Python II

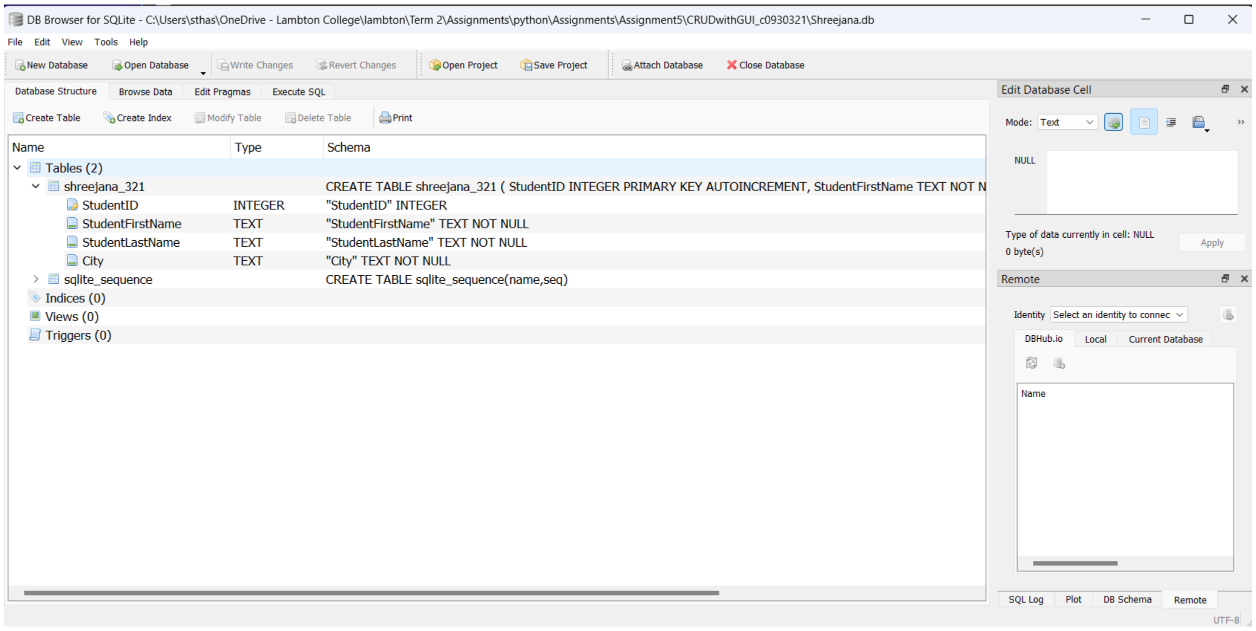
Student ID : C0930321

Student Name : Shreejana Shrestha

Assignment: 5

# GUI based python application to perform CRUD Operation using SQLITE3

# Creation of database and table

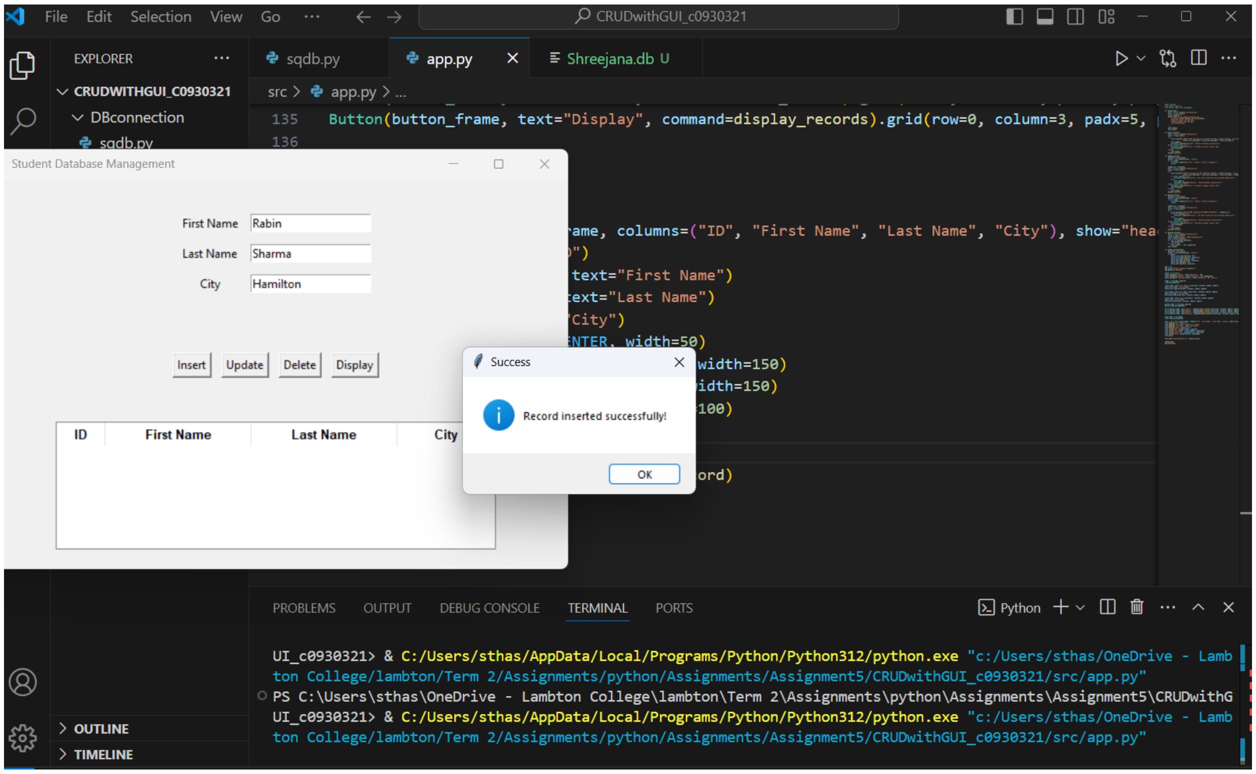


DB Browser for SQLite - C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithGUI_c0930321\Shreejana.db

Name	Type	Schema
Tables (2)		
shreejana_321		CREATE TABLE shreejana_321 (StudentID INTEGER PRIMARY KEY AUTOINCREMENT, StudentFirstName TEXT NOT N
StudentID	INTEGER	"StudentID" INTEGER
StudentFirstName	TEXT	"StudentFirstName" TEXT NOT NULL
StudentLastName	TEXT	"StudentLastName" TEXT NOT NULL
City	TEXT	"City" TEXT NOT NULL
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
Indices (0)		
Views (0)		
Triggers (0)		

Type of data currently in cell: NULL
0 byte(s)

# Inserting the record and showing the success message



```
Button(button_frame, text="Display", command=display_records).grid(row=0, column=3, padx=5,
```

```
ame, columns=("ID", "First Name", "Last Name", "City"), show="hea
")
 text="First Name")
ext="Last Name")
"City")
NTER, width=50)
width=150)
idth=150)
100)

ord)
```

Student Database Management

First Name	Rabin
Last Name	Sharma
City	Hamilton

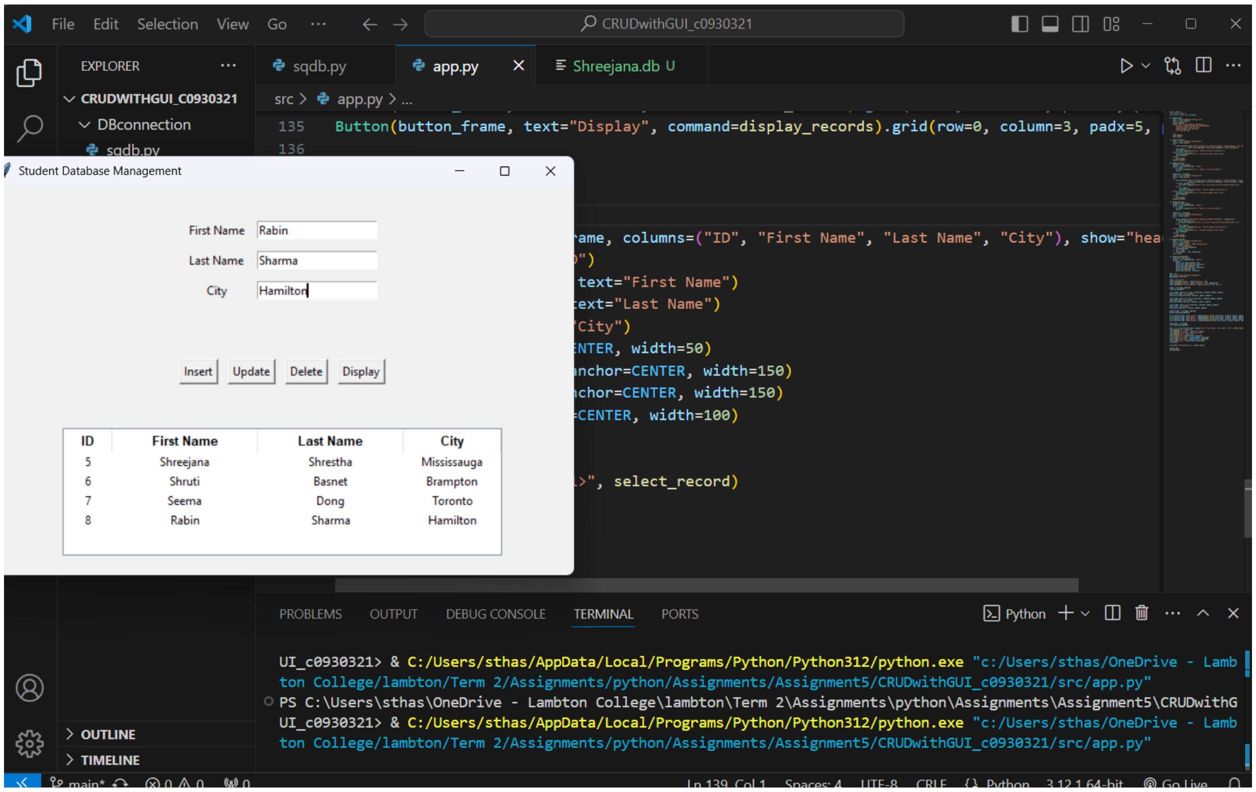
Success

Record inserted successfully!

ID	First Name	Last Name	City

```
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
PS C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithG
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
```

# Record displayed after insertion of data



```
Button(button_frame, text="Display", command=display_records).grid(row=0, column=3, padx=5,
```

```
ame, columns=("ID", "First Name", "Last Name", "City"), show="hea
")
text="First Name")
ext="Last Name")
"City")
NTER, width=50)
anchor=CENTER, width=150)
chor=CENTER, width=150)
CENTER, width=100)

>", select_record)
```

