Project Report

Project Name: **SMART SOLUTIONS FOR RAILWAYS**
Team ID: **PNT2022TMID11066**

TEAM LEAD:

SAHITH AFRIDI. S

TEAM MEMBERS:

1. SANJAY. B
2. PALANIYAPPAN. A
3. DIWAKAR. M

1.1 Project Abstract :

People use Indian Railways to travel even on daily basis and if the railways are not secure and prone to accident then life of many A lot of people in India travel to other places using railways and some people are at risk. A lot of railway accidents occur at level crossing that is the point of intersection of road and railway track and the reason in most of the cases in human error. So, to avoid the accidents caused due to human failure this model is to make level crossing unmanned and smart than can reduce the chances of accidents manifold. In this proposed paper we have implemented ideas such as pre-crashing using RFID senser. This model automatically closes the gates of railway crossing when the train is arriving near the crossing before a safe interval of time so that there is no chance of human error. Also, our model keeps a track of the train passed from the particular crossing along with exact time of passing so that the data is maintained that too without human effort

1.2 Introduction :

Railways have to continually ensure that the rolling stock and infrastructure are in good condition, with high resilience against failures. There are number of challenges in planning of high-quality maintenance that has to be organized on efficient and cost effective manner. We wanted to be apart of our surrounding with some change and advancement so that it can bring the better life of the middle class and lower class people to travel in high secutity and advanced locomotions .the train is one and only most widely used transportion and not only for this they are used for goods transportion also .Indian railways are not able to facilate the customer properly due to crowded amount of people. Statistics show that the leading cause of death by injury in railways traffic accidents(two train collision each other). There are number of causes for which an accident can occur, some of them are; lack of training for driving or less experinessed, use of mobile phone while driving, unskilled drivers, driving while intoxicated, bad railway tack condition, overloading in tain and negligence traffic management. In this survey paper, we briefly review selected railway accidents detection techniques and propose a solution. Rear end crashes occur mainly due to obstracle and crack in tracks. According to recent statistics, a major percentage of train accident happen due to not proper survillance of railway track The existing system in semi automated railway accidents are occuring at frequently, consideration this in mind we want to bring some change and make it effective so that it becomes a complsory and law for pratice.

1.3 Objective :

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 maximise connectivity and allow passengers to continue their activities on smart devices while travelling.

IoT technologies help railways successfully manage passenger safety, operational efficiency, and the passenger experience

Smart sensors can be used to track important assets, manage passenger flow, and enable predictive maintenance

Connect people, sensors, trains and automated train systems with the highest security. Transform your communications and operations from departure to destination and beyond. Secure communications. Enhancing overall service. Lower operational cost IoT applications.

The Corporate aim of the Indian Railways is to commit itself to ensuring that all its activities are managed to the highest level of safety which is pragmatic and reasonably practicable to achieve

2. LITERATURE SURVEY

2.1 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

1.Juyeop Kim
chnical
problems of Internet of Things and their solutions.
IEEE Internet of Things Journal 2018
Main problems in railways

2
Payal Srivastava

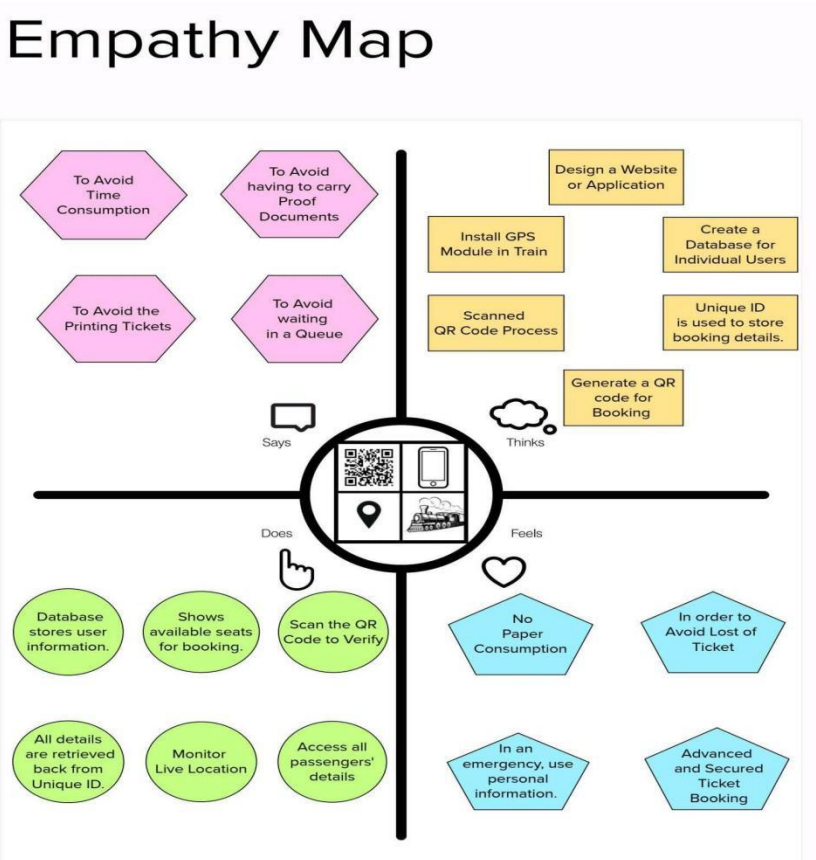
Construction and Building Materials Sañudo, Roberto, Marina
9th International Conference on Cloud
Computing, Data Science & Engineering 2019
Drainage in railways

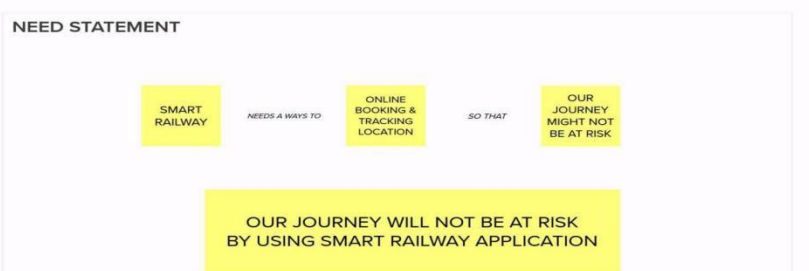
3
Ohyun Jo Published on "IEEE Internet of things journal 2018". Wide ranging developments in the realms of sensors, radio access, networks, and hardware/software platforms have been made possible by the Internet of Things' (IoT) rapidly expanding demand.

4
Y.Wang Published on "4th IET International Conference on Railway Condition Monitoring 2008". Train speed and density have been steadily rising over the past 20 years due to the increased demand for railroad services. As a result, stricter safety standards are required for the infrastructure, signalling, and control of railroads.
5
Bo Ai Published on "5G Key Technologies for Smart Railways 2022". Due to the rapid development of railways, particularly high-speed railways, railway communications have received considerable interest from both academia and industries (HSRs).The rail transportation sector needs to develop cutting-edge communication network architectures and critical technologies that guarantee high-quality transmissions for both passengers and railway operations and control systems in order to be in line with the goal of future smart rail communications.

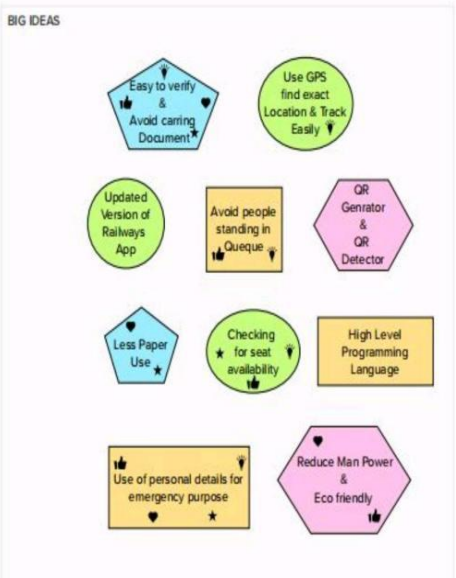
3. IDEATION & PROPOSED SOLUTION

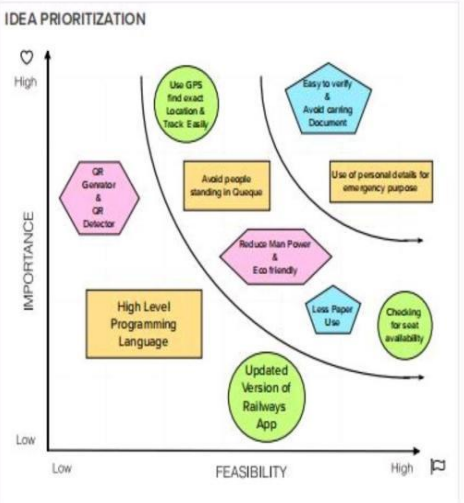
3.1 Empathy Map Canvas





3.2 Ideation & Brainstorming





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)**	• The goal of Smart Solutions for Railways is to reduce the amount of labour that users must do, as well as the use of paper, and to give real-time train location information. • People that travel quickly and have busy schedules need an online booking method. In recent years, there have been noticeably longer lines in front of the ticket booths in railroad stations. • The passenger experience is not sufficient or convenient with ticket reservations made at the counter. The passengers are trying to get tickets from ticket booths in a timely manner. They therefore choose to use online ticketing.
2.	**Idea / Solution description**	◆ The user can book tickets on a website, where they will also receive a QR code that they can provide to the ticket collector so that the ticket collector can scan it to retrieve the passenger's information. ◆ By installing a GPS module inside the train, the website also displays the train's real-time positions. The journey's location will be updated consistently on the website. ◆ The database will contain the user's booking information, which may be retrieved at any time.
3.	**Novelty / Uniqueness**	➢ The user will receive a QR code from the webpage, which will cut down on paperwork. ➢ All of the client booking information will be saved in the database with a special ID and may be retrieved when the ticket collector scans the QR Code.You may examine an interactive seat map as well.

4.	**Social Impact / Customer Satisfaction**	◆ There is no need to go to the station to book tickets, and the transaction process is made simple. One can update their online ticket booking and request a cancellation if their plans change. ◆ The consumer will receive notifications of all confirmations and cancellations through email and cell phone. ◆ In an emergency, we can quickly retrieve a doctor's passenger information.
5.	**Business Model (Revenue Model)**	• Using this application, the user can plan their trip, check the availability of a seat, examine an interactive seat map, and choose a seat that is most convenient for them. • Additionally, it makes it simple for your clients to plan trips and daily shuttles and it minimises the need to carry tickets. • Without this solution, customers would have to travel to the station to purchase tickets and would also need to carry their tickets with them to present to the ticket collector. • Customers could also view the train's current location.
6.	**Scalability of the Solution**	1. Printing Tickets is not required. 2. While handling counter tickets carefully is a must, text messages on a phone are more than sufficient. 3. By disregarding printouts, you are becoming more eco-friendly and helping to create a greener planet. 4. Tell TTR your name and that you are a passenger with a valid proof; there is no need to take your wallet out and display your ticket. 5. Booking an E-ticket instead of a counter ticket allows you to pay immediately from your bank account, making your work easier. Counter tickets required you to carry cash.

