**RAILWAY FINE SYSTEM**

****

**GROUP MEMBERS ROLL NO.**

Sayali Moghe 42

Jay Pakale 50

Mitali Ranawat 62

Ninad Wadode 77

**PROBLEM STATEMENT**

Railway fine system is a system that effectively and efficiently holds the details of people getting regularly fined and the details of the fines. It reduces the chances of corruption and other ill-practices.

Record of finer and the amount paid by him is store. Record of the finee is also kept.The system also records the place where the fine is collected, collected by whom, paid by whom and the place submitted.There are different pages for the finee and the finer depending on login credentials, the finee can get a display of his personal details and the data of all the fines that are due or that are paid by him.

The finer can get personal details and new fine generation page that displays the journey details as well as the amount and type of fine.

**MODULE DESCRIPTION**

**Login Page:**

The login page checks the credentials of the user and accordingly gives the access to the parts of the site that he has previliges to. It has two fields username and password, according to the role of the user they are given either passenger access or ticket collector access. If the user is not already existing,they are directed to the signup page.

**Signup Page:**

The signup page is for the new users, it has various feilds to take input of personal details namely Name,Age,Aadhar card number,Email ID,Username, Password,Contact etc of the user and the type of access to be given i.e. passenger or ticket collector.After signing up the user is redirected to the login page.

**Passenger logged in Page:**

The passenger logged in page is accessed after the user logs in and if he is given passenger privilege.This page has two parts, personal details where all the details of the passenger like username, name,id,contact etc are displayed and fine table that stores the data of total fines incurred by that passenger with details like fine type, fine amount,place collected at etc.There is option to logout from this page if the passenger is done checking out the details.

**Collector logged in Page:**

The passenger logged in page is accessed after the user logs in and if he is given passenger privilege.This page has two parts, personal details where all the details of the passenger like username, name,id,contact etc are displayed and fine generation table which is accessed only by the collector. In this table the collector generates fines depending on type and amount and the data is pushed into the database.

**Fine Generation Page:**

This is the main page of the system, this page is only accessed by the collector. It first validates the username of the passenger who is to be fined. Then it takes the place detail where the fine is incurred. It then takes the type of fine that is incurred and the amount of the fine is fetched from the database. The final fine detail is thus generated and then pushed into the database.

**PROGRAM CODE**

**Root:**

import tkinter as tk

import pymysql

import Login\_page

import Signup\_page

import Passenger\_Logged\_in

import TC\_Logged\_in

import Add\_fine

import Personal\_details

class Root(tk.Frame):

def \_\_init\_\_(self,master):

tk.Frame.\_\_init\_\_(self,master)

self.master.title("Railway Fine System")

self.master.resizable(False,False)

self.master.geometry("1200x800")

self.master.protocol("WM\_DELETE\_WINDOW", self.clickcancel)

self.master.bind("<Escape>", self.clickescape)

self.pack()

self.canvas = tk.Canvas(root).pack()

self.loginpage = Login\_page.Login\_page(self)

def database(self):

self.mydb=pymysql.connect(host='localhost',user='root',passwd='',db='railway\_fine\_system')

self.mycursor=self.mydb.cursor()

return self.mycursor

def clickcancel(self):

self.master.destroy()

def clickescape(self,event):

self.master.destroy()

def clicksignup(self):

self.signuppage = Signup\_page.Signup\_page(self)

def clickregister(self):

self.signuppage.error\_label.configure(text="")

name = self.signuppage.name\_entry.get()

aadhar = self.signuppage.aadhar\_entry.get()

email = self.signuppage.email\_entry.get()

username = self.signuppage.username\_entry.get()

password = self.signuppage.password\_entry.get()

confirm = self.signuppage.confirmpassword\_entry.get()

gender = self.signuppage.gender.get()

age = self.signuppage.age\_entry.get()

contact = self.signuppage.contact\_entry.get()

flag=1

if name=="" or aadhar=="" or email=="" or username=="" or password=="" or confirm=="" or gender=="" or age=="" or contact=="":

self.signuppage.error\_label.configure(text="One or more fields are empty!!")

flag=0

elif not name.isalnum() or not aadhar.isdigit() or not age.isdigit() or not contact.isdigit():

self.signuppage.error\_label.configure(text="Invalid data entry!!")

flag=0

elif len(aadhar)!=12:

self.signuppage.error\_label.configure(text="Invalid adhar number!!")

flag=0

elif len(username)>15:

self.signuppage.error\_label.configure(text="Username cannot have more than 15 characters!!")

flag=0

elif len(contact)!=10:

self.signuppage.error\_label.configure(text="Invalid contact number!!")

flag=0

elif self.signuppage.var.get() not in (1,2,3):

self.signuppage.error\_label.configure(text="User type not specified!!")

flag=0

if flag==1:

self.signuppage.error\_label.configure(text="")

if(confirm!=password):

self.signuppage.error\_label.configure(text="Password doesn't match!!")

