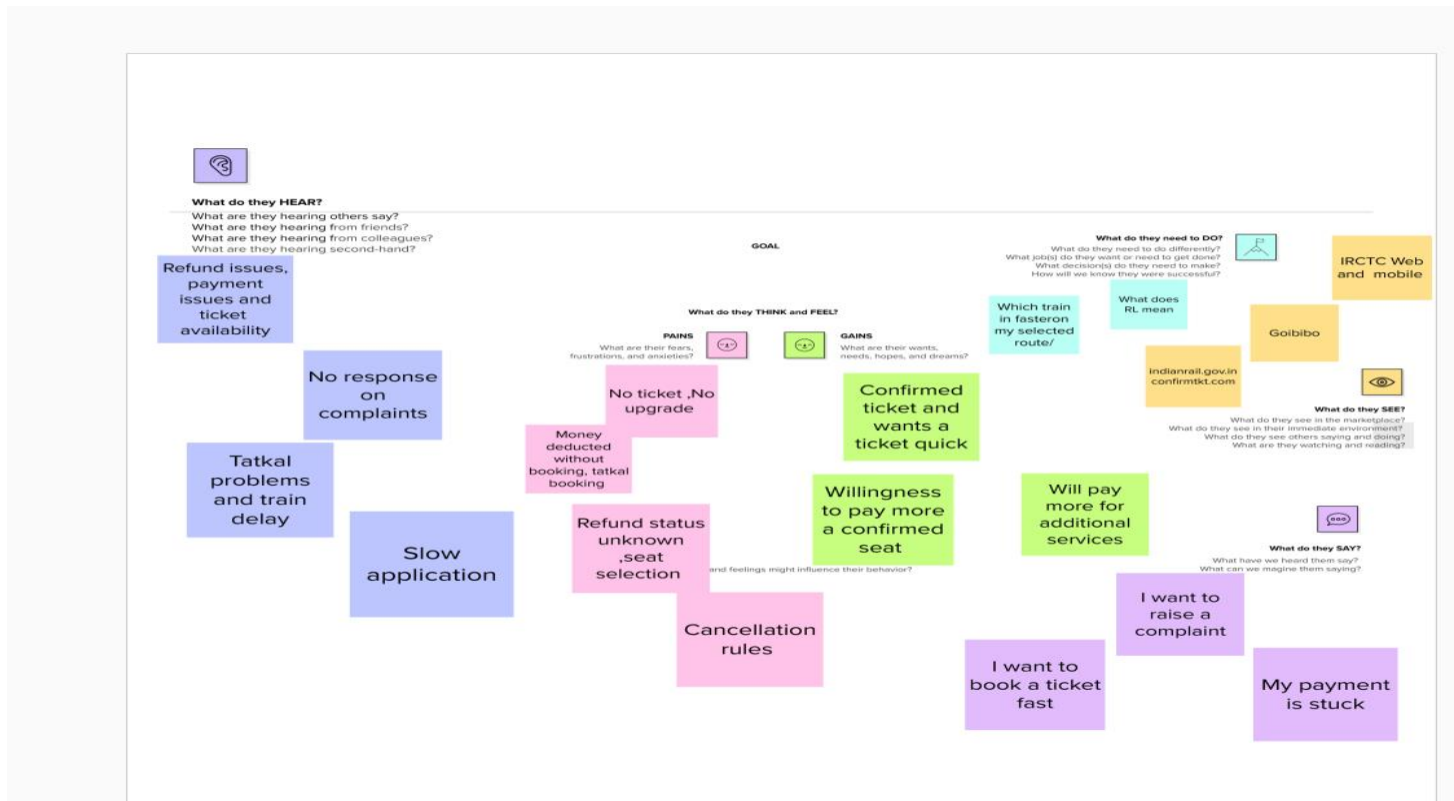
S.NO	TITLE	AUTHOR	YEAR	KEY TECHNOLOGY
1	Main geotechnical problems of railways and roads in kriolitozone and their solutions.	Kondratiev,Valentin G	2017	Main problems in railways
2	Construction and Building Materials	Sañudo, Roberto, Marina Miranda, Carlos García, and David García-Sanchez	2019	Drainage in railways
3	Problems of Indian Railways	Benjamin	2021	Common problems in Indian railways
4	A comparative study of Indian and worldwiderailways.	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

## Problem Statement Definition

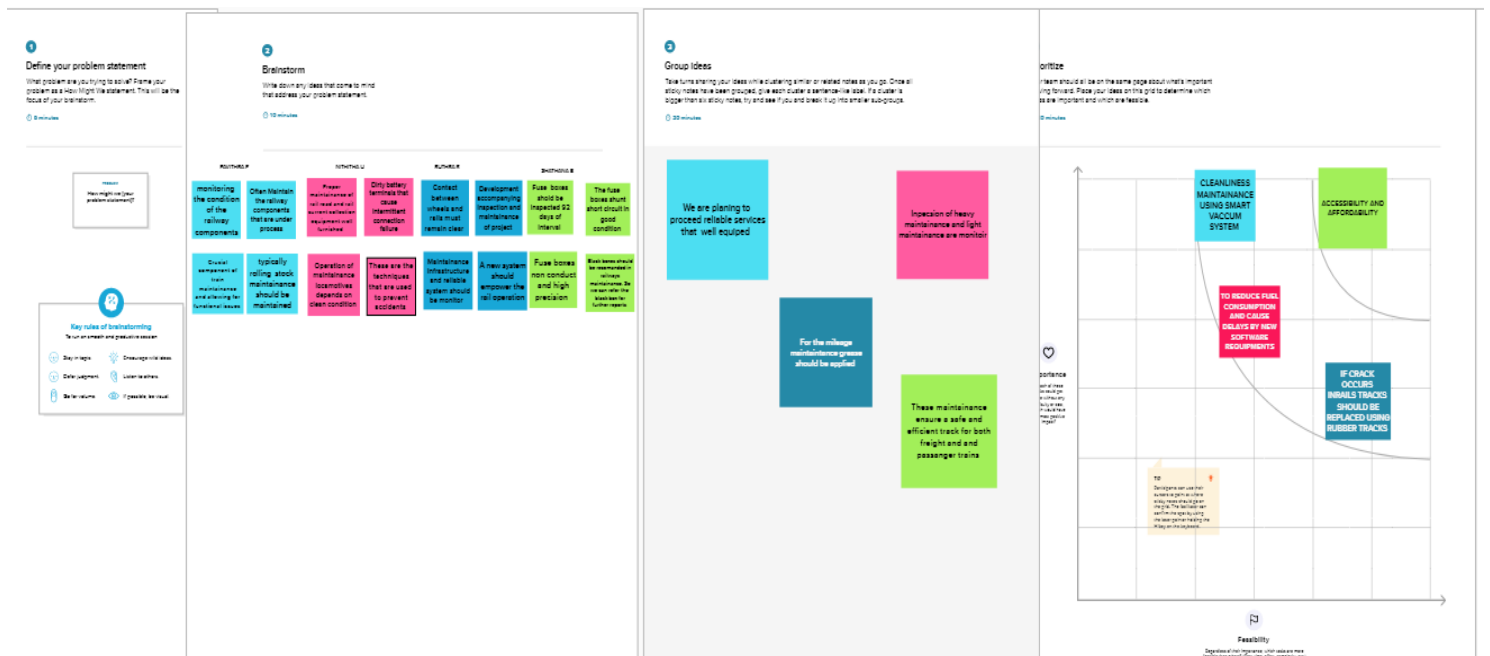
Smart Solutions for railways are designed to reduce the work load of the user and the use of paper.

### 3. IDEATION & PROPOSED SOLUTION

### 3.1 Empathy Map Canvas



## 3.2 Ideation & Brainstorming



### 3.3 Proposed Solution

	Parameter	Description
1.	Problem Statement (Problem to be solved)	Problems in the railways like robbery, fire accidents etc..
2.	Idea / Solution description	Developing an app for the passengers.
3.	Novelty / Uniqueness	The passengers can send an alert to the respective officials during the travel time Through the app when they are in trouble so that they can easily solve it.
4.	Social Impact / Customer Satisfaction	Usage of this app can be a great relief to the passengers, so that they can travel without any Fear.
5.	Business Model (Revenue Model)	5000
6.	Scalability of the Solution	This solution will be useful for passengers while travelling. They can use the app between the times of their travel. The users will feel more secured, in-case of an emergency by simply clicking on a button the alert signal will be sent to the respective officials and the corresponding measures will be taken.

## 3.4 Problem Solution fit

Project Title: Smart solution for railways

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID46747

Define CS, JR into CC	<b>1. CUSTOMER SEGMENT(S)</b> <small>Who is your customer? i.e. working parents of 0-5 y.o. kids</small>	<b>6. CUSTOMER CONSTRAINTS</b> <small>What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available services</small>	<b>5. AVAILABLE SOLUTIONS</b> <small>Which solutions are available to the customers when they face the problem? or need to get the job done? What have they tried in the past? What pros &amp; cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking</small>	Explore AS, differentiate
	<ul style="list-style-type: none"> <li>I am a train Passenger</li> </ul>	<ul style="list-style-type: none"> <li>Spending power</li> <li>budget</li> <li>tidiness of passengers</li> </ul>	<ul style="list-style-type: none"> <li>The smart sensor and analytics across the train engine and coaches and we are planning to determine significant service to passengers</li> </ul>	
Focus on JR, up to CS	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <small>Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.</small>	<b>9. PROBLEM ROOT CAUSE</b> <small>What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.</small>	<b>7. BEHAVIOUR</b> <small>What does your customer do to address the problem and get the job done? Directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)</small>	Focus on JR, up to CS
	<ul style="list-style-type: none"> <li>I am expecting waterless &amp; odourless toilet &amp; clean rail track box maintenance.</li> </ul>	<ul style="list-style-type: none"> <li>Geographical factors the North Indian plain with its level land, high density of population and rich agriculture presents the most favourable conditions for the development of railways.</li> </ul>	<ul style="list-style-type: none"> <li>Smart sensors and analytics across the train engine and coaches and we are planning to determine significant service to passengers</li> </ul>	
Identify strong TR & EM	<b>3. TRIGGERS</b> <small>What triggers customers to act? Looking for their current solution, existing solar panels, seeking about a more efficient solution for the same.</small>	<b>10. YOUR SOLUTION</b> <small>If you are working on an existing business, re-engineer your current solution. If you are working on a new business proposition, then keep it ideal until you figure out a concrete and come up with a solution that this world is customer friendly, solves a problem and matches customer expectations.</small>	<b>8. CHANNELS of BEHAVIOUR</b> <small>What kind of nature do customers who online? Direct: offline channels from #? OFFLINE: What kind of nature do customers who offline? Direct: offline channels from #? Indirect: offline channels from #? Indirect: offline channels from #?</small>	Identify strong TR & EM
	<ul style="list-style-type: none"> <li>Not burned locomotives exhaust stack fires can occur when diesel fuel is not burned properly or the stack and spark arresters are not cleaned periodically.</li> <li>In such cases, sparks may come out of the stack and could drift into flammable material on the right-of-way or beyond the right-of-way</li> </ul>	<ul style="list-style-type: none"> <li>The automation of toilet can significantly reduce the cost incurred by the train operator and at the same time provide a better service to passenger who will likely find a toilet out of</li> </ul>	<ul style="list-style-type: none"> <li>In online mode we monitor the system using IOT</li> <li>In offline mode the train passenger should use and keep the surrounding clean</li> </ul>	
<b>4. EMOTIONS: BEFORE / AFTER</b> <small>How do customers feel when they face a problem or a job not done right? How do customers feel when they face a problem or a job not done right? How do customers feel when they face a problem or a job not done right?</small>				
determine the status of on board toilet in real time and nowadays we are using new facilities				

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

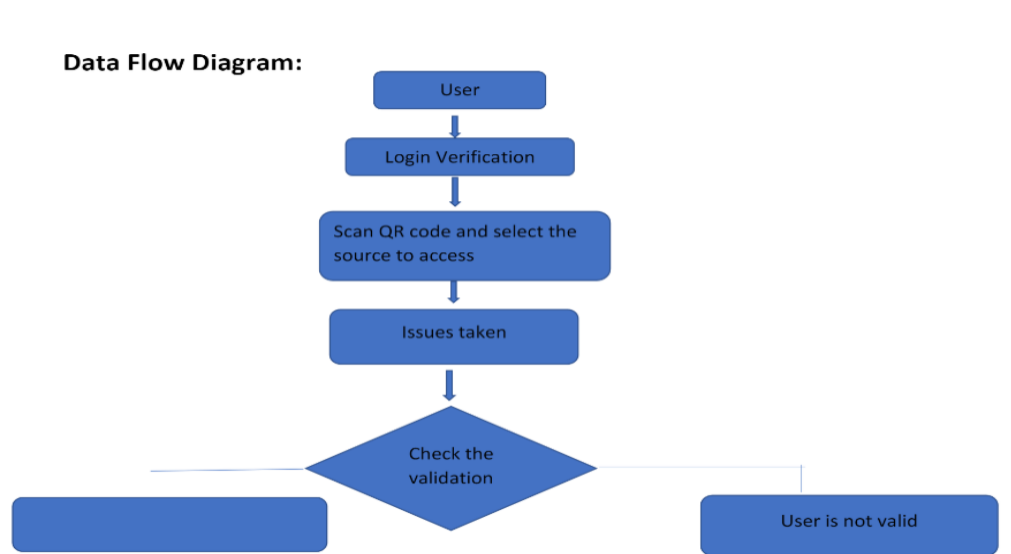
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Online Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Application installation	The application is installed through the given link
FR-4	User access	Access the app requirements

## 4.2 Non-Functional requirement

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	<ul style="list-style-type: none"><li>• The app can be used during the travelling time</li><li>• Easy and simple</li><li>• Efficiency is high</li></ul>
NFR-2	Security	By clicking on the icon, the alert 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 enough to support many users at the same time

## 5. PROJECT DESIGN

### 5.1 Data Flow Diagrams



### 5.2 Solution Architecture

**Digital platforms and integrated solutions for**

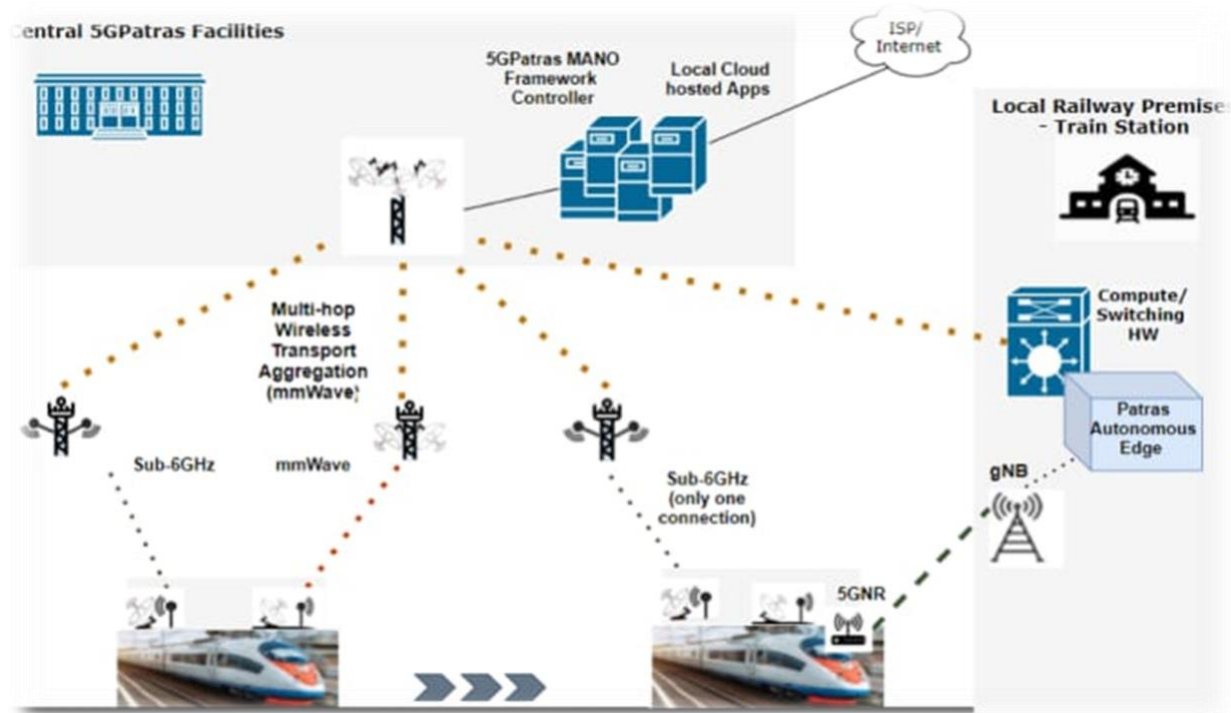
- **railway transport related services**
  - ✓ Electronic trade platform “Freight Transportation”
- **in Russian Federation (e-sales)**
  - ✓ Corridor One-Stop Shops (C-OSSs) on European Rail
- **Freight Corridors (RFC)**

