#importing required libraries

Import tkinter as tk

From tkinter import messagebox

From customer\_account import CustomerAccount

From admin import Admin

Import csv

#Creating Empty list to store data

Accounts\_list = []

Admins\_list = []

Class BankSystem(object):

Def \_\_init\_\_(self):

#Initialization

Self.accounts\_list = []

Self.admins\_list = []

#Loading Bank Data

Self.load\_bank\_data()

#Interest Rate

Self.business\_rate=0

Self.savings\_rate=4

#Overdraft Limit

Self.business\_overdraft\_limit=25

Self.savings\_overdraft\_limit=50

# Function to load bank data from csv

Def load\_bank\_data(self):

#Read Customer CSV File

With open(‘Customer.csv’) as f:

Reader = csv.reader(f)

Next(reader)

For row in reader:

Customer=CustomerAccount(row[0],row[1],[row[2],row[3],row[4],row[5]],row[6],row[7],row[8])

Self.accounts\_list.append(customer)

#Read Admin Csv File

With open(‘Admin.csv’) as file:

Reader = csv.reader(file)

Next(reader)

For row in reader:

Adm=Admin(row[0],row[1],[row[2],row[3],row[4],row[5]],row[6],row[7],row[8])

Self.admins\_list.append(adm)

# Function for searching Admins by username

Def search\_admins\_by\_name(self, admin\_username):

#STEP A.2

Found\_admin = None

#searching in the admins list

For a in self.admins\_list:

Username = a.get\_username()

If username == admin\_username:

Found\_admin = a

Break

Return found\_admin

# Function for searching Customers by username

Def search\_customers\_by\_name(self, customer\_lname):

#STEP A.3

Found\_customer = None

For a in self.accounts\_list:

Username = a.get\_last\_name()

If username == customer\_lname:

Found\_customer = a

Break

Return found\_customer

# Function for searching Customers by account number

Def search\_customers\_by\_acc(self, customer\_acc):

Found\_customer = None

For a in self.accounts\_list:

Acc = a.get\_acc\_no()

If acc == customer\_acc:

Found\_customer = a

Return found\_customer

If found\_customer == None:

Messagebox.showinfo(“Error”,”Invalid Credentials!\nTry Again!”)

Return found\_customer

# Helper Function for transferring money

Def transfer\_money\_helper(self, master,sender\_acc\_no, receiver\_account\_no, amount):

#ToDo

Try:

#Destroying Window

Master.destroy()

#Getting Customer object

Sender = self.search\_customers\_by\_acc(sender\_acc\_no)

If sender==None:

Return

Reciever = self.search\_customers\_by\_acc(receiver\_account\_no)

Amount=float(amount)

Overdraft\_limit=0

If sender.get\_account\_type()==’business’:

Overdraft\_limit=self.business\_overdraft\_limit

Elif sender.get\_account\_type()==’savings’:

Overdraft\_limit=self.savings\_overdraft\_limit

#Checking for overdraft limit

If reciever != None and sender != None and sender.get\_balance()+overdraft\_limit>=amount:

Reciever.deposit(amount)

Sender.withdraw(amount)

Messagebox.showinfo(“Success”,”\nSender Updated Balance:%s \nReceivers’ Updated Balance: %s “ %(sender.get\_balance(),reciever.get\_balance()))

Else:

Messagebox.showinfo(“Failure”,”Insufficient Balance in the account”)

Except Exception as e:

Messagebox.showinfo(“Error”,e)

#Function for transferring Money

Def transferMoney(self):

#Creating a new window

Tfwn=tk.Tk()

Tfwn.geometry(“600x300”)

Tfwn.title(“Transfer Money”)

Tfwn.configure(bg=”orange”)

Fr1=tk.Frame(tfwn,bg=”blue”)

L\_title=tk.Message(tfwn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

L1=tk.Label(tfwn,text=”Sender’s Account Number”,relief=”raised”)

L1.pack(side=”top”)

E1=tk.Entry(tfwn)

E1.pack(side=”top”)

L2=tk.Label(tfwn,text=”Receiver Account Number”,relief=”raised”)

L2.pack(side=”top”)

E2=tk.Entry(tfwn)

E2.pack(side=”top”)