3.4 Problem Solution fit

PROBLEM-SOLUTION FIT

Smart Solutions For Railways

Define CS,fit into CC

CUSTOMER SEGMENT(S) CS

Who is your customer?

According to our problem statement, the customer prefers trains as a form of transportation

CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

The passengers are able to handle our application with ease. The App, which is available on all smart devices and can only be used with a network connection, causes some difficulty for older persons to handle or use.

AVAILABLE SOLUTIONS AS

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 & cons do these solutions have?

If the app crashes or an error occurs while using it, passengers can book their tickets through the website. Previously, passengers had to travel to their nearest rail station to resolve such issues, but our solution is now quite simple and convenient.

Explore AS, differentiate

Focus on J&P, tap into

JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers?

The client information is kept in a database that can be accessed by scanning the QR code, and an application that allows users to book tickets based on available seats should be created.

PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists ? What is the back story behind the need to do this job?

Our application often requires an internet connection, therefore when neither is present, the functionality of the QR code scanner or the ability to book tickets are interrupted.

BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

If there is an issue, customers can contact customer support, and they will receive a prompt response. Customers can also provide feedback on the app for it to be improved even more.

Focus on J&P, tap int C

Identify strong TR & EM

TRIGGERS TR

What triggers customers to act?

If a person needs to go a long way, he or she can use this app to reserve tickets, find out the location of the train using GPS, and share all of the information with family members. This causes the app to be installed and used.

EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?

After using this application, the customer feels comfortable buying tickets. Elderly people are free to choose their own pleasant seats. Verifying tickets using a QR code can save you a bunch of time.

YOUR SOLUTION SL

Our solution is to develop a mobile application that allows users to book tickets while also viewing available seats. Along with a database where customer information is saved, it also includes Smart QR verification.

CHANNELS OF BEHAVIOUR CH

ONLINE :

What kind of actions do customers take online?

Customers can submit feedback online in the setting option's support section.

OFFLINE :

What kind of actions do customers take offline?

Customers can send a message or email to the appropriate official immediately in offline mode.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1. Functional Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Requirements**	1.Mobile Phone 2.Internet 3.QR Code Scanner
FR-2	**User Registration**	1.Manual Registration 2.Registration through web page 3.Registration through Application
FR-3	**User Confirmation**	1.Confirmation via Phone 2.Confirmation via Email 3.Confirmation via OTP 4.Confirmation via SMS
FR-4	**Payment Options**	1.Net Banking/UPI 2.Credit/Debit/ATM Card 3.Digital Wallet
FR-5	**Application Installation**	1.Free Installation via Play Store and App store 2.Website available for free and always functional
FR-6	**Application Feedback**	1.Through Web page 2.Through Phone calls

