ATM MACHINE

CLASS XI COMPUTER SCIENCE PROJECT

SUBMITTED BY : RUPANJAN BHATTACHARYYA

ROLL NO.-27

2021 — 2022

TOPIC OF THE PROJECT

A Python application for demonstrating the working principle of an ATM Machine

Submitted by: RUPANJAN BHATTACHARYYA
Class XI, Roll No.:27
Under the supervision of : Mr. Swarup Kumar Maity
DAV Model School,IIT Kharagpur-721302

CERTIFICATE

This is to certify that the project entitled "ATM " is a bona fide record of the work carried out by **RUPANJAN BHATTACHARYYA, Class XI, Roll No.-27** under my guidance and supervision.

Mr. Swarup Kumar Maity
PGT,Computer Science

Mrs.Lopa Chatterjee
Principal

External Examiner

Acknowledgement

I would like to express my special thanks of gratitude to my teacher **Mr. Swarup Kumar Maity** as well as our Principal **Mrs. Lopa Chatterjee** who gave me a wonderful opportunity to do this amazing project on the topic **ATM** which has really helped me in doing a lot of research and I came to know about so many new things and I am really thankful to them. I would also like to thank my parents and friends who helped me a lot in completing the project within the given time .

Rupanjan Bhattacharyya

DAV MODEL SCHOOL, IIT KHARAGPUR

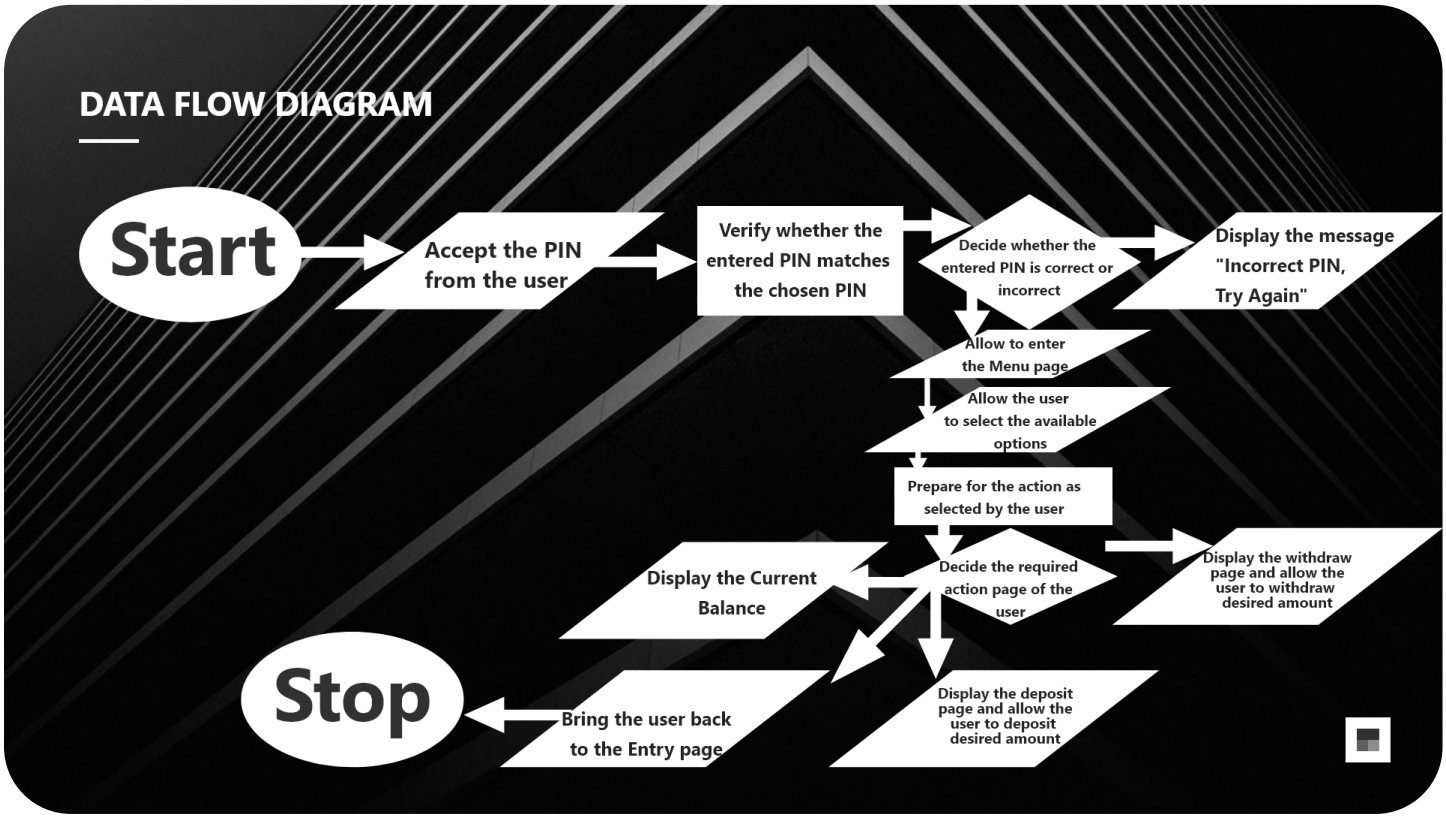
Date:

