

RAILWAY RESERVATION SYSTEM

Submitted by

NITHEESH S [RA2111047010194]

THIVAKAR R[RA2111047010197]

Under the Guidance of

Dr. SUMATHY G

Assistant Professor, Department of Computational Intelligence

In partial satisfaction of the requirements for the degree of

**BACHELORS OF TECHNOLOGY
in
ARTIFICIAL INTELLIGENCE**



**SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR - 603203**

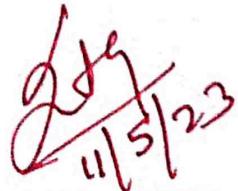
April 2023



SRM INSTITUTION OF SCIENCE AND TECHNOLOGY KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Course Project Report titled "**RAILWAY RESERVATION SYSTEM**" is the bonafide work done by **NITHEESH S [RA2111047010194], THIVAKAR R[RA2111047010197]** of II Year/ IV Sem B.Tech (AI) who carried out under my supervision for the course **18AIC207J - Database Management Systems for Artificial Intelligence**. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.



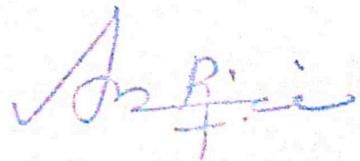
SIGNATURE

Faculty In-Charge

Dr.Sumathy G

Assistant Professor

Department of Computational Intelligence
SRM Institute of Science and Technology
Kattankulathur Campus, Chennai



HEAD OF THE DEPARTMENT

Dr. R Annie Uthra

Professor and Head ,

Department of Computational Intelligence,
SRM Institute of Science and Technology
Kattankulathur Campus, Chennai

TABLE OF CONTENTS

- **ABSTRACT**
- **INTRODUCTION**
- **SYSTEM CONFIGURATION**
- **HARDWARE AND SOFTWARE REQUIREMENTS**
- **FUNCTIONAL REQUIREMENTS(MODULES)**
- **ENTITIES AND ATTRIBUTES**
- **DIAGRAM DATA**
- **DATABASE DESIGN**
- **CODING SCREENSHOTS**
- **GRAPHICAL USER INTERFACE SCREENSHOTS**
- **BIBLIOGRAPHY**

ABSTRACT

The “**Railway Reservation System**” facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, etc.

The aim of this project is to design and develop a database maintaining the records of different trains, train status, and passengers. This project contains Introduction to the Railways reservation system .It is the computerized system of reserving the seats of train seats in advance.

It is mainly used for long routes. On-line reservation has made the process for the reservation of seats very much easier than ever before. In our country India, there are a number of counters for the reservation of the seats and one can easily make reservations and get tickets.

The record of train includes real time data of its number, name, source, destination, and days on which it is available, whereas record of train status includes dates for which tickets can be booked, total number of seats available, and number of seats already booked.

This project is done using Python and MySQL. Here this Python as a frontend and MySQL as a back end. These are making a best connectivity of front end and back end.Using Python GUI library named ‘TKINTER’, we can make a gui interface.SQL queries are used to retrieve data from a railway management database

INTRODUCTION

Railway Reservation System is developed to automate the railways reservation system. It includes modules required to successfully operate railways reversion process smoothly. It has a train master to add modified train information, train schedule to enter train journey details including all the station name, arrival time and departure time.

It includes automatic fare calculation as per the distance between two stations. Reservation module consists of an automatic seat number and coaches no allocation system. Daily schedule for updating of not conform seat and coach no. All masters like train master, Train schedule, reservation fees, cancellation fees, charges can be modified individually from front end and changes reflect in all modules immediately.

Few advantages are Passengers can book their tickets online or offline,Customer,User Details,Passengers Details,Train and Train Status,Station,Ticket Details will be stored in the database,The allocation of ticket information will also be stored in the database,Paperwork is reducing, Manager workload is reduced,If the selected Ticket is available, then the passenger will book the Ticket.

Therefore the proposed “Railway reservation system” has been designed to automate the process of railways for ticket reservation and back office activities. System can make the daily activities efficient and provide a fast response.

HARDWARE AND SOFTWARE REQUIREMENTS OF SYSTEM:

SOFTWARE SPECIFICATION

COMPONENTS	MINIMUM REQUIREMENTS
OPERATING SYSTEM	: Microsoft windows 7 and above
GRAPHICS	: Intel HD graphics
FRONT- END	: PYTHON VERSION 3.10
BACK-END	: MySQL VERSION 1.7.0

HARDWARE CONFIGURATION

COMPONENTS	MINIMUM REQUIREMENTS
PROCESSOR	Intel® Pentium® CPU P6200 @2-13GHz
RAM	2.00GB(1.74GB usable)
HARD DISK DRIVE	HARD DISK DRIVE 320GB
CACHE MEMORY	2MB
INTERNET :	Modem
BASIC COMPONENTS	Basic Pc or Laptop

FUNCTIONAL REQUIREMENTS(MODULES):

1. USER Details
2. PASSENGER Details
3. TRAIN Details
4. STATION Details
5. TICKET Details

ENTITY AND ATTRIBUTES:

Module 1: USER Details Entity: USER Table name Attributes: User_id, Password, First_name, Last_name, Gender, Age, Email, Aadhar_no, Mobile_no, City, State, Pin Code, Security_ques, Security_ans.

This module contains the login details of the user. It is required to login to access the application.

Module 2: PASSENGER Details Entity: PASSENGER Table name Attributes: Passenger_id, Name, Gender, Age, Pnr_no, Seat_no, Berth_np, Reservation_status

This module contains the passenger details of who booked a ticket with the help of a user.

Module 3: TRAIN Details Entity: TRAIN AND TRAIN STATUS Table name Attributes: Train_no, Train_name, Source, Destination, Arrival_time, Departure_time, Availability_of_seats, Train_no, A_seats1, A_seats2, A_seats3, B_seats1, B_seats2, B_seats3, W_Seats1, W_seats2, W_seats3.

This module contains the Train and its status details.

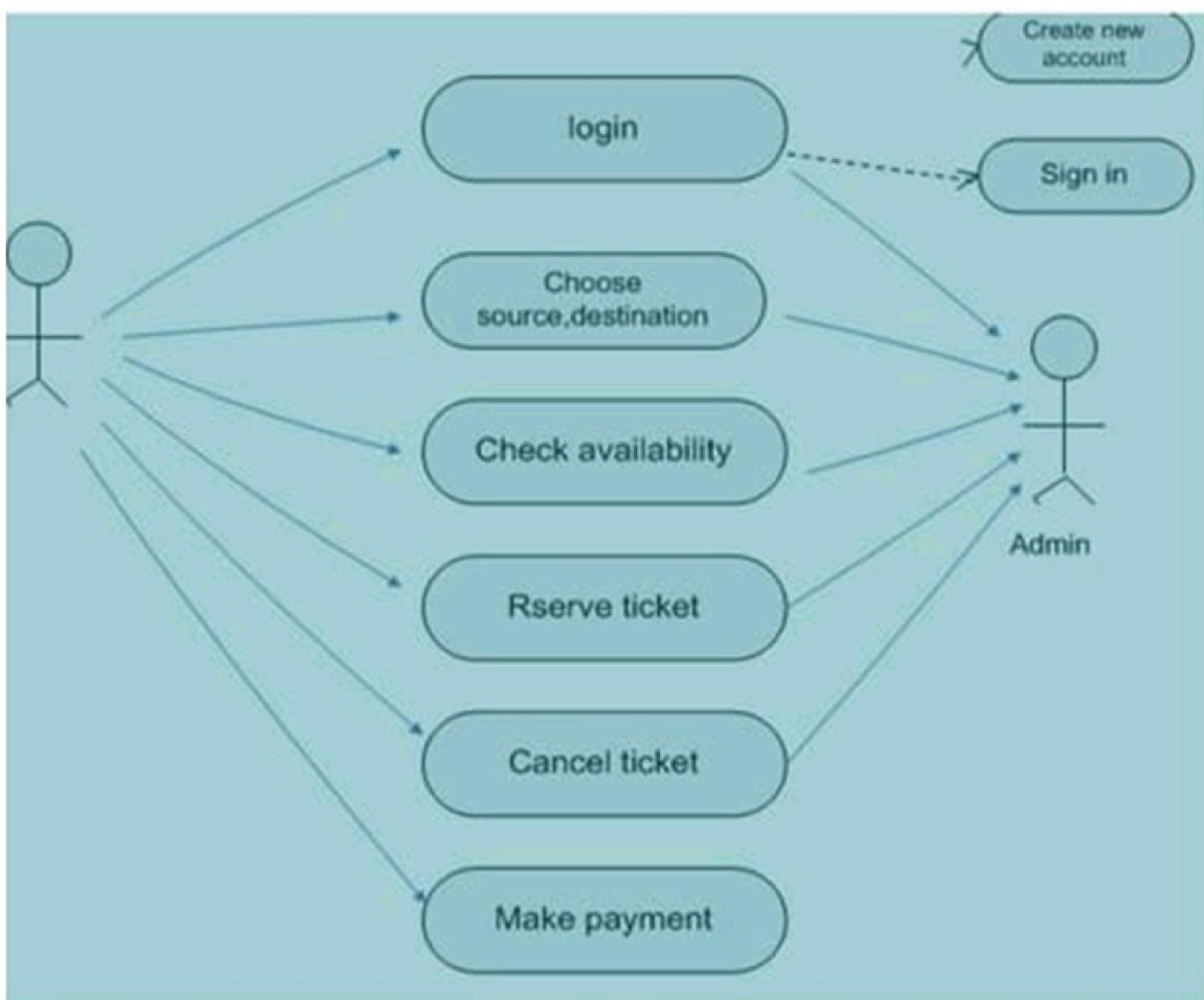
Module 4: STATION Details Entity: STATION Table name Attributes: Name, No, Train_no, Arrival_time, Hault.

This module contains the details of every train that passes through the station.

Module 5: TICKET Details Entity: TICKET Table name Attributes: Id, Train_no, Booked_user, Status, No_of_passengers.