else:

self.mycursor = self.database()

self.button\_value = self.signuppage.var.get()

if self.button\_value==1:

self.mycursor.execute('INSERT INTO finee(finee\_name, finee\_username, finee\_password, email, aadhar\_card, gender, age, contact\_no) VALUES ("%s", "%s", "%s", "%s", "%s", "%s", "%s", "%s")'%(name, username, password, email, aadhar, gender, age, contact))

self.mydb.commit()

elif self.button\_value==2:

self.mycursor.execute('INSERT INTO finer(finer\_name, finer\_username, finer\_password, email, aadhar\_card, gender, age, contact\_no, post) VALUES ("%s", "%s", "%s", "%s", "%s", "%s", "%s", "%s", "Ticket Collector")'%(name, username, password, email, aadhar, gender, age, contact))

self.mydb.commit()

elif self.button\_value==3:

self.mycursor.execute('INSERT INTO finer(finer\_name, finer\_username, finer\_password, email, aadhar\_card, gender, age, contact\_no, post) VALUES ("%s", "%s", "%s", "%s", "%s", "%s", "%s", "%s", "Station Incharge")'%(name, username, password, email, aadhar, gender, age, contact))

self.mydb.commit()

self.loginpage = Login\_page.Login\_page(self)

def clickregisternew(self):

self.signuppage.error\_label.configure(text="")

name = self.signuppage.name\_entry.get()

aadhar = self.signuppage.aadhar\_entry.get()

email = self.signuppage.email\_entry.get()

username = self.signuppage.username\_entry.get()

password = self.signuppage.password\_entry.get()

confirm = self.signuppage.confirmpassword\_entry.get()

gender = self.signuppage.gender.get()

age = self.signuppage.age\_entry.get()

contact = self.signuppage.contact\_entry.get()

flag=1

if name=="" or aadhar=="" or email=="" or username=="" or password=="" or confirm=="" or gender=="" or age=="" or contact=="":

self.signuppage.error\_label.configure(text="One or more fields are empty!!")

flag=0

elif not name.isalnum() or not aadhar.isdigit() or not age.isdigit() or not contact.isdigit():

self.signuppage.error\_label.configure(text="Invalid data entry!!")

flag=0

elif len(aadhar)!=12:

self.signuppage.error\_label.configure(text="Invalid adhar number!!")

flag=0

elif len(username)>15:

self.signuppage.error\_label.configure(text="Username cannot have more than 15 characters!!")

flag=0

elif len(contact)!=10:

self.signuppage.error\_label.configure(text="Invalid contact number!!")

flag=0

elif self.signuppage.var.get() not in (1,2,3):

self.signuppage.error\_label.configure(text="User type not specified!!")

flag=0

if flag==1:

self.signuppage.error\_label.configure(text="")

if(confirm!=password):

self.signuppage.error\_label.configure(text="Password doesn't match!!")

else:

self.mycursor = self.database()

self.button\_value = self.signuppage.var.get()

if self.button\_value==1:

self.mycursor.execute('INSERT INTO finee(finee\_name, finee\_username, finee\_password, email, aadhar\_card, gender, age, contact\_no) VALUES ("%s", "%s", "%s", "%s", "%s", "%s", "%s", "%s")'%(name, username, password, email, aadhar, gender, age, contact))

self.mydb.commit()

elif self.button\_value==2:

self.mycursor.execute('INSERT INTO finer(finer\_name, finer\_username, finer\_password, email, aadhar\_card, gender, age, contact\_no, post) VALUES ("%s", "%s", "%s", "%s", "%s", "%s", "%s", "%s", "Ticket Collector")'%(name, username, password, email, aadhar, gender, age, contact))

self.mydb.commit()

elif self.button\_value==3:

self.mycursor.execute('INSERT INTO finer(finer\_name, finer\_username, finer\_password, email, aadhar\_card, gender, age, contact\_no, post) VALUES ("%s", "%s", "%s", "%s", "%s", "%s", "%s", "%s", "Station Incharge")'%(name, username, password, email, aadhar, gender, age, contact))

self.mydb.commit()

self.addfine = Add\_fine.Add\_fine(self)

def clicklogin(self):

self.loginpage.error\_login.configure(text="")

username = self.loginpage.username\_entry.get()

password = self.loginpage.password\_entry.get()

if username=="" or password=="":

self.loginpage.error\_login.configure(text="Username or Password not entered!!")

else:

self.loginpage.error\_login.configure(text="")

self.mycursor = self.database()

self.mycursor.execute('SELECT finee\_username FROM finee')

self.finee\_list = self.mycursor.fetchall()

self.finee\_list = list(sum(self.finee\_list, ()))

self.mycursor.execute('SELECT finer\_username FROM finer')

self.finer\_list = self.mycursor.fetchall()

self.finer\_list = list(sum(self.finer\_list, ()))

self.final\_list = self.finee\_list+self.finer\_list

if username not in self.final\_list: #or self.finer\_list:

self.loginpage.error\_login.configure(text="Invalid username!!")

else:

if username in self.finee\_list:

self.query1 = 'SELECT finee\_password FROM finee WHERE finee\_username="%s" '

self.mycursor.execute(self.query1 % username)

self.finee\_password = self.mycursor.fetchall()

self.finee\_password = list(sum(self.finee\_password, ()))

if password not in self.finee\_password:

self.loginpage.error\_login.configure(text="Invalid password!!")

else:

self.user = username

self.passenger\_loggedin = Passenger\_Logged\_in.Passenger\_Logged\_in(self)

self.passengerfines()

elif username in self.finer\_list:

self.query1 = 'SELECT finer\_password FROM finer WHERE finer\_username="%s" '

self.mycursor.execute(self.query1 % username)

self.finer\_password = self.mycursor.fetchall()

self.finer\_password = list(sum(self.finer\_password, ()))

if password not in self.finer\_password:

self.loginpage.error\_login.configure(text="Invalid password!!")