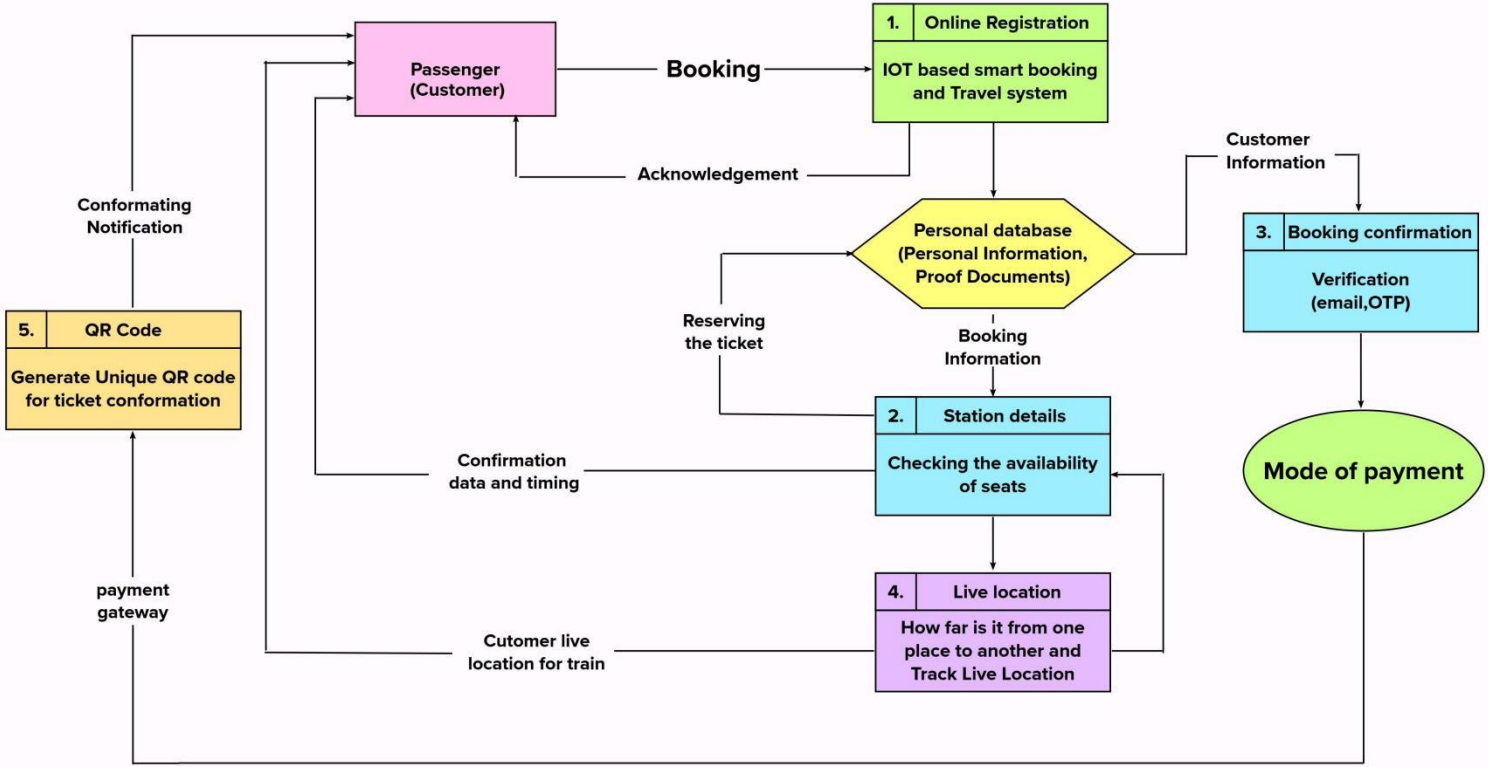
4.2 Non-Functional requirement

Non-functional Requirements:

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

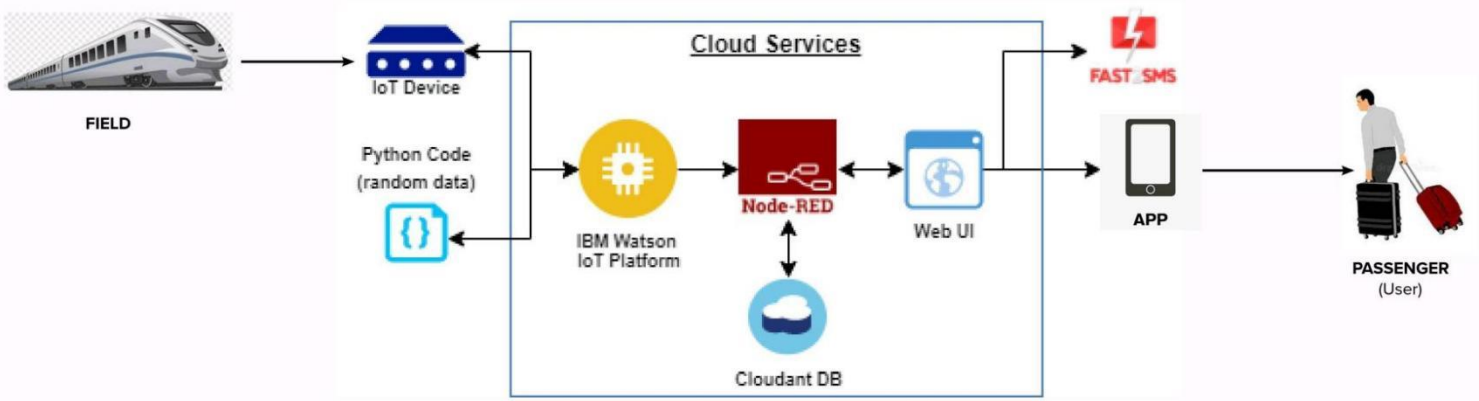
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	1.Have a Simple and Efficient application demo Video. 2.Easier to use. 3.If a Traveller has a Mobile Phone,they may easily Understand the procedure and make Reservations.
NFR-2	**Security**	1.Two-step authorization is required to secure the application. 2.Username and password will be assigned in accordance with user requirements.
NFR-3	**Reliability**	1.Periodic updates should be made to websites and applications. 2.If the booking process is interrupted by an internet outage, we offer an offline mode to complete the detail process.
NFR-4	**Performance**	1.The user interface of the web application must be user-friendly. 2.Moreover, payment should be quick and easy.
NFR-5	**Availability**	1.Provided with the proper train location. 2.Databases are maintained for passenger history. 3.Anytime and Anywhere for online ticket booking

Data Flow Diagrams



5. PROJECT DESIGN

Technology Architecture Of The Solution:



Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering user data	10	High	Sahith Afridi Palaniyappan
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application and can login to the application	10	High	Sanjay Diwakar
Sprint-2	Ticket Reservation and tracking	USN-3	As a user I can login and book Tickets.	15	High	Sahith Afridi Palaniyappan
Sprint-2		USN-4	As a user, I can track the exact location of the train	5	Medium	Sanjay Diwakar
Sprint-3	Connection with service provider	USN-5	As a User ,I can use services like payment gateways as a User by receiving OTP.	20	High	Sahith Afridi Palaniyappan Sanjay Diwakar
Sprint-4	QR code generation	USN-6	As a user , I am able to get a QR code for ticket verification	20	High	Sahith Afridi Palaniyappan Sanjay Diwakar

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

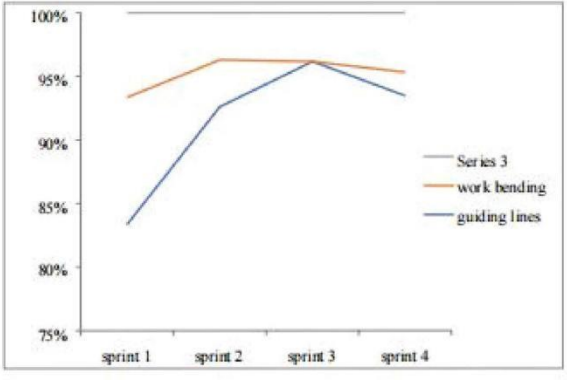
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

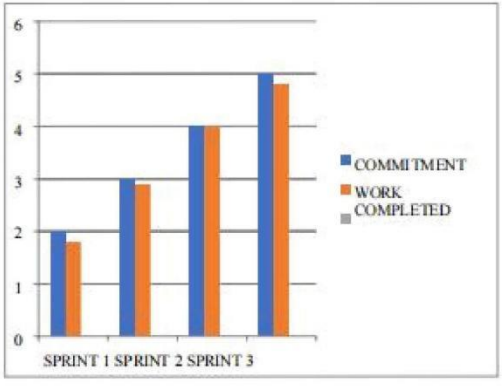
We have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). The team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{Sprint duration}}{\text{velocity}} = \frac{20}{6} = 3.33$$

Burndown Chart:



Velocity :



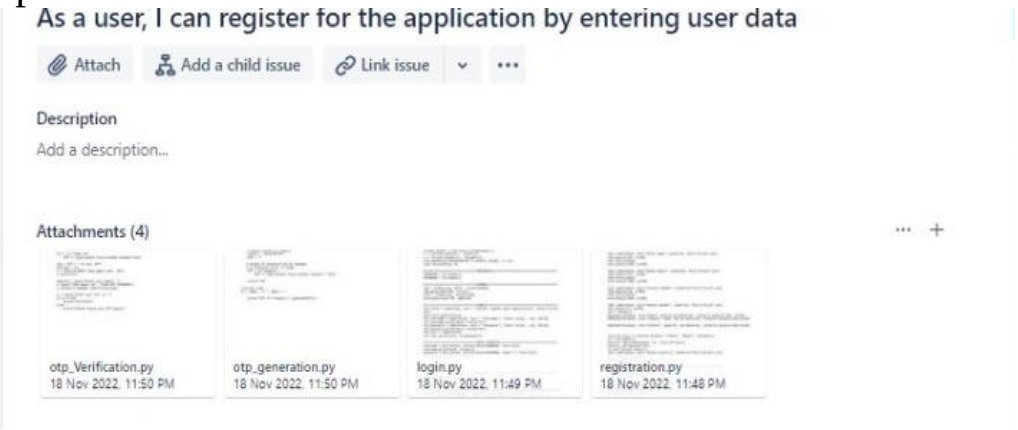
6. PROJECT PLANNING & SCHEDULING

TITLE	DESCRIPTION	DATE
Literature Survey& Information Gathering	A literature review is a comprehensive summary of previous researches on the topic. The literature review surveys scholarly articles, books, and other sources relevant to a particular area of research.	3 September 2022
Prepare Empathy Map	An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. It helps us to understand the customer's pain, gain and difficulties from their point of view.	10 September 2022
Ideation - Brainstorming	Brainstorming is a group problem-solving method that helped us to gather and organize various ideas and thoughts from team members.	17 September 2022
Define Problem statement	The Customer Problem Statement helps us to focus on what matters to create experiences people will love. A well-articulated customer problem statement allowed us to find the ideal solution for the challenges customers face.	19 September 2022

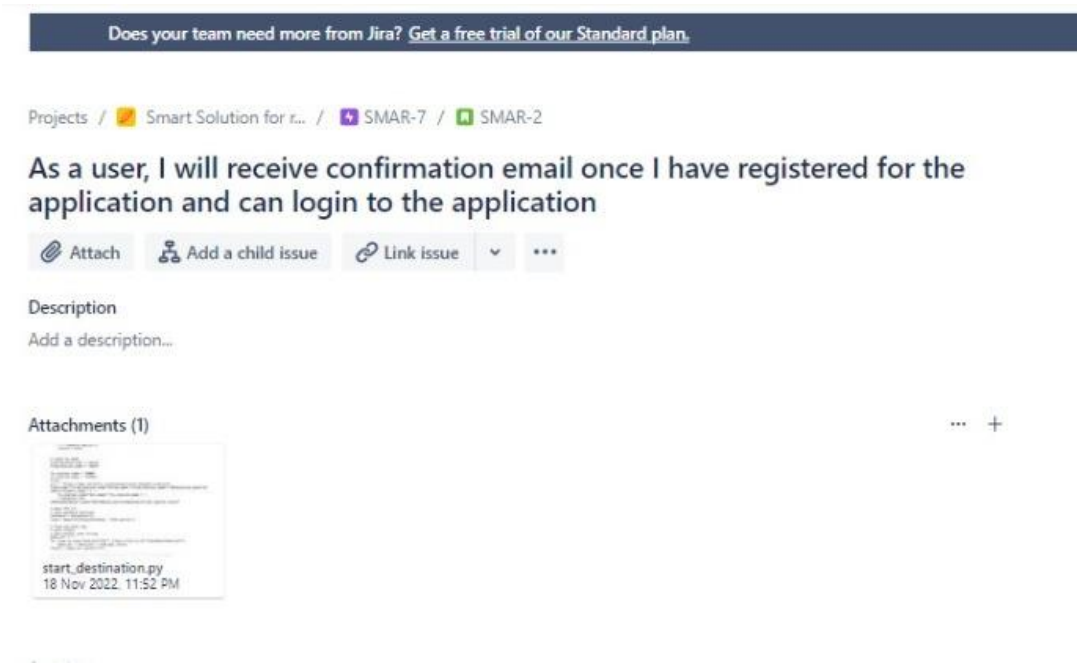
Data flow	A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.	11 October 2022
Sprint Delivery plan	Sprint Planning is an event in scrum that defines what can be delivered in the upcoming sprint and how that work will be achieved. It helps us to organize and complete the work effectively and efficiently.	22 October 2022
Prepare milestone and activity list	Helps us understand and evaluate our progress and accuracy so far.	23 October 2022
Project Development - Delivery of Sprint-1	Develop and submit the developed code by testing it.	In progress