INDEX

About The Project

- This project mainly outlines the working principle of an ATM machine and can be managed through computer system in organizations like banks.
- This project is made with the help of python programming language and tkinter system to store and extract the records related to the managing an account.
- The data in this project includes objectives like cash withdrawal, checking the balance, deposting cash within your bank account, transaction details, etc. Packages are also installed for further changes.
- All the commmands are run and are maintained by the python and thus it is a complete secure process. The user just needs to make certain choices at different steps and the work will be done as per the choices.

Data Flow Diagram



Program Code

February 13, 2022

[ ]:
```
from tkinter import *
from PIL import ImageTk, Image
import time

current_balance = 1000
password=0
class SampleApp(Tk):

    def __init__(self, *args, **kwargs):
        Tk.__init__(self, *args, **kwargs)
        self.shared_data = {'Balance':IntVar(),'Password':IntVar()}
        container = Frame(self)
        container.pack(side="top", fill="both", expand=True)
        container.grid_rowconfigure(0, weight=1)
        container.grid_columnconfigure(0, weight=1)

        self.frames = {}
        for F in (SelectionPage, StartPage, MenuPage, WithdrawPage,␣
 ↪DepositPage, BalancePage):
            page_name = F.__name__
            frame = F(parent=container, controller=self)
            self.frames[page_name] = frame
            frame.grid(row=0, column=0, sticky="nsew")

        self.show_frame("SelectionPage")

    def show_frame(self, page_name):
        '''Show a frame for the given page name'''
        frame = self.frames[page_name]
        frame.tkraise()

class SelectionPage(Frame):

    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        self.controller = controller
        global bgi
```

```
        global n1i
        global n2i
        global n3i
        global n4i
        global n5i
        global n6i
        global n7i
        global n8i
        global n9i
        global n0i
        global nci
        global nei
        self.controller.title('ATM-Machine')
        self.controller.state('zoomed')
        self.controller.iconphoto(False,PhotoImage(file='images/atm-machine.
↪png'))
        bgi=ImageTk.PhotoImage(Image.open("images/ATM software layout.png"))
        my_canvas = Canvas(self, width=960, height=700, bd=0,␣
↪highlightthickness=0)
        my_canvas.pack()
        my_canvas.create_image(0,0, image=bgi, anchor="nw")
        my_canvas.create_text(390, 230, text="Welcome !", font=("Helvetica",␣
↪26, "bold"))
        my_canvas.create_text(390, 335, text="Please enter a PIN of your choice␣
↪with maximum 9 digits", font=("Helvetica", 12, "bold"))
        my_canvas.create_text(390, 360, text="Click on the input box to hide␣
↪your PIN", font=("Helvetica", 14))
        my_canvas.create_text(390, 390, text="Please use the KEY PAD to enter␣
↪PIN", font=("Helvetica", 18))
        pw=StringVar()
        pin_select= Entry(self, textvariable=pw, font=("Helvetica", 24,␣
↪"bold"), width=6, bd=5)
        def handle_focus_in(_):
            pin_select.configure(fg='black',show='*')
        pin_select.bind('<FocusIn>',handle_focus_in)

        my_canvas.create_window(330,270, anchor="nw", window=pin_select)



        def button_click(number):
            current=pin_select.get()
            pin_select.delete(0, END)
            pin_select.insert(0, str(current)+str(number))
        def button_clear():
            pin_select.delete(0, END)
```

```
        n1i=ImageTk.PhotoImage(Image.open("images/1.png"))
        img_label1= Label(image=n1i)
        b1= Button(self, image=n1i, borderwidth=0, command=lambda:␣
↪button_click(1))
        l1= Label(self, text='')
        window_1=my_canvas.create_window(275,550, anchor="nw", window=b1)

        n2i=ImageTk.PhotoImage(Image.open("images/2.png"))
        img_label2= Label(image=n2i)
        b2= Button(self, image=n2i, borderwidth=0, command=lambda:␣
↪button_click(2))
        l2= Label(self, text='')
        window_2=my_canvas.create_window(355,550, anchor="nw", window=b2)

        n3i=ImageTk.PhotoImage(Image.open("images/3.png"))
        img_label3= Label(image=n3i)
        b3= Button(self, image=n3i, borderwidth=0, command=lambda:␣
↪button_click(3))
        l3= Label(self, text='')
        window_3=my_canvas.create_window(435,550, anchor="nw", window=b3)

        n4i=ImageTk.PhotoImage(Image.open("images/4.png"))
        img_label4= Label(image=n4i)
        b4= Button(self, image=n4i, borderwidth=0, command=lambda:␣
↪button_click(4))
        l4= Label(self, text='')
        window_4=my_canvas.create_window(275,585, anchor="nw", window=b4)

        n5i=ImageTk.PhotoImage(Image.open("images/5.png"))
        img_label5= Label(image=n5i)
        b5= Button(self, image=n5i, borderwidth=0, command=lambda:␣
↪button_click(5))
        l5= Label(self, text='')
        window_5=my_canvas.create_window(355,585, anchor="nw", window=b5)

        n6i=ImageTk.PhotoImage(Image.open("images/6.png"))
        img_label6= Label(image=n6i)
        b6= Button(self, image=n6i, borderwidth=0, command=lambda:␣
↪button_click(6))
        l6= Label(self, text='')
        window_6=my_canvas.create_window(435,585, anchor="nw", window=b6)

        n7i=ImageTk.PhotoImage(Image.open("images/7.png"))
        img_label7= Label(image=n7i)
        b7= Button(self, image=n7i, borderwidth=0, command=lambda:␣
↪button_click(7))
```

```
        l7= Label(self, text='')
        window_7=my_canvas.create_window(275,620, anchor="nw", window=b7)

        n8i=ImageTk.PhotoImage(Image.open("images/8.png"))
        img_label8= Label(image=n8i)
        b8= Button(self, image=n8i, borderwidth=0, command=lambda:␣
↪button_click(8))
        l8= Label(self, text='')
        window_8=my_canvas.create_window(355,620, anchor="nw", window=b8)

        n9i=ImageTk.PhotoImage(Image.open("images/9.png"))
        img_label9= Label(image=n9i)
        b9= Button(self, image=n9i, borderwidth=0, command=lambda:␣
↪button_click(9))
        l9= Label(self, text='')
        window_9=my_canvas.create_window(435,620, anchor="nw", window=b9)

        n0i=ImageTk.PhotoImage(Image.open("images/0.png"))
        img_label0= Label(image=n0i)
        b0= Button(self, image=n0i, borderwidth=0, command=lambda:␣
↪button_click(0))
        l0= Label(self, text='')
        window_0=my_canvas.create_window(355,655, anchor="nw", window=b0)

        nci=ImageTk.PhotoImage(Image.open("images/clear.png"))
        img_labelc= Label(image=nci)
        bc= Button(self, image=nci, borderwidth=0, command=button_clear)
        lc= Label(self, text='')
        window_c=my_canvas.create_window(275,655, anchor="nw", window=bc)

        def button_enter():
            global password
            password += int(pw.get())
            controller.shared_data['Password'].set(password)
            controller.show_frame('StartPage')

        nei=ImageTk.PhotoImage(Image.open("images/enter.png"))
        img_labele= Label(image=nei)
        be= Button(self, image=nei, borderwidth=0, command=button_enter)
        le= Label(self, text='')
        window_e=my_canvas.create_window(435,655, anchor="nw", window=be)


class StartPage(Frame):