L3=tk.Label(tfwn,text=”Enter Amount to be transferred”,relief=”raised”)

L3.pack(side=”top”)

E3=tk.Entry(tfwn)

E3.pack(side=”top”)

#Creating buttons

B1=tk.Button(tfwn,text=”Submit”,command=lambda: self.transfer\_money\_helper(tfwn,e1.get().strip(),e2.get().strip(),e3.get().strip()))

B1.pack(side=”top”)

B2=tk.Button(tfwn,text=”Exit”,command=tfwn.destroy)

B2.pack(side=”top”)

Return

#Function for displaying Management Report

Def get\_management\_report(self):

Try:

Total\_customers=0

Total\_interest\_payable=0

Total\_sum=0

Total\_overdrafts=0

Rate=0

For c in self.accounts\_list:

Total\_customers+=1

If c.get\_account\_type()==’business’:

Rate=self.business\_rate

Elif c.get\_account\_type()==’savings’:

Rate=self.savings\_rate

Total\_interest\_payable+=c.get\_balance()\* rate/100

Balance=c.get\_balance()

If balance>=0:

Total\_sum+=balance

Else:

Total\_overdrafts+=abs(balance)

Messagebox.showinfo(“Info”,”Total Number of Customers: %s \nTotal Money in bank:%s \nTotal Interest Payable:%s \nTotal Overdrafts taken by customers:%s” %(total\_customers,total\_sum,total\_interest\_payable,total\_overdrafts))

Except Exception as e:

Print€

# Helper Function for updating customer name

Def update\_customer\_name\_helper(self,master,sender,fname,sname):

Master.destroy()

Sender.update\_first\_name(fname)

Sender.update\_last\_name(sname)

Messagebox.showinfo(“Name Update Successful”)

# Helper Function for updating customer address

Def update\_customer\_address\_helper(self,master,sender,addr):

Master.destroy()

Sender.update\_address(addr)

Messagebox.showinfo(“Address Update Successful”)

# Function for updating customer name

Def update\_customer\_name(self,sender):

#Creating Window, buttons and labels

Ndwn=tk.Tk()

Ndwn.geometry(“600x300”)

Ndwn.title(“Update Customer Name”)

Ndwn.configure(bg=”orange”)

Fr1=tk.Frame(ndwn,bg=”blue”)

L\_title=tk.Message(ndwn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

L1=tk.Label(ndwn,text=”Enter Customer New First Name:”,relief=”raised”)

L1.pack(side=”top”)

E1=tk.Entry(ndwn)

E1.pack(side=”top”)

L2=tk.Label(ndwn,text=”Enter Customer New Last Name:”,relief=”raised”)

L2.pack(side=”top”)

E2=tk.Entry(ndwn)

E2.pack(side=”top”)

B1=tk.Button(ndwn,text=”Update”,command=lambda: self.update\_customer\_name\_helper(ndwn,sender,e1.get().strip(),e2.get().strip()))

B1.pack(side=”top”)

B2=tk.Button(ndwn,text=”Exit”,command=ndwn.destroy)

B2.pack(side=”top”)

Return

# Function for updating customer address

Def update\_customer\_address(self,sender):

#Creating Window, buttons and labels

Nawn=tk.Tk()

Nawn.geometry(“600x300”)

Nawn.title(“Update Customer Address”)

Nawn.configure(bg=”orange”)

Fr1=tk.Frame(nawn,bg=”blue”)

L\_title=tk.Message(nawn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

Addr=[]

L1=tk.Label(nawn,text=”Enter Customer New House Number:”,relief=”raised”)

L1.pack(side=”top”)

E1=tk.Entry(nawn)

E1.pack(side=”top”)

Addr.append(e1.get().strip())

L2=tk.Label(nawn,text=”Enter Customer New Street name:”,relief=”raised”)

L2.pack(side=”top”)

E2=tk.Entry(nawn)

E2.pack(side=”top”)

Addr.append(e2.get().strip())

L3=tk.Label(nawn,text=”Enter Customer New City Name:”,relief=”raised”)

L3.pack(side=”top”)

E3=tk.Entry(nawn)

E3.pack(side=”top”)

Addr.append(e3.get().strip())

L4=tk.Label(nawn,text=”Enter Customer New Post Code:”,relief=”raised”)

