PRACTICAL NO 10
DESIGN THE DATABASE APPLICATIONS

a) Design a simple database application that stores the records and retrieve the same.

Code :

```
import mysql.connector
from tkinter import *
from customtkinter import *
from tkinter import messagebox

root=CTk()
root.title("Database management")
root.geometry("600x690")
root.resizable(False,False)
global mydb

mydb=mysql.connector.connect(host="127.0.0.1",user="root",
password="")
cur=mydb.cursor()
cur.execute("create database if not exists productstock")
mydb.commit()
cur.close()
mydb.close()
mydb=mysql.connector.connect(host="127.0.0.1",user="root",
password="",database="productstock")
cur=mydb.cursor()
sql="create table if not exists prod(pid int primary key
auto_increment,pname char(20) NOT NULL,pqty int NOT NULL)"
cur.execute(sql)

def insert():
    name=pname_entry.get()
    qty=pqty_entry.get()
    if name :
        if qty :
            #sql=f"insert (pname,pqty) into prod({name},{qty})"
            try :
                cur.execute("insert into prod(pname,pqty)
                 values(%s,%s)",(name,qty))
                messagebox.showinfo("Successful","This record can
                 insert to the database successfully")
                pname_entry.delete(0,END)
                pqty_entry.delete(0,END)
                mydb.commit()
                retrive()
            except :
                messagebox.showerror("Error 4433642","error occor
                 while inserting a record")
        else :
            messagebox.showerror("Invalid Input","The product
             quantity can not be blank")
    else :
        messagebox.showerror("Invalid Input","The product name can
        not be blank")
```

```
        def retrive():
            textbox.delete(0.0,END)
            sql="select pid,pname,pqty from prod"
            cur.execute(sql)
            rec=cur.fetchall()
            for r in rec :
                textbox.insert(END,f"Product Id   : {r[0]}\n")
                textbox.insert(END,f"Product Name : {r[1]}\n")
                textbox.insert(END,f"Product Qty  : {r[2]}\n\n")

        pname_text=CTkLabel(root,text="Product
        Name",font=("arial",22,"bold")).pack()
        pname_entry=CTkEntry(root,font=("arial",22,"bold"))
        pname_entry.pack()
        pqty_text=CTkLabel(root,text="Product
        Quantity",font=("arial",22,"bold")).pack()
        pqty_entry=CTkEntry(root,font=("arial",22,"bold"))
        pqty_entry.pack()
        insbtn=CTkButton(root,text="Insert",command=insert,font=("arial",
        22,"bold")).place(x=150,y=150)
        insbtn=CTkButton(root,text="Retrive",command=retrive,font=("arial
        ",22,"bold")).place(x=300,y=150)
        textbox = CTkTextbox(root,height=450,width=520)
        textbox.place(x=40,y=200)

        root.mainloop()
```

Output :

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

MySQL Query :

```
CREATE DATABASE bookdb;
USE bookdb;

CREATE TABLE books (
    id INT PRIMARY KEY AUTO_INCREMENT,
    title VARCHAR(255) NOT NULL,
    author VARCHAR(255) NOT NULL
);
```

Code :

```
import customtkinter as ctk
from tkinter import *
import mysql.connector
from mysql.connector import Error
from tkinter import messagebox

# Function to connect to the MySQL database
def create_connection():
    try:
        connection = mysql.connector.connect(host='localhost',
        user='root',password='',database='bookdb')
        return connection
    except Error as e:
        print(f"Error connecting to database: {e}")
        return None

def show_books(query=None):
    connection=create_connection()
    if connection:
        try:
            cursor=connection.cursor()
            if query:
                cursor.execute("SELECT title, author FROM books
                WHERE title LIKE %s OR author LIKE %s",
                (f'%{query}%', f'%{query}%'))
                results=cursor.fetchall()
            else :
                cursor.execute("SELECT title,author FROM books")
                results=cursor.fetchall()
            if results:
                clear_search()
                for result in results:
                    listbox.insert(ctk.END, f"Book Name :
                    {result[0]} Written by {result[1]}")
            else:
                listbox.insert(ctk.END, "No matching books found.")
            cursor.close()
        except Error as e:
            listbox.insert(ctk.END, f"Error: {e}")
```

```
        def add_book():
            name=book_entry.get()
            write=write_entry.get()
            if name :
                if write :
                    connection=create_connection()
                    if coznnection:
                        cursor=connection.cursor()
                        try :
                            cursor.execute("insert into books(title,author)
                             values(%s,%s)",(name,write))
                            messagebox.showinfo("Successful","Book can
                             inserted to the database successfully")
                            clear_entry()
                        except :
                            messagebox.showerror("Error: 404 ","error occor
                             while inserting a record")
                        finally :
                            show_books()
                            cursor.close()
                            connection.commit()
                            connection.close()
                else :
                    messagebox.showerror("Invalid Input","The product
                     quantity can not be blank")
            else :
                messagebox.showerror("Invalid Input","The product name can
                 not be blank")
        def search_books():
            query = search_entry.get()
            listbox.delete(0,ctk.END)
            if query:
                connection=create_connection()
                if connection:
                    cursor=connection.cursor()
                    show_books(query)
                    connection.close()
            else:
                listbox.insert(ctk.END,"Please enter.")

        def clear_search():
            search_entry.delete(0, ctk.END)
            listbox.delete(0, ctk.END)

        def clear_entry():
            book_entry.delete(0,ctk.END)
            write_entry.delete(0,ctk.END)

        root= ctk.CTk()
        root.title("Book Search")
        root.geometry("800x600")
        root.resizable(False,False)

        font1=font=('arial',20)
```

```
        book_text=ctk.CTkLabel(root,text="Book Name",font=font1).pack()
        book_entry=ctk.CTkEntry(root,font=font1,width=300)
        book_entry.pack(pady=5)

        write_text=ctk.CTkLabel(root,text="Book Writer",font=font1).pack()
        write_entry=ctk.CTkEntry(root,font=font1,width=300)
        write_entry.pack(pady=5)

        add_button = ctk.CTkButton(root, text="Add Book",
        command=add_book,font=font1,width=170)
        add_button.pack(pady=5)

        search_label = ctk.CTkLabel(root, text="Enter Book Name Or
        Title",font=font1)
        search_label.pack()
        search_entry = ctk.CTkEntry(root, width=400,font=('arial',20))
        search_entry.pack(pady=5)

        search_button = ctk.CTkButton(root, text="Search",
        command=search_books,font=font1)
        search_button.pack(pady=5)

        listbox =Listbox(root, width=60, height=10,font=('arial',20))
        listbox.pack(pady=5)

        clear_button = ctk.CTkButton(root, text="Clear",
        command=clear_search,font=font1)
        clear_button.pack(pady=5)

        root.mainloop()
```

Output :

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

Code :

Output :