```
  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-between-
stations?from_code="+from_Station_code+"&from_name="+from_Station_n
ame+
"+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_tra
ins"


# 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

```
berth_type(s):
    if s>0 and s<73:  if s % 8 == 1 or s % 8 == 4:          print (s), "is lower berth"
elif s % 8 == 2 or s % 8 == 5:                          print (s), "is middle berth"
elif s % 8 == 3 or s % 8 == 6:          print (s), "is upper berth"        elif s % 8 ==
7: print (s), "is side lower berth"     else:
print (s), "is side upper berth"     else: print (s), "invalid seat number"

# Driver code s = 10
berth_type(s)      # fxn call for berth type


s = 7
berth_type(s)     # fxn call for berth type
s = 0
berth_type(s)      # fxn call for berth type
```

CONFIRMATION

```
  # import module import requests from bs4 import BeautifulSoup import
pandas as pd

  # user define function # Scrape the data def getdata(url): r =
requests.get(url)
  return r.text

  # input by geek
  train_name = "03391-rajgir-new-delhi-clone-special-rgd-to-ndls"

  # url
  url = "https:/ www.railyatri.in/live-train-status/"+train_name
  # pass the url # into getdata function htmldata = getdata(url) soup =
  BeautifulSoup(htmldata, 'html.parser')

  # traverse the live status from # this Html code data = [] for item in
soup.find_all('script', type="application/ld+json"):
  data.append(item.get_text())

  # convert into dataframe df = pd.read_json(data[2])

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

  print(df["mainEntity"][0]['acceptedAnswer']['text'])
```

TICKET GENERATION

```
   class Ticket:     counter=0
  def__init_(self,passenger_name,source,destination): self._
passenger_name=passenger_name
  self.__________source=source               self.__________destination=destination
self.Counter=Ticket.counter                    Ticket.counter+=1                    def
validate_source_destination(self):
  if (self._source=="Delhi" and (self.__destination=="Pune" or self._
destination=="Mumbai" or self._destination=="Chennai" or self._
destination=="Kolkata")):                    return True        else:
                               return False def generate_ticket(self ):
 if True:
__ticket_id=self._source[0]+self._destination[0]+"0"+str(self. Counter)
```

```
print( "Ticket id will be:",__ticket_id)            else: return False        def
get_ticket_id(self):                                return self.ticket_id     def
get_passenger_name(self):        return self.__passenger_name     def
get_source(self):        if self._source=="Delhi":
return self._source else:
print("you have written invalid soure option")           return None     def
get_destination(self):        if self.__destination=="Pune":            return self.__
destination                  elif self._destination=="Mumbai":
return self.__destination         elif self._destination=="Chennai": return self.__
destination                            elif self._destination=="Kolkata": return self._
destination
else:
        return None
```

OTP GENERATION

```
import os import math import random import smtplib

digits = "0123456789" OTP = ""

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

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

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

```
s.sendmail('&&&&&&',emailid,message)

a = input("Enter your OTP >>: ") if a == OTP: print("Verified") else:
    print("Please Check your OTP again")
```

OTP VERIFICATION

```
import os import math import random import smtplib

digits = "0123456789" OTP = ""
   for i in range (6):
       OTP += digits[math.floor(random.random()*10)] otp = OTP + " is your
       OTP" message = otp
   s = smtplib.SMTP('smtp.gmail.com', 587) s.starttls()

      emailid = input("Enter your email: ")
      s.login("YOUR Gmail ID", "YOUR APP PASSWORD")
      s.sendmail('&&&&&&',emailid,message)

         a = input("Enter your OTP >>: ") if a == OTP: print("Verified") else:
             print("Please Check your OTP again")
```

GitHub link:

https://github.com/IBM-EPBL/IBM-Project-37692-1660317989

Demo Video Link

https://drive.google.com/drive/folders/1jmM3gzodBKQeg3lEd6dEURjcOAZ4urk0