L4.pack(side=”top”)

E4=tk.Entry(nawn)

E4.pack(side=”top”)

Addr.append(e4.get().strip())

B1=tk.Button(nawn,text=”Update”,command=lambda:self.update\_customer\_address\_helper(nawn,sender,addr))

B1.pack(side=”top”)

B2=tk.Button(nawn,text=”Exit”,command=nawn.destroy)

B2.pack(side=”top”)

Return

# Helper Function for depositing money

Def deposit\_money\_helper(self,master,sender,amt):

Master.destroy()

Sender.deposit(float(amt))

Messagebox.showinfo(‘Money deposited’)

# Helper Function for withdrawing money

Def withdraw\_money\_helper(self,master,sender,amt):

Master.destroy()

Overdraft\_limit=0

If sender.get\_account\_type()==’business’:

Overdraft\_limit=25

Elif sender.get\_account\_type()==’savings’:

Overdraft\_limit=50

If sender.get\_balance()+overdraft\_limit<float(amt):

Messagebox.showinfo(“Failure”,”Insufficient Balance in the account”)

Else:

Sender.withdraw(float(amt))

Messagebox.showinfo(‘Money Withdrawn’)

# Function for depositing money

Def deposit\_Money(self,sender):

#Creating Window, buttons and labels

Tfwn=tk.Tk()

Tfwn.geometry(“600x300”)

Tfwn.title(“Deposit Money”)

Tfwn.configure(bg=”orange”)

Fr1=tk.Frame(tfwn,bg=”blue”)

L\_title=tk.Message(tfwn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

L2=tk.Label(tfwn,text=”Enter Amount to be deposited”,relief=”raised”)

L2.pack(side=”top”)

E2=tk.Entry(tfwn)

E2.pack(side=”top”)

B1=tk.Button(tfwn,text=”Deposit”,command=lambda:self.deposit\_money\_helper(tfwn,sender,e2.get().strip()))

B1.pack(side=”top”)

B2=tk.Button(tfwn,text=”Exit”,command=tfwn.destroy)

B2.pack(side=”top”)

Return

# Function for withdrawing money

Def Withdraw(self,sender):

#Creating Window, buttons and labels

Tfwn=tk.Tk()

Tfwn.geometry(“600x300”)

Tfwn.title(“Withdraw Money”)

Tfwn.configure(bg=”orange”)

Fr1=tk.Frame(tfwn,bg=”blue”)

L\_title=tk.Message(tfwn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

L2=tk.Label(tfwn,text=”Enter Amount to be withdrawn”,relief=”raised”)

L2.pack(side=”top”)

E2=tk.Entry(tfwn)

E2.pack(side=”top”)

B1=tk.Button(tfwn,text=”Withdraw”,command=lambda: self.withdraw\_money\_helper(tfwn,sender,e2.get().strip()))

B1.pack(side=”top”)

B2=tk.Button(tfwn,text=”Exit”,command=tfwn.destroy)

B2.pack(side=”top”)

Return

# Function for checking customer account balance

Def check\_balance(self,sender):

Messagebox.showinfo(‘Balance Info’,”Balance: %s” %sender.get\_balance())

Return

# Function for printing the details of particular customer

Def print\_details(self,c):

#Creating Window, buttons and text widget

Root1=tk.Tk()

# specify size of window.

Root1.geometry(“250x170”)

# Create text widget and specify size.

T = tk.Text(root1, height = 5, width = 52)

# Create label

L = tk.Label(root1, text = “Customer Details”)

l.config(font =(“Courier”, 14))

# Create an Exit button.

B2 = tk.Button(root1, text = “Exit”,command = root1.destroy)

l.pack()

T.pack()

B2.pack()

Data=””

Data+= “\nAccount Number:”+ str(c.get\_acc\_no())

Data+=”\n”+ “First name:”+ str(c.get\_first\_name())

Data+=”\n”+”Last name:” +str(c.get\_last\_name())

Address=c.get\_address()

Data+=”\n”+”Address:”+str(address[0])

Data+=”\n”+str(c.address[1])

Data+=”\n”+str(c.address[2])

Data+=”\n”+str(c.address[3])

Data+=’\n’