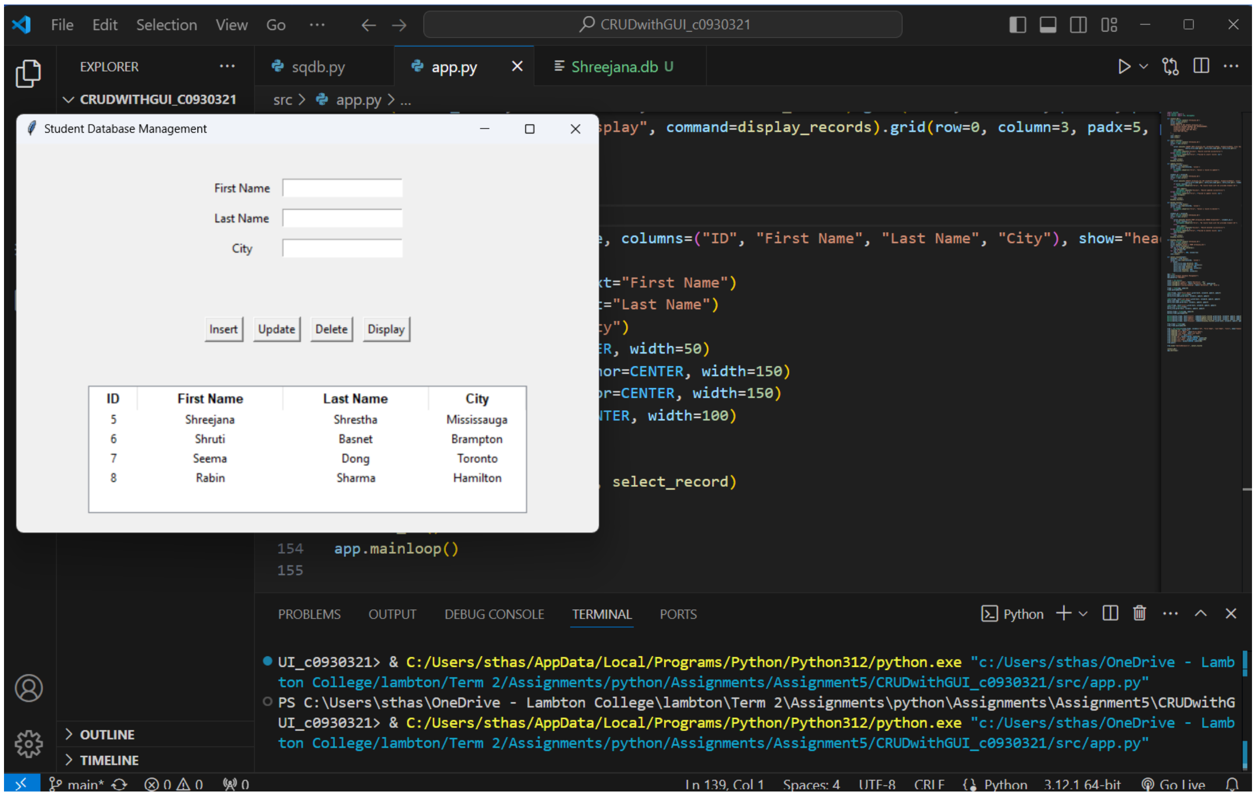
Student Database Management

First Name	Rabin
Last Name	Sharma
City	Hamilton

ID	First Name	Last Name	City
5	Shreejana	Shrestha	Mississauga
6	Shruti	Basnet	Brampton
7	Seema	Dong	Toronto
8	Rabin	Sharma	Hamilton

```
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
PS C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithG
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
```