else:

self.user = username

self.mycursor.execute('SELECT post FROM finer WHERE finer\_username="%s" ' %(username))

self.post = self.mycursor.fetchall()

self.post = list(sum(self.post, ()))

self.tc\_loggedin = TC\_Logged\_in.TC\_Logged\_in(self)

self.finedetails()

def clicklogout(self):

self.loginpage = Login\_page.Login\_page(self)

def clickpersonaldetails(self):

self.personaldetails = Personal\_details.Personal\_details(self)

def clickaddfine(self):

self.addfine = Add\_fine.Add\_fine(self)

def checkvalidity(self):

fineename = self.addfine.fineename\_entry.get()

fineename = str.lower(fineename)

self.mycursor = self.database()

self.mycursor.execute('SELECT finee\_username FROM finee')

self.finee\_list = self.mycursor.fetchall()

self.finee\_list = list(sum(self.finee\_list, ()))

if fineename not in self.finee\_list:

self.signuppage = Signup\_page.Signup\_page(self)

else:

self.addfine.fineename\_error.configure(text="USER EXISTS.")

def displayfine(self):

self.mycursor = self.database()

self.mycursor.execute('SELECT amount FROM fine WHERE type\_of\_fine = "%s"'%(self.addfine.finetype.get()))

self.otherfines = self.mycursor.fetchall()

self.otherfines = list(sum(self.otherfines, ()))

self.addfine.amount\_entry.configure(text=self.otherfines)

def firstfine(self):

self.mycursor = self.database()

self.mycursor.execute('SELECT relative\_distance FROM station\_coordinates WHERE station\_name="%s" '%(self.addfine.fromstation.get()))

self.f = self.mycursor.fetchall()

self.f = list(sum(self.f, ()))

self.mycursor.execute('SELECT relative\_distance FROM station\_coordinates WHERE station\_name="%s" '%(self.addfine.station.get()))

self.t = self.mycursor.fetchall()

self.t = list(sum(self.t, ()))

self.distance = abs(self.t[0] - self.f[0])

self.amount = 250 + self.distance

self.addfine.amount\_entry.configure(text=self.amount)

def clicksubmit(self):

self.mycursor = self.database()

finerusername = self.loginpage.username\_entry.get()

fineeusername = self.addfine.fineename\_entry.get()

fine\_at = self.addfine.station.get()

fine\_type = self.addfine.finetype.get()

if self.addfine.finetype.get()=="Travelling without ticket":

fine\_amt = self.amount

else:

self.mycursor.execute('SELECT amount FROM fine WHERE type\_of\_fine = "%s" '%(self.addfine.finetype.get()))

amt = self.mycursor.fetchall()

amt=list(sum(amt, ()))

fine\_amt = amt[0]

self.mycursor.execute('INSERT INTO fine\_generated(finer\_username, finee\_username, type\_of\_fine, fine\_amt, fine\_collected\_at) VALUES ("%s", "%s", "%s", "%s", "%s")'%(finerusername, fineeusername, fine\_type, fine\_amt, fine\_at))

self.mydb.commit()

self.tc\_loggedin = TC\_Logged\_in.TC\_Logged\_in(self)

self.finedetails()

def finedetails(self):

self.mycursor = self.database()

self.query1 = 'SELECT finee\_username FROM fine\_generated WHERE finer\_username="%s" '

self.mycursor.execute(self.query1 % self.loginpage.username\_entry.get())

self.username1 = self.mycursor.fetchall()

self.username1 = list(sum(self.username1, ()))

self.query2 = 'SELECT fine\_collected\_at FROM fine\_generated WHERE finer\_username="%s" '

self.mycursor.execute(self.query2 % self.loginpage.username\_entry.get())

self.station1 = self.mycursor.fetchall()

self.station1 = list(sum(self.station1, ()))

self.query3 = 'SELECT fine\_amt FROM fine\_generated WHERE finer\_username="%s" '

self.mycursor.execute(self.query3 % self.loginpage.username\_entry.get())

self.amount1 = self.mycursor.fetchall()

self.amount1 = list(sum(self.amount1, ()))

self.query5 = 'SELECT count(id) FROM fine\_generated WHERE finer\_username="%s" '

self.mycursor.execute(self.query5 % self.loginpage.username\_entry.get())

self.count = self.mycursor.fetchall()

self.count = list(sum(self.count, ()))

if(self.count[0]==0):

self.tc\_loggedin.error\_label.configure(text="NO FINES TAKEN YET")

else:

k=0.30

for i in range(0, self.count[0]):

j=0.15

self.u = tk.Label(self.tc\_loggedin.details\_frame, text=self.username1[i], fg="black", bg="white", font=("Verdana", "13"))

self.u.place(relx=j, rely=k)

j+=0.30

self.s = tk.Label(self.tc\_loggedin.details\_frame, text=self.station1[i], fg="black", bg="white", font=("Verdana", "13"))

self.s.place(relx=j, rely=k)

j+=0.30

self.a = tk.Label(self.tc\_loggedin.details\_frame, text=self.amount1[i], fg="black", bg="white", font=("Verdana", "13"))

self.a.place(relx=j, rely=k)

j+=0.30

k+=0.1

def passengerfines(self):

self.mycursor = self.database()

self.query0 = 'SELECT type\_of\_fine FROM fine\_generated WHERE finee\_username = "%s" '

self.mycursor.execute(self.query0 % self.loginpage.username\_entry.get())

self.type = self.mycursor.fetchall()

self.type = list(sum(self.type, ()))

self.query1 = 'SELECT fine\_collected\_at FROM fine\_generated WHERE finee\_username = "%s" '

self.mycursor.execute(self.query1 % self.loginpage.username\_entry.get())

self.place = self.mycursor.fetchall()

self.place = list(sum(self.place, ()))

