ABSTRACT

Our project is based on Dress Stock Management. Users can sign in to the page of DRESS STOCK MANAGEMENT for registering a details of an outfit, Updating the details of outfit, Viewing the data of specific outfit , Deleting data of specific outfit , Rating our application. The Dress Stock Management main page can be entered using their username and password provided. It is accessible by the authorized users themselves. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

REQUIREMENT ANALYSIS

1.Hardware Requirement:

- A Computer Which Can Run Python
- 64-bit CPU (Intel / AMD architecture)
- 8.00 GB RAM
- 5 GB free disk space

2. Software Requirement:

- Python (Version Above 2.9)
- Windows 10 or Windows 11

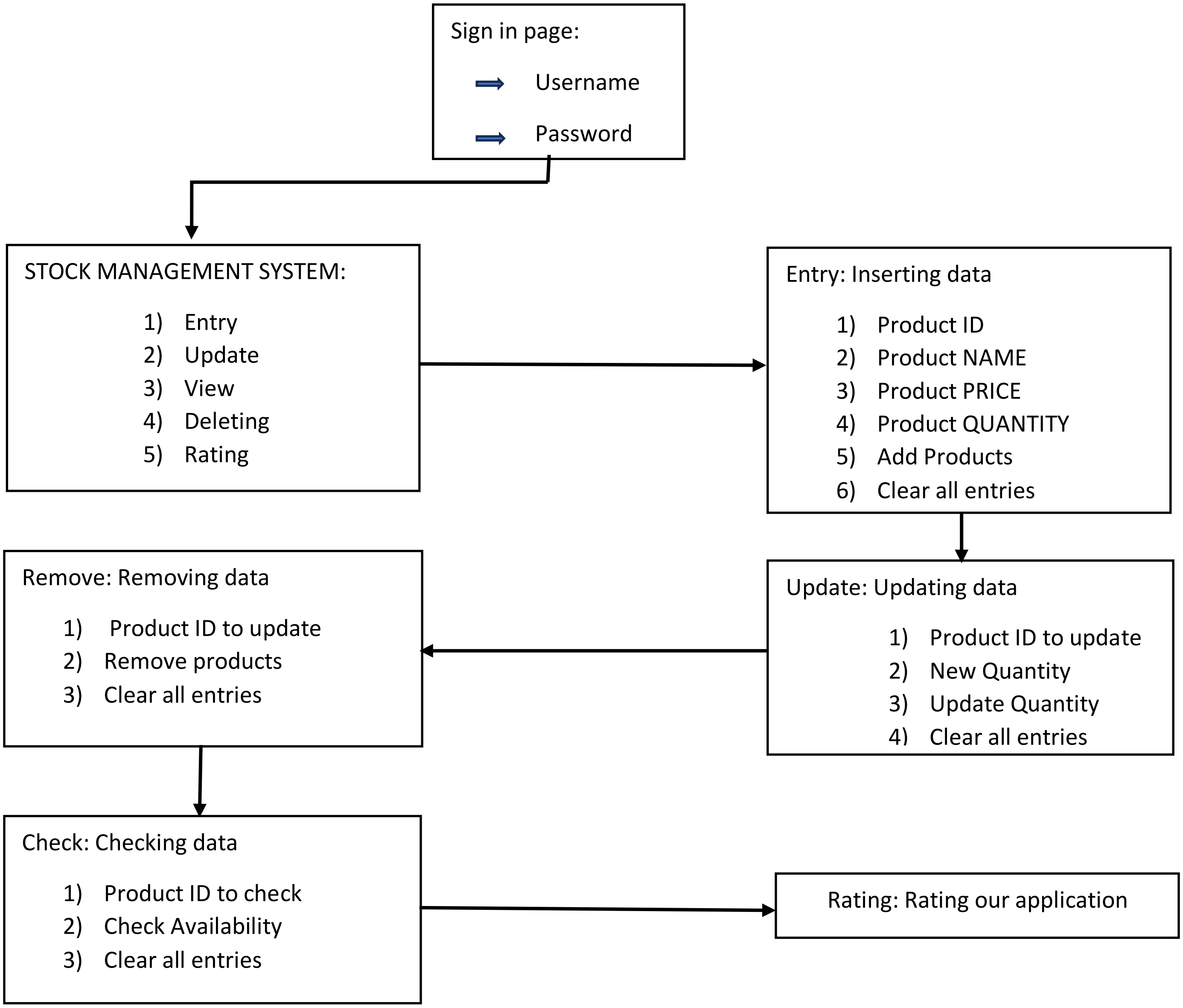
APPLICATION

Stock management system was introduced to solve the complications coming from managing all the details associated with the various outfits along with many other details for maintain a ordered record of data for efficient management with confidentiality. Stock management system provides the ability to manage all the complexity of registering the details of the outfit in one place, reducing the work of staff in arranging and analyzing the details of the outfit