Displaying all the record on click of display button



First Name	
Last Name	
City	

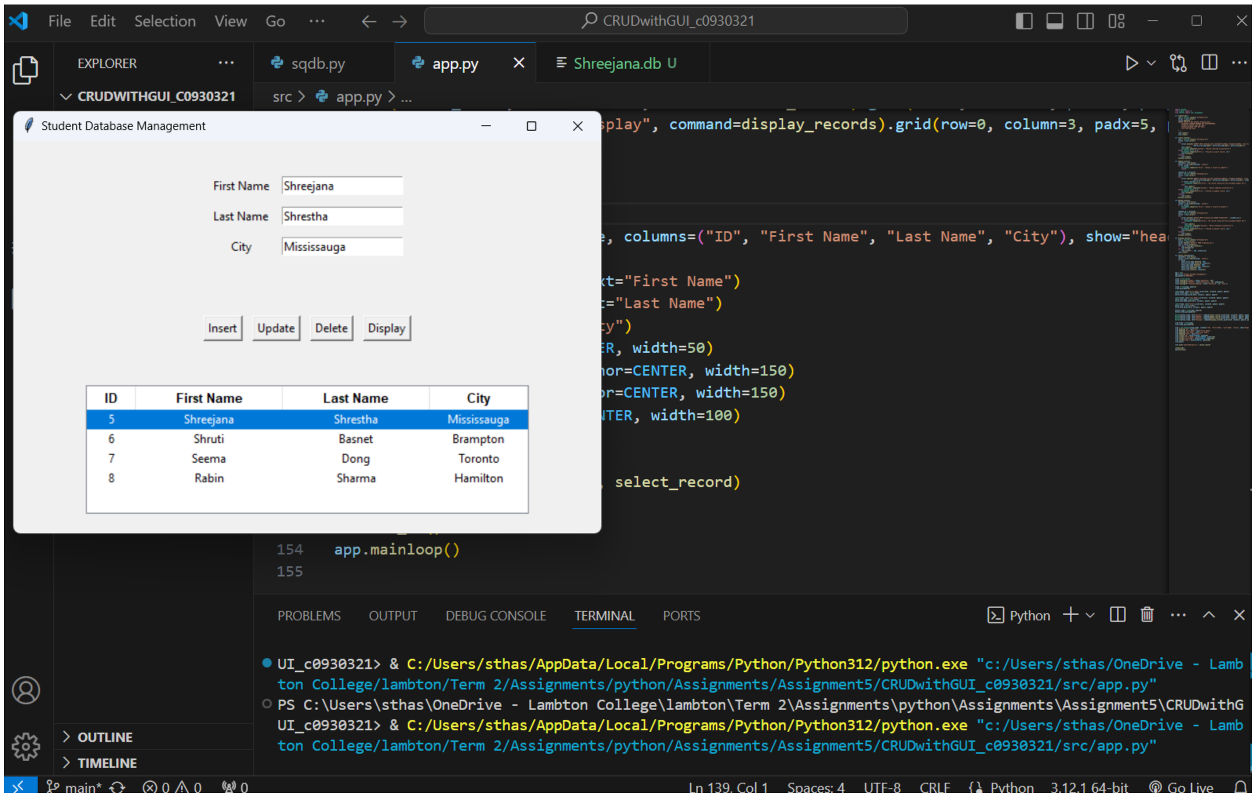
ID	First Name	Last Name	City
5	Shreejana	Shrestha	Mississauga
6	Shruti	Basnet	Brampton
7	Seema	Dong	Toronto
8	Rabin	Sharma	Hamilton

```
splay", command=display_records).grid(row=0, column=3, padx=5,

e, columns=("ID", "First Name", "Last Name", "City"), show="hea

xt="First Name")
t="Last Name")
ty")
ER, width=50)
hor=CENTER, width=150)
or=CENTER, width=150)
NTER, width=100)

, select_record)

app.mainloop()
```

```
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lambton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
PS C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithGUI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lambton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
```

```
main*  ⊗ 0 ⚠ 0  0    Ln 139, Col 1  Spaces: 4  UTF-8  CRLF  {} Python  3.12.1 64-bit  Go Live
```

Selecting the record to perform update or delete operation



File Edit Selection View Go ...
CRUDwithGUI_c0930321
EXPLORER
CRUDWITHGUI_C0930321
sqdb.py
app.py
Shreejana.db U
src > app.py > ...
Student Database Management
Insert
Update
Delete
Display
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
main* ⊗ 0 ⚠ 0 ⑇ 0
Ln 139, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.1 64-bit Go Live