self.query2 = 'SELECT fine\_amt FROM fine\_generated WHERE finee\_username = "%s" '

self.mycursor.execute(self.query2 % self.loginpage.username\_entry.get())

self.fineamount = self.mycursor.fetchall()

self.fineamount = list(sum(self.fineamount, ()))

self.mycursor.execute('SELECT count(id) FROM fine\_generated WHERE finee\_username = "%s" ' % self.loginpage.username\_entry.get())

self.count1 = self.mycursor.fetchall()

self.count1 = list(sum(self.count1, ()))

if(self.count1[0]==0):

self.passenger\_loggedin.error\_label.configure(text="NO FINES PAID YET")

else:

k=0.3

for i in range(0, self.count1[0]):

j=0.05

self.u = tk.Label(self.passenger\_loggedin.finedetails\_frame, text=self.type[i], fg="black", bg="white", font=("Verdana", "13"))

self.u.place(relx=j, rely=k)

j+=0.4

self.s = tk.Label(self.passenger\_loggedin.finedetails\_frame, text=self.place[i], fg="black", bg="white", font=("Verdana", "13"))

self.s.place(relx=j, rely=k)

j+=0.4

self.a = tk.Label(self.passenger\_loggedin.finedetails\_frame, text=self.fineamount[i], fg="black", bg="white", font=("Verdana", "13"))

self.a.place(relx=j, rely=k)

k+=0.1

def clickbacktc(self):

self.tc\_loggedin = TC\_Logged\_in.TC\_Logged\_in(self)

self.finedetails()

def clickbackpassenger(self):

self.passenger\_loggedin = Passenger\_Logged\_in.Passenger\_Logged\_in(self)

if \_\_name\_\_=="\_\_main\_\_":

root = tk.Tk()

r = Root(root)

r.mainloop()

**Login\_page:**

import tkinter as tk

class Login\_page(tk.Frame):

def \_\_init\_\_(self,master):

mainlogin\_frame = tk.Frame(master.canvas, bg="white")

mainlogin\_frame.place(relheight=1, relwidth=1)

title\_frame = tk.Frame(mainlogin\_frame, bg="#003145", bd=13)

title\_frame.place(relx=0.05, rely=0.1, relwidth=0.9, relheight=0.2)

title\_label = tk.Label(title\_frame, text="RAILWAY FINE SYSTEM", font=("Comic Sans MS", "62"), bg="#003145", fg="white", )

title\_label.place(relheight=1, relwidth=1)

login\_frame = tk.Frame(mainlogin\_frame, bg="#003145", bd=13)

login\_frame.place(relx=0.5, rely=0.4, relwidth=0.4, relheight=0.45, anchor="n")

login\_label = tk.Frame(login\_frame, bg="white")

login\_label.place(relheight=1, relwidth=1)

username\_label = tk.Label(login\_label, text="Username", fg="black", bg="white", font=("Times", "17", "italic"))

username\_label.place(relx=0.1, rely=0.25)

self.username\_entry = tk.Entry(login\_label, font=("Courier", "10"))

self.username\_entry.place(relx=0.4, rely=0.25)

password\_label = tk.Label(login\_label, text="Password", fg="black", bg="white", font=("Times", "17", "italic"))

password\_label.place(relx=0.1, rely=0.4)

self.password\_entry = tk.Entry(login\_label, font=("Courier", "10"), show="\*")

self.password\_entry.place(relx=0.4, rely=0.4)

self.error\_login = tk.Label(login\_label, text="", fg="red", bg="white", font=("Times", "13"))

self.error\_login.place(relx=0.5, rely=0.53, relwidth=0.5, anchor="n")

login\_button = tk.Button(login\_label, text="Login", font=("Bradley Hand ITC", "14"), bg="#003145", fg="white", command=master.clicklogin)

login\_button.place(relx=0.5, relwidth=0.35, anchor="n", rely=0.60)

signup\_label = tk.Label(login\_label, text="Not registered yet? ", bg="white", font=("Times", "14", "italic"))

signup\_label.place(relx=0.5, rely=0.75, relwidth=0.5, anchor="n")

signup\_button = tk.Button(login\_label, text="Signup", bg="#003145", fg="white", font=("Bradley Hand ITC","13"), command=master.clicksignup)

signup\_button.place(relx=0.5, rely=0.85, relwidth=0.2, anchor="n")

**Signup\_page:**

import tkinter as tk

import pymysql

class Signup\_page(tk.Frame):

def \_\_init\_\_(self,master):

self.var=tk.IntVar()

mainsignup\_frame = tk.Frame(master.canvas, bg="white")

mainsignup\_frame.place(relheight=1, relwidth=1)

signup\_frame = tk.Frame(master.canvas, bg="#003145", bd=13)

signup\_frame.place(relx=0.5, rely=0.1, relwidth=0.4, relheight=0.8, anchor="n")

signup\_label = tk.Frame(signup\_frame, bg="white")

signup\_label.place(relheight=1, relwidth=1)

name\_label = tk.Label(signup\_label, text="Name", fg="black",bg="white", font=("Georgia", "14"))

name\_label.place(relx=0.15, rely=0.05)

self.name\_entry = tk.Entry(signup\_label)

self.name\_entry.place(relx=0.45, rely=0.05)

aadhar\_label = tk.Label(signup\_label, text="Aadhar No.", fg="black", bg="white", font=("Georgia", "14"))

aadhar\_label.place(relx=0.11, rely=0.125)

self.aadhar\_entry = tk.Entry(signup\_label)

self.aadhar\_entry.place(relx=0.45, rely=0.125)

