# F.CRUD_operations_GUI_SQL

June 17, 2024

[1]:
```
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
import mysql.connector

db_host = "localhost"
db_user = "root"
db_password = ",428A3B2UDpY"

root = None
feedback_table = None
selected_row = None

def quit():
    global root
    if messagebox.askokcancel("Quit", "Are you sure you want to quit?"):
        root.destroy()


def home():
    global root
    if root:
        root.destroy()
    root = Tk()
    root.title("Registration Form")
    root.resizable(False, False)
    frm = ttk.Frame(root, padding=10)
    frm.grid()

    global name_var, email_var, number_var, gender_var, feedback_var
    name_var = StringVar()
    email_var = StringVar()
    number_var = StringVar()
    gender_var = StringVar()
    feedback_var = StringVar()

    ttk.Label(frm, text="Name:").grid(column=0, row=0, sticky=W)
```

```
    name_entry = ttk.Entry(frm, textvariable=name_var)
    name_entry.grid(column=1, row=0, sticky=(W, E))
    name_entry.focus()

    ttk.Label(frm, text="Email:").grid(column=0, row=1, sticky=W)
    email_entry = ttk.Entry(frm, textvariable=email_var)
    email_entry.grid(column=1, row=1, sticky=(W, E))

    ttk.Label(frm, text="Number:").grid(column=0, row=2, sticky=W)
    number_entry = ttk.Entry(frm, textvariable=number_var)
    number_entry.grid(column=1, row=2, sticky=(W, E))

    ttk.Label(frm, text="Gender:").grid(column=0, row=3, sticky=W)
    ttk.Radiobutton(frm, text="Male", variable=gender_var, value="M").grid(
        column=1, row=3, sticky=W
    )
    ttk.Radiobutton(frm, text="Female", variable=gender_var, value="F").grid(
        column=2, row=3, sticky=W
    )

    ttk.Label(frm, text="Feedback:").grid(column=0, row=4, sticky=W)
    feedback_entry = ttk.Entry(frm, textvariable=feedback_var)
    feedback_entry.grid(column=1, row=4, sticky=(W, E))

    ttk.Button(frm, text="Submit", command=submit_form).grid(column=1, row=5,␣
 ↪sticky=E)
    ttk.Button(frm, text="Quit", command=quit).grid(column=2, row=5, sticky=E)

    ttk.Button(frm, text="View Feedback", command=view_feedback).grid(
        column=0, row=5, sticky=W
    )

    root.mainloop()


def submit_form():
    name = name_var.get()
    email = email_var.get()
    number = number_var.get()
    gender = gender_var.get()
    feedback = feedback_var.get()

    try:
        db = mysql.connector.connect(
            host=db_host,
            user=db_user,
            password=db_password,
```

```
        )
        cursor = db.cursor()

        cursor.execute("CREATE DATABASE IF NOT EXISTS forms")
        cursor.execute("USE forms")

        cursor.execute(
            "CREATE TABLE IF NOT EXISTS feedback (id INT AUTO_INCREMENT PRIMARY␣
↪KEY, name VARCHAR(255), email VARCHAR(255), number VARCHAR(20), gender␣
↪VARCHAR(1), feedback TEXT)"
        )

        sql = "INSERT INTO feedback (name, email, number, gender, feedback)␣
↪VALUES (%s, %s, %s, %s, %s)"
        val = (name, email, number, gender, feedback)
        cursor.execute(sql, val)

        db.commit()
        db.close()

        messagebox.showinfo("Success", "Form submitted successfully!")
        home()

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


def view_feedback():
    global feedback_table, root, tree
    try:
        if feedback_table:
            feedback_table.destroy()
    except:
        pass

    root.withdraw()
    feedback_table = Tk()
    feedback_table.title("Feedback Table")
    feedback_table.resizable(False, False)

    tree = ttk.Treeview(feedback_table)
    tree["columns"] = ("id", "Name", "Email", "Number", "Gender", "Feedback")
    tree.heading("id", text="ID")
    tree.heading("Name", text="Name")
    tree.heading("Email", text="Email")
    tree.heading("Number", text="Number")
    tree.heading("Gender", text="Gender")
```

```
    tree.heading("Feedback", text="Feedback")
    tree.bind("<<TreeviewSelect>>", on_tree_select)

    tree.grid()

    try:
        db = mysql.connector.connect(
            host=db_host, user=db_user, password=db_password, database="forms"
        )
        cursor = db.cursor()

        cursor.execute("SELECT * FROM feedback")
        rows = cursor.fetchall()
        for row in rows:
            tree.insert("", "end", values=row)

        db.close()

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

    ttk.Button(feedback_table, text="Close", command=close_feedback).grid()
    ttk.Button(feedback_table, text="Delete Feedback", command=delete_feedback).
 ↪grid()
    ttk.Button(feedback_table, text="Update Feedback", command=update_feedback).
 ↪grid()


def on_tree_select(event):
    global selected_row
    selected_item = tree.focus()
    selected_row = None
    if selected_item:
        selected_row = tree.item(selected_item, "values")

    global updated_name, updated_email, updated


def delete_feedback():
    global feedback_table

    selected_item = tree.focus()
    if selected_item:
        if_confirm = messagebox.askyesno(
            "Confirm Deletion", "Are you sure you want to delete this feedback?"
        )
        if if_confirm:
```

```
            db = mysql.connector.connect(
                host=db_host, user=db_user, password=db_password,␣
 ↪database="forms"
            )
            cursor = db.cursor()

            item_id = tree.item(selected_item)["values"][0]
            cursor.execute("DELETE FROM feedback WHERE id = %s", (item_id,))
            db.commit()
            db.close()

            messagebox.showinfo("Success", "Feedback deleted successfully!")
            view_feedback()


def update_feedback():
    global feedback_table, root, selected_row
    if selected_row:
        # Destroy the feedback table window
        feedback_table.destroy()

        # Hide the main window
        root.withdraw()
        global update_window
        update_window = Tk()
        update_window.title("Update Feedback")
        update_window.resizable(False, False)
        global row_id

        # Define StringVar variables for each entry field
        row_id = selected_row[0]
        name_var = StringVar(value=selected_row[1])  # Index 1 contains the name
        email_var = StringVar(value=selected_row[2])  # Index 2 contains the␣
 ↪email
        number_var = StringVar(value=selected_row[3])  # Index 3 contains the␣
 ↪number
        gender_var = StringVar(value=selected_row[4])  # Index 4 contains the␣
 ↪gender
        feedback_var = StringVar(value=selected_row[5])  # Index 5 contains the␣
 ↪feedback

        ttk.Label(update_window, text="Name:").grid(column=0, row=0, sticky=W)
        name_entry = ttk.Entry(update_window, textvariable=name_var)
        name_entry.insert(0, selected_row[1])
        name_entry.grid(column=1, row=0, sticky=(W, E))

        ttk.Label(update_window, text="Email:").grid(column=0, row=1, sticky=W)
```

```
        email_entry = ttk.Entry(update_window, textvariable=email_var)
        email_entry.insert(0, selected_row[2])
        email_entry.grid(column=1, row=1, sticky=(W, E))

        ttk.Label(update_window, text="Number:").grid(column=0, row=2, sticky=W)
        number_entry = ttk.Entry(update_window, textvariable=number_var)
        number_entry.insert(0, selected_row[3])
        number_entry.grid(column=1, row=2, sticky=(W, E))

        ttk.Label(update_window, text="Gender:").grid(column=0, row=3, sticky=W)
        ttk.Radiobutton(
            update_window, text="Male", variable=gender_var, value="M"
        ).grid(column=1, row=3, sticky=W)
        ttk.Radiobutton(
            update_window, text="Female", variable=gender_var, value="F"
        ).grid(column=2, row=3, sticky=W)

        ttk.Label(update_window, text="Feedback:").grid(column=0, row=4,␣
 ↪sticky=W)
        feedback_entry = ttk.Entry(update_window, textvariable=feedback_var)
        feedback_entry.insert(0, selected_row[5])
        feedback_entry.grid(column=1, row=4, sticky=(W, E))

        ttk.Button(
            update_window,
            text="Update",
            command=lambda: update_form(
                name_entry.get(),
                email_entry.get(),
                number_entry.get(),
                gender_var.get(),
                feedback_entry.get(),
            ),
        ).grid(column=1, row=5, sticky=E)

        update_window.mainloop()
    else:
        messagebox.showwarning("No Selection", "Please select a feedback to␣
 ↪update.")


def update_form(
    updated_name, updated_email, updated_number, updated_gender,␣
 ↪updated_feedback
):
    global selected_row
    # Retrieve updated values from the entry fields
```

```
    window = Tk()
    window.withdraw()

    try:
        db = mysql.connector.connect(
            host=db_host, user=db_user, password=db_password, database="forms"
        )
        cursor = db.cursor()


        cursor.execute(
            "UPDATE feedback SET name = %s, email = %s, number = %s, gender =␣
 ↪%s, feedback = %s WHERE id = %s",
            (
                updated_name,
                updated_email,
                updated_number,
                updated_gender,
                updated_feedback,
                row_id,
            ),
        )
        db.commit()
        db.close()

        messagebox.showinfo("Success", "Feedback updated successfully!")
        window.destroy()
        view_feedback()
        update_window.destroy()

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


def close_feedback():
    global feedback_table, root
    feedback_table.destroy()
    root.deiconify()


home()
```