student-database

June 12, 2023

[1]:
```
import tkinter as tk
from tkinter import messagebox
import mysql.connector


connection = None


def create_student_table(cursor):
    create_table_query = """
        CREATE TABLE IF NOT EXISTS student_data (
            student_id INT PRIMARY KEY,
            name VARCHAR(255),
            age INT,
            courses VARCHAR(255),
            father_name VARCHAR(255),
            mother_name VARCHAR(255),
            address VARCHAR(255),
            phone_num VARCHAR(255)
        )
    """
    cursor.execute(create_table_query)

def display_student_info(student):
    messagebox.showinfo("Student Information",
                        f"Student ID: {student[0]}\n"
                        f"Name: {student[1]}\n"
                        f"Age: {student[2]}\n"
                        f"Courses: {student[3]}\n"
                        f"Father Name: {student[4]}\n"
                        f"Mother Name: {student[5]}\n"
                        f"Address: {student[6]}\n"
                        f"Phone Num: {student[7]}\n")

def save_student():
    global connection
```

```
    student_id = entry_student_id.get()
    name = entry_name.get()
    age = entry_age.get()
    course = var_course.get()
    father_name = entry_father_name.get()
    mother_name = entry_mother_name.get()
    address = entry_address.get()
    phone_num = entry_phone_num.get()

    try:

        connection = mysql.connector.connect(
            host="localhost",
            user="root",
            database="student",
            password="12345"
        )

        cursor = connection.cursor()

        create_student_table(cursor)

        query = "SELECT * FROM student_data WHERE name = %s"
        cursor.execute(query, (name,))
        student = cursor.fetchone()

        if student:
            display_student_info(student)
        else:
            insert_query = "INSERT INTO student_data (student_id, name, age,␣
↪courses, father_name, mother_name, address, phone_num) VALUES (%s, %s, %s,␣
↪%s, %s, %s, %s, %s)"
            values = (student_id, name, age, course, father_name, mother_name,␣
↪address, phone_num)
            cursor.execute(insert_query, values)
            connection.commit()

            messagebox.showinfo("Success", "Student data saved successfully!")

            entry_student_id.delete(0, tk.END)
            entry_name.delete(0, tk.END)
            entry_age.delete(0, tk.END)
            entry_father_name.delete(0, tk.END)
            entry_mother_name.delete(0, tk.END)
            entry_address.delete(0, tk.END)
            entry_phone_num.delete(0, tk.END)
```

```
    except mysql.connector.Error as error:
        messagebox.showerror("Error", f"An error occurred: {error}")

    finally:
        if connection and connection.is_connected():
            cursor.close()
            connection.close()

def search_student():
    global connection
    name = entry_search_name.get()

    try:
        connection = mysql.connector.connect(
            host="localhost",
            user="root",
            database="student",
            password="12345"
        )

        cursor = connection.cursor()

        create_student_table(cursor)

        query = "SELECT * FROM student_data WHERE name = %s"
        cursor.execute(query, (name,))
        student = cursor.fetchone()

        if student:
            display_student_info(student)
        else:
            messagebox.showinfo("Not Found", "Student not found!")

        entry_search_name.delete(0, tk.END)

    except mysql.connector.Error as error:
        messagebox.showerror("Error", f"An error occurred: {error}")

    finally:
        if connection and connection.is_connected():
            cursor.close()
            connection.close()

window = tk.Tk()
window.title("Student Information App")

label_student_id = tk.Label(window, text="Student ID:")
```

```
label_student_id.pack()
entry_student_id = tk.Entry(window)
entry_student_id.pack()


label_name = tk.Label(window, text="Name:")
label_name.pack()
entry_name = tk.Entry(window)
entry_name.pack()


label_age = tk.Label(window, text="Age:")
label_age.pack()
entry_age = tk.Entry(window)
entry_age.pack()

label_course = tk.Label(window, text="Course:")
label_course.pack()
var_course = tk.StringVar(window)
var_course.set("CSE")  # Default selection
dropdown_course = tk.OptionMenu(window, var_course, "CSE", "EEE", "EC", "BTECH")
dropdown_course.pack()


label_father_name = tk.Label(window, text="Father's Name:")
label_father_name.pack()
entry_father_name = tk.Entry(window)
entry_father_name.pack()

label_mother_name = tk.Label(window, text="Mother's Name:")
label_mother_name.pack()
entry_mother_name = tk.Entry(window)
entry_mother_name.pack()


label_address = tk.Label(window, text="Address:")
label_address.pack()
entry_address = tk.Entry(window)
entry_address.pack()


label_phone_num = tk.Label(window, text="Phone Number:")
label_phone_num.pack()
entry_phone_num = tk.Entry(window)
entry_phone_num.pack()

btn_save = tk.Button(window, text="Save", command=save_student)
```

```
btn_save.pack()


label_search_name = tk.Label(window, text="Search Name:")
label_search_name.pack()
entry_search_name = tk.Entry(window)
entry_search_name.pack()


btn_search = tk.Button(window, text="Search", command=search_student)
btn_search.pack()


window.mainloop()
```

[ ]: