

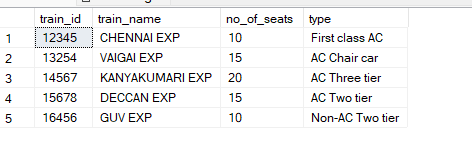
**CODE:**

**­­**import tkinter as tk  
from typing import List  
import pyodbc  
from tkinter import \*  
from tkinter import ttk  
import tkinter.messagebox as messagebox  
  
conn = pyodbc.connect(  
 "Driver={SQL Server Native Client 11.0};"  
 "Server=(LocalDb)\MSSQLLocalDB;"  
 "Database=test;"  
 "Trusted\_Connection=yes;")  
  
  
def inquire():  
 def last\_exit():  
 messagebox.showinfo("Checked!!", "Thank you!!")  
  
 def search():  
 originn = origin\_id.get()  
 destinationn = destination\_id.get()  
 type = type\_id.get()  
 date = date\_id.get()  
 nooftickets = nooftickets\_id.get()  
 *# origin\_id.set("")  
 # destination\_id.set("")* curr\_search = conn.cursor()  
 s1 = "select train.train\_id,Time from train,weekly\_schedule where From\_place='%s'and To\_place='%s' and type='%s' and DateofJourney='%s' and train.train\_id=weekly\_schedule.train\_id " % (  
 originn, destinationn, type, date)  
 curr\_search.execute(s1)  
 count1 = 0  
 anss = 0  
 name\_id = 0  
 timee = 0  
 depart\_time = 0  
 for i in curr\_search:  
 count1 += 1  
 anss = list(i)  
 name\_id = anss[0]  
 timee = anss[1]  
 if count1 > 0:  
 root5.destroy()  
 root6 = Tk()  
 root6.geometry('200x300')  
 root6.title('Search results')  
 curr\_last = conn.cursor()  
 curr\_neww = conn.cursor()  
 s2 = "select total\_fare,train\_name,train.train\_id from train,fare where train.train\_id=fare.train\_id and train.train\_id='%d'" % (  
 name\_id)  
 s3 = "select departure\_time from station where train\_id='%d'" % (name\_id)  
 curr\_last.execute(s2)  
  
 for i in curr\_last:  
 checkk = list(i)  
 fare\_no = checkk[0]  
 train\_namee = checkk[1]  
 train\_idd = checkk[2]  
 tot\_fare = fare\_no \* nooftickets  
 last\_head1 = Label(root6, text='Train Details').place(x=50, y=10)  
 last\_head2 = Label(root6, text='Train Name-').place(x=10, y=40)  
 last\_head3 = Label(root6, text='Train ID-').place(x=10, y=70)  
 last\_head4 = Label(root6, text='Total Fare-').place(x=10, y=100)  
 last\_head5 = Label(root6, text='Train Starting time-').place(x=10, y=130)  
 last\_head6 = Label(root6, text=train\_namee).place(x=100, y=40)  
 last\_head7 = Label(root6, text=train\_idd).place(x=100, y=70)  
 last\_head8 = Label(root6, text=tot\_fare).place(x=100, y=100)  
 last\_head9 = Label(root6, text=timee).place(x=120, y=130)  
 Button(root6, text='Done', borderwidth=4, command=last\_exit).place(x=100, y=190)  
 curr\_last.close()  
 curr\_neww.execute(s3)  
 for i in curr\_neww:  
 checkk\_1 = list(i)  
 depart\_time = checkk\_1[0]  
 last\_head10 = Label(root6, text="Train departure time-").place(x=10, y=160)  
 last\_head11 = Label(root6, text=depart\_time).place(x=130, y=160)  
  
 else:  
 messagebox.showinfo("message,No trains available!!!")  
 root5.destroy()  
  
 def inquexit():  
 messagebox.showinfo("Team login", "THANK YOU!!!")  
 root5.destroy()  
  