# Insert The Fact.

T.insert(tk.END,data)

# Function for signing out

Def sign\_out(self,master,admin):

Master.destroy()

Self.run\_admin\_options(admin)

# Function for choosing options for particular customer

Def run\_account\_options(self,sender,admin):

#Creating Window, buttons and labels

Cawn=tk.Tk()

Cawn.geometry(“700x400”)

Cawn.title(“Customer Account cations”)

Cawn.configure(bg=”orange”)

Fr1=tk.Frame(cawn,bg=”blue”)

L\_title=tk.Message(cawn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

B1=tk.Button(text=”Deposit Money”,command=lambda: self.deposit\_Money(sender))

B2=tk.Button(text=”Withdraw”,command=lambda:self.Withdraw(sender))

B3=tk.Button(text=”Check balance”,command=lambda: self.check\_balance(sender))

B4=tk.Button(text=”Show customer details”,command=lambda: self.print\_details(sender))

B5=tk.Button(text=”Update Customer Name”,command=lambda: self.update\_customer\_name(sender))

B6=tk.Button(text=”Update Customer Address”,command=lambda: self.update\_customer\_address(sender))

B7=tk.Button(text=”Sign out”,command=lambda: self.sign\_out(cawn,admin))

B1.place(x=100,y=100)

B2.place(x=400,y=100)

B3.place(x=100,y=200)

B4.place(x=400,y=200)

B5.place(x=100,y=300)

B6.place(x=300,y=300)

B7.place(x=500,y=300)

Cawn.mainloop()

# Helper Function for customer operations

Def helper(self,master,master1,acc,admin):

Master.destroy()

Sender = self.search\_customers\_by\_acc(acc)

If sender==None:

Return

Master1.destroy()

Self.run\_account\_options(sender,admin)

# Function for customer operations

Def customer\_operations(self,master,admin):

#Creating Window, buttons and labels

Tfwn=tk.Tk()

Tfwn.geometry(“600x300”)

Tfwn.title(“Customer Operations”)

Tfwn.configure(bg=”orange”)

Fr1=tk.Frame(tfwn,bg=”blue”)

L\_title=tk.Message(tfwn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

L2=tk.Label(tfwn,text=”Enter Account Number”,relief=”raised”)

L2.pack(side=”top”)

E2=tk.Entry(tfwn)

E2.pack(side=”top”)

B1=tk.Button(tfwn,text=”Submit”,command=lambda:self.helper(tfwn,master,e2.get().strip(),admin))

B1.pack(side=”top”)

B2=tk.Button(tfwn,text=”Exit”,command=tfwn.destroy)

B2.pack(side=”top”)

Return

# Helper Function for deleting record

Def delete\_helper(self,master,acc\_no):

Master.destroy()

Customer\_account = self.search\_customers\_by\_acc(acc\_no)

If customer\_account != None:

Self.accounts\_list.remove(customer\_account)

Messagebox.showinfo(“Success”,”%s was deleted successfully!” %customer\_account.get\_first\_name())

# Function for deleting customer from records

Def delete\_customer(self,admin\_obj):

#Creating Window, buttons and labels

If admin\_obj.has\_full\_admin\_right():

Dcwn=tk.Tk()

Dcwn.geometry(“600x300”)

Dcwn.title(“Delete Customer Account”)

Dcwn.configure(bg=”orange”)

Fr1=tk.Frame(dcwn,bg=”blue”)

L\_title=tk.Message(dcwn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

L1=tk.Label(dcwn,text=”Enter Customer Account Number:”,relief=”raised”)

L1.pack(side=”top”)

E1=tk.Entry(dcwn)

E1.pack(side=”top”)

B1=tk.Button(dcwn,text=”Delete”,command=lambda: self.delete\_helper(dcwn,e1.get().strip()))

B1.pack(side=”top”)

B2=tk.Button(dcwn,text=”Exit”,command=dcwn.destroy)

B2.pack(side=”top”)

Return

Else:

Messagebox.showinfo(“Error”,”Admin %s %s does not have full admin rights” %(admin\_obj.get\_first\_name(), admin\_obj.get\_last\_name()))

Return

# Helper Function for updating the admin name

Def update\_admin\_name\_helper(self,master,admin\_obj,fname,sname):