- ✓ ETA electronic information exchange (ELETA Project)
- ✓ National Transport and Logistics Public Information
- **Platform (LOGINK) in China (B2B, B2G & G2B)**

### Visibility an electronic tracking

- ✓ GN SS electronic tracking options
- ✓ Indian Railways Real-time Train Information System

### Example - Solution Architecture Diagram



### Smart railways with 4G and 5G crision

## 5.3 User Stories

User Type	Functional Requirement(Epic)	User Story Number	Acceptance criteria	Priority	Release
PASSENGER (Mobile user)	Booking registration	USN-1	I can access the web link to install the application.	High	Sprint-1
	Confirmation	USN-2	I can receive confirmation email & click confirm.	High	Sprint-1

	Applic ation regist ration	USN-3	As a passenger, I can register for the application through the Web link.	I can register & access the application through google login.	Low	Sprint- 2
	Applica tion access	USN-4	As a passenger, I can access the application during my travel for resolving my issues.		Mediu m	Sprint- 1

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

<b>STEP 1</b>	Identify the problem
<b>STEP 2</b>	Prepare an abstract, problem statement
<b>STEP 3</b>	List required objects needed
<b>STEP 4</b>	Create a code and run it
<b>STEP 5</b>	Make a prototype
<b>STEP 6</b>	Test with the created code and check the designed prototype is working



## 6.3 Reports from JIRA

### SPRINT 1

The screenshot displays a GitHub repository interface. At the top, the repository name is **IBM-EPBL / IBM-Project-44610-1660725541**, marked as **Public**. The navigation bar includes links for **Code**, **Issues**, **Pull requests**, **Actions**, **Projects**, **Wiki**, **Security**, and **Insights**. The current view is the **main** branch, showing the directory **Project development phase /**. Below this, a commit by **pavithra20112** is shown, with the message **Add files via upload** and commit hash **06ed0d6**, dated **5 hours ago**. The commit details show a directory structure with four files: **SPRINT PLAN-1**, **SPRINT PLAN-2**, **SPRINT PLAN-3**, and **SPRINT PLAN-4**, each with the action **Add files via upload** and a timestamp of **5 hours ago**. The footer of the page includes the GitHub logo, copyright notice **© 2022 GitHub, Inc.**, and various links such as **Terms**, **Privacy**, **Security**, **Status**, **Docs**, **Contact GitHub**, **Pricing**, **API**, **Training**, **Blog**, and **About**.

## 7. CODING & SOLUTIONING

### 7.1 Feature 1

- IoT device
- IBM Watson Platform
- Node red
- Cloudant DB
- Web UI
- MIT App Inventor
- Python code

### 7.2 Feature 2

- Login
- Verification

- Ticket Booking
- Adding rating

## 8 TESTING AND RESULTS

## 8.1 Test Cases

## Test case 1

Testcases- Sprint 1 - Excel

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Paste Clipboards Font Alignment Number Styles Cells Editing

Calibri 11 A<sup>A</sup>

B I U % .00

Custom

Conditional Formatting Format as Table Cell Styles

Insert Delete Format

Share  
Share this document, and see who it's shared with.

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation Y/N	BUG ID	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				P.PAVITHRA
2	UI	Generating OTP	Generating the otp for further process		1.Generating of OTP number		user can register through phone	Working as expected	PASS				U.NITHITHA
3	Functional	OTP verification	Verify user otp using mail		1.Enter gmail id and enter password 2.click submit	Username: railvags password: admin	OTP verified is to be displayed	Working as expected	FAIL				B.SHATANA
4	Functional	Login page	Verify user is able to log into application with valid credentials		1.Enter into login page 2.Click on My Account dropdown button 3.Enter Invalid username/email in Email test box 4.Enter valid password in password test box 5.Click on login button	Username: railvags password: admin	Application should show 'Incorrect email or password' validation message.	Working as expected	FAIL				R.RUTHIRA
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	Username: railvags password: admin	A user can view about the available trains to enter start and destination details	Working as expected	PASS				P.PAVITHRA