 root5 = Tk()  
 root5.geometry('400x500')  
 root5.title('To search trains')  
 origin\_id = StringVar(root5)  
 destination\_id = StringVar(root5)  
 type\_id = StringVar(root5)  
 date\_id = StringVar(root5)  
 nooftickets\_id = IntVar(root5)  
 headd1 = Label(root5, text='Enter detials').place(x=150, y=20)  
 headd = Label(root5, text='Enter origin').place(x=75, y=100)  
 entry\_1 = Entry(root5, textvariable=origin\_id).place(x=150, y=100)  
 headd2 = Label(root5, text='Enter destination').place(x=60, y=150)  
 entry\_2 = Entry(root5, textvariable=destination\_id).place(x=160, y=150)  
 headd\_3 = Label(root5, text='Enter type').place(x=77, y=200)  
 entry\_3 = Entry(root5, textvariable=type\_id).place(x=150, y=200)  
 headd\_4 = Label(root5, text='Enter date').place(x=77, y=250)  
 entry\_4 = Entry(root5, textvariable=date\_id).place(x=150, y=250)  
 search\_bu = Button(root5, text='SEARCH', borderwidth=4, command=search).place(x=130, y=350)  
 headd\_5 = Label(root5, text="Enter no of tickets").place(x=55, y=300)  
 entry\_5 = Entry(root5, textvariable=nooftickets\_id).place(x=160, y=300)  
 exit\_bu = Button(root5, text='EXIT', borderwidth=4, command=inquexit).place(x=200, y=350)  
  
  
root = tk.Tk()  
root.geometry("400x300")  
root.title("Input User Details")  
  
  
def login():  
 root3 = Tk()  
 root3.geometry('400x250')  
  
 head3 = Label(root3, text="PASSENGER login").place(x=120, y=50)  
 head3\_1 = Label(root3, text="Passenger ID").place(x=75, y=100)  
 head3\_2 = Label(root3, text="Password").place(x=75, y=150)  
  
 def logininto():  
 passenger\_id = p\_id.get()  
 p\_word = pass\_word.get()  
 p\_id.set("")  
 pass\_word.set("")  
 cur3 = conn.cursor()  
 s1 = "select id from user\_reg where id='%s' and r\_password='%s'" % (passenger\_id, p\_word)  
 cur3.execute(s1)  
 c = 0  
 for i in cur3:  
 ans = list(i)  
 c += 1  
 if c == 1:  
 messagebox.showinfo("Team login", "successfully logged In")  
 root3.destroy()  
 root4 = Tk()  
 root4.geometry('400x200')  
  
 head4 = Label(root4, text="PASSENGER DETAIL").place(x=120, y=40)  
 head4\_1 = Label(root4, text=ans).place(x=175, y=80)  
 fin\_b = Button(root4, text='Search trains', borderwidth=4, command=inquire)  
 fin\_b.place(x=145, y=120)  
 else:  
 messagebox.showinfo("Team login", "Invalid credentials!")  
 root3.destroy()  
  
 def loginexit():  
 messagebox.showinfo("Team login", "THANK YOU!!!")  
 root3.destroy()  
  
 p\_id = StringVar(root3)  
 pass\_word = StringVar(root3)  
  
 entry3\_1 = Entry(root3, textvariable=p\_id).place(x=175, y=100)  
 entry3\_2 = Entry(root3, textvariable=pass\_word, show='\*').place(x=175, y=150)  
  
 b3\_1 = Button(root3, text=" LOGIN ", borderwidth=4, command=logininto)  
 b3\_1.place(x=75, y=200)  
  
 b3\_2 = Button(root3, text=" EXIT ", borderwidth=4, command=loginexit)  
 b3\_2.place(x=175, y=200)  
  
 root3.mainloop()  
  
  
def register():  
 def exitt():  
 messagebox.showinfo("THANK YOU FOR REGISTERING", "THANKS")  
 root2.destroy()  
  
 def passengerregister():  
  