Stock management system does many works like:

1. Registering the ID of the product , NAME of the product , QUANTITY of the product and RATE of the product
2. Updating the existing record of product
3. Removing the record of existing product
4. Viewing the record of existing product

BLOCK DIAGRAM



Sign in page:

⟹ Username

⟹ Password

STOCK MANAGEMENT SYSTEM:

1) Entry
2) Update
3) View
4) Deleting
5) Rating

Entry: Inserting data

1) Product ID
2) Product NAME
3) Product PRICE
4) Product QUANTITY
5) Add Products
6) Clear all entries

Update: Updating data

1) Product ID to update
2) New Quantity
3) Update Quantity
4) Clear all entries

Remove: Removing data

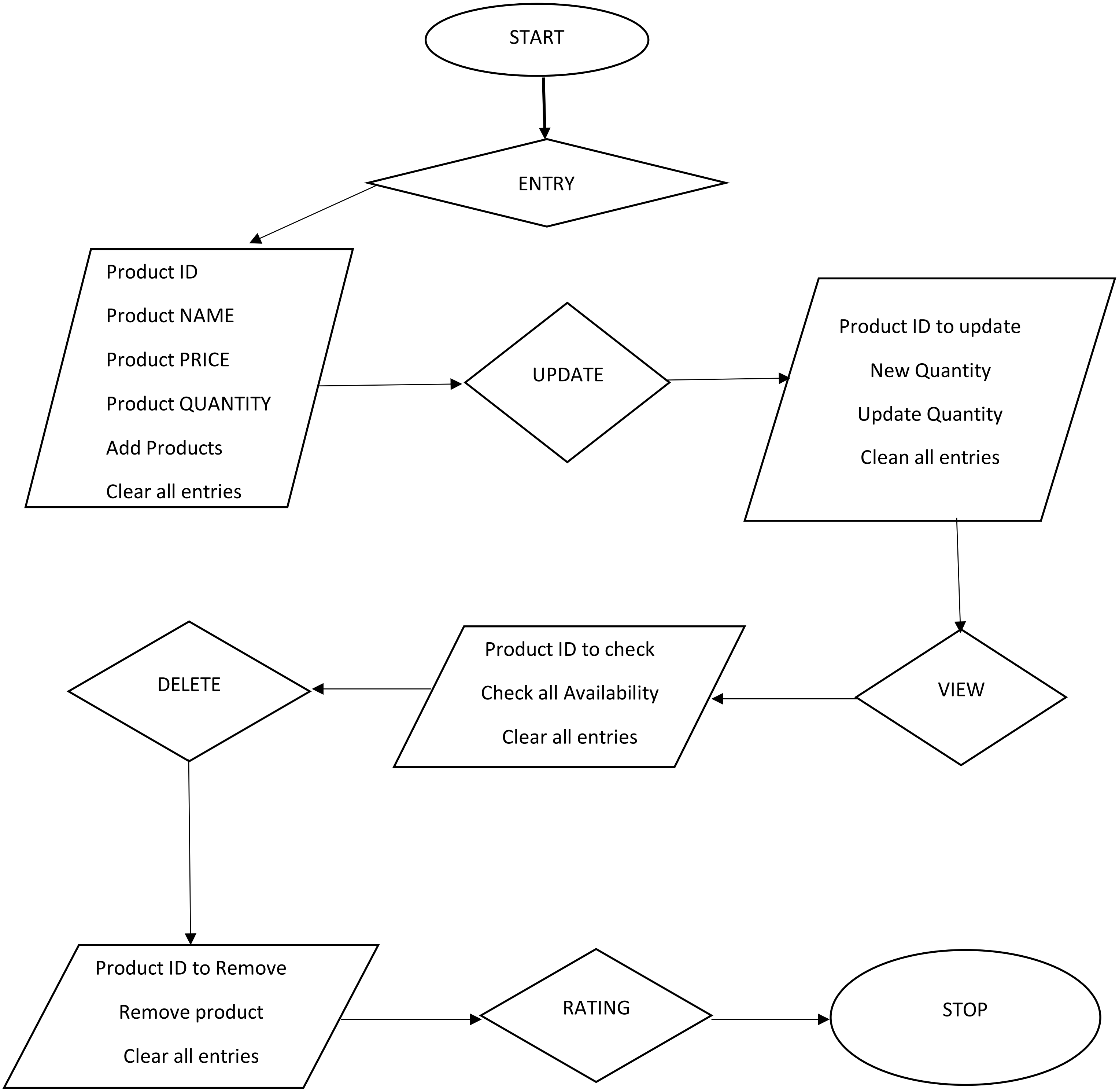
1) Product ID to update
2) Remove products
3) Clear all entries

Check: Checking data

1) Product ID to check
2) Check Availability
3) Clear all entries

Rating: Rating our application

FLOW CHART


START
ENTRY
Product ID
Product NAME
Product PRICE
Product QUANTITY
Add Products
Clear all entries
UPDATE
Product ID to update
New Quantity
Update Quantity
Clean all entries
VIEW
Product ID to check
Check all Availability
Clear all entries
DELETE
Product ID to Remove
Remove product
Clear all entries
RATING
STOP

ALGORITHM

1)Start

2)Users can insert data of any outfit product.