Shopenzer Testcases | Testscenarios

Ready Accessibility: Investigate

Type here to search

29°C Haze 09:58 19-11-2022 ENG

## TEST CASE 2

File

Home

Insert

Page Layout

Formulas

Data

Review

View

Help

Tell me what you want to do

Cut

Copy

Paste

Format Painter

Clipboard

Calibri

11

A

A

B

I

U

Font

Wrap Text

Merge & Center

Alignment

General

Number

Conditional Formatting

Format as Table

Cell Styles

Styles

Insert

Delete

Format

Cells

AutoSum

Fill

Clear

Sort & Filter

Find & Select

Editing

E16

	A	B	C	D	E	F	G	H	I	J	K
1					Date	14-Nov-22					
2					Team ID	PNT2022TMID07171					
3					Project Name	Smart Solutions for Railways					
4					Maximum Marks	4 marks					
5	Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Expected Result	Actual Result	Status	Executed By	
6	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	
7	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	
8	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	
9	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	
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											

Shopenzer Testcases

Testscenarios

Ready Accessibility: Investigate

28°C Cloudy

ENG

11:37

14-11-2022

## TEST CASE3

Testcases Report sprint3 - Excel (Product Activation Failed)

File Home Insert Page Layout Formulas Data Review View Tell me what you want to do... Sign In Share

Paste Cut Copy Format Painter Clipboard Font Alignment Number Conditional Formatting Styles Cell Styles Insert Delete Format Cells Editing AutoSum Fill Clear Sort & Find & Filter Select

A1 X ✓ fx

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1				Date	11-Nov-22												
2				Team ID	PNT2022TMD46747												
3				Project Name	smart solutions for railways												
4				Maximum Marks	4 marks												
5	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By				
6	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				U.Nithitha				
7	UI	Ticket status	a user can see the status of my ticket Whether it's confirmed/pending/RAC		1.Amopen to the status of the tickets booked		known to the status of the tickets booked	Working as expected	pass				P.PAWITHRA				
8	Functional	Reminder notification	a user, I get reminders about my journey A day before my actual journey		1.user can get reminder notification		user can get reminder notification	Working as expected	pass				B.Shathana				
9	Functional	GPS tracking	user can track the train using GPS and can get information such as ETA, Current stop and delay		1.tracking train for getting information		tracking process through GPS	Working as expected	pass				R.Puthra				
10																	
11																	
12																	
13																	
14																	
15																	
16																	
17																	
18																	
19																	
20																	
21																	
22																	
23																	
24																	

Shopenzer Testcases | Testscenarios

Ready

## Test case 4

Testcases- Sprint 4 - Excel										
File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do										
Clipboard Font Alignment Number Styles Cells Editing										
G10										
A B C D E F G H I J K										
1					Date	14-Nov-22				
2					Team ID	PNT2022TMID07171				
3					Project Name	Smart Solutions for Railways				
4					Maximum Marks	4 marks				
5	Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Expected Result	Actual Result	Status	Executed By
6	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
7	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
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										

## 9. ADVANTAGES

- The passengers can use this application, while they are travelling alone to ensure their safety.
- It is easy to use.
- It has minimized error rate.

## 10. DISADVANTAGES

- Network issues may arise.

## II.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 industry struggles to meet the increasing demand for freight and passenger transportation due to lack of optimized use 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 smarter and 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 passenger data, 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 intelligent processing and timely retrieval would help backing up the security breaches. Here we've explored different issues of implementing smart computing in railway systems pertaining to reservation models besides pointing out some future scopes 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 purchases and timetable information implemented at stations and stops. With the rise of Industry, railway companies can now ensure that they are prepared to avoid the surprise of equipment downtime. Like above mentioned, the developed application of our project can lead the passenger who travel can travel safely without any fear.

## 12. FUTURE SCOPE

- This application is ensured for safety for the passengers while they are travelling alone as well as they travel with their family or 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 features regarding their safety.

# 13. APPENDIX

## Source Code

### LOGIN

```
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()

```

## START AND DESTINATION

```

# import module
import requests
from bs4 import BeautifulSoup
# user define function
# Scrape the data
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.raillyatri.in/booking/trains-between-
stations?from_code="+from_Station_code+"&from_name="+from_Station_name+"+JN+&journey_date="+Wed&src=tbs&t
o_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)

```

## TICKET BOOKING

```

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_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)
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

```

## SEATS BOOKING

```

def berth_type(s):
    if s>0 and s<73:
        if s % 8 == 1 or s % 8 == 4:
            print (s), "is lower berth"
        elif s % 8 == 2 or s % 8 == 5:
            print (s), "is middle berth"
        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), "is side upper berth"
        else:
            print (s), "invalid seat number"
# Driver code
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 module
import requests
from bs4 import BeautifulSoup
import pandas as pd
# convert into dataframe
df = pd.read_json(data[2])

```

```

# display this column of#
dataframe
print(df["mainEntity"][0]['name'])
print(df["mainEntity"][0]['acceptedAnswer']['text'])

```

## TICKET GENERATION

```

class Ticke:
    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:
        return False

def generate_ticket(self):
    if True:
        __ticket_id=self._source[0]+self._destination[0]+"0"+str(self.Counter)
        print( "Ticket id will be:",__ticket_id)
    else:
        return None
    return False

def get_ticket_id(self):
    return self.ticket_id

def get_passenger_name(self):
    return self._passenger_name

def get_source(self):
    if self._source=="Delhi":
        return self._source
    else:
        print("you have written invalid source option")
        return None

def get_destination(self):
    if self._destination=="Pune":
        return self._destination
    elif self._destination=="Mumbai":
        return self._destination
    elif self._destination=="Chennai":
        return self._destination
    elif self._destination=="Kolkata":
        return self._destination

```

else:

```
    return None
    import pandas as pd

    import numpy as np

    import matplotlib.pyplot as plt

    from PIL import Image,
    ImageDraw

    data_path = 'data.csv'

    data = pd.read_csv(data_path, names=['LATITUDE', 'LONGITUDE'], sep=',')
    gps_data = tuple(zip(data['LATITUDE'].values, data['LONGITUDE'].values))

    image = Image.open('map.png', 'r') # Load map image.
    img_points = []

    for d in gps_data:
        x1, y1 = scale_to_img(d, (image.size[0], image.size[1])) # Convert GPS coordinates to image coordinates.
        img_points.append((x1, y1))

    draw =
    ImageDraw.Draw(image)

    draw.line(img_points, fill=(255, 0, 0), width=2) # Draw converted records to the map image.

    image.save('resultMap.png')

    x_ticks = map(lambda x: round(x, 4), np.linspace(lon1, lon2, num=7))

    y_ticks = map(lambda x: round(x, 4), np.linspace(lat1, lat2, num=8))

    y_ticks = sorted(y_ticks, reverse=True) # y ticks must be reversed due to conversion to image
    coordinates.

    fig, axis1 = plt.subplots(figsize=(10, 10))

    axis1.imshow(plt.imread('resultMap.png')) # Load the image to matplotlib plot.
    axis1.set_xlabel('Longitude')
```

## OTP VERIFICATION

```
import os
import math
import random
import smtplib
```

```
digits = "0123456789"
OTP = ""
```

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

```
otp = OTP + "is you
OTP"message = otp
s=smtplib.SMTP('smtp.gmail.com', 587)s.starttls()
```

```
emailid = input("Enter your email: ")
s.login("YOUR Gmail 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")
```

















## **SPRINT 4**

### **6. CODING & SOLUTIONING**

#### **a. Feature 1**

- i. IoT device
- ii. IBM Watson Platform
- iii. Node red
- iv. Cloudant DB
- v. Web UI
- vi. MIT App Inventor
- vii. Python code

#### **b. Feature 2**

- i. Login
- ii. Verification
- iii. Ticket Booking
- iv. Adding rating

### **7. TESTING AND RESULTS**

#### **a. Test Cases**

Chandrika Chennupalli

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Clipboard Font Alignment Number Styles Cells Editing

Calibri 11 Wrap Text General

B I U Conditional Formatting Format as Table Insert Delete Format AutoSum Fill Sort & Find & Filter Clear