 a = passenger\_id.get()  
 b = password.get()  
 cur\_1 = conn.cursor()  
 ins2 = "select \* from user\_reg where id= '%s' or r\_password='%s'" % (a, b)  
 cur\_1.execute(ins2)  
 count = 0  
 for i in cur\_1:  
 count += 1  
 if count > 0:  
 messagebox.showinfo("Unsuccesfull", "Team id or Team Name already exists! Try again")  
 root2.destroy()  
 return  
 else:  
 cur1 = conn.cursor()  
 ins = "insert into user\_reg(id,r\_password) values('%s','%s')" % (a, b)  
 cur1.execute(ins)  
 conn.commit()  
 passenger\_id.set("")  
 password.set("")  
  
 root2 = Tk()  
 root2.geometry('400x300')  
  
 passenger\_id = StringVar(root2)  
 password = StringVar(root2)  
  
 head2 = Label(root2, text="NEW PASSENGER REGISTRATION").place(x=110, y=50)  
 head2\_1 = Label(root2, text=" PASSENGER ID ").place(x=70, y=100)  
 head2\_2 = Label(root2, text=" PASSENGER PASSWORD ").place(x=30, y=150)  
  
 entry2\_1 = Entry(root2, textvariable=passenger\_id).place(x=175, y=100)  
 entry2\_2 = Entry(root2, textvariable=password, show='\*').place(x=175, y=150)  
  
 b2\_1 = Button(root2, text="Register", borderwidth=4, command=passengerregister)  
 b2\_1.place(x=120, y=200)  
  
 b2\_2 = Button(root2, text="Exit", borderwidth=4, command=exitt)  
 b2\_2.place(x=230, y=200)  
  
 root2.mainloop()  
  
  
def main\_exit():  
 messagebox.showinfo("Message", "Thank you")  
 root.destroy()  
  
  
head1 = Label(root, text='For Passenger registration')  
head1.place(x=20, y=50)  
  
b1\_1 = Button(root, text="Passenger regsitration", borderwidth=4, command=register)  
b1\_1.place(x=165, y=45)  
  
headd1 = Label(root, text='For passenger Login')  
headd1.place(x=40, y=120)  
  
b1\_2 = Button(root, text="Passenger Login", borderwidth=4, command=login)  
b1\_2.place(x=175, y=120)  
  
b1\_3 = Button(root, text="Exit", borderwidth=4, command=main\_exit).place(x=175, y=200)  
  
root.mainloop()

