**Practical No. 10**

1. **Design a simple database application that stores the records and retrieve the**

**same.**

**CODE:**

from tkinter import \*

import mysql.connector

from tkinter import messagebox

try :

db = mysql.connector.connect(host = "localhost" , user = "root" , password = "" , database = "product")

dbC = db.cursor()

except Exception as e :

print(str(e))

def insertRecord() :

nm = pnameEntry.get()

q = pqtyEntry.get()

if nm == "" or q == "":

messagebox.showwarning('warning','please enter both product name and quantity')

else:

try:

nm = str(nm)

q = int(q)

dbC.execute("insert into productt(productName , productQuantity) values(%s , %s)" , (nm , q))

db.commit()

pnameEntry.delete(0 , END)

pqtyEntry.delete(0 , END)

messagebox.showinfo('success','product name and quantity added successfully')

except ValueError:

messagebox.showwarning('warning','please enter both valid product name and quantity')

def disp():

try:

dbc2=db.cursor()

dbc2.execute("select \* from productt")

res = dbc2.fetchall()

for i in res :

tb.insert(END , f"Product ID :{i[0]},\t")

tb.insert(END , f"Product Name :{i[1]},\t")

tb.insert(END , f"Product Quantity :{i[2]}\n")

except:

db.rollback()

root = Tk()

pname = Label(root , text = "Product Name ")

pname.pack()

pnameEntry = Entry(root)

pnameEntry.pack()

pqty = Label(root , text = "Quantity")

pqty.pack()

pqtyEntry = Entry(root)

pqtyEntry.pack()

insButton = Button(root , text = "Add Product" , command = insertRecord)

insButton.pack()

dispButton = Button(root , text = "Display" , command = disp)

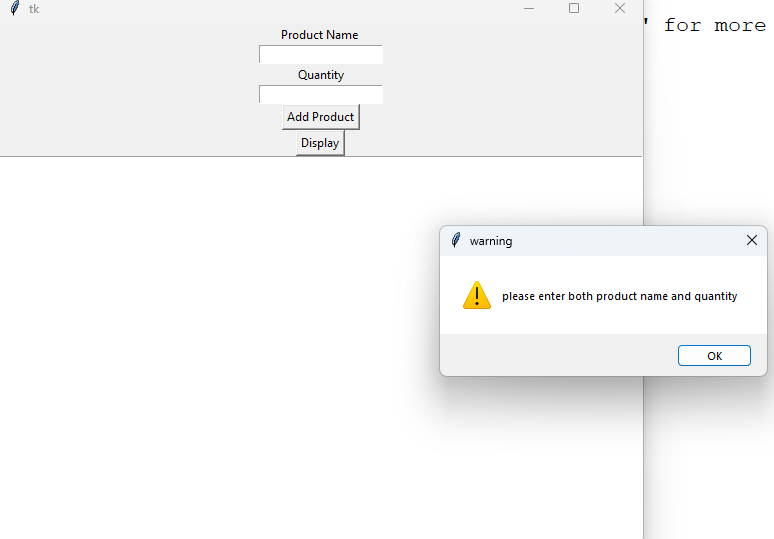
dispButton.pack()

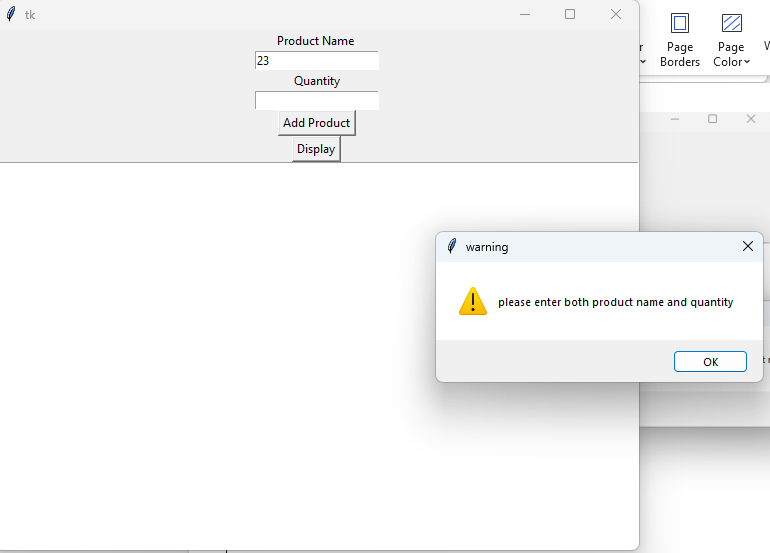
tb = Text(root)