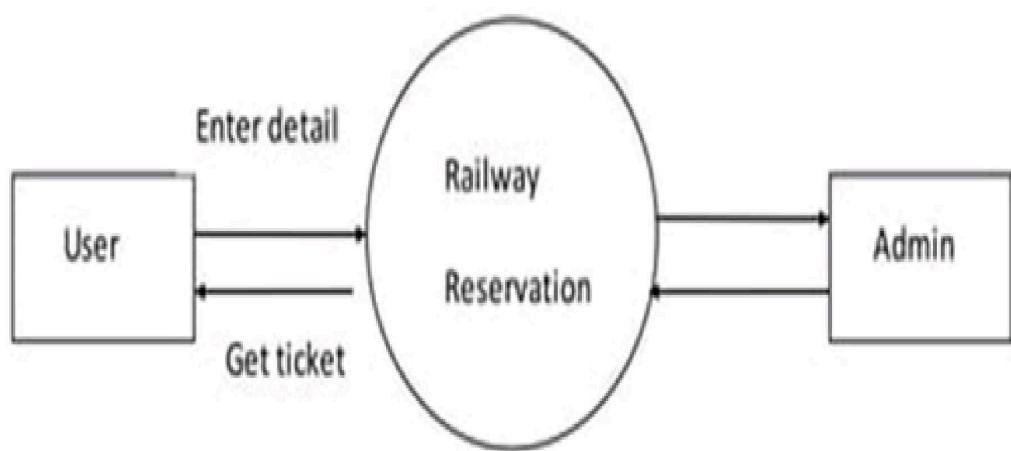
This module contains the details of the Ticket which has been booked by the user.

USE CASE DIAGRAM:

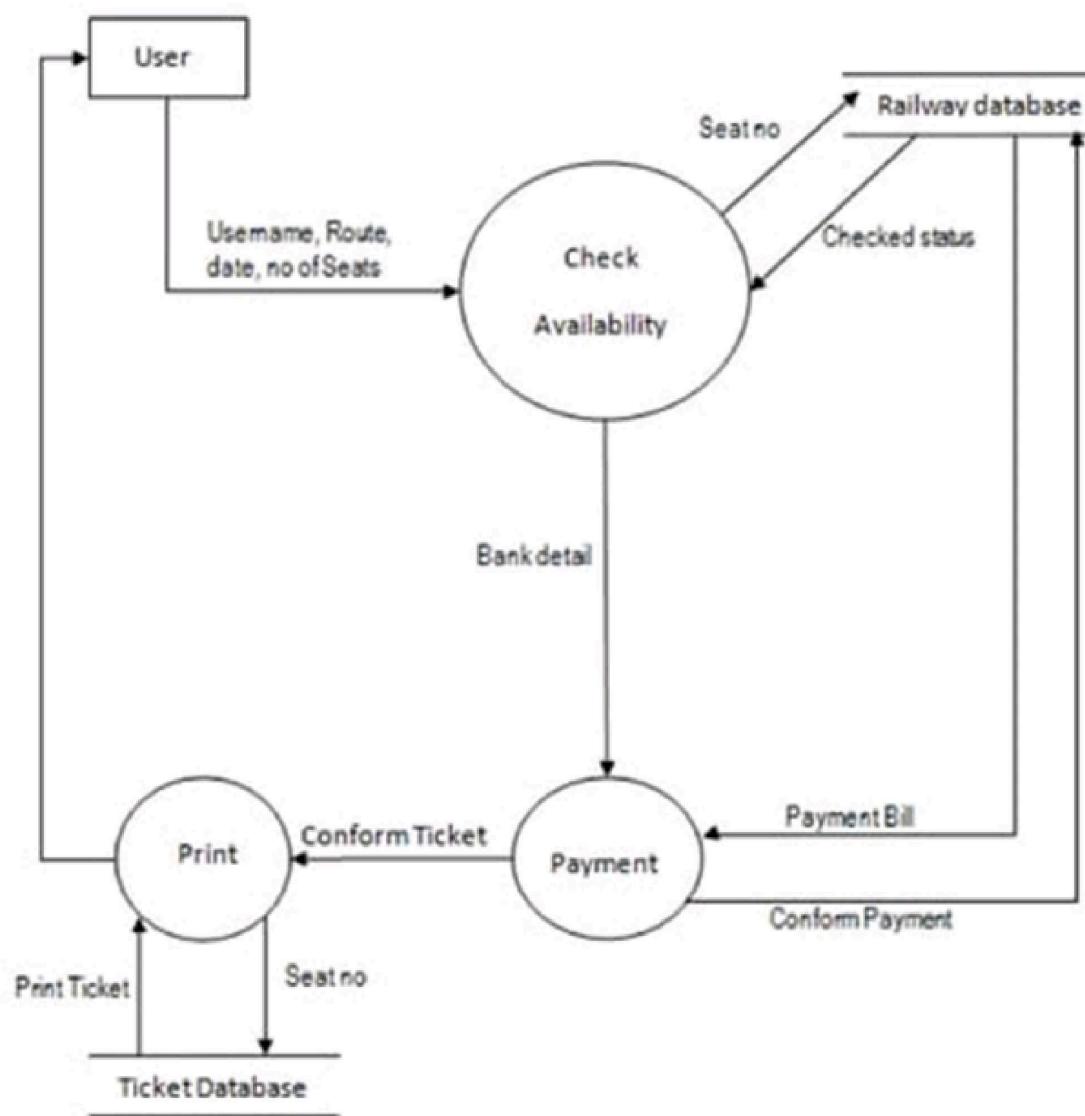


DATAFLOW DIAGRAM:

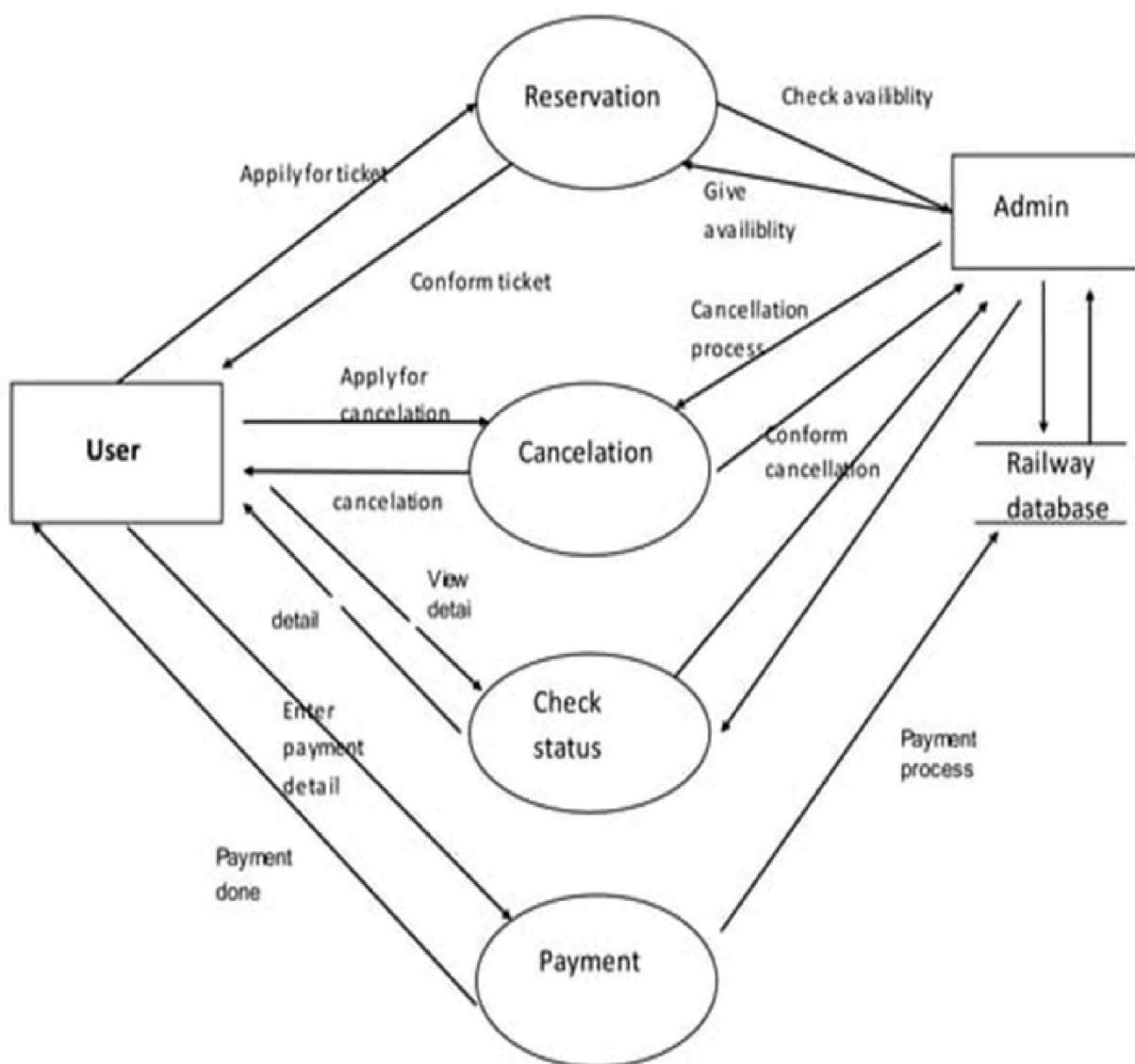
Level 0:



Level 1:

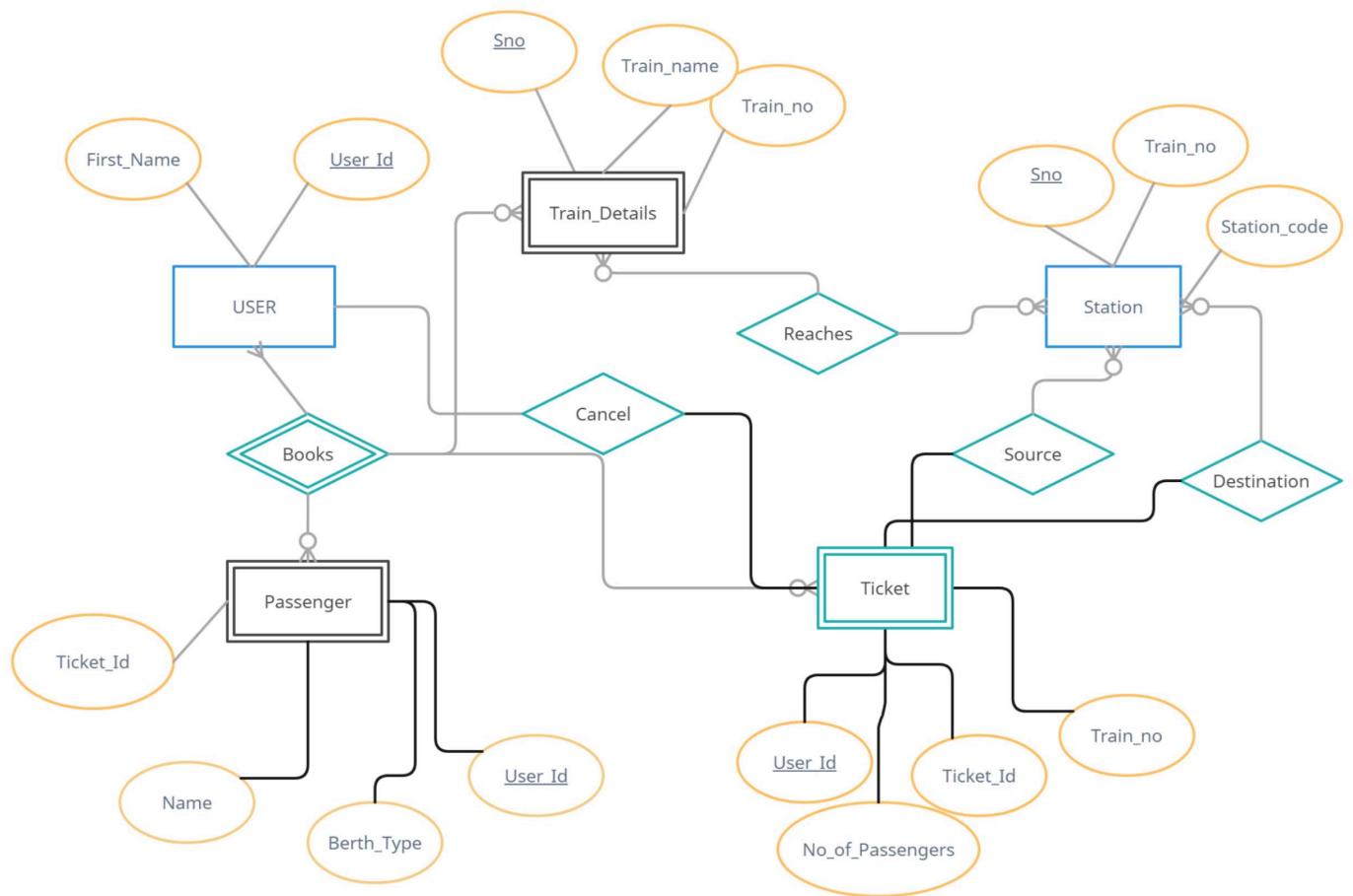


Level 2:



ENTITY RELATIONSHIP DIAGRAM:

An entity relationship diagram is a specialized graphic that illustrates the interrelationship between entities in a database. ER diagram often uses symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes. The below ER diagram shows us for the Railway Reservation System.



DATA DICTIONARY:

Data dictionary is the centralized collection of information about data. It stores the meaning and origin of data, its relationship with other data, data format for usage, etc. Data dictionary has rigorous definitions of all names to facilitate user and software designers.), Data dictionary is often referenced as meta-data (data about data) repository. It is created along with DFD (Data Flow Diagram) model of software program and is expected to be updated whenever DFD is changed or updated.

Table Index:

□ USER Details

Table name: USER

Field	Type	Null	Key	Default	Extra
user_id	varchar(30)	YES		NULL	
password	varchar(30)	YES		NULL	
first_name	char(30)	YES		NULL	
last_name	char(30)	YES		NULL	
gender	char(30)	YES		NULL	
age	int	YES		NULL	
email	varchar(30)	YES		NULL	
aadhar_no	int	YES		NULL	
mobile_no	int	YES		NULL	
city	char(30)	YES		NULL	
state	char(30)	YES		NULL	
pin_code	int	YES		NULL	
security_ques	varchar(30)	YES		NULL	
security_ans	varchar(30)	YES		NULL	

□ PASSENGER Details

Table Name: PASSENGER

Field	Type	Null	Key	Default	Extra
ticket_id	int	YES		NULL	
name	char(30)	YES		NULL	
gender	char(30)	YES		NULL	
age	int	YES		NULL	
berth_type	char(30)	YES		NULL	
user_id	varchar(60)	YES		NULL	

□ TRAIN Details

Table Name: TRAIN_DETAILS

Field	Type	Null	Key	Default	Extra
train_no	int	YES		NULL	
train_name	char(30)	YES		NULL	
source	char(30)	YES		NULL	
destination	char(30)	YES		NULL	
arrival_time	time	YES		NULL	
departure_time	time	YES		NULL	
seat_avail	varchar(30)	YES		NULL	
sno	int	NO	PRI	NULL	auto_increment

□ STATION Details

Table Name: STATION

Field	Type	Null	Key	Default	Extra
name	char(30)	YES		NULL	
train_no	int	YES		NULL	
arrival	time	YES		NULL	
departure	time	YES		NULL	
halt	time	YES		NULL	
station_code	char(30)	YES		NULL	
sno	int	NO	PRI	NULL	auto_increment

□ TICKET Details

Table Name: TICKET

Field	Type	Null	Key	Default	Extra
ticket_id	int	YES		NULL	
train_no	int	YES		NULL	
no_of_passengers	int	YES		NULL	
status	char(30)	YES		NULL	
user_id	varchar(60)	YES		NULL	

CODE SCREENSHOTS:

SIGN UP PAGE:

```
from tkinter import *
from tkinter import messagebox
from PIL import ImageTk
import pymysql

def clear():
    userIdEntry.delete(0,END)
    passwordEntry.delete(0,END)
    firstnameEntry.delete(0,END)
    lastnameEntry.delete(0,END)
    genderEntry.delete(0,END)
    ageEntry.delete(0,END)
    emailEntry.delete(0,END)
    aadharnoEntry.delete(0,END)
    mobilenoEntry.delete(0,END)
    cityEntry.delete(0,END)
    stateEntry.delete(0,END)
    pincodeEntry.delete(0,END)
    securityquesEntry.delete(0,END)
    securityanswerEntry.delete(0,END)
    check.set(0)

def connect_database():
    if userIdEntry.get() == "" or passwordEntry.get() == "" or
    firstnameEntry.get() == "" or lastnameEntry.get() == "" or
    genderEntry.get() == "" or ageEntry.get() == "" or emailEntry.get() == "" or
    aadharnoEntry.get() == "" or mobilenoEntry.get() == "" or cityEntry.get() == ""
    or stateEntry.get() == "" or pincodeEntry.get() == "" or
    securityquesEntry.get() == "" or securityanswerEntry.get() == "":
        messagebox.showerror('Error','All Fields Are Mandatory!')
    elif check.get() == 0:
        messagebox.showerror('Error','Please Accept Terms & Conditions')
    else:
        try:
            con=pymysql.connect(host='localhost',user='root',password='root')
```

```

mycursor=con.cursor()
except:
    messagebox.showerror('Error',"Database Connectivity
Issue,Please try again")
return
try:
    query='use railway_reservation_system'
    mycursor.execute(query)
except:
    mycursor.execute('use railway_resrvation_system')
query='select * from user where user_id=%s'
mycursor.execute(query,(userIdEntry.get()))

row=mycursor.fetchone()
if row !=None:
    messagebox.showerror('Error','User already exist')

else:
    query='insert      into
user(user_id,password,first_name,last_name,gender,age,email,aadhar_no
,mobile_no,city,state,pin_code,security_ques,security_ans)
values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)'

mycursor.execute(query,(userIdEntry.get(),passwordEntry.get(),firstname
Entry.get(),lastnameEntry.get(),

genderEntry.get(),ageEntry.get(),emailEntry.get(),aadharEntry.get(),m
obilenoEntry.get(),

cityEntry.get(),stateEntry.get(),pincodeEntry.get(),securityquesEntry.get(
),securityanswerEntry.get()))
    con.commit()
    con.close()
    messagebox.showinfo('Success','Registration is successful')
    clear()
    signup_window.destroy()
    import signin

```

```
def login_page():
    signup_window.destroy()
    import signin

signup_window=Tk()
signup_window.title('Railway Reservation System - Signup Page')
signup_window.resizable(False,False)
background=ImageTk.PhotoImage(file='bg.jpg')

bgLabel=Label(signup_window,image=background)
bgLabel.grid()

frame=Frame(signup_window,bg='white')
frame.place(x=540,y=50)

heading=Label(frame,text="CREATE AN ACCOUNT",font=('Microsoft Yahei UI Light',18,'bold'),bg='white',fg='firebrick1')
heading.grid(row=0,column=0,padx=10,pady=10)

userIdLabel=Label(frame,text="User ID",font=('Microsoft Yahei UI Light',10,'bold'),bg='white',
                  fg='firebrick1')
userIdLabel.grid(row=1,column=0,sticky='w',padx=5)
userIdEntry=Entry(frame,width=25,font=('Microsoft Yahei UI Light',10,'bold'),
                  fg='white',bg='firebrick1')

userIdEntry.grid(row=2,column=0,sticky='w',padx=5)

passwordLabel=Label(frame,text="Password",font=('Microsoft Yahei UI Light',10,'bold'),bg='white',
                     fg='firebrick1')
passwordLabel.grid(row=3,column=0,sticky='w',padx=5)
passwordEntry=Entry(frame,width=25,font=('Microsoft Yahei UI
```

```
Light',10,'bold'),  
    fg='white',bg='firebrick1')  
  
passwordEntry.grid(row=4,column=0,sticky='w',padx=5)  
  
firstnameLabel=Label(frame,text="First Name",font=('Microsoft Yahei  
UI Light',10,'bold'),bg='white',  
    fg='firebrick1')  
firstnameLabel.grid(row=5,column=0,sticky='w',padx=5)  
firstnameEntry=Entry(frame,width=10,font=('Microsoft Yahei UI  
Light',10,'bold'),  
    fg='white',bg='firebrick1')  
  
firstnameEntry.grid(row=6,column=0,sticky='w',padx=5)  
  
lastnameLabel=Label(frame,text="Last Name",font=('Microsoft Yahei  
UI Light',10,'bold'),bg='white',  
    fg='firebrick1')  
lastnameLabel.grid(row=5,column=1,padx=5)  
lastnameEntry=Entry(frame,width=10,font=('Microsoft Yahei UI  
Light',10,'bold'),  
    fg='white',bg='firebrick1')  
  
lastnameEntry.grid(row=6,column=1,padx=5)  
  
genderLabel=Label(frame,text="Gender",font=('Microsoft Yahei UI  
Light',10,'bold'),bg='white',  
    fg='firebrick1')  
genderLabel.grid(row=9,column=0,sticky='w',padx=5)  
genderEntry=Entry(frame,width=10,font=('Microsoft Yahei UI  
Light',10,'bold'),  
    fg='white',bg='firebrick1')  
  
genderEntry.grid(row=10,column=0,sticky='w',padx=5)
```

```
ageLabel=Label(frame,text="Age",font=('Microsoft Yahei UI Light',10,'bold'),bg='white',  
             fg='firebrick1')  
ageLabel.grid(row=9,column=1,sticky='w',padx=5)  
ageEntry=Entry(frame,width=10,font=('Microsoft Yahei UI Light',10,'bold'),  
             fg='white',bg='firebrick1')
```

```
ageEntry.grid(row=10,column=1,sticky='w',padx=25)
```

```
emailLabel=Label(frame,text="Email",font=('Microsoft Yahei UI Light',10,'bold'),bg='white',  
             fg='firebrick1')  
emailLabel.grid(row=11,column=0,sticky='w',padx=5)  
emailEntry=Entry(frame,width=25,font=('Microsoft Yahei UI Light',10,'bold'),  
             fg='white',bg='firebrick1')
```

```
emailEntry.grid(row=12,column=0,sticky='w',padx=5)
```

```
aadharnoLabel=Label(frame,text="Aadhar Number",font=('Microsoft Yahei UI Light',10,'bold'),bg='white',  
             fg='firebrick1')  
aadharnoLabel.grid(row=13,column=0,sticky='w',padx=5)  
aadharnoEntry=Entry(frame,width=10,font=('Microsoft Yahei UI Light',10,'bold'),  
             fg='white',bg='firebrick1')
```

```
aadharnoEntry.grid(row=14,column=0,sticky='w',padx=5)
```

```
mobilenoLabel=Label(frame,text="Mobile Number",font=('Microsoft
```

```
Yahei UI Light',10,'bold'),bg='white',
    fg='firebrick1')
mobilenoLabel.grid(row=13,column=1,padx=5)
mobilenoEntry=Entry(frame,width=10,font=('Microsoft Yahei UI
Light',10,'bold'),
    fg='white',bg='firebrick1')
```

```
mobilenoEntry.grid(row=14,column=1,padx=5)
```

```
cityLabel=Label(frame,text="City",font=('Microsoft Yahei UI
Light',10,'bold'),bg='white',
    fg='firebrick1')
cityLabel.grid(row=15,column=0,sticky='w',padx=5)
cityEntry=Entry(frame,width=10,font=('Microsoft Yahei UI
Light',10,'bold'),
    fg='white',bg='firebrick1')
```

```
cityEntry.grid(row=16,column=0,sticky='w',padx=5)
```

```
stateLabel=Label(frame,text="State",font=('Microsoft Yahei UI
Light',10,'bold'),bg='white',
    fg='firebrick1')
stateLabel.grid(row=15,column=0,padx=5)
stateEntry=Entry(frame,width=10,font=('Microsoft Yahei UI
Light',10,'bold'),
    fg='white',bg='firebrick1')
```

```
stateEntry.grid(row=16,column=0,padx=5)
```

```
pincodeLabel=Label(frame,text="Pincode",font=('Microsoft Yahei UI
Light',10,'bold'),bg='white',
    fg='firebrick1')
pincodeLabel.grid(row=15,column=1,padx=5)
pincodeEntry=Entry(frame,width=10,font=('Microsoft Yahei UI
Light',10,'bold'),
    fg='white',bg='firebrick1')
```

```
pincodeEntry.grid(row=16,column=1,padx=5)

securityquesLabel=Label(frame,text="Security  
Question",font=('Microsoft Yahei UI Light',10,'bold'),bg='white',  
fg='firebrick1')  
securityquesLabel.grid(row=17,column=0,sticky='w',padx=5)  
securityquesEntry=Entry(frame,width=10,font=('Microsoft Yahei UI  
Light',10,'bold'),  
fg='white',bg='firebrick1')

securityquesEntry.grid(row=18,column=0,sticky='w',padx=5)

securityanswerLabel=Label(frame,text="Security  
Answer",font=('Microsoft Yahei UI Light',10,'bold'),bg='white',  
fg='firebrick1')  
securityanswerLabel.grid(row=17,column=1,padx=5)  
securityanswerEntry=Entry(frame,width=10,font=('Microsoft Yahei UI  
Light',10,'bold'),  
fg='white',bg='firebrick1')

securityanswerEntry.grid(row=18,column=1,padx=5)  
check=IntVar()  
termsandconditions=Checkbutton(frame,text='I agree to the Terms &  
Conditions',font=('Microsoft Yahei UI Light',9,'bold'),  
fg='firebrick1',bg='white',activebackground='white',activeforeground='fir  
ebrick1',cursor='hand2',variable=check)  
termsandconditions.grid(row=19,column=0,pady=5)

signupButton=Button(frame,text="Signup",font=('Open  
Sans',16,'bold'),bd=0,  
bg='firebrick1',fg='white',activebackground='firebrick1',activeforeground
```

```

='white',width=17,
    command=connect_database)
signupButton.grid(row=20,column=0)

alreadyaccount=Label(frame,text='Already           have           an
account?',font=('Open Sans',9,'bold'),
                     bg='white',fg='firebrick1')
alreadyaccount.grid(row=21,column=0,sticky='w',padx=25)

loginButton=Button(frame,text='Log     in',font=('Open   Sans',9,'bold
underline'),
                     bg='white',fg='blue',bd=0,cursor='hand2',activebackground='white',
                     activeforeground='blue',command=login_page)
loginButton.grid(row=21,column=1)

signup_window.mainloop()

```

SIGN IN PAGE:

```

from tkinter import *
from tkinter import Tk,Canvas
import PIL.Image
from PIL import ImageTk,Image
from tkinter import messagebox
import pymysql

```

#Functionality Part

```
def forget_pass():
```

```
def change_password():
    if user_entry.get() == "" or newpass_entry.get() == "" or
    confirmpass_entry.get() == "":
        messagebox.showerror('Error', 'All fields are
    required', parent=window)
    elif newpass_entry.get() != confirmpass_entry.get():
        messagebox.showerror('Error', 'Password
    mismatched', parent=window)
    else:
        con=pymysql.connect(host='localhost',user='root',password='root',databa
        se='railway_reservation_system')
        mycursor=con.cursor()
        query='select * from user where user_id=%s'
        mycursor.execute(query,(user_entry.get()))
        row=mycursor.fetchone()
        if row==None:
            messagebox.showerror('Error', 'Incorrect
        Username', parent=window)
        else:
            query='update data set password=%s where username=%s'
            mycursor.execute(query,(newpass_entry.get(),user_entry.get()))
            con.commit()
            con.close()
            messagebox.showerror('Success', 'Password is reset, please
        login with new password', parent=window)
            window.destroy()
```

```
window = Toplevel()
window.title('Change Password')
```

```
bgPic=ImageTk.PhotoImage(file='background.jpg')
bglabel=Label(window,image=bgPic)
bglabel.grid()
```

```
heading_label=Label(window,text="RESET
PASSWORD",font=('arial',18,'bold'),
                    bg='white',fg='magenta2')
heading_label.place(x=480,y=60)
```

```
userLabel=Label(window,text='Username',font=('arial',12,'bold'),bg='white',fg='orchid1')
userLabel.place(x=470,y=130)
```

```
user_entry=Entry(window,width=25,fg='magenta2',font=('arial',11,'bold'),bd=0)
user_entry.place(x=470,y=160)
```

```
Frame(window,width=250,height=2,bg='orchid1').place(x=470,y=180)
```

```
passwordLabel=Label(window,text='New
Password',font=('arial',12,'bold'),bg='white',fg='orchid1')
passwordLabel.place(x=470,y=210)
```

```
newpass_entry=Entry(window,width=25,fg='magenta2',font=('arial',11,'bold'),bd=0)
newpass_entry.place(x=470,y=240)
```

```
Frame(window,width=250,height=2,bg='orchid1').place(x=470,y=260)
```

```
confirmpassLabel=Label(window,text='Confirm
Password',font=('arial',12,'bold'),bg='white',fg='orchid1')
confirmpassLabel.place(x=470,y=290)
```

```
confirmpass_entry=Entry(window,width=25,fg='magenta2',font=('arial',11,'bold'),bd=0)
confirmpass_entry.place(x=470,y=320)
```

```
Frame(window,width=250,height=2,bg='orchid1').place(x=470,y=340)
```

```
submitButton=Button(window,text='Submit',bd=0,bg='magenta2',fg='white',font=('Open Sans',16,'bold'),width=19,
cursor='hand2',activebackground='magenta2',activeforeground='white',command=change_password)
submitButton.place(x=470,y=390)
```

```
window.mainloop()

def login_user():
    if usernameEntry.get() == "" or passwordEntry.get() == "":
        messagebox.showerror('Error', 'All fields are required')
    else:
        try:
            con=pymysql.connect(host='localhost',user='root',password='root')
            mycursor=con.cursor()
        except:
            messagebox.showerror('Error', 'Connection is not established try again')
        return
        query='use railway_reservation_system'
        mycursor.execute(query)
        query='select * from user where user_id=%s and password=%s'
        mycursor.execute(query,(usernameEntry.get(),passwordEntry.get()))
        row=mycursor.fetchone()
        if row==None:
            messagebox.showerror('Error', 'Invalid username or password')
        else:
            messagebox.showinfo('Welcome', 'Login Successful')
            login_window.destroy()
            import main

def signup_page():
    login_window.destroy()
    import signup

def hide():
    openeye.config(file='closeye.png')
    passwordEntry.config(show='*')
    eyeButton.config(command=show)

def show():
    openeye.config(file='openeye.png')
    passwordEntry.config(show="")
    eyeButton.config(command=hide)
```

```
def user_enter(event):
    if usernameEntry.get()=='Username':
        usernameEntry.delete(0,END)

def password_enter(event):
    if passwordEntry.get()=='Password':
        passwordEntry.delete(0,END)

#GUI Part
login_window=Tk()
login_window.geometry('990x660+50+50')
login_window.resizable(False,False)
login_window.title('Railway Reservation System - Login Page')
image=PIL.Image.open('C:\\Users\\dbzna\\Desktop\\bg.jpg')
bgImage=ImageTk.PhotoImage(image)
bgLabel=Label(login_window,image=bgImage )
bgLabel.place(x=0,y=0)

heading=Label(login_window,text="USER LOGIN",font=('Microsoft Yahei UI Light',23,'bold'),bg='white',fg='firebrick1')
heading.place(x=605,y=120)

usernameEntry=Entry(login_window,width=25,font=('Microsoft Yahei UI Light',11,'bold'),bd=0,fg='firebrick1')
usernameEntry.place(x=580,y=200)
usernameEntry.insert(0,'Username')

usernameEntry.bind('<FocusIn>',user_enter)

frame1=Frame(login_window,width=250,height=2,bg='firebrick1')
frame1.place(x=580,y=222)

passwordEntry=Entry(login_window,width=25,font=('Microsoft Yahei UI Light',11,'bold'),bd=0,fg='firebrick1')
passwordEntry.place(x=580,y=260)
passwordEntry.insert(0,'Password')
```

```
passwordEntry.bind('<FocusIn>',password_enter)
frame2=Frame(login_window,width=250,height=2, bg='firebrick1')
frame2.place(x=580,y=282)

openeye=PhotoImage(file='openeye.png')
eyeButton=Button(login_window,image=openeye,bd=0, bg='white',activebackground='white'
                 ,cursor='hand2',command=hide)
eyeButton.place(x=800,y=255)

forgetButton=Button(login_window,text='Forgot
Password?',bd=0, bg='white',activebackground='white'
                   ,cursor='hand2',font=('Microsoft Yahei UI
Light',9,'bold'),fg='firebrick1',
                   activeforeground='firebrick1',command=forget_pass)
forgetButton.place(x=715,y=295)

loginButton=Button(login_window,text='Login',font=('Open
Sans',16,'bold'),fg='white',
                   bg='firebrick1',activeforeground='white',activebackground='firebrick1',c
ursor='hand2',bd=0,width=19,command=login_user)
loginButton.place(x=578,y=350)

orLabel=Label(login_window,text='----- OR
-----',font=("Open Sans",16),fg='firebrick1',bg='white')
orLabel.place(x=583,y=400)

signupLabel=Label(login_window,text='Dont have a
account?',font=("Open Sans",9,'bold'),fg='firebrick1',bg='white')
signupLabel.place(x=590,y=500)

newaccountButton=Button(login_window,text='Create New
One',font=('Open Sans',9,'bold underline'),fg='blue',
                   bg='white',activeforeground='blue',activebackground='white',cursor='han
d2',bd=0,
                   command=signup_page)
newaccountButton.place(x=727,y=500)

login_window.mainloop()
```

MAIN PAGE:

```
from tkinter import *
from tkinter import messagebox
from PIL import ImageTk,Image,ImageFilter
import pymysql
import pyautogui
import tkinter as tk
from tkinter import ttk

mydb=pymysql.connect(host='localhost',user='root',password='root')
mycursor=mydb.cursor()

def on_enter(event):
    if trainnoEntry.get()=='Train_no':
        trainnoEntry.delete(0,END)

def find_train():
    try:
        con=pymysql.connect(host='localhost',user='root',password='root')
        mycursor=con.cursor()
    except:
        messagebox.showerror('Error',"Database Connectivity Issue,Please
try again")
        return
    try:
        query='use railway_reservation_system'
        mycursor.execute(query)
    except:
        mycursor.execute('use railway_resrvation_system')
    query='select * from station where train_no=%s'
    data=mycursor.fetchall()
    print(data)
    con.close()
```

```

return data

def trch_page():
    main_window.destroy()
    import signin

def main_page():
    train_page.destory()

def show_trains():
    trainwindow = Toplevel()
    trainwindow.title('Train Schedule')

bgPic=ImageTk.PhotoImage(file='irctc_indian_railway_engine.jpg')
bglabel=Label(trainwindow,image=bgPic)
bglabel.grid()
#frame =Label(main_window,width=85,height=27,bg='white')
#frame.place(x=470,y=120)
BackButton=Button(trainwindow,text='Back',font=('Soul
Papa',12,'bold'),fg='white',
bg='dodgerblue4',activeforeground='white',activebackground='dodgerblu
e4',cursor='hand2',bd=0,width=15,
)
BackButton.place(x=0,y=0)
my_str=tk.StringVar()

l2=tk.Label(trainwindow,textvariable=my_str,width=50,fg='blue',font=1
0)
l2.place(x=100,y=100)

try:
    con=pymysql.connect(host='localhost',user='root',password='root')
    mycursor=con.cursor()
except:
    messagebox.showerror('Error',"Database Connectivity Issue,Please
try again")
    return
query='use railway_reservation_system'
mycursor.execute(query)
query='select * from train_details'
mycursor.execute(query)
results=mycursor.fetchall()
l1=[q for q in results.keys()]

```

```

r_set=list(results)

trv=ttk.Treeview(trainwindow,selectmode='browse',columns=11,show='headings',
height=len(r_set))
trv.place(x=100,y=100)
for i in 11:
    trv.column(i,anchor='c',width=70)
    trv.heading(i,text=i)
for row in r_set:
    trv.insert("","end",iid=row[0],text=row[0],values=list(row))
trainwindow.mainloop()

```

```

def train_page():

    trwindow = Toplevel()
    trwindow.title('Train Schedule')

    bgPic=ImageTk.PhotoImage(file='ircfc_indian_railway_engine.jpg')
    bglab=Label(trwindow,image=bgPic)
    bglab.grid()
    #frame =Label(main_window,width=85,height=27,bg='white')
    #frame.place(x=470,y=120)
    BackButton=Button(trwindow,text='Back',font=('Soul
    Papa',12,'bold'),fg='white',
    bg='dodgerblue4',activeforeground='white',activebackground='dodgerblu
    e4',cursor='hand2',bd=0,width=15,
    )
    BackButton.place(x=0,y=0)
    my_str=tk.StringVar()

    l2=tk.Label(trwindow,textvariable=my_str,width=50,fg='blue',font=10)
    l2.place(x=100,y=100)

try:
    con=pymysql.connect(host='localhost',user='root',password='root')
    mycursor=con.cursor()
except:
    messagebox.showerror('Error',"Database Connectivity Issue,Please
try again")
    return

```

```
query='use railway_reservation_system'
mycursor.execute(query)
query='select * from station where train_no=%s'
hey=mycursor.execute(query,(trainnoEntry.get()))
student=list(hey.fetchall())
my_str.set(student)
l=[r for r in hey.keys()]
print(l,student)
```

```
main_window=Tk()
main_window.title('Railway Reservation System')
main_window.geometry('1080x660+50+50')
main_window.iconbitmap("Citycons.ico")
main_window.resizable(False,False)
my_pic=Image.open('150168.jpg')
resized=my_pic.resize((1080,500),Image.LANCZOS)
#bgImage=ImageTk.PhotoImage(file='indian-railway-irctc-88.jpg')
new_pic=ImageTk.PhotoImage(resized)
bgLabel=Label(main_window,image=new_pic, bg='white')
heading=Label(main_window, text="Railway Reservation
System", font=('Soul Papa',50,'bold'), fg='dodgerblue4')
heading.place(x=100,y=0)
myicon=Image.open("Citycons.ico")
resized1=myicon.resize((100,80),Image.LANCZOS)
icon=ImageTk.PhotoImage(resized1)
bgicon=Label(main_window, image=icon)
bgicon.place(x=0,y=0)
```

```
#frame1=Frame(main_window, width=600, height=400, bg='white')
label8 = Label(main_window, width=85, height=27, bg='white')
label8.place(x=470, y=120)
#frame1.place(x=470, y=120)
```

```
LogoutButton=Button(main_window, text='Logout', font=('Soul
Papa',12,'bold'), fg='white',
bg='dodgerblue4', activeforeground='white', activebackground='dodgerblu
e4', cursor='hand2', bd=0, width=15,
command=trch_page)
LogoutButton.place(x=0, y=120)
```

```
bookticketButton=Button(main_window,text='Book a Ticket',font=('Soul  
Papa',12,'bold'),fg='white',  
  
bg='dodgerblue4',activeforeground='white',activebackground='dodgerblu  
e4',cursor='hand2',bd=0,width=15)  
bookticketButton.place(x=475,y=125)
```

```
cancelticketButton=Button(main_window,text='Cancel a  
Ticket',font=('Soul Papa',12,'bold'),fg='white',  
  
bg='dodgerblue4',activeforeground='white',activebackground='dodgerblu  
e4',cursor='hand2',bd=0,width=15)  
cancelticketButton.place(x=680,y=125)
```

```
viewticketButton=Button(main_window,text='View Ticket',font=('Soul  
Papa',12,'bold'),fg='white',  
  
bg='dodgerblue4',activeforeground='white',activebackground='dodgerblu  
e4',cursor='hand2',bd=0,width=15)  
viewticketButton.place(x=900,y=125)
```

```
ticketenqButton=Button(main_window,text='Ticket Enquiry',font=('Soul  
Papa',12,'bold'),fg='white',  
  
bg='dodgerblue4',activeforeground='white',activebackground='dodgerblu  
e4',cursor='hand2',bd=0,width=15)  
ticketenqButton.place(x=475,y=185)
```

```
availButton=Button(main_window,text='Available Trains',font=('Soul  
Papa',12,'bold'),fg='white',  
  
bg='dodgerblue4',activeforeground='white',activebackground='dodgerblu  
e4',cursor='hand2',bd=0,width=15,  
command=show_trains)  
availButton.place(x=680,y=185)
```

```
trch=Label(main_window,text="Train Schedule ",font=('Soul  
Papa',20,'bold'),bg='white',fg='dodgerblue4')  
trch.place(x=680,y=225)
```

```
trainnoEntry=Entry(main_window,width=15,font=('Soul
Papa',16,'bold'),fg='dodgerblue4')
trainnoEntry.place(x=600,y=300)
trainnoEntry.insert(0,'Train_no')
trainnoEntry.bind('<FocusIn>',on_enter)

submitbutton=Button(main_window,text='Submit',font=('Soul
Papa',12,'bold'),fg='white',
bg='orange',activeforeground='white',activebackground='darkorange',cur
sor='hand2',bd=0,
width=15,command=train_page)
submitbutton.place(x=900,y=300)
bgLabel.pack(pady=100)
main_window.mainloop()
```

SQL SCREENSHOTS:

```

mysql> desc passenger;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| ticket_id | int | YES | | NULL |
| name | char(30) | YES | | NULL |
| gender | char(30) | YES | | NULL |
| age | int | YES | | NULL |
| berth_type | char(30) | YES | | NULL |
| user_id | varchar(60) | YES | | NULL |
+-----+-----+-----+-----+-----+
6 rows in set (0.23 sec)

mysql> desc station;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| name | char(30) | YES | | NULL | |
| train_no | int | YES | | NULL |
| arrival | time | YES | | NULL |
| departure | time | YES | | NULL |
| hault | time | YES | | NULL |
| station_code | char(30) | YES | | NULL |
| sno | int | NO | PRI | NULL | auto_increment |
+-----+-----+-----+-----+-----+
7 rows in set (0.03 sec)

mysql> desc user;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| user_id | varchar(30) | YES | | NULL |
| password | varchar(30) | YES | | NULL |
| first_name | char(30) | YES | | NULL |
| last_name | char(30) | YES | | NULL |
| gender | char(30) | YES | | NULL |
| age | int | YES | | NULL |
| email | varchar(30) | YES | | NULL |
| aadhar_no | int | YES | | NULL |
| mobile_no | int | YES | | NULL |
| city | char(30) | YES | | NULL |
| state | char(30) | YES | | NULL |
| pin_code | int | YES | | NULL |
| security_ques | varchar(30) | YES | | NULL |
| security_ans | varchar(30) | YES | | NULL |
+-----+-----+-----+-----+-----+
14 rows in set (0.00 sec)

mysql> desc ticket;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| ticket_id | int | YES | | NULL |
| train_no | int | YES | | NULL |
| no_of_passengers | int | YES | | NULL |
| status | char(30) | YES | | NULL |
| user_id | varchar(60) | YES | | NULL |
+-----+-----+-----+-----+-----+
5 rows in set (0.07 sec)