    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
```

```
        self.controller = controller
        global bg
        global n1
        global n2
        global n3
        global n4
        global n5
        global n6
        global n7
        global n8
        global n9
        global n0
        global nc
        global ne
        self.controller.title('ATM-Machine')
        self.controller.state('zoomed')
        self.controller.iconphoto(False,PhotoImage(file='images/atm-machine.
↪png'))
        bg=ImageTk.PhotoImage(Image.open("images/ATM software layout.png"))
        my_canvas = Canvas(self, width=960, height=700, bd=0,␣
↪highlightthickness=0)
        my_canvas.pack()
        my_canvas.create_image(0,0, image=bg, anchor="nw")
        my_canvas.create_text(390, 230, text="Please enter your PIN",␣
↪font=("Helvetica", 24, "bold"))
        my_canvas.create_text(390, 200, text="*Enter the PIN you chose in the␣
↪welcome page*", font=("Helvetica", 14, "bold"))
        my_canvas.create_text(390, 380, text="Click on the input box to hide␣
↪your PIN", font=("Helvetica", 14))
        my_canvas.create_text(390, 410, text="Please use the KEY PAD to enter␣
↪PIN", font=("Helvetica", 14))
        pin_entry= Entry(self, font=("Helvetica", 24, "bold"), width=6, bd=5)
        def handle_focus_in(_):
            pin_entry.configure(fg='black',show='*')
        pin_entry.bind('<FocusIn>',handle_focus_in)

        my_canvas.create_window(330,270, anchor="nw", window=pin_entry)

        def button_click(number):
            current=pin_entry.get()
            pin_entry.delete(0, END)
            pin_entry.insert(0, str(current)+str(number))
        def button_clear():
            pin_entry.delete(0, END)
        n1=ImageTk.PhotoImage(Image.open("images/1.png"))
        img_label1= Label(image=n1)
```

```
        b1= Button(self, image=n1, borderwidth=0, command=lambda:␣
↪button_click(1))
        l1= Label(self, text='')
        window_1=my_canvas.create_window(275,550, anchor="nw", window=b1)

        n2=ImageTk.PhotoImage(Image.open("images/2.png"))
        img_label2= Label(image=n2)
        b2= Button(self, image=n2, borderwidth=0, command=lambda:␣
↪button_click(2))
        l2= Label(self, text='')
        window_2=my_canvas.create_window(355,550, anchor="nw", window=b2)

        n3=ImageTk.PhotoImage(Image.open("images/3.png"))
        img_label3= Label(image=n3)
        b3= Button(self, image=n3, borderwidth=0, command=lambda:␣
↪button_click(3))
        l3= Label(self, text='')
        window_3=my_canvas.create_window(435,550, anchor="nw", window=b3)

        n4=ImageTk.PhotoImage(Image.open("images/4.png"))
        img_label4= Label(image=n4)
        b4= Button(self, image=n4, borderwidth=0, command=lambda:␣
↪button_click(4))
        l4= Label(self, text='')
        window_4=my_canvas.create_window(275,585, anchor="nw", window=b4)

        n5=ImageTk.PhotoImage(Image.open("images/5.png"))
        img_label5= Label(image=n5)
        b5= Button(self, image=n5, borderwidth=0, command=lambda:␣
↪button_click(5))
        l5= Label(self, text='')
        window_5=my_canvas.create_window(355,585, anchor="nw", window=b5)

        n6=ImageTk.PhotoImage(Image.open("images/6.png"))
        img_label6= Label(image=n6)
        b6= Button(self, image=n6, borderwidth=0, command=lambda:␣
↪button_click(6))
        l6= Label(self, text='')
        window_6=my_canvas.create_window(435,585, anchor="nw", window=b6)

        n7=ImageTk.PhotoImage(Image.open("images/7.png"))
        img_label7= Label(image=n7)
        b7= Button(self, image=n7, borderwidth=0, command=lambda:␣
↪button_click(7))
        l7= Label(self, text='')
        window_7=my_canvas.create_window(275,620, anchor="nw", window=b7)
```

```
        n8=ImageTk.PhotoImage(Image.open("images/8.png"))
        img_label8= Label(image=n8)
        b8= Button(self, image=n8, borderwidth=0, command=lambda:␣
↪button_click(8))
        l8= Label(self, text='')
        window_8=my_canvas.create_window(355,620, anchor="nw", window=b8)

        n9=ImageTk.PhotoImage(Image.open("images/9.png"))
        img_label9= Label(image=n9)
        b9= Button(self, image=n9, borderwidth=0, command=lambda:␣
↪button_click(9))
        l9= Label(self, text='')
        window_9=my_canvas.create_window(435,620, anchor="nw", window=b9)

        n0=ImageTk.PhotoImage(Image.open("images/0.png"))
        img_label0= Label(image=n0)
        b0= Button(self, image=n0, borderwidth=0, command=lambda:␣
↪button_click(0))
        l0= Label(self, text='')
        window_0=my_canvas.create_window(355,655, anchor="nw", window=b0)

        nc=ImageTk.PhotoImage(Image.open("images/clear.png"))
        img_labelc= Label(image=nc)
        bc= Button(self, image=nc, borderwidth=0, command=button_clear)
        lc= Label(self, text='')
        window_c=my_canvas.create_window(275,655, anchor="nw", window=bc)


        def button_enter():
            number=pin_entry.get()
            global password
            controller.shared_data['Password'].set(password)
            pin=int(number)
            if pin==password:
                ip=Label(self, text="", font=("Helvetica", 24, "bold"),␣
↪foreground="red", background="#FFFFFF")
                my_canvas.create_window(390,340, width=400, height=40,␣
↪window=ip)
                controller.show_frame('MenuPage')
                pin_entry.delete(0, END)
            else:
                ip=Label(self, text="Incorrect PIN! Try Again",␣
↪font=("Helvetica", 24, "bold"), foreground="red", background="#FFFFFF")
                pin_entry.delete(0, END)
                my_canvas.create_window(390,340, window=ip)
```

```
        ne=ImageTk.PhotoImage(Image.open("images/enter.png"))
        img_labele= Label(image=ne)
        be= Button(self, image=ne, borderwidth=0, command=button_enter)
        le= Label(self, text='')
        window_e=my_canvas.create_window(435,655, anchor="nw", window=be)

class MenuPage(Frame):

    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        self.controller = controller
        global c2
        c2=ImageTk.PhotoImage(Image.open("images/canvas2.png"))
        canvas = Canvas(self, width=960, height=700, bd=0, highlightthickness=0)
        canvas.create_image(0,0, image=c2, anchor="nw")
        canvas.pack()

        frame1 = Frame(self,width=696, height=430, bg='#FF7F50' )
        frame1.pack()
        canvas.create_window(486,300, window=frame1)

        heading_label = Label(frame1,
                                                text='NEUTRINOVAULT ATM',
                                                font=('orbitron',40,'bold'),
                                                foreground='#ffffff',
                                                background='#FF7F50')
        heading_label.pack(pady=18)

        main_menu_label = Label(frame1,
                                                    text='Main Menu',
                                                    font=('orbitron',13),
                                                    fg='white',
                                                    bg='#FF7F50')
        main_menu_label.pack()

        selection_label = Label(frame1,
                                                    text='Please make a␣
 ↪selection',
                                                    font=('orbitron',13),
                                                    fg='white',
                                                    bg='#FF7F50',
                                                    anchor='w')
        selection_label.pack(fill='x')

        button_frame = Frame(frame1,bg='#FF4040')
```

```
        button_frame.pack(fill='both',expand=True)

        def withdraw():
            controller.show_frame('WithdrawPage')

        withdraw_button = Button(button_frame,
                                                    text='Withdraw',
                                                    command=withdraw,
                                                    relief='raised',
                                                    borderwidth=3,
                                                    width=30,
                                                    height=3)
        withdraw_button.grid(row=0,column=0,pady=5)

        def deposit():
            controller.show_frame('DepositPage')

        deposit_button = Button(button_frame,
                                                    text='Deposit',
                                                    command=deposit,
                                                    relief='raised',
                                                    borderwidth=3,
                                                    width=30,
                                                    height=3)
        deposit_button.grid(row=1,column=0,pady=5)

        def balance():
            controller.show_frame('BalancePage')

        balance_button = Button(button_frame,
                                                    text='Balance',
                                                    command=balance,
                                                    relief='raised',
                                                    borderwidth=3,
                                                    width=30,
                                                    height=3)
        balance_button.grid(row=2,column=0,pady=5)

        def exit():
            controller.show_frame('StartPage')

        exit_button = Button(button_frame,
                                                    text='Exit',
                                                    command=exit,
                                                    relief='raised',
                                                    borderwidth=3,
                                                    width=30,
```

```
                                                        height=3)
        exit_button.grid(row=3,column=0,pady=5)


        bottom_frame = Frame(frame1,relief='raised',borderwidth=3)
        bottom_frame.pack(fill='x',side='bottom')

        visa_photo = PhotoImage(file='images/visa.png')
        visa_label = Label(bottom_frame,image=visa_photo)
        visa_label.pack(side='left')
        visa_label.image = visa_photo

        mastercard_photo = PhotoImage(file='images/mastercard.png')
        mastercard_label = Label(bottom_frame,image=mastercard_photo)
        mastercard_label.pack(side='left')
        mastercard_label.image = mastercard_photo

        american_express_photo = PhotoImage(file='images/american-express.png')
        american_express_label =␣
↪Label(bottom_frame,image=american_express_photo)
        american_express_label.pack(side='left')
        american_express_label.image = american_express_photo

        def tick():
            current_time = time.strftime('%I:%M %p').lstrip('0').replace(' 0','␣
↪')
            time_label.config(text=current_time)
            time_label.after(200,tick)

        time_label = Label(bottom_frame,font=('orbitron',12))
        time_label.pack(side='right')

        tick()


class WithdrawPage(Frame):

    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        self.controller = controller
        global c3
        c3=ImageTk.PhotoImage(Image.open("images/canvas2.png"))
        canvas = Canvas(self, width=960, height=700, bd=0, highlightthickness=0)
        canvas.create_image(0,0, image=c2, anchor="nw")
        canvas.pack()

        frame1 = Frame(self,width=696, height=430, bg='#FF7F50' )
```

```
        frame1.pack()
        canvas.create_window(486,300, window=frame1)

        heading_label = Label(frame1,
                                                    text='NEUTRINOVAULT ATM',
                                                   ␣
↪font=('orbitron',40,'bold'),
                                                    foreground='#ffffff',
                                                    background='#FF7F50')
        heading_label.pack(pady=32)
        global choose_amount_label
        choose_amount_label = Label(self,
                                                          text='Choose the␣
↪amount you want to withdraw',
                                                          font=('orbitron',13),
                                                          fg='white',
                                                          bg='#FF7F50')
        canvas.create_window(475,180, width=400, height=40,␣
↪window=choose_amount_label)

        button_frame = Frame(frame1,bg='#FF4040')
        button_frame.pack(fill='both',expand=True)

        def withdraw(amount):
            global current_balance
            if amount<current_balance:
                choose_amount_label = Label(self,
                                                          text='Choose the␣
↪amount you want to withdraw',
                                                          font=('orbitron',13),
                                                          fg='white',
                                                          bg='#FF7F50')
                canvas.create_window(475,180, width=440, height=40,␣
↪window=choose_amount_label)
                current_balance -= amount
                controller.shared_data['Balance'].set(current_balance)
                controller.show_frame('MenuPage')
            else:
                choose_amount_label = Label(self,
                                                          text='Your demand␣
↪exceeds your current balance!',
                                                         ␣
↪font=('orbitron',13,'bold'),
                                                          fg='red',
                                                          bg='#FF7F50')
```

```
            canvas.create_window(475,180, width=440, height=40,␣
↪window=choose_amount_label)



        twenty_button = Button(button_frame,
                                                text='20',
                                                command=lambda:
↪withdraw(20),
                                                relief='raised',
                                                borderwidth=3,
                                                width=15,
                                                height=3)
        twenty_button.grid(row=0,column=0,pady=5)

        forty_button = Button(button_frame,
                                                text='40',
                                                command=lambda:
↪withdraw(40),
                                                relief='raised',
                                                borderwidth=3,
                                                width=15,
