import mysql.connector  
import tkinter as tk  
from tkinter import ttk, messagebox  
  
# Function to connect to MySQL  
def connect\_to\_db():  
 try:  
 conn = mysql.connector.connect(  
 host="127.0.0.1",  
 user="root",  
 password="",  
 database="student\_db"  
 )  
 if conn.is\_connected():  
 messagebox.showinfo("Success", "Connected to MySQL successfully!")  
 conn.close()  
 else:  
 messagebox.showerror("Error", "Connection to MySQL failed!")  
 except mysql.connector.Error as err:  
 messagebox.showerror("Database Error", f"Error: {err}")  
  
class StudentApp:  
 def \_\_init\_\_(self, root):  
 self.root = root  
 self.root.title("Student Tracker")  
 self.root.geometry("900x600")  
  
 # === Title ===  
 title = tk.Label(self.root, text="STUDENT TRACKER", font=("Arial", 20, "bold"),  
 bg="#ff5733", fg="white", pady=10)  
 title.pack(side=tk.TOP, fill=tk.X)  
  
 # === Left Frame (Form) ===  
 form\_frame = tk.Frame(self.root, bd=4, bg="#f6e0b5", relief=tk.RIDGE)  
 form\_frame.place(x=20, y=60, width=350, height=500)  
  
 lbl\_frame = tk.LabelFrame(form\_frame, text="Student Details", bg="#eea990",  
 font=("Arial", 12, "bold"))  
 lbl\_frame.pack(fill="both", expand=True, padx=10, pady=10)  
  
 # Input Fields  
 tk.Label(lbl\_frame, text="Student No.", bg="#eea990",  
 font=("Arial", 12)).grid(row=0, column=0, padx=10, sticky="w")  
 self.txt\_No = tk.Entry(lbl\_frame, font=("Arial", 10))  
 self.txt\_No.grid(row=1, column=0, padx=10, sticky="we")  
  
 tk.Label(lbl\_frame, text="Name", bg="#eea990",  
 font=("Arial", 12)).grid(row=2, column=0, padx=10, sticky="w")  
 self.txt\_name = tk.Entry(lbl\_frame, font=("Arial", 10))  
 self.txt\_name.grid(row=3, column=0, padx=10, sticky="we")  
  
 tk.Label(lbl\_frame, text="Attendance", bg="#eea990",  
 font=("Arial", 12)).grid(row=4, column=0, padx=10, sticky="w")  
 self.txt\_attendance = tk.Entry(lbl\_frame, font=("Arial", 10))  
 self.txt\_attendance.grid(row=5, column=0, padx=10, sticky="we")  
  
 tk.Label(lbl\_frame, text="Quiz Score", bg="#eea990",  
 font=("Arial", 12)).grid(row=6, column=0, padx=10, sticky="w")  
 self.txt\_quiz = tk.Entry(lbl\_frame, font=("Arial", 10))  
 self.txt\_quiz.grid(row=7, column=0, padx=10, sticky="we")  
  
 tk.Label(lbl\_frame, text="Exam Score", bg="#eea990",  
 font=("Arial", 12)).grid(row=8, column=0, padx=10, sticky="w")  
 self.txt\_exam = tk.Entry(lbl\_frame, font=("Arial", 10))  
 self.txt\_exam.grid(row=9, column=0, padx=10, sticky="we")  
  
 # === Buttons ===  
 btn\_frame = tk.Frame(lbl\_frame, bg="#eea990")  
 btn\_frame.grid(row=10, column=0, pady=10)  
  
 tk.Button(btn\_frame, text="Add Student", width=12,  
 command=self.add\_student).grid(row=0, column=0, padx=10, pady=5)  
 tk.Button(btn\_frame, text="Edit Info", width=12,  
 command=self.update\_student).grid(row=1, column=0, padx=10, pady=5)  
 tk.Button(btn\_frame, text="Clear Form", width=12,  
 command=self.clear\_form).grid(row=2, column=0, padx=10, pady=5)  
  
 # === Right Frame (Student Table) ===  
 details\_frame = tk.Frame(self.root, bd=4, bg="#eea990", relief=tk.RIDGE)  
 details\_frame.place(x=400, y=60, width=480, height=500)  
  
 tk.Label(details\_frame, text="Students Information",  
 bg="#eea990", font=("Arial", 14, "bold")).pack(pady=10)  
  
 self.Student\_table = ttk.Treeview(details\_frame,  
 columns=("Student No.", "Name", "Attendance",  
 "Quiz Score", "Exam Score"), show="headings")  
 self.Student\_table.pack(fill=tk.BOTH, expand=1)  
  
 for col in ("Student No.", "Name", "Attendance", "Quiz Score", "Exam Score"):  
 self.Student\_table.heading(col, text=col)  
 self.Student\_table.column(col, width=100)  
  
 # Bind row click to select\_student function  
 self.Student\_table.bind("<ButtonRelease-1>", self.select\_student)  
  
 # Load stored students  
 self.fetch\_students()  
  
 # === Add Student ===  
 def add\_student(self):  
 if not all([self.txt\_No.get(), self.txt\_name.get(), self.txt\_attendance.get(),  
 self.txt\_quiz.get(), self.txt\_exam.get()]):  
 messagebox.showwarning("Warning", "All fields are required!")  
 return  
  
 try:  
 conn = mysql.connector.connect(host="127.0.0.1", user="root", password="",  
 database="student\_db")  
 cursor = conn.cursor()  
 query = "INSERT INTO students (student\_no, name, attendance, quiz\_score, exam\_score) VALUES (%s, %s, %s, %s, %s)"  
 values = (self.txt\_No.get(), self.txt\_name.get(), self.txt\_attendance.get(),  
 self.txt\_quiz.get(), self.txt\_exam.get())  
 cursor.execute(query, values)  
 conn.commit()  
 cursor.close()  
 conn.close()  
  
 messagebox.showinfo("Success", "Student added successfully!")  
 self.fetch\_students() # Refresh the table  
 except mysql.connector.Error as e:  
 messagebox.showerror("Error", f"Database Error: {e}")  
  
 # === Fetch Students ===  
 def fetch\_students(self):  
 try:  
 conn = mysql.connector.connect(host="127.0.0.1", user="root", password="", database="student\_db")  
 cursor = conn.cursor()  
 cursor.execute("SELECT \* FROM students")  
 rows = cursor.fetchall()  
 cursor.close()  
 conn.close()  
  
 self.Student\_table.delete(\*self.Student\_table.get\_children())  
 for row in rows:  
 self.Student\_table.insert("", "end", values=row)  
  
 except mysql.connector.Error as e:  
 messagebox.showerror("Error", f"Database Error: {e}")  
  
 # === Select Student (Click Row) ===  
 def select\_student(self, event):  
 selected = self.Student\_table.focus()  
 values = self.Student\_table.item(selected, "values")  
 if values:  
 self.txt\_No.delete(0, tk.END)  
 self.txt\_name.delete(0, tk.END)  
 self.txt\_attendance.delete(0, tk.END)  
 self.txt\_quiz.delete(0, tk.END)  
 self.txt\_exam.delete(0, tk.END)  
  
 self.txt\_No.insert(0, values[0])  
 self.txt\_name.insert(0, values[1])  
 self.txt\_attendance.insert(0, values[2])  
 self.txt\_quiz.insert(0, values[3])  
 self.txt\_exam.insert(0, values[4])  
  
 # === Update Student Info ===  
 def update\_student(self):  
 try:  
 conn = mysql.connector.connect(host="127.0.0.1", user="root", password="", database="student\_db")  
 cursor = conn.cursor()  
 query = """UPDATE students SET name=%s, attendance=%s, quiz\_score=%s, exam\_score=%s WHERE student\_no=%s"""  
 values = (self.txt\_name.get(), self.txt\_attendance.get(), self.txt\_quiz.get(), self.txt\_exam.get(), self.txt\_No.get())  
 cursor.execute(query, values)  
 conn.commit()  
 cursor.close()  
 conn.close()  
  
 messagebox.showinfo("Success", "Student updated successfully!")  
 self.fetch\_students()  
  
 except mysql.connector.Error as e:  
 messagebox.showerror("Error", f"Database Error: {e}")  
  
 # === Clear Form ===  
 def clear\_form(self):  
 for entry in [self.txt\_No, self.txt\_name, self.txt\_attendance, self.txt\_quiz, self.txt\_exam]:  
 entry.delete(0, tk.END)  
  
  
root = tk.Tk()  
app = StudentApp(root)  
root.mainloop()