6.1 Reports from JIRA

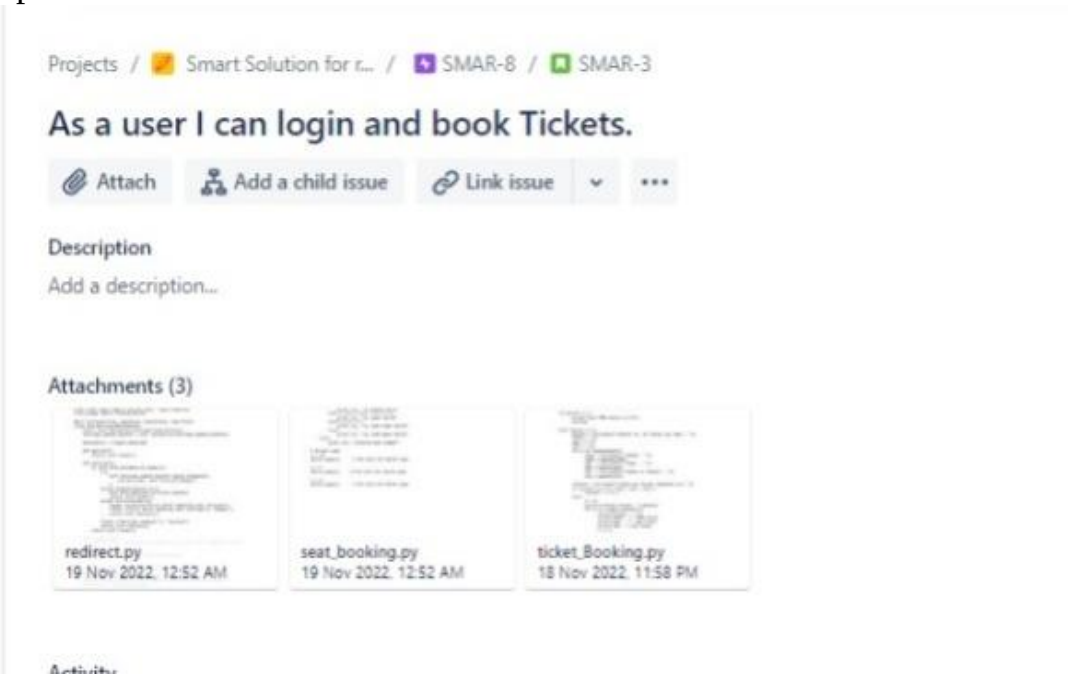
Sprint 1:



Sprint 2:



Sprint 3:



Sprint 4:

7. CODING & SOLUTIONING

7.1 Feature 1

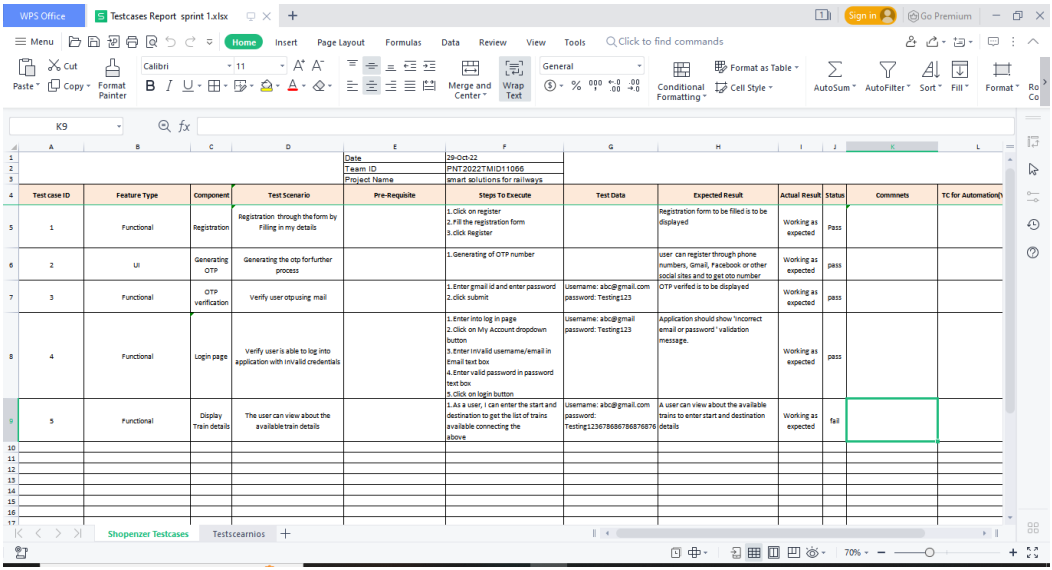
- IoT device
- IBM Watson Platform
- Node red
- Cloudant DB
- Web UI
- MIT App Inventor
- Python code