                                                height=3)
        forty_button.grid(row=1,column=0,pady=5)

        sixty_button = Button(button_frame,
                                                text='60',
                                                command=lambda:
↪withdraw(60),
                                                relief='raised',
                                                borderwidth=3,
                                                width=15,
                                                height=3)
        sixty_button.grid(row=2,column=0,pady=5)

        eighty_button = Button(button_frame,
                                                text='80',
                                                command=lambda:
↪withdraw(80),
                                                relief='raised',
                                                borderwidth=3,
                                                width=15,
                                                height=3)
        eighty_button.grid(row=3,column=0,pady=5)

        one_hundred_button = Button(button_frame,
```

```
                                                    text='100',
                                                    command=lambda:
↪withdraw(100),
                                                    relief='raised',
                                                    borderwidth=3,
                                                    width=15,
                                                    height=3)
        one_hundred_button.grid(row=0,column=1,pady=5,padx=200)

        two_hundred_button = Button(button_frame,
                                                    text='200',
                                                    command=lambda:
↪withdraw(200),
                                                    relief='raised',
                                                    borderwidth=3,
                                                    width=15,
                                                    height=3)
        two_hundred_button.grid(row=1,column=1,pady=5)

        three_hundred_button = Button(button_frame,
                                                    text='300',
                                                    command=lambda:
↪withdraw(300),
                                                    relief='raised',
                                                    borderwidth=3,
                                                    width=15,
                                                    height=3)
        three_hundred_button.grid(row=2,column=1,pady=5)

        cash = StringVar()
        other_amount_entry = Entry(button_frame,
                                                        textvariable=cash,
                                                        width=20,
                                                        justify='right')
        other_amount_entry.grid(row=3,column=1,pady=5,ipady=30)

        def other_amount(_):
            global current_balance
            if int(cash.get())<current_balance:
                choose_amount_label = Label(self,
                                                            text='Choose the␣
↪amount you want to withdraw',
                                                            font=('orbitron',13),
                                                            fg='white',
                                                            bg='#FF7F50')
                canvas.create_window(475,180, width=440, height=40,␣
↪window=choose_amount_label)
```

```
                current_balance -= int(cash.get())
                controller.shared_data['Balance'].set(current_balance)
                cash.set('')
                controller.show_frame('MenuPage')
            else:
                choose_amount_label = Label(self,
                                                            text='Your demand␣
↪exceeds your current balance!',
                                                           ␣
↪font=('orbitron',13,'bold'),
                                                            fg='red',
                                                            bg='#FF7F50')
                canvas.create_window(475,180, width=440, height=40,␣
↪window=choose_amount_label)
                cash.set('')


        other_amount_entry.bind('<Return>',other_amount)

        bottom_frame = Frame(frame1,relief='raised',borderwidth=3)
        bottom_frame.pack(fill='x',side='bottom')

        visa_photo = PhotoImage(file='images/visa.png')
        visa_label = Label(bottom_frame,image=visa_photo)
        visa_label.pack(side='left')
        visa_label.image = visa_photo

        mastercard_photo = PhotoImage(file='images/mastercard.png')
        mastercard_label = Label(bottom_frame,image=mastercard_photo)
        mastercard_label.pack(side='left')
        mastercard_label.image = mastercard_photo

        american_express_photo = PhotoImage(file='images/american-express.png')
        american_express_label =␣
↪Label(bottom_frame,image=american_express_photo)
        american_express_label.pack(side='left')
        american_express_label.image = american_express_photo

        def tick():
            current_time = time.strftime('%I:%M %p').lstrip('0').replace(' 0','␣
↪')
            time_label.config(text=current_time)
            time_label.after(200,tick)

        time_label = Label(bottom_frame,font=('orbitron',12))
        time_label.pack(side='right')
```

```
        tick()


class DepositPage(Frame):

    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        self.controller = controller
        global c3
        c3=ImageTk.PhotoImage(Image.open("images/canvas2.png"))
        canvas = Canvas(self, width=960, height=700, bd=0, highlightthickness=0)
        canvas.create_image(0,0, image=c2, anchor="nw")
        canvas.pack()

        frame1 = Frame(self,width=696, height=430, bg='#FF7F50' )
        frame1.pack()
        canvas.create_window(486,300, window=frame1)

        heading_label = Label(frame1,
                                                text='NEUTRINOVAULT ATM',
                                               ␣
 ↪font=('orbitron',40,'bold'),
                                                foreground='#ffffff',
                                                background='#FF7F50')
        heading_label.pack(pady=29)

        space_label = Label(frame1,height=4,bg='#FF7F50')
        space_label.pack()

        enter_amount_label = Label(frame1,
                                                 text='Enter amount',
                                                 font=('orbitron',13),
                                                 bg='#FF7F50',
                                                 fg='white')
        enter_amount_label.pack(pady=20)

        cash = StringVar()
        deposit_entry = Entry(frame1,
                                               textvariable=cash,
                                               font=('orbitron',12),
                                               width=22)
        deposit_entry.pack(ipady=14)

        def deposit_cash():
            global current_balance
            current_balance += int(cash.get())
            controller.shared_data['Balance'].set(current_balance)
```

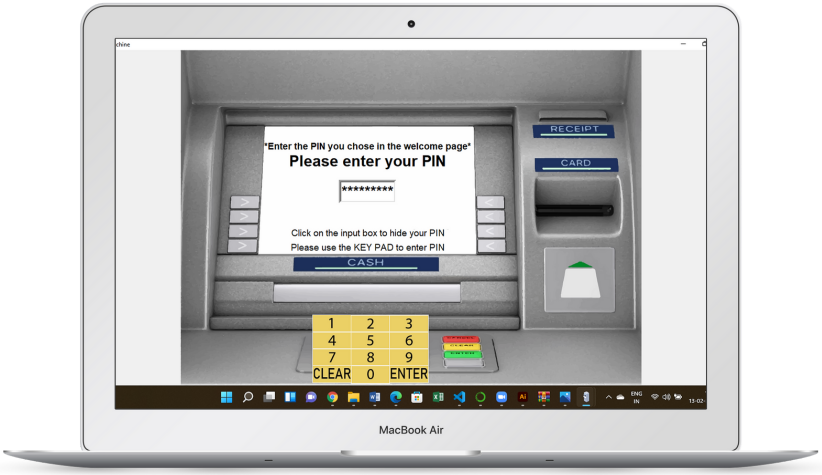
```
            controller.show_frame('MenuPage')
            cash.set('')

        enter_button = Button(frame1,
                                                text='Enter',
                                                command=deposit_cash,
                                                relief='raised',
                                                borderwidth=3,
                                                width=40,
                                                height=3)
        enter_button.pack(pady=20)

        two_tone_label = Label(frame1,bg='#FF7F50')
        two_tone_label.pack(fill='both',expand=True)

        bottom_frame = Frame(frame1,relief='raised',borderwidth=3)
        bottom_frame.pack(fill='x',side='bottom')

        visa_photo = PhotoImage(file='images/visa.png')
        visa_label = Label(bottom_frame,image=visa_photo)
        visa_label.pack(side='left')
        visa_label.image = visa_photo

        mastercard_photo = PhotoImage(file='images/mastercard.png')
        mastercard_label = Label(bottom_frame,image=mastercard_photo)
        mastercard_label.pack(side='left')
        mastercard_label.image = mastercard_photo

        american_express_photo = PhotoImage(file='images/american-express.png')
        american_express_label =␣
↪Label(bottom_frame,image=american_express_photo)
        american_express_label.pack(side='left')
        american_express_label.image = american_express_photo

        def tick():
            current_time = time.strftime('%I:%M %p').lstrip('0').replace(' 0','␣
↪')
            time_label.config(text=current_time)
            time_label.after(200,tick)

        time_label = Label(bottom_frame,font=('orbitron',12))
        time_label.pack(side='right')

        tick()


class BalancePage(Frame):
```

```
   def __init__(self, parent, controller):
       Frame.__init__(self, parent)
       self.controller = controller
       global c4
       c4=ImageTk.PhotoImage(Image.open("images/canvas2.png"))
       canvas = Canvas(self, width=960, height=700, bd=0, highlightthickness=0)
       canvas.create_image(0,0, image=c2, anchor="nw")
       canvas.pack()

       frame1 = Frame(self,width=696, height=430, bg='#FF7F50' )
       frame1.pack()
       canvas.create_window(486,300, window=frame1)


       heading_label = Label(frame1,
                                                text='NEUTRINOVAULT ATM',
                                               ␣
↪font=('orbitron',40,'bold'),
                                                foreground='#ffffff',
                                                background='#FF7F50')
       heading_label.pack(pady=35)

       global current_balance
       controller.shared_data['Balance'].set(current_balance)
       balance_label = Label(frame1,
                                               textvariable=controller.
↪shared_data['Balance'],
                                               font=('orbitron',13),
                                               fg='white',
                                               bg='#FF7F50',
                                               anchor='w')
       balance_label.pack(fill='x')

       button_frame = Frame(frame1,bg='#FF4040')
       button_frame.pack(fill='both',expand=True)

       def menu():
           controller.show_frame('MenuPage')

       menu_button = Button(button_frame,
                                                 command=menu,
                                                 text='Menu',
                                                 relief='raised',
                                                 borderwidth=3,
                                                 width=40,
                                                 height=4)
```

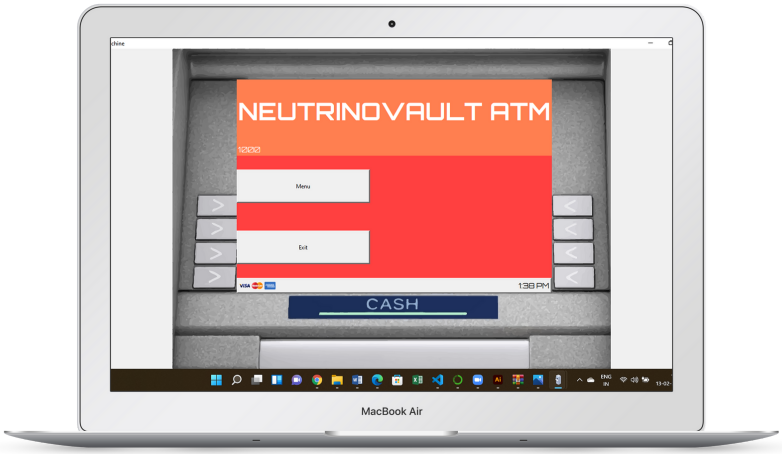
```
        menu_button.grid(row=0,column=0,pady=30)

        def exit():
            controller.show_frame('StartPage')

        exit_button = Button(button_frame,
                                                text='Exit',
                                                command=exit,
                                                relief='raised',
                                                borderwidth=3,
                                                width=40,
                                                height=4)
        exit_button.grid(row=1,column=0,pady=30)

        bottom_frame = Frame(frame1,relief='raised',borderwidth=3)
        bottom_frame.pack(fill='x',side='bottom')

        visa_photo = PhotoImage(file='images/visa.png')
        visa_label = Label(bottom_frame,image=visa_photo)
        visa_label.pack(side='left')
        visa_label.image = visa_photo

        mastercard_photo = PhotoImage(file='images/mastercard.png')
        mastercard_label = Label(bottom_frame,image=mastercard_photo)
        mastercard_label.pack(side='left')
        mastercard_label.image = mastercard_photo

        american_express_photo = PhotoImage(file='images/american-express.png')
        american_express_label =␣
 ↪Label(bottom_frame,image=american_express_photo)
        american_express_label.pack(side='left')
        american_express_label.image = american_express_photo

        def tick():
            current_time = time.strftime('%I:%M %p').lstrip('0').replace(' 0','␣
 ↪')
            time_label.config(text=current_time)
            time_label.after(200,tick)

        time_label = Label(bottom_frame,font=('orbitron',12))
        time_label.pack(side='right')

        tick()


if __name__ == "__main__":
    app = SampleApp()
```

```
    app.geometry("960x700")
    app.mainloop()
```