3)Users can make changes in any of the inserted record on their will.

4)Users can view the data that they need.

5)Users can delete their desired data when they want.

6)Users can rate the application which they seem fit.

7)End

FUNCTIONS IN THE PROGRAM

1) add_product(): To add the details of the product in mysql database.
2) update_product(): To update the product details in mysql database.
3) check_availability(): To check availability of the product in mysql database.
4) remove_product(): To delete data of the prodcuts from mysql database.
5) entry(): To provide input of details of the product for adding in an application.
6) remove(): To remove a particular data by specifying its ID in an application.
7) update(): To update a particular data by specifying its ID in an application.
8) avail(): To check availability of product by specifying its ID in an application.
9) clear_ent(): To clear all entries types in an application.
10) rating(): To provide rating for our application.
11) signin(): To create a sign in page for authorized user.

SOURCE CODE

```
from tkinter import *
import mysql.connector as ms
from tkinter import messagebox
from tkinter import PhotoImage
from time import *

# Create the database and table if they don't exist
conn = ms.connect(host="localhost",user="root",password="123456")
cursor = conn.cursor()

cursor.execute("create database if not exists design")
cursor.execute("use design")
cursor.execute("CREATE TABLE IF NOT EXISTS designult"
          "("
          "id INTEGER PRIMARY KEY,"
          "name varchar(20),"
          "price int(10),"
          "quantity int(10))")
```

```
conn.commit()

def add_product():

    global entry_id,entry_name,entry_price,entry_quantity
    id=entry_id.get()
    name=entry_name.get()
    price=entry_price.get()
    quantity=entry_quantity.get()

    if id and name and price and quantity:
        query='INSERT INTO designult (id, name, price, quantity) VALUES ("{}", "{}", "{}", "{}")'.format(id,name,price,quantity)
        cursor.execute(query)
        conn.commit()

        messagebox.showinfo("Success", "Product added successfully!")

    else:
        messagebox.showwarning("Error", "Please fill in all fields.")

def update_product():
    global entry_update_id,entry_new_quantity
    id = entry_update_id.get()
    quantity = entry_new_quantity.get()

    if id and quantity:
        cursor.execute("update designult set quantity={} where id={}".format(quantity,id))

        conn.commit()
        messagebox.showinfo("Success", "Product quantity updated successfully!")

    else:
        messagebox.showwarning("Error", "Please enter a product ID and new quantity.")

def check_availability():

    global entry_check_id
    id = entry_check_id.get()

    if id:
        cursor.execute("SELECT quantity AS totalnumberis FROM
designult WHERE id=(%s)",(id,))


        dat=cursor.fetchall()


        a=[]
        for i in dat:
            a.append(i)
        if len(a)==0:
            messagebox.showwarning("ERROR", "NO DATA FOUND!!")
        else:
            det=("QUNATITY OF ITEMS AVAILABLE IS",a)
            messagebox.showinfo("TOTAL COUNT IS", det)


    else:
        messagebox.showwarning("Error", "Please enter a product ID to
check.")


def remove_product():


    global entry_remove_id
    id = entry_remove_id.get()
    if id:
        query="delete from designult where id={}".format(id)
        cursor.execute(query)
        conn.commit()
        messagebox.showinfo("Success", "Product removed successfully!")

    else:
        messagebox.showwarning("Error", "Please enter a product ID to remove.")

# Product Entry Fields

def entry():

    root1=Tk()
    root1.title("INSERTING DATA")
    root1.geometry("925x500+300+200")
    root1.config(bg="white")

    frame = Frame(root1)
    frame.pack(pady=20)

    global entry_id,entry_name,entry_price,entry_quantity
```

```
label_1 = Label(frame,text="INSERTING DATA")
label_1.grid(row=0, column=0)

label_id = Label(frame, text="Product ID:")
label_id.grid(row=1, column=0)

entry_id = Entry(frame)
entry_id.grid(row=1, column=1)

label_name = Label(frame, text="Product Name:")
label_name.grid(row=2, column=0)

entry_name = Entry(frame)
entry_name.grid(row=2, column=1)

label_price = Label(frame, text="Product Price:")
label_price.grid(row=3, column=0)

entry_price = Entry(frame)
entry_price.grid(row=3, column=1)
```

```
    label_quantity = Label(frame, text="Product Quantity:")
    label_quantity.grid(row=4, column=0)

    entry_quantity = Entry(frame)
    entry_quantity.grid(row=4, column=1)

    add_button = Button(frame, text="Add Product", command=add_product)
    add_button.grid(row=5, column=0, columnspan=2)

    def clear_ent():
        entry_id.delete(0,END)
        entry_name.delete(0,END)
        entry_price.delete(0,END)
        entry_quantity.delete(0,END)

    clear_button = Button(frame,text="Clear all entries",command=clear_ent)
    clear_button.grid(row=6, column=0, columnspan=2)

    root1.mainloop()

# Remove Product
```

```
def remove():

    root2=Tk()
    frame = Frame(root2)
    frame.pack(pady=20)