email\_label = tk.Label(signup\_label, text="Email", fg="black", bg="white", font=("Georgia", "14"))

email\_label.place(relx=0.13, rely=0.2)

self.email\_entry = tk.Entry(signup\_label)

self.email\_entry.place(relx=0.45, rely=0.2)

username\_label = tk.Label(signup\_label, text="Username", fg="black", bg="white", font=("Georgia", "14"))

username\_label.place(relx=0.11, rely=0.275)

self.username\_entry = tk.Entry(signup\_label)

self.username\_entry.place(relx=0.45, rely=0.275)

password\_label = tk.Label(signup\_label, text="Password", fg="black", bg="white", font=("Georgia", "14"))

password\_label.place(relx=0.11, rely=0.35)

self.password\_entry = tk.Entry(signup\_label, show="\*")

self.password\_entry.place(relx=0.45, rely=0.35)

confirmpassword\_label = tk.Label(signup\_label, text="Confirm Password", fg="black", bg="white", font=("Georgia", "14"))

confirmpassword\_label.place(relx=0.05, rely=0.425)

self.confirmpassword\_entry = tk.Entry(signup\_label, show="\*")

self.confirmpassword\_entry.place(relx=0.45, rely=0.425)

gender\_label = tk.Label(signup\_label, text="Gender", fg="black", bg="white", font=("Georgia", "14"))

gender\_label.place(relx=0.13, rely=0.5)

self.gender = tk.StringVar(signup\_label)

choices = ("Male", "Female", "Other")

self.gender\_option = tk.OptionMenu(signup\_label, self.gender, \*choices)

self.gender\_option.place(relx=0.475, rely=0.5, width=100)

self.gender.set("Male")

age\_label = tk.Label(signup\_label, text="Age", fg="black", bg="white", font=("Georgia", "14"))

age\_label.place(relx=0.16, rely=0.575)

self.age\_entry = tk.Entry(signup\_label)

self.age\_entry.place(relx=0.45, rely=0.575)

contact\_label = tk.Label(signup\_label, text="Contact No.", fg="black", bg="white", font=("Georgia", "14"))

contact\_label.place(relx=0.10, rely=0.65)

self.contact\_entry = tk.Entry(signup\_label)

self.contact\_entry.place(relx=0.45, rely=0.65)

self.passenger\_button = tk.Radiobutton(signup\_label, variable=self.var, value=1, text="Passenger", bg="white")

self.passenger\_button.place(relx=0.15, rely=0.75)

self.ticketcollector\_button = tk.Radiobutton(signup\_label, variable=self.var, value=2, text="Ticket Collector", bg="white")

self.ticketcollector\_button.place(relx=0.15, rely=0.80)

self.error\_label = tk.Label(signup\_label, text="", fg="red", bg="white", font=("Times", "13"))

self.error\_label.place(relx=0.5, rely=0.85, relwidth=1, anchor="n")

self.register\_button = tk.Button(signup\_label, text="Register", fg="white", bg="#003145", font=("Georgia", "14"), command=master.clickregister)

self.register\_button.place(relx=0.5, rely=0.90, relwidth=0.4, anchor="n")

**Add\_fine:**

import tkinter as tk

import tkcalendar

# import TC\_Logged\_in

class Add\_fine(tk.Frame):

def \_\_init\_\_(self,master):

self.master = master

main\_frame = tk.Frame(master.canvas, bg="white")

main\_frame.place(relheight=1, relwidth=1)

title\_frame = tk.Frame(main\_frame, bg="#003145", bd=13)

title\_frame.place(relx=0.05, rely=0.05, relwidth=0.9, relheight=0.2)

title\_label = tk.Label(title\_frame, text="RAILWAY FINE SYSTEM", font=("Comic Sans MS", "62"), bg="#003145", fg="white", )

title\_label.place(relheight=1, relwidth=1)

self.fine\_frame = tk.Frame(main\_frame, bg="white")

self.fine\_frame.place(relx=0.05, rely=0.30, relwidth=0.9, relheight=0.6)

name\_label = tk.Label(self.fine\_frame, bg="white", text="Name of Finee ", fg="black", font=("Courier", "14", "bold"))

name\_label.place(relx=0.05, rely=0.05)

self.fineename\_entry = tk.Entry(self.fine\_frame)

self.fineename\_entry.place(relx=0.25, rely=0.05)

check\_button = tk.Button(self.fine\_frame, fg="white", bg="#003145", text="Check Validity", font=("Bradley Hand ITC", "11", "bold"), command=master.checkvalidity)

check\_button.place(relx=0.7, rely=0.05)

self.fineename\_error = tk.Label(self.fine\_frame, bg="white", fg="red", text="")

self.fineename\_error.place(relx=0.5, rely=0.075)

station\_label = tk.Label(self.fine\_frame, bg="white", fg="black", text="Station ", font=("Courier", "14", "bold"))

station\_label.place(relx=0.05, rely=0.20)

self.station = tk.StringVar(self.fine\_frame)

self.stations = ("Chhatrapati Shivaji Maharaj Terminus", "Sandhurst Road", "Byculla", "Currey Road", "Dadar", "Matunga", "Sion", "Kurla",

"Vidyavihar", "Ghatkopar", "Vikhroli", "Kanjurmarg", "Bhandup", "Nahur", "Mulund", "Thane", "Kalwa", "Mumbra", "Diva",

"Kopar", "Dombivali", "Kalyan", "Vithalwadi", "Ulhasnagar", "Ambernath", "Badlapur", "Vangani", "Neral", "Karjat",