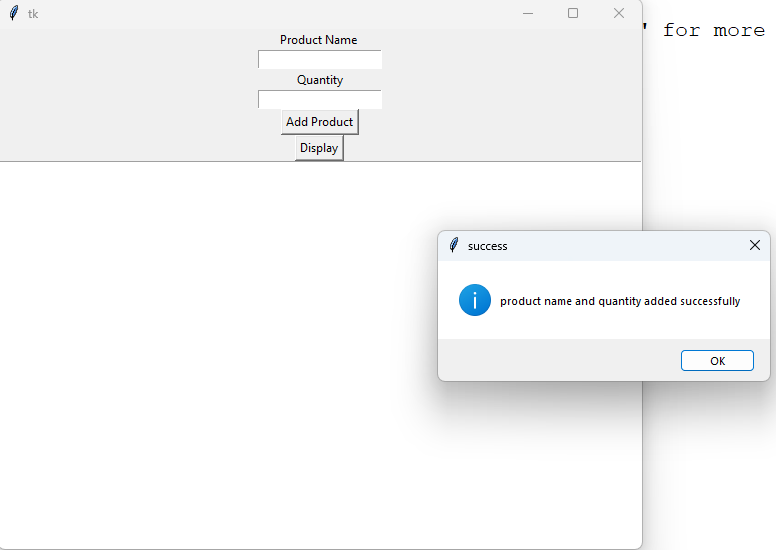
tb.pack()

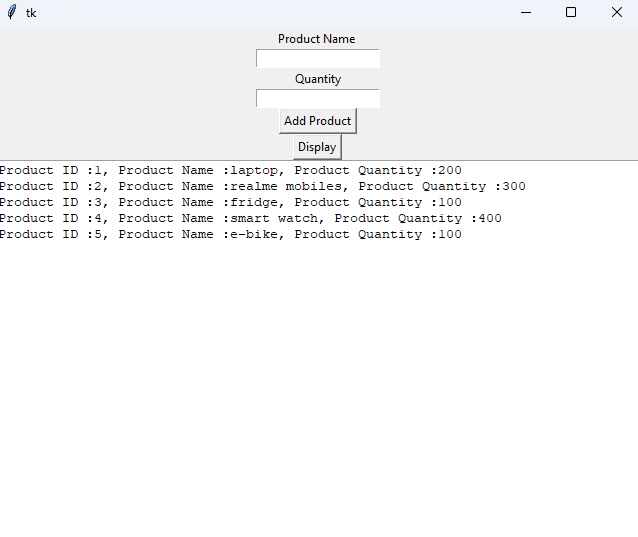
root.mainloop()

**OUTPUT:**









**b. Design a database application to search the specified record from the database.**

**CODE:**

from tkinter import \*

from tkinter import messagebox

import mysql.connector

# Connect to the database

try:

db = mysql.connector.connect(

host="localhost",

user="root",

password="",

database="product"

)

except Exception as e:

print(f"Error: {str(e)}")

# Function to search for a product by name

def searchRecord():

pname = pnameEntry.get()

if pname:

dbC = db.cursor()

dbC.execute("select \* from product where productName = %s", (pname,))

result = dbC.fetchone()

lb.delete(0, END) # Clear the listbox before displaying new data

if result:

lb.insert(END, f"Product ID: {result[0]}")

lb.insert(END, f"Product Name: {result[1]}")

lb.insert(END, f"Product Quantity: {result[2]}")

else:

messagebox.showinfo("Not Found", "Product not exists")

else:

messagebox.showwarning("Input Error", "Please enter a product name to search")

# Create the main window

win = Tk()

win.title("Product Search")

# Labels and entries for product name

pname = Label(win, text="Product Name")

pname.pack()

pnameEntry = Entry(win)

pnameEntry.pack()

# Search button

searchButton = Button(win, text="Search", command=searchRecord)

searchButton.pack()

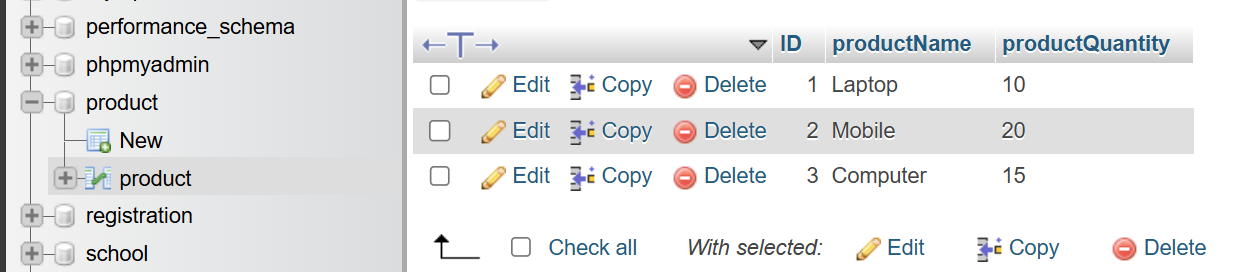
# Listbox to display product information