JS Executed By

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	<div> <div>Date</div> <div>14-Nov-22</div> </div> <div> <div>Team ID</div> <div>PH12022TMO07171</div> </div> <div> <div>Project Name</div> <div>Smart Solutions for Railways</div> </div> <div> <div>Maximum Marks</div> <div>4 marks</div> </div>	Registration form to be filled is to be displayed	Working as expected	PASS	VAISHNAVY
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 verified is to be displayed	Working as expected	FAIL	JESLINE
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	VAISHNAVY

Shopenzer Testcases Testscenarios

Ready Accessibility: Investigate

Count: 6 28°C Cloudy 11:29 14-11-2022

Testcases - Sprint 2 - Excel

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Clipboard Font Alignment Number Styles Cells Editing

Calibri 11 Wrap Text General

B I U Bold Italic Underline Font Color Background Color

Conditional Formatting Table Cell Styles

Insert Delete Format

AutoSum Fill Sort & Filter Find & Select

Clear

E16

	A	B	C	D	E	F	G	H	I	J	K
1					Date	14-Nov-22					
2					Team ID	PNT2022TMID07171					
3					Project Name	Smart Solutions for Railways					
4					Maximum Marks	4 marks					
5	Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Expected Result	Actual Result	Status	Executed By	
6	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	
7	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	
8	3	Functional	Payment	user, i can choose to pay through credit Card/debit card/UPL.		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	
9	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	Valshnavi	
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											

Shopenzer Testcases Testscenarios

Ready Accessibility: Investigate

28°C Cloudy 11:37 14-11-2022

### Test case 4

### Test case 4

## **8. ADVANTAGES**

- The passengers can use this application, while they are travelling alone to ensure their safety.
- It is easy to use.
- It has minimized error rate.

## **9. DISADVANTAGES**

- Network issues may arise.

## **10. 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 industry struggles to meet the increasing demand for freight and passenger transportation due to lack of optimized use 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 smarter and 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 passenger data, 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 intelligent processing and timely retrieval would help backing up the security breaches. Here we've explored different issues of implementing smart computing in railway systems pertaining to reservation models besides pointing out some future scopes 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 purchases and timetable information implemented at stations and stops. With the rise of Industry, railway companies can now ensure that they are prepared to avoid the surprise of equipment downtime. Like above mentioned, the developed application of our project can lead the passenger who travel can travel safely without any fear.

## **11. FUTURE SCOPE**

This application is ensured for safety for the passengers while they are travelling alone as well as they travel with their family or 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 features regarding their safety.

## 12. APPENDIX

### a. Source Code

#### LOGIN

```
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()
```

## **START AND DESTINATION**

```
# import module
import requests
from bs4 import BeautifulSoup

# user define function
# Scrape the data
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.raillyatri.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_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)

```

### **TICKET BOOKING**

```

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_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)
    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

```

### **SEATS BOOKING**

```

def berth_type(s):

    if s>0 and s<73:
        if s % 8 == 1 or s % 8 == 4:
            print (s), "is lower berth"
        elif s % 8 == 2 or s % 8 == 5:
            print (s), "is middle berth"
        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), "is side upper berth"
    else:
        print (s), "invalid seat number"

```

# Driver code

```

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 module
import requests
from bs4 import BeautifulSoup
import pandas as 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.raillyatri.in/live-train-status/"+train_name

# pass the url
# into getdata function
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
# dataframe
print(df["mainEntity"][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:
        return False

def generate_ticket(self ):
    if True:
        __ticket_id=self._source[0]+self._destination[0]+"0"+str(self.Counter)
        print( "Ticket id will be:",_ticket_id)
    else:
        return False
def get_ticket_id(self):
    return self.ticket_id
def get_passenger_name(self):
    return self._passenger_name
def get_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":
        return self._destination
    elif self._destination=="Mumbai":
        return self._destination
    elif self._destination=="Chennai":
        return self._destination
    elif self._destination=="Kolkata":
        return self._destination

```

```
else:  
    return None
```

### **OTP GENERATION**

```
import os  
import math  
import random  
import smtplib
```

```
digits = "0123456789"  
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("YOUR Gmail 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  
import math  
import random  
import smtplib
```

```
digits = "0123456789"  
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("YOUR Gmail 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")
```

## **b. GitHub**

### **GitHub**

#### **link:**

<https://github.com/IBM-EPBL/IBM-Project-19581-1659700929>