```

```
mysql> desc train_details;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| train_no | int | YES | NO | NULL |          |
| train_name | char(30) | YES | NO | NULL |          |
| source | char(30) | YES | NO | NULL |          |
| destination | char(30) | YES | NO | NULL |          |
| arrival_time | time | YES | NO | NULL |          |
| departure_time | time | YES | NO | NULL |          |
| seat_avail | varchar(30) | YES | NO | NULL |          |
| sno | int | NO | PRI | NULL | auto_increment |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

```
mysql> select * from station;
+-----+-----+-----+-----+-----+-----+-----+-----+
| name | train_no | arrival | departure | hault | station_code | sno |
+-----+-----+-----+-----+-----+-----+-----+-----+
| chennai_egmore | 16823 | 00:00:00 | 20:10:00 | 00:00:00 | ms | 1 |
| tambaram | 16823 | 20:38:00 | 20:40:00 | 00:02:00 | tbm | 2 |
| jolarpettai | 12640 | 17:28:00 | 17:30:00 | 00:02:00 | jtj | 3 |
| chengalpattu | 16823 | 21:08:00 | 21:10:00 | 00:02:00 | cgl | 4 |
| ambur | 12640 | 17:58:00 | 18:00:00 | 00:02:00 | ab | 5 |
| madurantakam | 16823 | 21:28:00 | 21:30:00 | 00:02:00 | mmk | 6 |
| melmaruvathur | 16823 | 21:38:00 | 21:40:00 | 00:02:00 | mlmr | 7 |
| tindivanam | 16823 | 22:03:00 | 22:05:00 | 00:02:00 | tmv | 8 |
| villupuram_jn | 16823 | 22:48:00 | 22:50:00 | 00:02:00 | vm | 9 |
| vridhachalam_jn | 16823 | 23:30:00 | 23:32:00 | 00:02:00 | vri | 10 |
| tiruchirappalli | 16823 | 04:30:00 | 01:35:00 | 00:05:00 | tpj | 11 |
| dindigul_jn | 16823 | 02:52:00 | 02:55:00 | 00:03:00 | dg | 12 |
| madurai_jn | 16823 | 04:30:00 | 03:55:00 | 00:05:00 | mdu | 13 |
| vaniyambadi | 12640 | 17:43:00 | 17:45:00 | 00:02:00 | vn | 14 |
| tirumangalam | 16823 | 04:14:00 | 04:15:00 | 00:01:00 | tmq | 15 |
| katpadi_jn | 12640 | 18:43:00 | 18:45:00 | 00:02:00 | kpd | 16 |
| walajah_road_jn | 12640 | 19:03:00 | 19:05:00 | 00:02:00 | wjr | 17 |
| sholinghur | 12640 | 19:18:00 | 19:20:00 | 00:02:00 | shu | 18 |
| virudunagar_jn | 16823 | 04:33:00 | 04:35:00 | 00:02:00 | vpt | 19 |
| satur | 16823 | 04:57:00 | 04:58:00 | 00:01:00 | srt | 20 |
| kovilpatti | 16823 | 05:18:00 | 05:20:00 | 00:02:00 | cvp | 21 |
| arakkonam | 12640 | 19:43:00 | 19:45:00 | 00:02:00 | ajj | 22 |
| tirunelveli | 16823 | 04:30:00 | 06:50:00 | 00:05:00 | ten | 23 |
| nanguneri | 16823 | 07:17:00 | 07:18:00 | 00:01:00 | nnn | 24 |
| perambur | 12640 | 20:33:00 | 20:35:00 | 00:02:00 | per | 25 |
| mgr_chennai_central | 12640 | 21:10:00 | 00:00:00 | 00:00:00 | mas | 26 |
| tambaram | 12759 | 00:00:00 | 17:10:00 | 00:00:00 | tbm | 27 |
| chennai_egmore | 12759 | 17:40:00 | 17:10:00 | 00:05:00 | ms | 28 |
| sullurupeta | 12759 | 19:24:00 | 19:25:00 | 00:01:00 | spe | 29 |
| nayadupeta | 12759 | 19:48:00 | 19:50:00 | 00:02:00 | nyp | 30 |
| gudur_jn | 12759 | 20:23:00 | 20:25:00 | 00:02:00 | gdr | 31 |
| nellore | 12759 | 20:53:00 | 20:55:00 | 00:02:00 | nlr | 32 |
| kavali | 12759 | 21:33:00 | 21:35:00 | 00:02:00 | kvz | 33 |
| ongole | 12759 | 22:18:00 | 22:20:00 | 00:02:00 | ogl | 34 |
| valliyur | 16823 | 07:29:00 | 07:30:00 | 00:01:00 | vly | 35 |
| aralvayamozhi | 16823 | 07:49:00 | 07:50:00 | 00:01:00 | aay | 36 |
| nagercoil_town | 16823 | 08:37:00 | 08:40:00 | 00:03:00 | njt | 37 |
| eraniel | 16823 | 08:59:00 | 09:00:00 | 00:01:00 | erl | 38 |
| kulitthurai | 16823 | 09:15:00 | 09:18:00 | 00:03:00 | kzt | 39 |
| parassala | 16823 | 09:28:00 | 09:29:00 | 00:01:00 | pasa | 40 |
| neyyattinkara | 16823 | 09:41:00 | 09:42:00 | 00:01:00 | nyy | 41 |
| trivandrum_central | 16823 | 04:30:00 | 10:15:00 | 00:05:00 | tvc | 42 |
| chirala | 12759 | 23:02:00 | 23:05:00 | 00:02:00 | clx | 43 |
| varkala | 16823 | 10:54:00 | 10:55:00 | 00:01:00 | vak | 44 |
| tenali_jn | 12759 | 23:55:00 | 23:57:00 | 00:02:00 | tel | 45 |
| paravur | 16823 | 11:06:00 | 11:07:00 | 00:01:00 | pvu | 46 |
| kollam_jn | 16823 | 11:45:00 | 00:00:00 | 00:00:00 | qln | 47 |
| vijayawada_jn | 12759 | 00:45:00 | 00:55:00 | 00:10:00 | bza | 48 |
| kollam_jn | 16824 | 00:00:00 | 00:00:00 | 15:40:00 | qln | 49 |
| khammam | 12759 | 02:18:00 | 02:20:00 | 00:02:00 | kmt | 50 |
| paravur | 16824 | 15:53:00 | 15:54:00 | 00:01:00 | pvv | 51 |
| mahbubabad | 12759 | 03:04:00 | 03:05:00 | 00:01:00 | mabd | 52 |
| varkala | 16824 | 16:04:00 | 16:05:00 | 00:01:00 | vak | 53 |
| warangal | 12759 | 03:55:00 | 03:57:00 | 00:02:00 | wl | 54 |
| trivandrum_central | 16824 | 04:30:00 | 16:50:00 | 00:05:00 | tvc | 55 |
| kazipet_jn | 12759 | 04:13:00 | 04:15:00 | 00:02:00 | kzj | 56 |
| neyyattinkara | 16824 | 17:12:00 | 17:13:00 | 00:01:00 | nyy | 57 |
| secunderabad_jn | 12759 | 06:35:00 | 06:40:00 | 00:05:00 | sc | 58 |
| parassala | 16824 | 17:25:00 | 17:26:00 | 00:01:00 | pasa | 59 |
```

MySQL 8.0 Command Line Client

hyderabad_deccan	12759	07:50:00	00:00:00	00:00:00	hyb	60
kulitthurai	16824	17:37:00	17:40:00	00:03:00	kzt	61