Master.destroy()

Admin\_obj.update\_first\_name(fname)

Admin\_obj.update\_last\_name(sname)

Messagebox.showinfo(“Name Update Successful”)

# Helper Function for updating the admin address

Def update\_admin\_address\_helper(self,master,admin\_obj,addr):

Master.destroy()

Admin\_obj.update\_address(addr)

Messagebox.showinfo(“Address Update Successful”)

# Function for updating the admin name

Def update\_admin\_name(self,admin\_obj):

#Creating Window, buttons and labels

Ndwn=tk.Tk()

Ndwn.geometry(“600x300”)

Ndwn.title(“Update Admin Name”)

Ndwn.configure(bg=”orange”)

Fr1=tk.Frame(ndwn,bg=”blue”)

L\_title=tk.Message(ndwn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

L1=tk.Label(ndwn,text=”Enter Admin New First Name:”,relief=”raised”)

L1.pack(side=”top”)

E1=tk.Entry(ndwn)

E1.pack(side=”top”)

L2=tk.Label(ndwn,text=”Enter Admin New Last Name:”,relief=”raised”)

L2.pack(side=”top”)

E2=tk.Entry(ndwn)

E2.pack(side=”top”)

B1=tk.Button(ndwn,text=”Update”,command=lambda: self.update\_admin\_name\_helper(ndwn,admin\_obj,e1.get().strip(),e2.get().strip()))

B1.pack(side=”top”)

B2=tk.Button(ndwn,text=”Exit”,command=ndwn.destroy)

B2.pack(side=”top”)

Return

# Function for updating the admin address

Def update\_admin\_address(self,admin\_obj):

#Creating Window, buttons and labels

Nawn=tk.Tk()

Nawn.geometry(“600x300”)

Nawn.title(“Update Admin Address”)

Nawn.configure(bg=”orange”)

Fr1=tk.Frame(nawn,bg=”blue”)

L\_title=tk.Message(nawn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

Addr=[]

L1=tk.Label(nawn,text=”Enter Admin New House Number:”,relief=”raised”)

L1.pack(side=”top”)

E1=tk.Entry(nawn)

E1.pack(side=”top”)

Addr.append(e1.get().strip())

L2=tk.Label(nawn,text=”Enter Admin New Street name:”,relief=”raised”)

L2.pack(side=”top”)

E2=tk.Entry(nawn)

E2.pack(side=”top”)

Addr.append(e2.get().strip())

L3=tk.Label(nawn,text=”Enter Admin New City Name:”,relief=”raised”)

L3.pack(side=”top”)

E3=tk.Entry(nawn)

E3.pack(side=”top”)

Addr.append(e3.get().strip())

L4=tk.Label(nawn,text=”Enter Admin New Post Code:”,relief=”raised”)

L4.pack(side=”top”)

E4=tk.Entry(nawn)

E4.pack(side=”top”)

Addr.append(e4.get().strip())

B1=tk.Button(nawn,text=”Update”,command=lambda:self.update\_admin\_address\_helper(nawn,admin\_obj,addr))

B1.pack(side=”top”)

B2=tk.Button(nawn,text=”Exit”,command=nawn.destroy)

B2.pack(side=”top”)

# nawn.destroy()

Return

# Function for Printing all account details

Def print\_all\_accounts\_details(self):

#Creating Window, buttons and labels

Root=tk.Tk()

# specify size of window.

Root.geometry(“250x170”)

# Create text widget and specify size.

T = tk.Text(root, height = 5, width = 52)

# Create label

L = tk.Label(root, text = “Customer Details”)

l.config(font =(“Courier”, 14))

# Create an Exit button.

B2 = tk.Button(root, text = “Exit”,command = root.destroy)

l.pack()

T.pack()

B2.pack()

Data=””

I = 0

For c in self.accounts\_list:

I+=1

Data=data+’\n’+ str(i)

Data+= “\nAccount Number:”+ str(c.get\_acc\_no())

Data+=”\n”+ “First name:”+ str(c.get\_first\_name())

Data+=”\n”+”Last name:” +str(c.get\_last\_name())

Address=c.get\_address()

Data+=”\n”+”Address:”+str(address[0])

Data+=”\n”+str(c.address[1])

Data+=”\n”+str(c.address[2])

Data+=”\n”+str(c.address[3])

Data+=’\n’

# Insert The Data.

T.insert(tk.END,data)

# Function for Checking admin credentials

Def admin\_login(self,master,username,password):

Found\_admin = self.search\_admins\_by\_name(username)

If found\_admin == None or found\_admin.get\_password() != password:

Found\_admin=None

Messagebox.showinfo(“Login Failed”,”Invalid Credentials\nPlease try again.”)