Output Snapshots




NEUTRINOVAULT ATM
Choose the amount you want to withdraw
20
40
60
80
100
200
300
CASH
MacBook Air


NEUTRINOVAULT ATM
1000
Menu
Exit
CASH
MacBook Air


NEUTRINOVAULT ATM
Enter amount
5000
Enter
CASH
MacBook Air

Variable Description

Name	Description
Current balance	stores the value of curent amount
pin _entry	stores the value of the pin entered in the machine
Withdraw_Button	stoes the amount of value to be withdrawn
Deposit_Button	stores the amount of value to be deposited
Exit Button	takes back to the menu page

Function Description

Name	Description
tkinter	Main Tkinter module
tkinter.colorchooser	Dialog to let the user choose a color
Button	To add a button in your application, this widget is used.
Entry	It is used to input the single line text entry from the user. For multi-line text input, Text widget is used.
Label	It refers to the display box where you can put any text or image which can be updated

Utility Of The Project

- This project can be utilized to maintain the transactions and for checking the balances within the banks which would have taken a much longer time operated manually.
- As it is an interface based, a person who does not have much knowledge about computer can also easily use it with just the help of the pin of the user.
- It is a completely safe process since no other person can make the transactions . The user only has an access to it at a particular time.
- It can solve the work efficiently without data redundancy, duplicity and many others.
- This project can alsobe impoved later in a multiple ways, like opening a bank account or creating a joint account, give loans with proper amount of interest and many more.

Bibliography

- https://docs.python.org/3/library/tkinter.html
- https://www.geeksforgeeks.org/python-gui- tkinter/t
- https://docs.python.org/3/
- https://www.w3schools.com/python
- Computer Science for Class 11-PreetiA rora