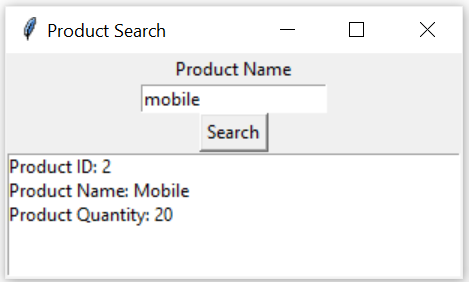
lb = Listbox(win, width=50, height=5)

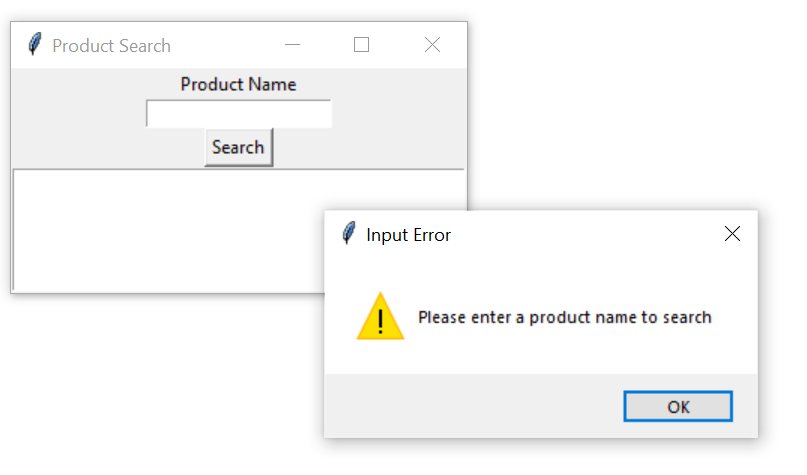
lb.pack()

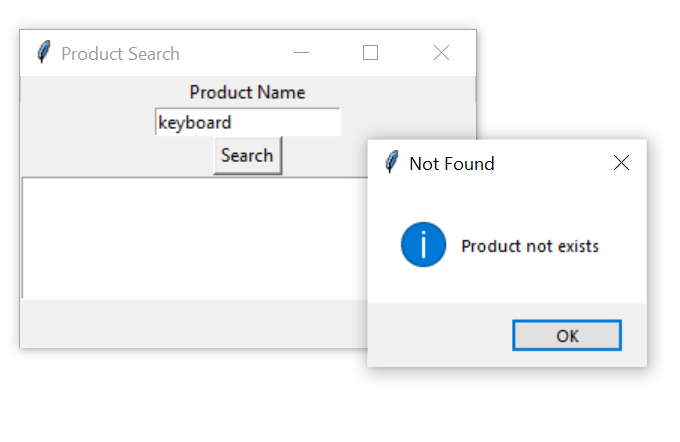
win.mainloop()

**OUTPUT:**









**c. Design a database application to that allows the user to add, delete and modify the records.**

**Code:**

import tkinter as tk

from tkinter import messagebox

import mysql.connector

conn = mysql.connector.connect(

host="localhost",

user="root",

password="",

database="product\_database"

)

def add\_product():

pname = pname\_entry.get()

quantity = quantity\_entry.get()

try:

c = conn.cursor()

c.execute("INSERT INTO prod (pname, quantity) VALUES (%s, %s)", (pname, quantity))

conn.commit()

update\_listbox()

clear\_entries()

except Exception as e:

messagebox.showerror("Database Error", str(e))

def delete\_product():

try:

c1 = conn.cursor()

selected\_index = listbox.curselection()[0]

selected\_product = listbox.get(selected\_index).split(' ')[0]

c1.execute("DELETE FROM prod WHERE id = %s", (selected\_product,))

conn.commit()

update\_listbox()

except IndexError:

messagebox.showwarning("Selection Error", "Please select a product to delete.")

except Exception as e:

messagebox.showerror("Database Error", str(e))

def modify\_product():

try:

c2 = conn.cursor()

selected\_index = listbox.curselection()[0]

sel\_pid=listbox.get(selected\_index).split(' ')[0]

new\_pname = pname\_entry.get()

new\_quantity = quantity\_entry.get()

c2.execute("UPDATE prod SET pname = %s, quantity = %s WHERE id = %s", (new\_pname, new\_quantity, sel\_pid))

conn.commit()

update\_listbox()

clear\_entries()

except IndexError:

messagebox.showwarning("Selection Error", "Please select a product to modify.")