    global entry_remove_id

    label_1 = Label(frame,text="REMOVING DATA")
    label_1.grid(row=0, column=0)

    label_remove_id = Label(frame, text="Product ID to Remove:")
    label_remove_id.grid(row=0, column=0)

    entry_remove_id = Entry(frame)
    entry_remove_id.grid(row=0, column=1)

    remove_button = Button(frame, text="Remove Product",
command=remove_product)
    remove_button.grid(row=1, column=0, columnspan=2)

    def clear_ent():
        entry_remove_id.delete(0,END)

    clear_button = Button(frame,text="Clear all entries",command=clear_ent)
    clear_button.grid(row=5, column=0, columnspan=2)

    root2.mainloop()

# Update Product Quantity

def update():

    root3=Tk()
    update_frame = Frame(root3)
    update_frame.pack(pady=20)

    global entry_update_id,entry_new_quantity

    label_1 = Label(update_frame,text="UPDATING DATA")
    label_1.grid(row=0, column=0)

    label_update_id = Label(update_frame, text="Product ID to Update:")
    label_update_id.grid(row=0, column=0)
    entry_update_id = Entry(update_frame)
    entry_update_id.grid(row=0, column=1)

    label_new_quantity = Label(update_frame, text="New Quantity:")
    label_new_quantity.grid(row=1, column=0)

    entry_new_quantity = Entry(update_frame)
    entry_new_quantity.grid(row=1, column=1)

    update_button = Button(update_frame, text="Update Quantity",
command=update_product)
    update_button.grid(row=2, column=0, columnspan=2)

    def clear_ent():
        entry_update_id.delete(0,END)
        entry_new_quantity.delete(0,END)

    clear_button = Button(update_frame,text="Clear all
entries",command=clear_ent)
    clear_button.grid(row=5, column=0, columnspan=2)

    root3.mainloop()
```

```
# Check Availability

def avail():

    root4=Tk()
    availability_frame = Frame(root4)
    availability_frame.pack(pady=20)

    global entry_check_id

    label_1 = Label(availability_frame,text=" DATA")
    label_1.grid(row=0, column=0)

    label_check_id = Label(availability_frame, text="Product ID to Check:")
    label_check_id.grid(row=0, column=0)

    entry_check_id = Entry(availability_frame)
    entry_check_id.grid(row=0, column=1)

    check_button = Button(availability_frame, text="Check Availability", command=check_availability)
    check_button.grid(row=1, column=0, columnspan=2)

    result_text = Label(availability_frame, text="")
    result_text.place(x=20,y=20)

    def clear_ent():
        entry_check_id.delete(0,END)

    clear_button = Button(availability_frame,text="Clear all entries",command=clear_ent)
    clear_button.grid(row=5, column=0, columnspan=2)

    root4.mainloop()

def rating():

     root5=Tk()
     rating_frame = Frame(root5)
     rating_frame.pack(pady=20)
     label26 = Label(rating_frame,text="RATING",font=("ALGERIAN 40 bold"))
     label26.pack()
    scale = Scale(rating_frame,from_=100,to=0,length=600,orient=VERTICAL,font=("arial 20 bold",20),tickinterval=10,troughcolor="light blue",fg="green",bg="black")
    scale.pack()
    def submit():
        messagebox.showinfo("RATING",scale.get())

    button21=Button(rating_frame,text="Submit",command=submit)
    button21.pack()

root = Tk()
root.title("DRESS RETAIL SHOP")
root.geometry('925x500+300+200')
root.config(bg='white')

def signin():
    username=user.get()
    password=code.get()

    if username=='a' and password=='0':

        root1 = Tk()
        root1.title("Shop Management System")
        root1.geometry('925x500+300+200')
        root1.config(bg='white')
        label=Label(root1,text="STOCK MANAGEMENT",font="arial 40 bold",bg='light blue')
        label.place(x=450,y=70)

        def update_1():
            time_string = strftime("%I:%M:%S %p")
            time_label.config(text=time_string)

            day_string = strftime("%A")
            day_label.config(text=day_string)

            date_string = strftime("%B %d, %Y")
            date_label.config(text=date_string)

            root1.after(1000,update_1)

        time_label = Label(root1,font=("Arial",25),fg="black",bg="white")
        time_label.place(x=1200,y=640)

        day_label = Label(root1,font=("Arial",25))
```

```
        day_label.place(x=1240,y=690)


        date_label = Label(root1,font=("Arial",25))
        date_label.place(x=1170,y=730)
        update_1()


        b1=Button(root1,text="ENTRY",font="arial 20 bold",bg='yellow',command=entry)
        b2=Button(root1,text="UPDATE",font="arial 20 bold",bg='yellow',command=update)
        b3=Button(root1,text="VIEWING",font="arial 20 bold",bg='yellow',command=avail)
        b4=Button(root1,text="DELETING",font='arial 20 bold',bg='yellow',command=remove)
        b5=Button(root1,text="RATING",font='arial 20 bold',bg='yellow',command=rating)


        l1=Label(root1,text=" <- Click to enter outfit drtails in  its respective section",font="arial 20 bold")
        l2=Label(root1,text=" <- Click to update the existing record of outfit",font="arial 20 bold")
        l3=Label(root1,text=" <- Click to view paricular record using its ID",font="arial 20 bold")
        l4=Label(root1,text=" <- Click to delete unwanted record",font="arial 20 bold")
```