7.2 Feature 2

- Login
- Verification
- Ticket Booking
- Adding rating & Feedback

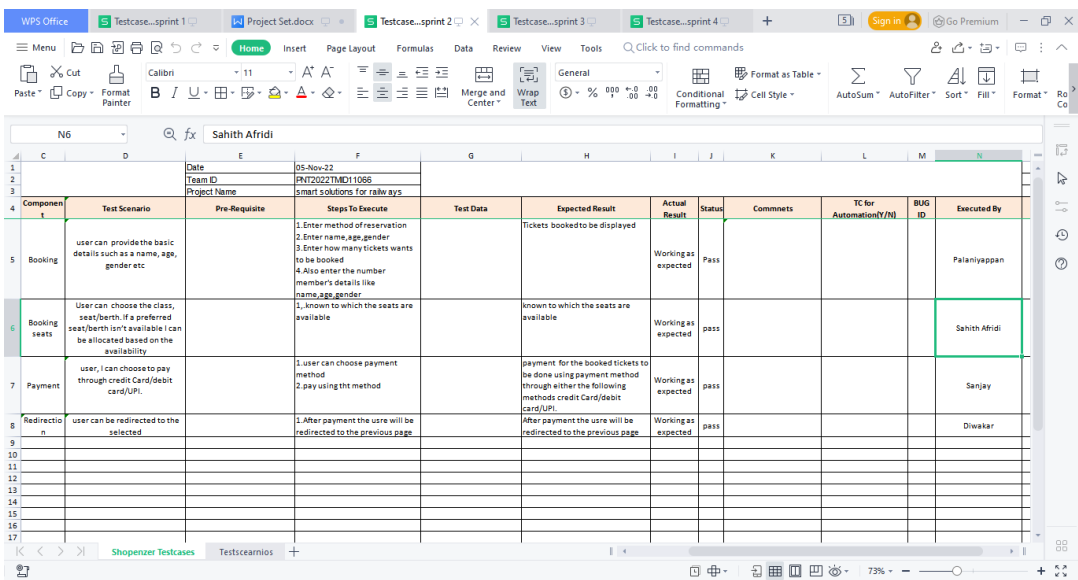
8. TESTING AND RESULTS

8.1 Test Cases 1



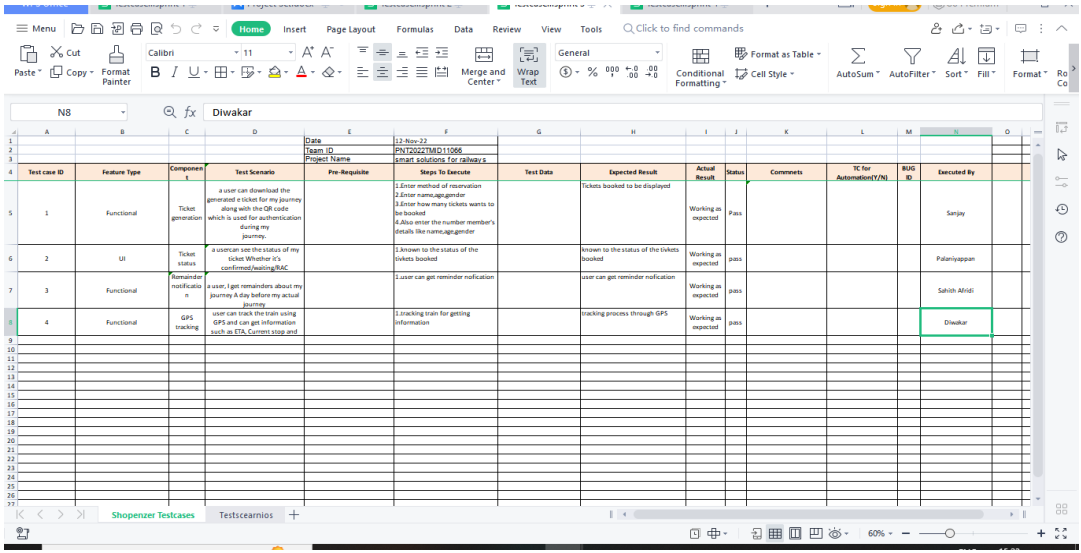
				Date	29-Oct-22					
				Team ID	PNT2022TMID11066					
				Project Name	smart solutions for railways					
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y
1	Functional	Registration	Registration 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		
2	UI	Generating OTP	Generating the otp forfurther process		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		
3	Functional	OTP verification	Verify user otpusing mail		1.Enter gmail id and enter password 2.click submit	Username: abc@gmail.com password: Testing123	OTP verified is to be displayed	Working as expected	pass		
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 5.Click on login button	Username: abc@gmail password: Testing123	Application should show 'Incorrect email or password ' validation message.	Working as expected	pass		
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: abc@gmail.com password: Testing123678686786876876	A user can view about the available trains to enter start and destination details	Working as expected	fail		