mgr_chennai_central	22625	00:00:00	07:25:00	00:00:00	mas	62
arakkonam	22625	08:23:00	08:25:00	00:02:00	ajj	63
eraniel	16824	17:55:00	17:56:00	00:01:00	erl	64
nagercoil_town	16824	18:10:00	18:13:00	00:03:00	njt	65
aralvaymozhi	16824	18:39:00	18:40:00	00:01:00	aay	66
katpadi_jn	22625	09:13:00	09:15:00	00:02:00	kpd	67
ambur	22625	09:53:00	09:55:00	00:02:00	ab	68
vaniyambadi	22625	10:08:00	10:10:00	00:02:00	vn	69
valliyur	16824	18:59:00	19:00:00	00:01:00	vly	70
jolarpettai	22625	10:28:00	10:30:00	00:02:00	jtj	71
nanguneri	16824	19:10:00	19:11:00	00:01:00	nnn	72
kuppam	22625	11:11:00	11:12:00	00:01:00	kpn	73
tirunelveli	16824	04:30:00	20:05:00	00:05:00	ten	74
kovilpatti	16824	20:58:00	21:00:00	00:02:00	cvp	75
satur	16824	21:18:00	21:20:00	00:02:00	srt	76
virudunagar_jn	16824	22:18:00	22:20:00	00:02:00	vpt	77
tirumangalam	16824	22:39:00	22:40:00	00:01:00	tmq	78
madurai_jn	16824	04:30:00	23:20:00	00:05:00	mdu	79
dindigul_jn	16824	00:32:00	00:35:00	00:03:00	dg	80
tiruchirappalli	16824	04:30:00	01:55:00	00:05:00	tpj	81
vridhachalam_jn	16824	03:23:00	03:25:00	00:02:00	vri	82
villupuram_jn	16823	04:30:00	04:25:00	00:05:00	vm	83
tindivanam	16824	05:05:00	05:07:00	00:02:00	tmv	84
bangarapet	22625	11:39:00	11:41:00	00:02:00	bwt	85
melmaruvathur	16824	05:28:00	05:30:00	00:02:00	mlmr	86
madurantakam	16824	05:43:00	05:45:00	00:02:00	mmk	87
chengalpattu	16824	06:13:00	06:15:00	00:02:00	cgl	88
krishnarajapuram	22625	12:24:00	12:26:00	00:02:00	kjm	89
tambaram	16824	06:43:00	06:45:00	00:02:00	tbm	90
bangalore_cant	22625	12:28:00	12:30:00	00:02:00	bnc	91
mambalam	16824	07:04:00	07:05:00	00:00:00	mbm	92
bangalore_city_jn	22625	13:15:00	00:00:00	00:00:00	sbc	93
chennai_egmore	16824	07:40:00	00:00:00	00:00:00	ms	94
coimbatore_jn	12244	00:00:00	15:05:00	00:00:00	cbe	95
mgr_chennai_central	12639	00:00:00	00:00:00	07:40:00	mas	96
tiruppur	12244	15:43:00	15:45:00	00:02:00	tup	97
arakkonam	12639	08:38:00	08:40:00	00:02:00	ajj	98
erode_jn	12244	16:32:00	16:35:00	00:03:00	ed	99
salem_jn	12244	17:23:00	17:25:00	00:02:00	sa	100
jolarpettai	12244	18:54:00	18:55:00	00:01:00	jtj	101
sholinghur	12639	08:58:00	09:00:00	00:02:00	shu	102
walajah_road_jn	12639	09:13:00	09:15:00	00:02:00	wjr	103
katpadi_jn	12639	09:38:00	09:40:00	00:02:00	kpd	104
katpadi_jn	12244	19:58:00	20:00:00	00:02:00	kpd	105
ambur	12639	10:18:00	10:20:00	00:02:00	ab	106
mgr_chennai_central	12244	22:05:00	00:00:00	00:00:00	mas	107
vaniyambadi	12639	10:33:00	10:35:00	00:02:00	vn	108
guruvayur	16342	00:00:00	03:25:00	00:00:00	guv	109
jolarpettai	12639	10:58:00	11:00:00	00:02:00	jtj	110
thrissur	16342	03:50:00	03:52:00	00:02:00	tcr	111
kuppam	12639	11:39:00	11:40:00	00:01:00	kpn	112
irinjalkuda	16342	04:15:00	04:16:00	00:01:00	ijk	113
bangarapet	12639	12:09:00	12:11:00	00:02:00	bwt	114
chalakudi	16342	04:23:00	04:24:00	00:01:00	cki	115
krishnarajapuram	12639	12:50:00	12:52:00	00:02:00	kjm	116
angamali	16342	04:39:00	04:40:00	00:01:00	afk	117
aluva	16342	04:51:00	04:53:00	00:02:00	awy	118
ernakulam_jn	16342	05:15:00	05:20:00	00:05:00	ers	119
banglore_cant	12639	13:08:00	13:10:00	00:02:00	bnc	120
banglore_city_jn	12639	13:40:00	00:00:00	00:00:00	sbc	121
banglore_city_jn	12640	00:00:00	15:10:00	00:00:00	sbc	122

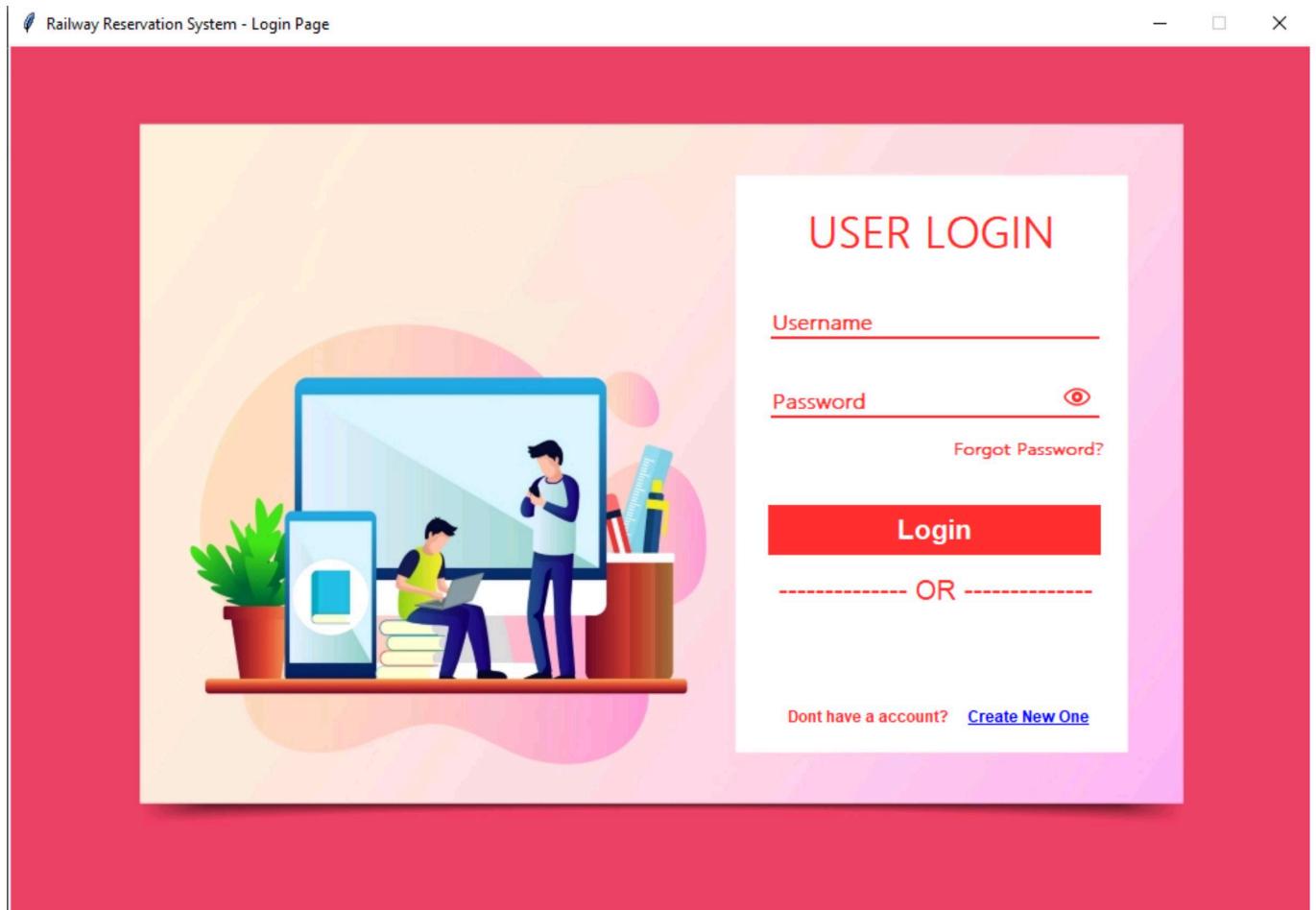
parassala	16381	10:09:00	10:10:00	00:01:00	pasa		200
kulitthurai	16381	10:20:00	10:21:00	00:01:00	kzt		201
eraniel	16381	10:39:00	10:40:00	00:01:00	erl		202
nagercoil_jn	16381	11:35:00	11:40:00	00:05:00	ncj		203
kanyakumari	16381	12:30:00	00:00:00	00:00:00	cape		204
yesvantpur_jn	12257	00:00:00	20:45:00	00:00:00	ypr		205
banaswadi	12257	21:07:00	21:08:00	00:01:00	band		206
hosur	12257	21:58:00	22:00:00	00:02:00	hsra		207
dharmapuri	12257	23:28:00	23:30:00	00:02:00	dpj		208
salem_jn	12257	01:10:00	01:12:00	00:02:00	sa		209
erode_jn	12257	02:05:00	02:10:00	00:05:00	ed		210
coimbatore_jn	12257	03:57:00	04:00:00	00:03:00	cbe		211
palakkad	12257	05:05:00	05:10:00	00:05:00	pgt		212
thrissur	12257	06:27:00	06:30:00	00:03:00	tcr		213
aluva	12257	07:38:00	07:40:00	00:02:00	awy		214
ernakulam_town	12257	08:17:00	08:20:00	00:03:00	ern		215
kottayam	12257	09:49:00	09:50:00	00:01:00	ktym		216
changanaseri	12257	10:09:00	10:10:00	00:01:00	cgy		217
tiruvalla	12257	10:19:00	10:20:00	00:01:00	trvl		218
chengannur	12257	10:29:00	10:30:00	00:01:00	cngr		219
mavelikara	12257	10:44:00	10:45:00	00:01:00	mvlk		220
kayankulam_jn	12257	10:58:00	11:00:00	00:02:00	kyj		221
kollam_jn	12257	11:37:00	11:40:00	00:03:00	qln		222
kochuveli	12257	13:10:00	00:00:00	00:00:00	kcvl		223
mumbai_csmt	16339	00:00:00	20:35:00	00:00:00	csmt		224
mumbai_dadar_central	16339	20:48:00	20:50:00	00:02:00	dr		225
thane	16339	21:13:00	21:15:00	00:02:00	tna		226
kalyan_jn	16339	21:27:00	21:30:00	00:03:00	kyn		227
karjat	16339	22:27:00	22:30:00	00:02:00	kjt		228
pune_jn	16339	00:15:00	00:20:00	00:05:00	pune		229
daund_jn	16339	01:30:00	01:35:00	00:05:00	dd		230
kurduvadi	16339	02:48:00	02:50:00	00:02:00	kvv		231