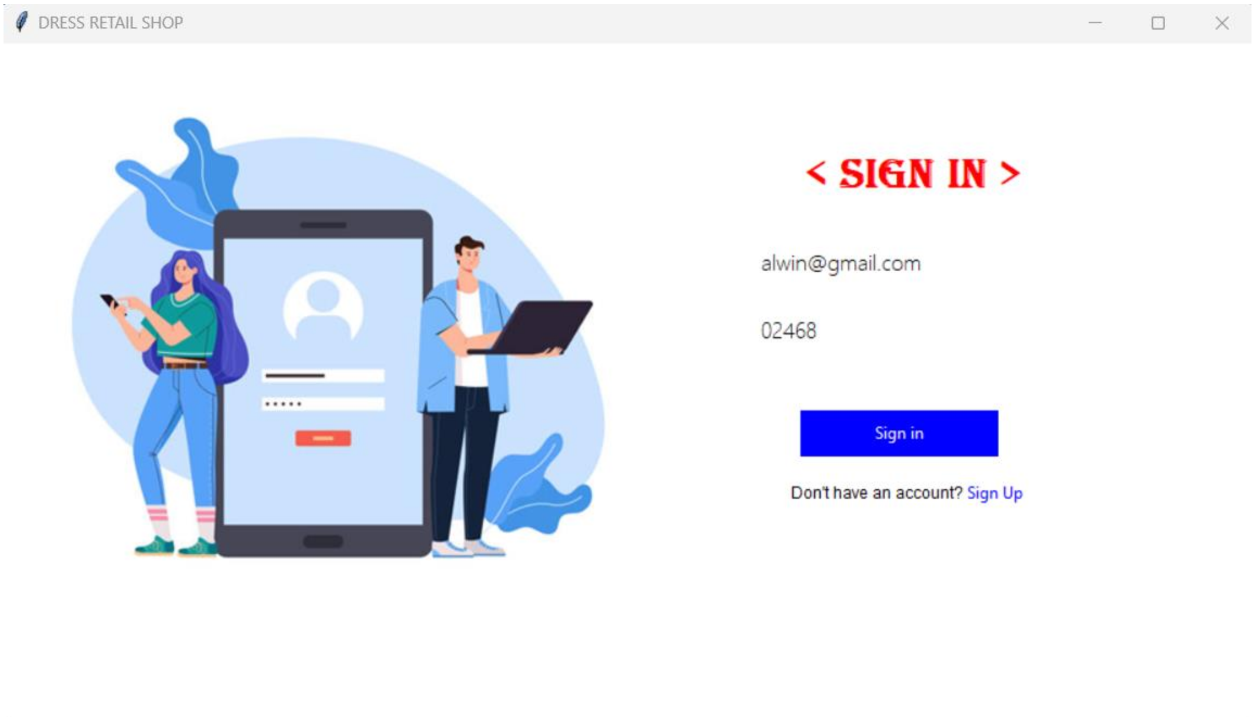
```
    l5=Label(root1,text=" <- Click to rate our page",font="arial 20 bold")

    l6=Label(root1,text="<1>",font="arial 20 bold")
    l7=Label(root1,text="<2>",font="arial 20 bold")
    l8=Label(root1,text="<3>",font="arial 20 bold")
    l9=Label(root1,text="<4>",font="arial 20 bold")
    l0=Label(root1,text="<5>",font="arial 20 bold")

    b1.place(x=200,y=200)
    b2.place(x=200,y=300)
    b3.place(x=200,y=400)
    b4.place(x=200,y=500)
    b5.place(x=200,y=600)

    l1.place(x=400,y=200)
    l2.place(x=400,y=300)
    l3.place(x=400,y=400)
    l4.place(x=400,y=500)
    l5.place(x=400,y=600)

    l6.place(x=100,y=200)
    l7.place(x=100,y=300)
```

```
        l8.place(x=100,y=400)
        l9.place(x=100,y=500)
        l0.place(x=100,y=600)

        root1.mainloop()

    elif username!='0' and password!='1':
        messagebox.showerror("Invalid","invalid username and password")
        messagebox.showerror('Error','Please enter valid username and
password')

photo = PhotoImage(file='C:\\Users\\manis\\Videos\\login.png')
label = Label(root,image = photo,bg='white')
label.place(x=50,y=50)

frame = Frame(root,width=350,height=350,bg="white")
frame.place(x=480,y=70)

heading = Label(frame,text='< Sign in
>',fg='red',bg='white',font=('ALGERIAN',23,'bold'))
heading.place(x=110,y=5)

def on_enter(e):
    user.delete(0,'end')


def on_leave(e):
    name=user.get()
    if name==":
        user.insert(0,'USERNAME')


user = Entry(frame,width=25,fg='black',border=0,bg='white',font=('Microsoft YaHei UI Light',11))
user.place(x=80,y=80)
user.insert(0,'USERNAME-->')
user.bind('<FocusIn>',on_enter)
user.bind('<FocusOut>',on_leave)


def on_enter(e):
    code.delete(0,'end')


def on_leave(e):
    name=code.get()
    if name==":
        code.insert(0,'PASSWORD')
```

```
code=Entry(frame,width=25,fg='black',border=0,bg='white',font=('Microsoft YaHei UI Light',11))

code.place(x=80,y=130)

code.insert(0,'PASSWORD-->')

code.bind('<FocusIn>',on_enter)

code.bind('<FocusOut>',on_leave)

Button(frame,width=20,pady=7,text='Sign in',bg='blue',fg='white',border=0,command=signin).place(x=110,y=200)

label = Label(frame,text="Don't have an account?",fg='black',bg='white',font=('Arial',9))

label.place(x=100,y=250)

sign_up=Button(frame,width=6,text='Sign Up',border=0,bg='white',cursor='hand2',fg='blue')

sign_up.place(x=230,y=250)

root.mainloop()
```