"Khopoli")

self.station\_option = tk.OptionMenu(self.fine\_frame, self.station, \*self.stations)

self.station\_option.place(relx=0.25, rely=0.20, width=200)

self.station.set("Select station")

finetype\_label = tk.Label(self.fine\_frame, bg="white", fg="black", text="Fine type ", font=("Courier", "14", "bold"))

finetype\_label.place(relx=0.05, rely=0.35)

self.finetype = tk.StringVar(self.fine\_frame)

choices = ("Travelling without ticket", "Travelling Fraudulently", "Alarm chain pulling", "Travelling in coach reserved for handicapped",

"Travelling on roof top", "Trespassing", "Nuisance and littering", "Bill pasting", "Touting", "Unauthorised Hawking")

self.finetype\_option = tk.OptionMenu(self.fine\_frame, self.finetype, \*choices, command=self.places)

self.finetype\_option.place(relx=0.25, rely=0.35, width=300)

self.finetype.set("Select your fine type")

def places(self,master):

if self.finetype.get()=="Travelling without ticket":

blank\_frame = tk.Frame(self.fine\_frame, bg="white")

blank\_frame.place(relx=0.05, rely=0.45, relwidth=0.9, relheight=0.4)

from\_label = tk.Label(self.fine\_frame, text="From station ", bg="white", font=("Courier", "13", "bold"))

from\_label.place(relx=0.1, rely=0.475)

self.fromstation = tk.StringVar(self.fine\_frame)

self.fromstation\_option = tk.OptionMenu(self.fine\_frame, self.fromstation, \*self.stations, command=self.fromto)

self.fromstation\_option.place(relx=0.25, rely=0.475)

self.fromstation.set("Select station")

self.to\_label = tk.Label(self.fine\_frame, text="To station ", bg="white", font=("Courier", "13", "bold"))

self.to\_label.place(relx=0.1, rely=0.575)

self.to\_entry = tk.Label(self.fine\_frame, text=self.station.get(), bg="white", fg="black", font=("Courier", "13"))

self.to\_entry.place(relx=0.25, rely=0.575)

else:

blank\_frame = tk.Frame(self.fine\_frame, bg="white")

blank\_frame.place(relx=0.05, rely=0.45, relwidth=0.9, relheight=0.4)

amount\_label = tk.Label(self.fine\_frame, bg="white", fg="black", text="Fine Amount ", font=("Courier", "14", "bold"))

amount\_label.place(relx=0.05, rely=0.55)

self.amount\_entry = tk.Label(self.fine\_frame, text="", fg="black", bg="white", font=("Courier", "14"))

self.amount\_entry.place(relx=0.25, rely=0.55)

self.master.displayfine()

self.submit\_button = tk.Button(self.fine\_frame, text="Next >", fg="white", bg="#003145", width=20, font=("Courier", "13"), command=self.master.clicksubmit)

self.submit\_button.place(relx=0.5, rely=0.70)

def fromto(self,master):

amount\_label = tk.Label(self.fine\_frame, bg="white", fg="black", text="Fine Amount ", font=("Courier", "14", "bold"))

amount\_label.place(relx=0.05, rely=0.65)

self.amount\_entry = tk.Label(self.fine\_frame, text="", fg="black", bg="white", font=("Courier", "14"))

self.amount\_entry.place(relx=0.25, rely=0.65)

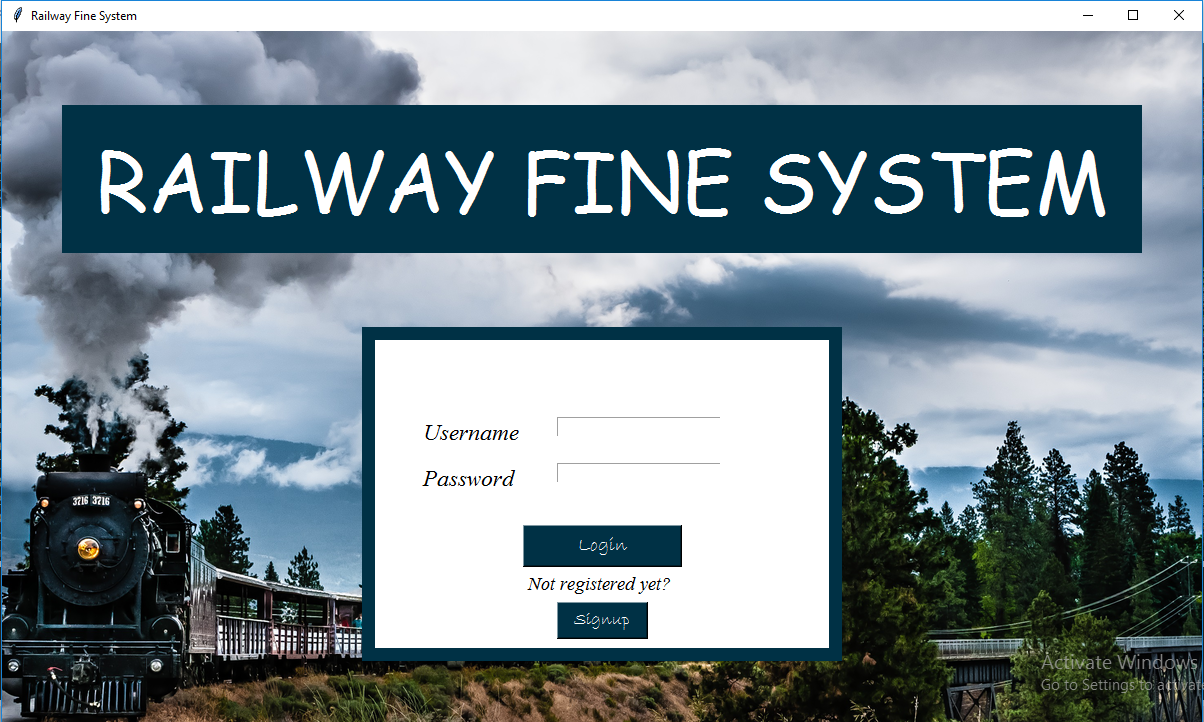
self.master.firstfine()

self.submit\_button = tk.Button(self.fine\_frame, text="Next >", fg="white", bg="#003145", width=20, font=("Courier", "13"), command=self.master.clicksubmit)

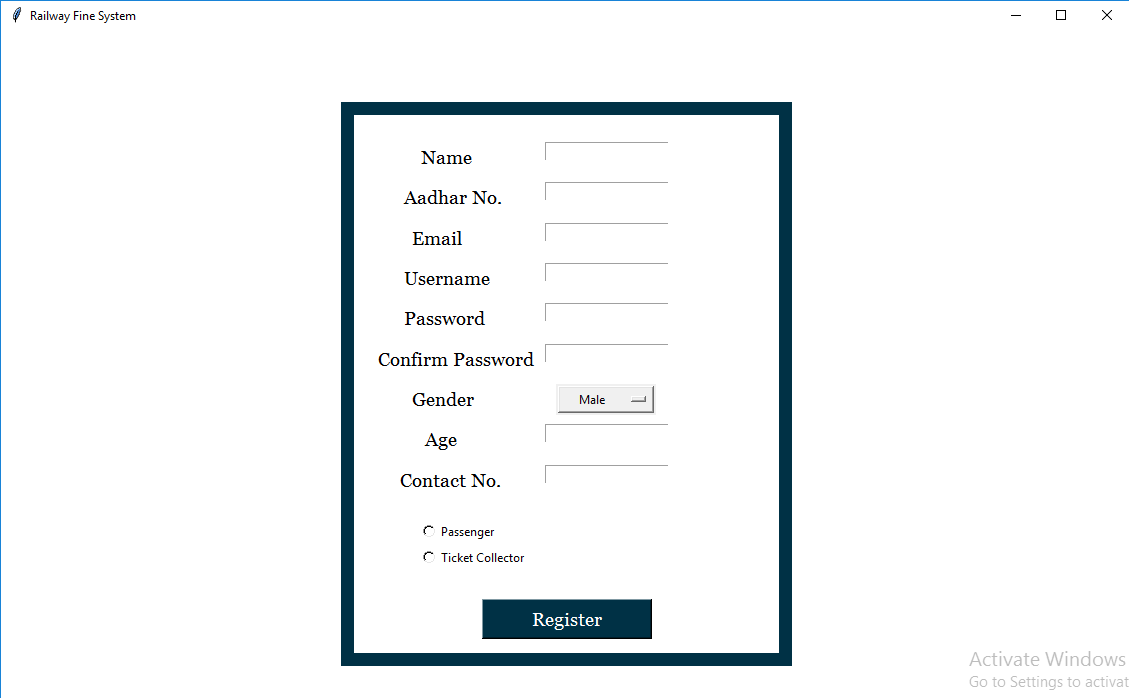
self.submit\_button.place(relx=0.5, rely=0.70)