solapur_jn	16339	03:45:00	03:50:00	00:05:00	sur		232
kalaburagi_junction	16339	05:37:00	05:40:00	00:03:00	klbg		233
shahabad	16339	06:00:00	06:02:00	00:02:00	sbd		234
wadi	16339	06:50:00	06:55:00	00:05:00	wadi		235
yadgir	16339	07:29:00	07:30:00	00:01:00	yg		236
raichur	16339	08:43:00	08:45:00	00:02:00	rc		237
manthralayam_rd	16339	09:09:00	09:10:00	00:01:00	malm		238
adoni	16339	09:52:00	09:53:00	00:01:00	ad		239
guntakal_jn	16339	10:40:00	10:45:00	00:05:00	gtl		240
anantapur	16339	11:59:00	12:00:00	00:01:00	atp		241
dharmaaram_jn	16339	12:43:00	12:45:00	00:02:00	dmm		242
kadiri	16339	13:39:00	13:40:00	00:01:00	kry		243
madanapalle_rd	16339	14:39:00	14:45:00	00:01:00	mpl		244
pakala_jn	16339	16:20:00	16:30:00	00:10:00	pak		245
chittoor	16339	16:54:00	16:55:00	00:01:00	cto		246
katpadi_jn	16339	18:10:00	18:30:00	00:20:00	kpd		247
tirupattur_jn	16339	20:19:00	20:20:00	00:01:00	tpt		248
salem_jn	16339	22:40:00	22:45:00	00:05:00	sa		249
karur	16339	00:48:00	00:50:00	00:02:00	krr		250
dindigul_jn	16339	03:10:00	03:15:00	00:05:00	dg		251
madurai_jn	16339	04:10:00	04:15:00	00:05:00	mdu		252
virudunagar_jn	16339	05:28:00	05:30:00	00:02:00	vpt		253
satur	16339	05:51:00	05:53:00	00:02:00	srt		254
kovlpatti	16339	06:13:00	06:15:00	00:02:00	cvp		255
tirunelveli	16339	08:25:00	08:30:00	00:05:00	ten		256
nanguneri	16339	08:59:00	09:00:00	00:01:00	nnn		257
valliyur	16339	09:10:00	09:11:00	00:01:00	vly		258
nagercoil_jn	16339	10:15:00	00:00:00	00:00:00	ncj		259

259 rows in set (0.48 sec)

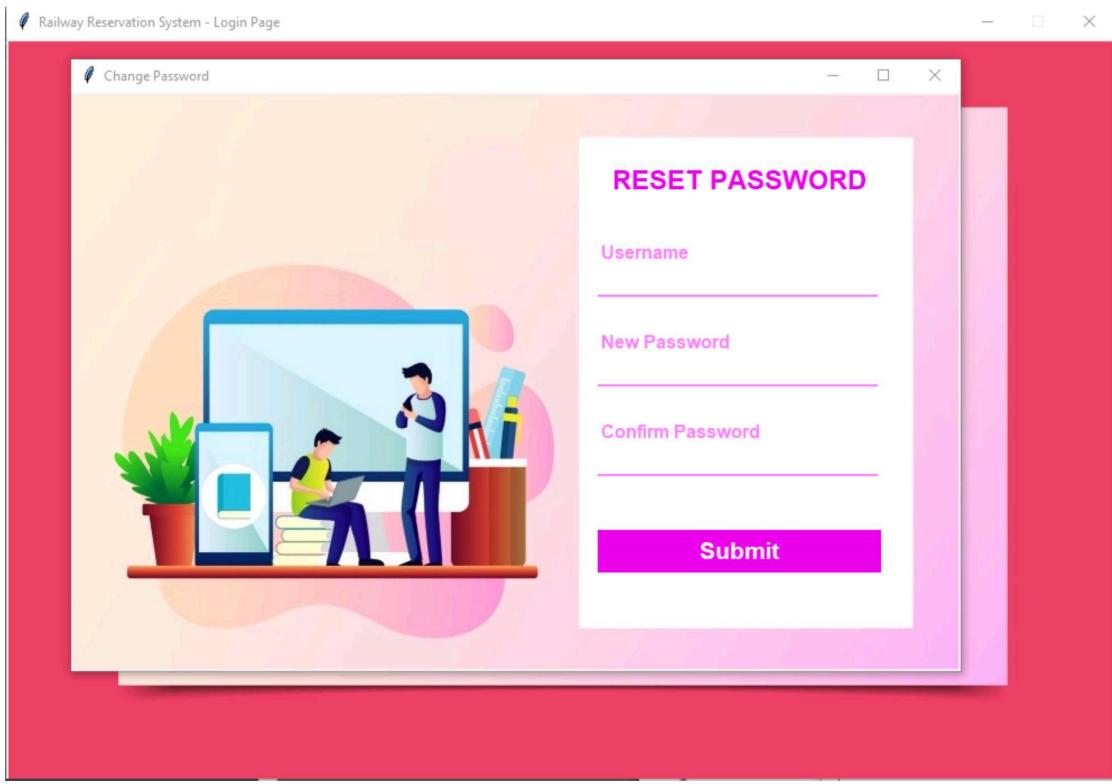
train_no	train_name	source	destination	arrival_time	departure_time	seat_avail	sno
16823	ananthapuri_express	chennai_egmore	kollam	11:45:00	20:10:00	half	1
16824	ananthapuri_express	kollam	chennai_egmore	15:40:00	07:40:00	half	2
12639	brindavan_sf_express	mgr_chennai_central	banglore_city_jn	07:40:00	13:40:00	half	3
12640	brindavan_sf_express	banglore_city_jn	mgr_chennai_central	15:00:00	21:10:00	full	4
12759	charminar_express	tambaram	hyderabad	17:10:00	07:50:00	half	5
22625	sbc_double_decker	mgr_chennai_central	banglore_city_jn	07:25:00	13:15:00	half	6
12244	shatabdi_express	coimbatore_jn	mgr_chennai_central	15:05:00	22:05:00	full	7
16342	guruvayur_intercity_express	guruvayur	trivandrum_central	03:25:00	09:55:00	full	8
12434	chennai_rajdhani	delhi_hazrat_nizamuddin	mgr_chennai_central	15:25:00	20:55:00	empty	9
16381	kanyakumari_express	pune_jn	kanyakumari	23:50:00	12:30:00	empty	10
12257	kochuveli_garib_rath	yeshvantpur_jn	kochuveli	20:45:00	13:10:00	empty	11
16339	nagercoil_express	mumbai_csmt	nagercoil_jn	20:35:00	10:10:00	empty	12
16787	ten_jammu_express	tirunelveli	smvd_katra	17:35:00	10:35:00	empty	13
12605	pallavan_express	chennai_egmore	karaikudi	15:45:00	22:35:00	empty	14
16090	yelagiri_express	jolarpettai	mgr_chennai_central	05:00:00	09:10:00	half	15

GRAPHICAL USER INTERFACE:

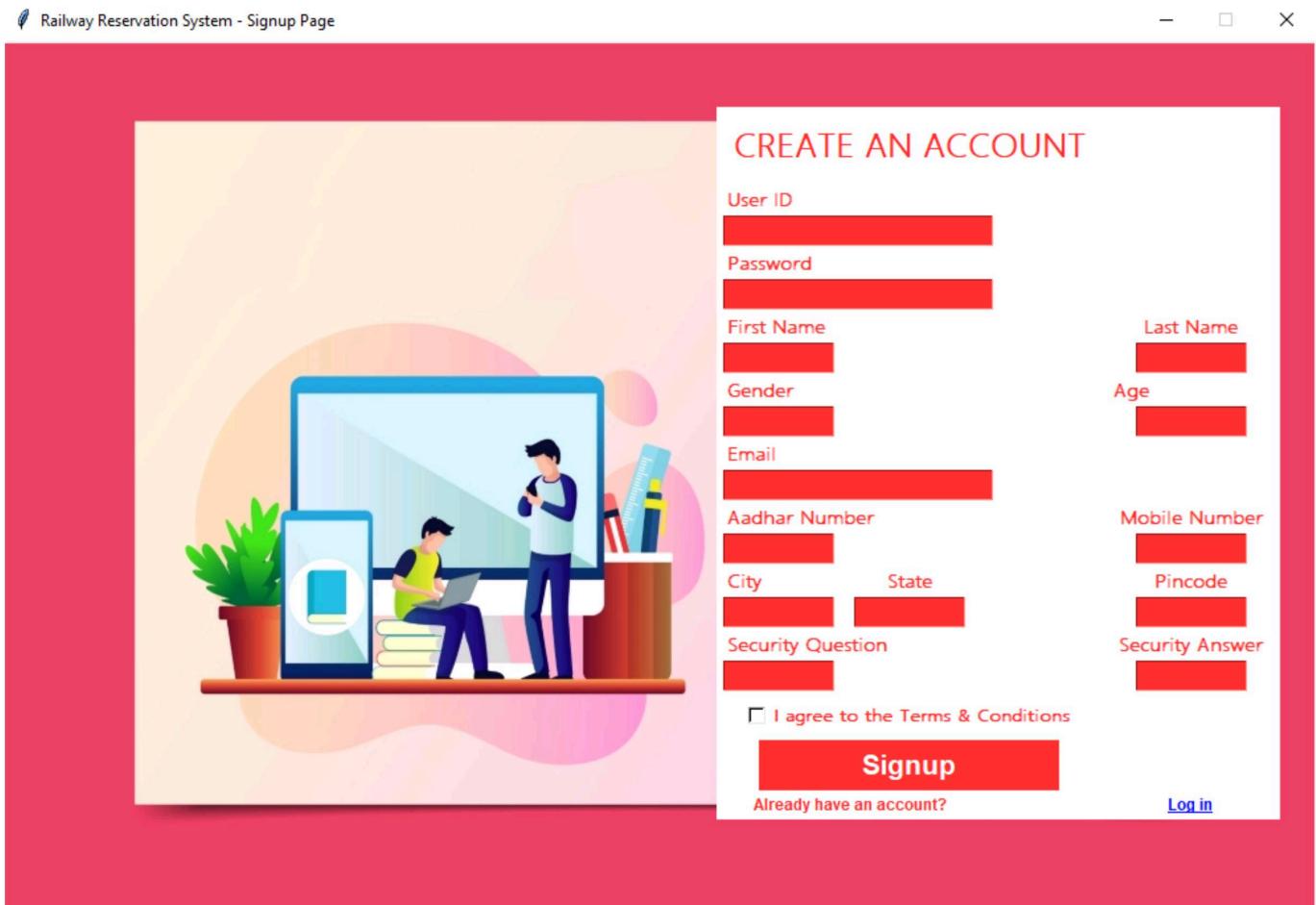
SIGN IN PAGE:



FORGOT PASSWORD PAGE:



SIGNUP PAGE:



MAIN PAGE:

The main interface features a large banner image of a black steam locomotive emitting smoke. Overlaid on the banner are several blue rectangular buttons with white text: "Logout", "Book a Ticket", "Cancel a Ticket", "View Ticket", "Ticket Enquiry", and "Available Trains". To the right of these buttons is the text "Train Schedule". Below the banner is a search bar with the placeholder "Train_no" and a yellow "Submit" button.

SEARCH TRAINS:

The search form is titled "Search trains" and is associated with "Indian Railways IRCTC Train Tickets Reservation". It includes fields for "From" (with placeholder "Enter a city or station") and "To" (with placeholder "Enter a city or station"). There are dropdown menus for "Class" (with placeholder "Select class") and "Date" (with placeholder "dd/mm/yyyy"). Below these are four dropdown menus for passenger counts: "Adults" (set to 1), "Children" (set to 0), "Senior men" (set to 0), and "Senior women" (set to 0). At the bottom is a large orange "Search trains" button.

BIBLIOGRAPHY:

REFERENCE SITES:

1. www.free-code-camp.com
2. www.Irctc.co.in
3. www.Indianrail.gov.in
4. www.w3schools.com
5. www.icons-icons.com
6. www.plus2net.com
7. www.python.org



SRM

INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Course Project Report titled “RAILWAY RESERVATION SYSTEM” is the bonafide work done by NITHEESH S [RA2111047010194], THIVAKAR R[RA2111047010197] of II Year/ IV Sem B.Tech (AI) who carried out under my supervision for the course 18AIC207J - Database Management Systems for Artificial Intelligence. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE

Faculty In-Charge

Dr.Sumathy G

Assistant Professor

Department of Computational Intelligence

SRM Institute of Science and Technology

Kattankulathur Campus, Chennai

HEAD OF THE DEPARTMENT

Dr. R Annie Uthra

Professor and Head ,

Department of Computational Intelligence,

SRM Institute of Science and Technology

Kattankulathur Campus, Chennai