except Exception as e:

messagebox.showerror("Database Error", str(e))

def on\_select(event):

selected\_index = listbox.curselection()[0]

sel\_pd,pname,qua = listbox.get(selected\_index).split(' ')

quan = ''.join([x for x in qua if x.isdigit()])

clear\_entries()

pname\_entry.insert(tk.END, pname)

quantity\_entry.insert(tk.END, quan)

def clear\_entries():

pname\_entry.delete(0, tk.END)

quantity\_entry.delete(0, tk.END)

def update\_listbox():

listbox.delete(0, tk.END)

c = conn.cursor()

c.execute("SELECT \* FROM prod")

products = c.fetchall()

for product in products:

listbox.insert(tk.END, f"{product[0]} {product[1]} (Quantity:{product[2]})")

root = tk.Tk()

root.title("Product Manager")

tk.Label(root, text="Product Name:").grid(row=0, column=0, padx=10, pady=10)

pname\_entry = tk.Entry(root, width=30)

pname\_entry.grid(row=0, column=1, padx=10, pady=10)

tk.Label(root, text="Quantity:").grid(row=1, column=0, padx=10, pady=10)

quantity\_entry = tk.Entry(root, width=30)

quantity\_entry.grid(row=1, column=1, padx=10, pady=10)

add\_button = tk.Button(root, text="Add Product", command=add\_product)

add\_button.grid(row=2, column=0, padx=10, pady=10)

modify\_button = tk.Button(root, text="Modify Product", command=modify\_product)

modify\_button.grid(row=2, column=1, padx=10, pady=10)

delete\_button = tk.Button(root, text="Delete Product", command=delete\_product)

delete\_button.grid(row=2, column=2, padx=10, pady=10)

listbox = tk.Listbox(root, height=10, width=60)

listbox.grid(row=3, column=0, columnspan=3, padx=10, pady=10)

update\_listbox()

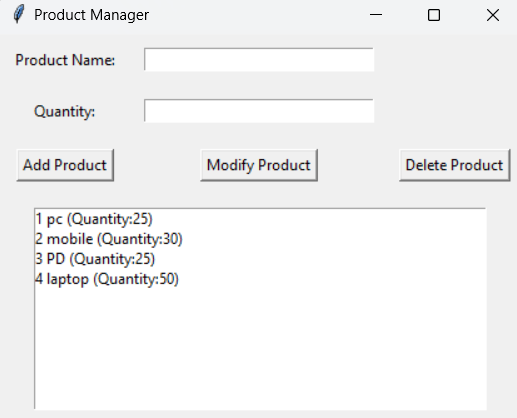
listbox.bind('<<ListboxSelect>>', on\_select)

root.mainloop()

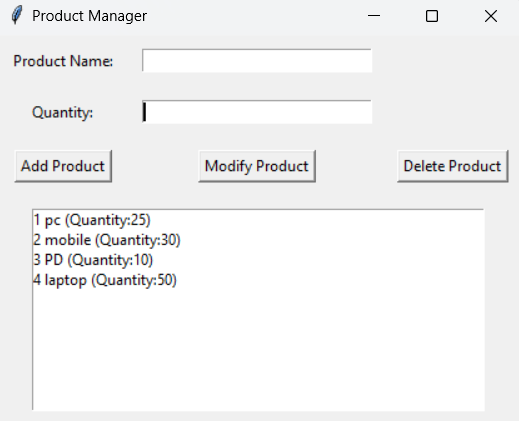
conn.close()

**Output:**

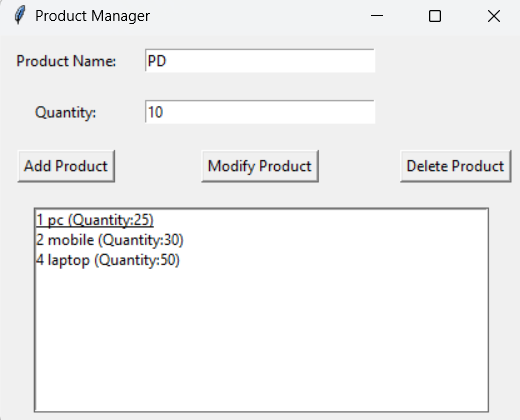
**Add Product:**

****

**Modify Product:**

****

**Delete Product:**

****