SAMPLE OUTPUT

SIGN IN PAGE:



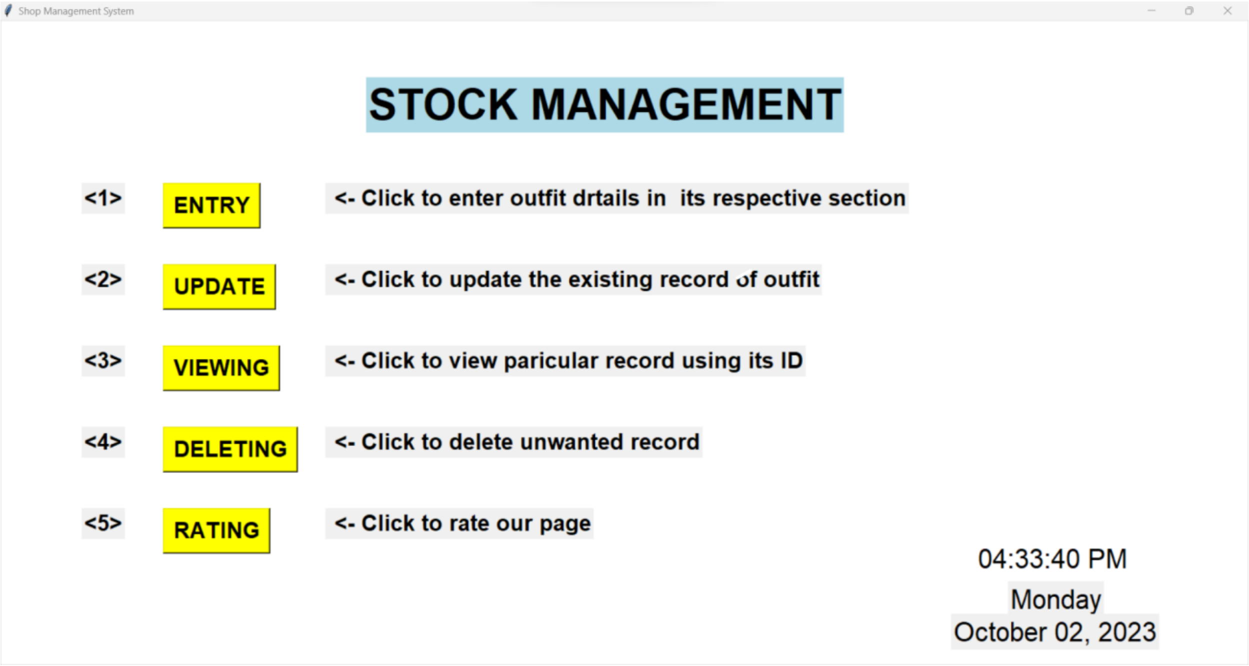
DRESS RETAIL SHOP

< SIGN IN >

alwin@gmail.com

02468

After Sign In Page:



Shop Management System

STOCK MANAGEMENT

<1> ENTRY <- Click to enter outfit drtails in its respective section

<2> UPDATE <- Click to update the existing record of outfit

<3> VIEWING <- Click to view paricular record using its ID

<4> DELETING <- Click to delete unwanted record