8.2 Test Cases 2



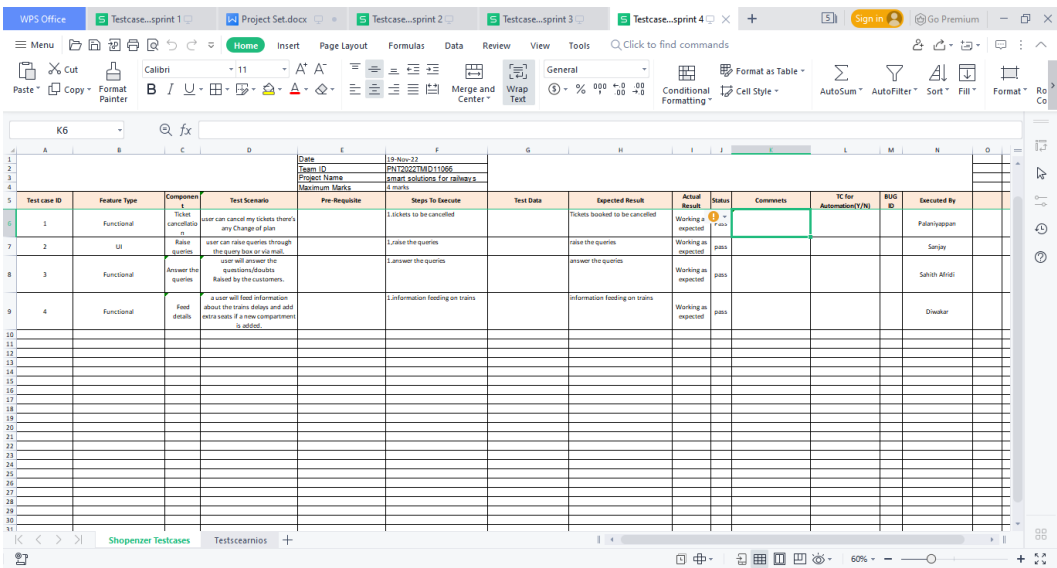
			Date	05-Nov-22							
			Team ID	PNT2022TMID11066							
			Project Name	smart solutions for railw ays							
Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
Booking	user can provide the basic details such as a name, age, gender etc		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 bookedto be displayed	Working as expected	Pass				Palaniyappan
Booking seats	User can choose the class, seat/berth. If a preferred seat/berth isn't available I can be allocated based on the availability		1,.known to which the seats are available		known to which the seats are available	Working as expected	pass				Sahith Afridi
Payment	user, i can choose to pay through credit Card/debit card/UPI.		1.user can choose payment method 2.pay using tht method		payment for the booked tickets to be done using payment method through either the following methods credit Card/debit card/UPI.	Working as expected	pass				Sanjay
Redirectio n	user can be redirected to the selected		1.After payment the usre will be redirected to the previous page		After payment the usre will be redirected to the previous page	Working as expected	pass				Diwakar

8.3 Test Cases 3



				Date	12-Nov-22									
				Team ID	PNT2022TMID11066									
				Project Name	smart solutions for railways									
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By	
1	Functional	Ticket generation	a user can download the generated e tickes 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				Sanjay	
2	UI	Ticket status	a usercan see the status of my ticket Whether it's confirmed/waiting/RAC		1.known to the status of the tivkets booked		known to the status of the tivkets booked	Working as expected	pass				Palaniyappan	
3	Functional	Remainder notification	a user, I get remainders about my journey A day before my actual journey		1.user can get reminder nofication		user can get reminder nofication	Working as expected	pass				Sahith Afridi	
4	Functional	GPS tracking	user can track the train using GPS and can get information such as ETA, Current stop and		1.tracking train for getting information		tracking process through GPS	Working as expected	pass				Diwakar	

8.4 Test Cases 4



				Date	19-Nov-22									
				Team ID	PNT2022TMID11066									
				Project Name	smart solutions for railways									
				Maximum Marks	4 marks									
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	Ticket cancellation	user can cancel my tickets there's any Change of plan		1.tickets to be cancelled		Tickets booked to be cancelled	Working as expected	Pass				Palaniyappan	
2	UI	Raise queries	user can raise queries through the query box or via mail.		1.raise the queries		raise the queries	Working as expected	pass				Sanjay	
3	Functional	Answer the queries	user will answer the questions/doubts Raised by the customers.		1.answer the queries		answer the queries	Working as expected	pass				Sahith Afridi	
4	Functional	Feed details	a user will feed information about the trains delays and add extra seats if a new compartment is added.		1.information feeding on trains		information feeding on trains	Working as expected	pass				Diwakar	

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.

11. 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 userfriendly websites, mobile applications for real-time information about vehicles in motion, and eticket 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
13.1 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.railyatri.in/booking/trains-
betweenstations?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-
clonespecial-rgd-to-ndls"

# url url = "https://www.railyatri.in/live-
trainstatus/"+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(s
elf.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")
```

13.2 GitHub
GitHub link:

https://github.com/IBM-EPBL/IBM-Project-5751-1658814251

Application Install through Link:

https://drive.google.com/file/d/1qlzj5tG7MV0lJlit2szkgTYgXA2QHYx0/view?usp=sharing