Master.destroy()

Self.Main\_Menu()

# messagebox.showinfo(“Login Succesfull”)

Master.destroy()

Self.run\_admin\_options(found\_admin)

# Function for Calling the menu of the options available with admin

Def run\_admin\_options(self,admin\_obj):

#Creating Window, buttons and labels

Opwn=tk.Tk()

Opwn.geometry(“700x400”)

Opwn.title(“Admin Options”)

Opwn.configure(bg=”orange”)

Fr1=tk.Frame(opwn,bg=”blue”)

L\_title=tk.Message(opwn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

B1=tk.Button(text=”Tranfer Money”,command=self.transferMoney)

B2=tk.Button(text=”Customer account operations & profile settings”,command=lambda: self.customer\_operations(opwn,admin\_obj))

B3=tk.Button(text=”Delete customer”,command=lambda: self.delete\_customer(admin\_obj))

B4=tk.Button(text=”Print all customers details”,command=self.print\_all\_accounts\_details)

B5=tk.Button(text=”Update Admin Name”,command=lambda: self.update\_admin\_name(admin\_obj))

B6=tk.Button(text=”Update Admin Address”,command=lambda: self.update\_admin\_address(admin\_obj))

B7=tk.Button(text=”Get Management Report”,command=self.get\_management\_report)

B8=tk.Button(text=”Sign out”,command=opwn.destroy)

B1.place(x=100,y=100)

B2.place(x=200,y=100)

B3.place(x=500,y=100)

B4.place(x=100,y=200)

B5.place(x=300,y=200)

B6.place(x=500,y=200)

B7.place(x=100,y=300)

B8.place(x=400,y=300)

Opwn.mainloop()

# Function for Calling the login windoe for admin

Def login(self,master):

#Creating Window, buttons and labels

Master.destroy()

Crwn=tk.Tk()

Crwn.geometry(“600x300”)

Crwn.title(“Admin Login”)

Crwn.configure(bg=”orange”)

Fr1=tk.Frame(crwn,bg=”blue”)

L\_title=tk.Message(crwn,text=”Python Banking System”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

L1=tk.Label(crwn,text=”Enter Admin username:”,relief=”raised”)

L1.pack(side=”top”)

E1=tk.Entry(crwn)

E1.pack(side=”top”)

L2=tk.Label(crwn,text=”Enter Admin password”,relief=”raised”)

L2.pack(side=”top”)

E2=tk.Entry(crwn,show=”\*”)

E2.pack(side=”top”)

B=tk.Button(crwn,text=”Submit”,command=lambda: self.admin\_login(crwn,e1.get().strip(),e2.get().strip()))

b.pack(side=”top”)

# Function for Calling the main menu

Def Main\_Menu(self):

#Creating Window, buttons and labels

Rootwn=tk.Tk()

Rootwn.geometry(“700x300”)

Rootwn.title(“Python Banking System”)

Rootwn.configure(background=’orange’)

Fr1=tk.Frame(rootwn)

Fr1.pack(side=”top”)

L\_title=tk.Message(text=”Python BANKING\n SYSTEM”,relief=”raised”,width=2000,padx=600,pady=0,fg=”white”,bg=”black”,justify=”center”,anchor=”center”)

L\_title.config(font=(“Courier”,”50”,”bold”))

L\_title.pack(side=”top”)

B1=tk.Button(text=”Admin Login”,command=lambda: self.login(rootwn))

B2=tk.Button(text=”Quit Python Bank System”,command=rootwn.destroy)

B1.place(x=100,y=200)

B2.place(x=400,y=200)

Rootwn.mainloop()

#Creating instance of BankSystem Class

App = BankSystem()

#Calling Main\_Menu function

App.Main\_Menu()