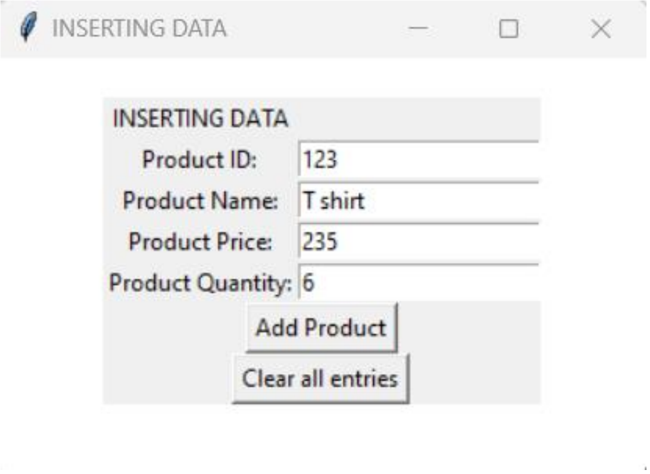
<5> RATING <- Click to rate our page

04:33:40 PM

Monday

October 02, 2023

<1> ENTRY: Inserting data



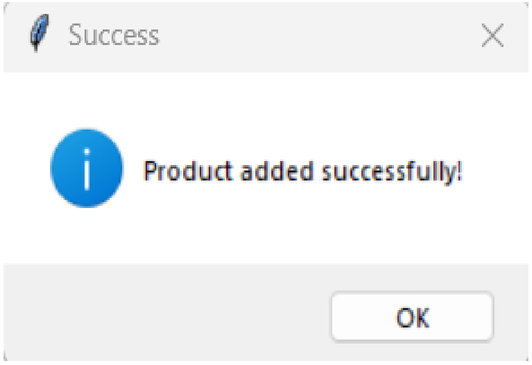
INSERTING DATA

Product ID: 123

Product Name: T shirt

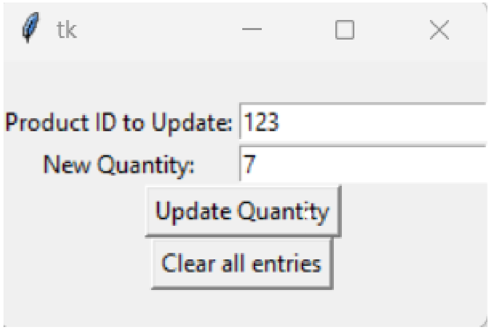
Product Price: 235

Product Quantity: 6



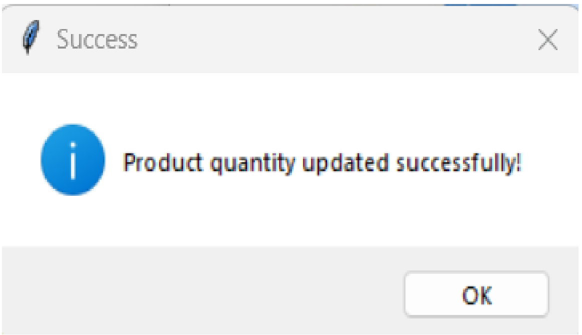
Product added successfully!