**OUTPUT**

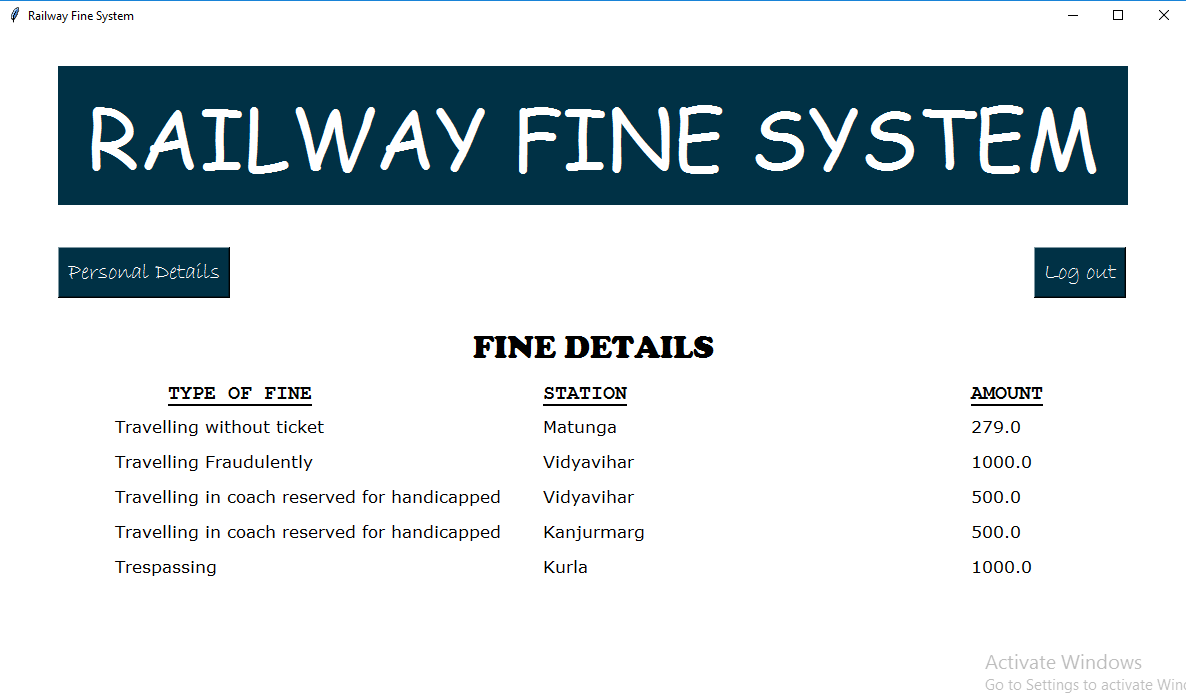
**Login\_page:**

****

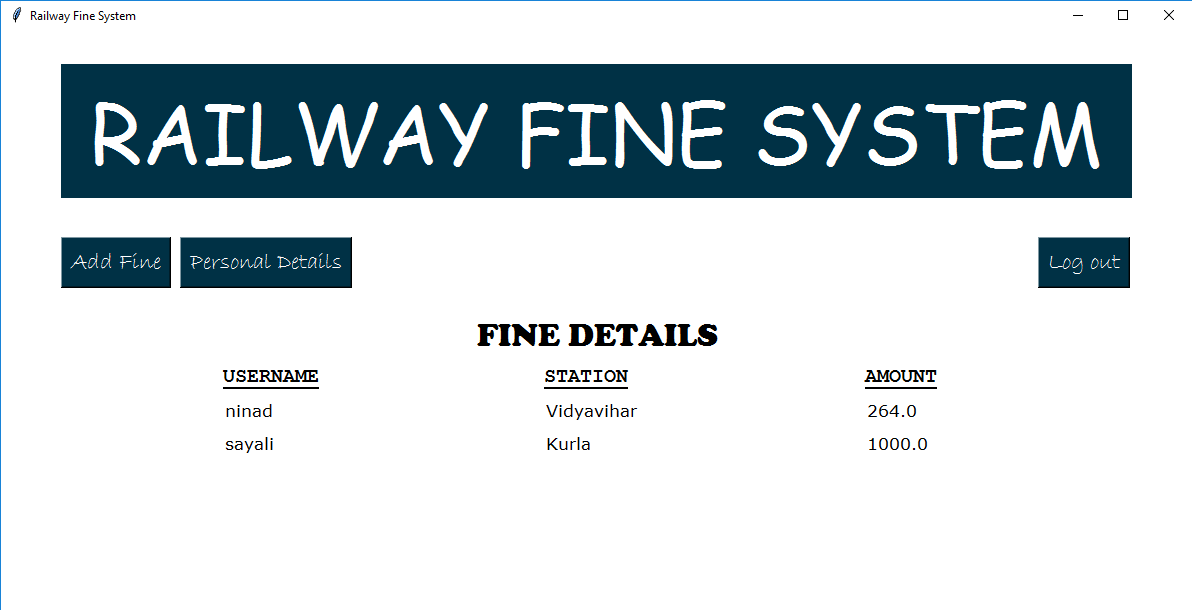
**Signup\_page:**

****

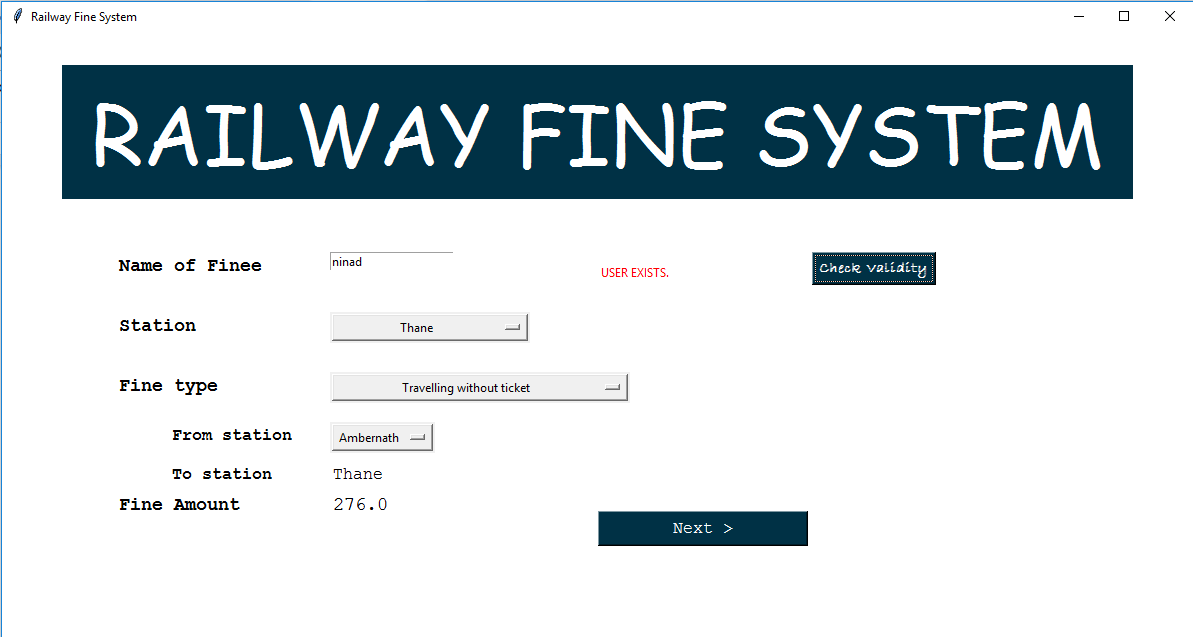
**Passenger\_Logged\_in:**

****

**TC\_Logged\_in:**

****

**Add\_fine:**

****