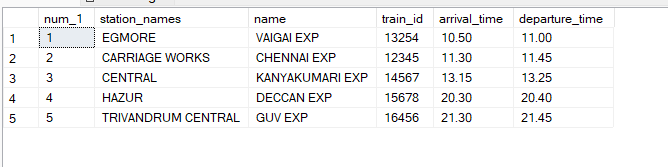
**OUTPUT SCREENSHOTS:**

**TABLES:**

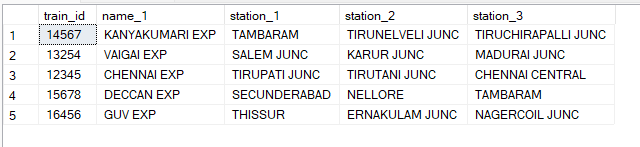
**Train:**

****

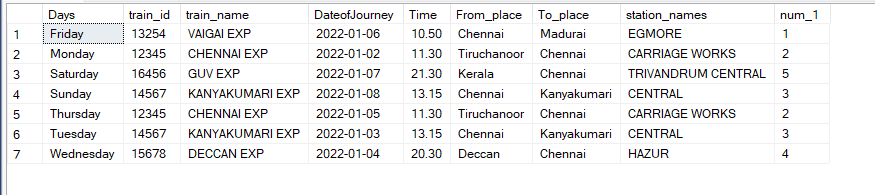
**Station:**



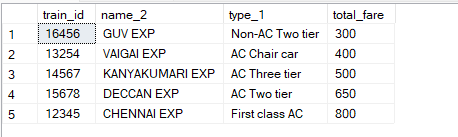
**Passes Through:**



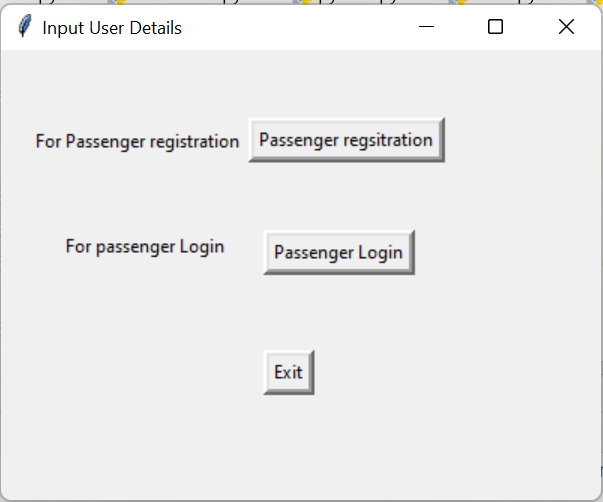
**Weekly Schedule:**

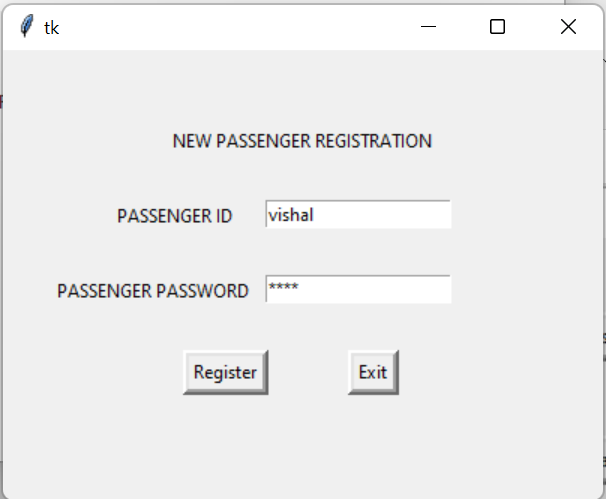


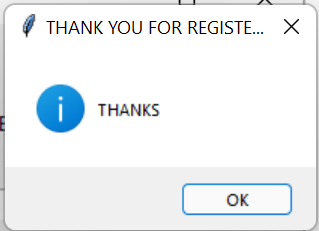
**Fare:**

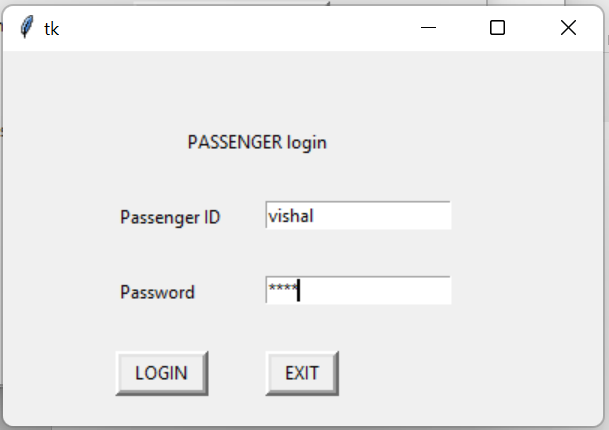


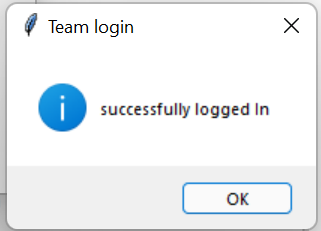
**WINDOWS:**

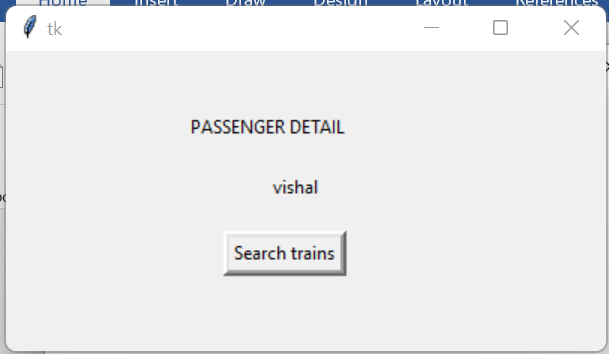
****

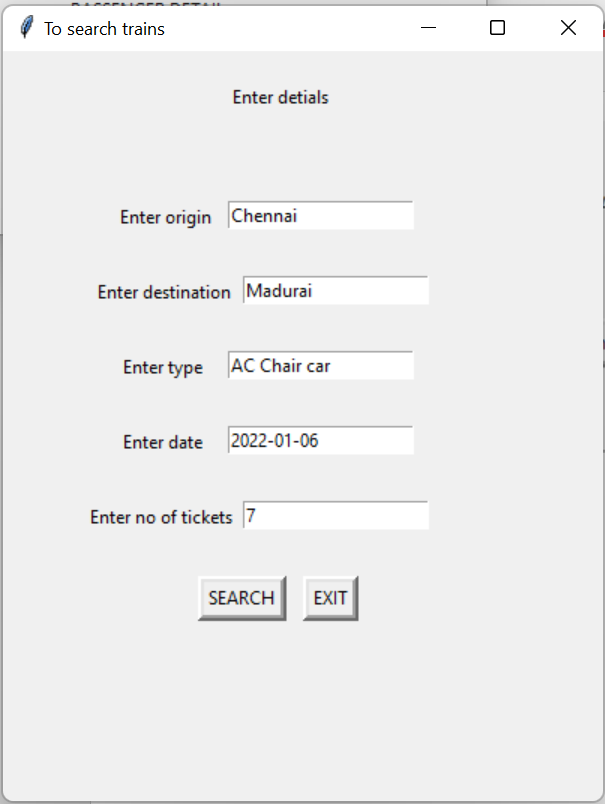
****

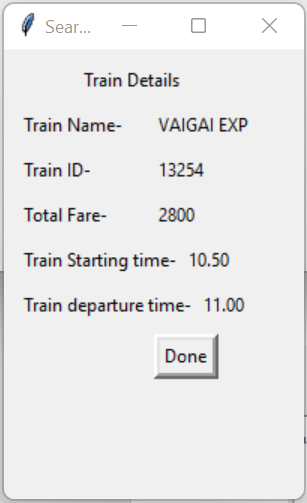
****

****

****

****

****

****

**APPENDIX:**

The [tkinter](https://docs.python.org/3/library/tkinter.html" \l "module-tkinter" \o "tkinter: Interface to Tcl/Tk for graphical user interfaces) package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and [tkinter](https://docs.python.org/3/library/tkinter.html" \l "module-tkinter" \o "tkinter: Interface to Tcl/Tk for graphical user interfaces) are available on most Unix platforms, including macOS, as well as on Windows systems.

Running python -m tkinter from the command line should open a window demonstrating a simple Tk interface, letting you know that [tkinter](https://docs.python.org/3/library/tkinter.html" \l "module-tkinter" \o "tkinter: Interface to Tcl/Tk for graphical user interfaces) is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.

Tkinter supports a range of Tcl/Tk versions, built either with or without thread support. The official Python binary release bundles Tcl/Tk 8.6 threaded. See the source code for the \_tkinter module for more information about supported versions.

Tkinter is not a thin wrapper, but adds a fair amount of its own logic to make the experience more pythonic. This documentation will concentrate on these additions and changes, and refer to the official Tcl/Tk documentation for details that are unchanged.

Button Insertion:

w = Button ( master, option=value, ... )

Entry:

w = Entry( master, option, ... )

Label:

w = Label ( master, option, ... )

MessageBox:

tkMessageBox.FunctionName(title, message [, options])