<2> UPDATE: Updating data



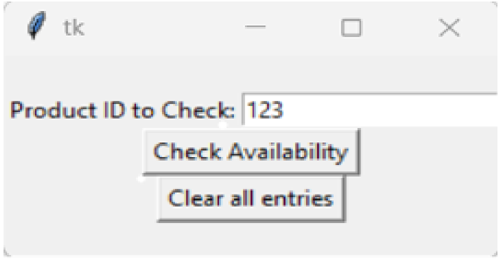
Product ID to Update: 123

New Quantity: 7

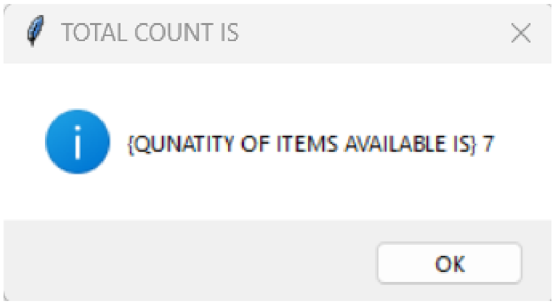


Product quantity updated successfully!

<3> VIEWING: Viewing data

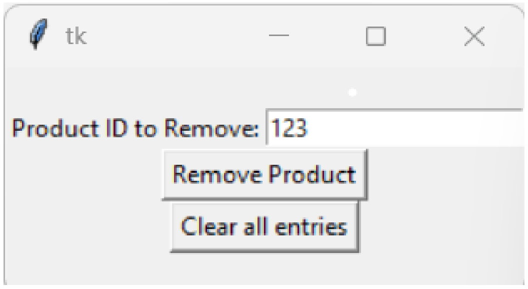


Product ID to Check: 123

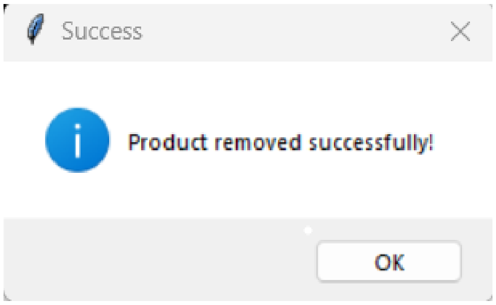


{QUNATITY OF ITEMS AVAILABLE IS} 7

<4> DELETING: Deleting data

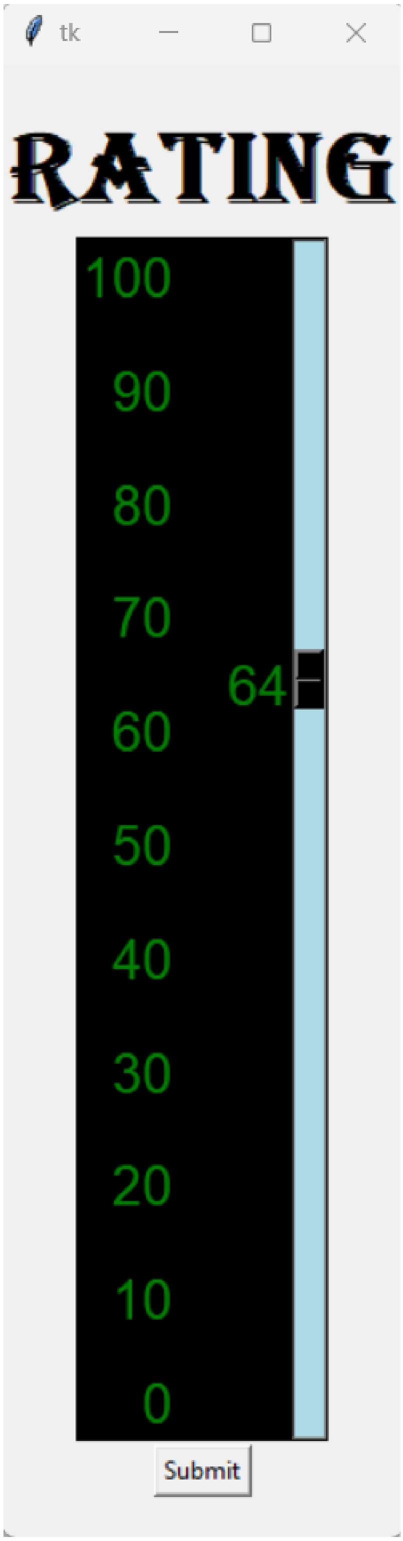


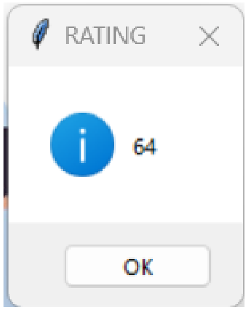
Product ID to Remove: 123



Product removed successfully!

<5> RATING: Rating our application





64

FUTURE ENHANCEMENT

1) It can be further enhanced by adding a drop down list in various fields which shows by default.

2) Other additional enhancements like showing pop up menu of the details which is given a input

BIBLIOGRAPHY AND REFERENCES

1) NCERT class 12 computer science

2) Computer science with python class 12-sumita arora

3) https://www.geeksforgeeks.org/

4) https://www.youtube.com/watch?v=X9reTl_Mckk