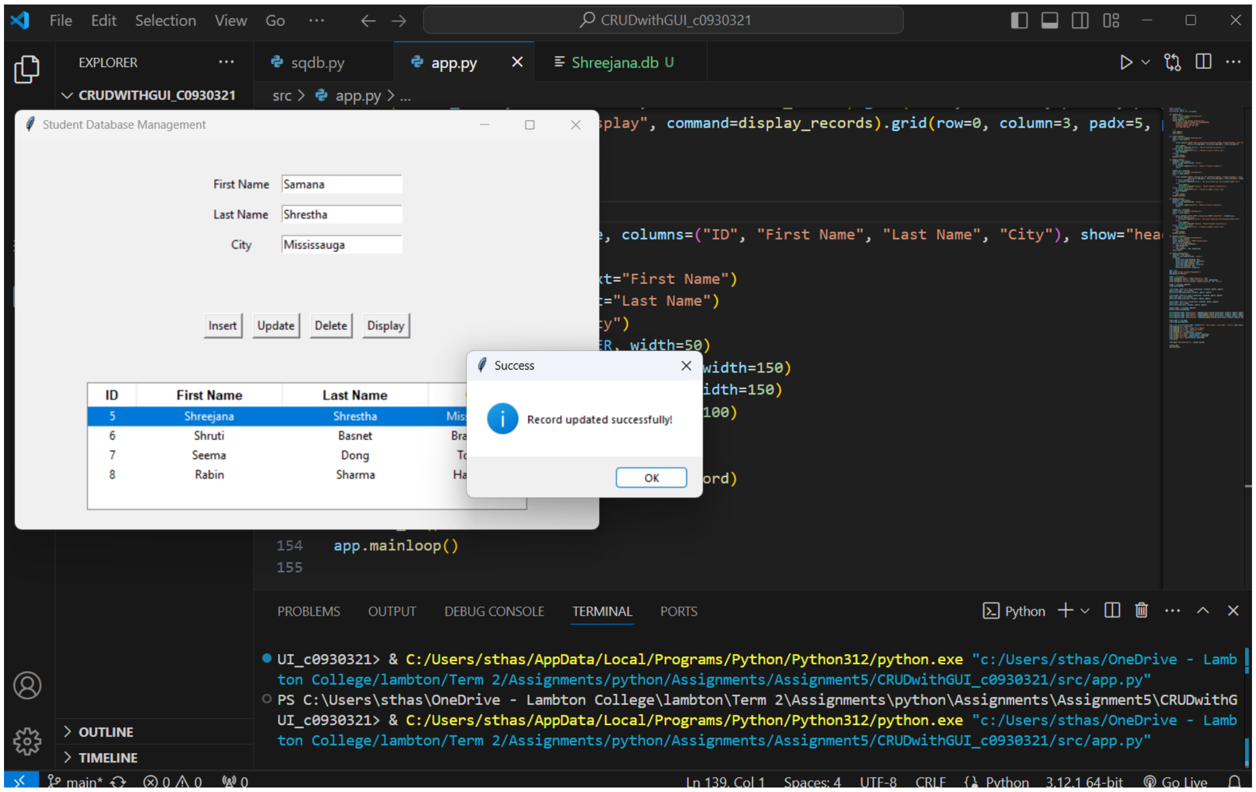
First Name: Shreejana
Last Name: Shrestha
City: Mississauga

ID	First Name	Last Name	City
5	Shreejana	Shrestha	Mississauga
6	Shruti	Basnet	Brampton
7	Seema	Dong	Toronto
8	Rabin	Sharma	Hamilton

```
splay", command=display_records).grid(row=0, column=3, padx=5,

, columns=("ID", "First Name", "Last Name", "City"), show="hea

xt="First Name")
t="Last Name")
ty")
ER, width=50)
hor=CENTER, width=150)
or=CENTER, width=150)
NTER, width=100)

, select_record)

app.mainloop()
```

```
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
PS C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithG
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
```

# Updating of the record



Student Database Management

First Name: Samana
Last Name: Shrestha
City: Mississauga

ID	First Name	Last Name	
5	Shreejana	Shrestha	Mis
6	Shruti	Basnet	Bra
7	Seema	Dong	To
8	Rabin	Sharma	Ha

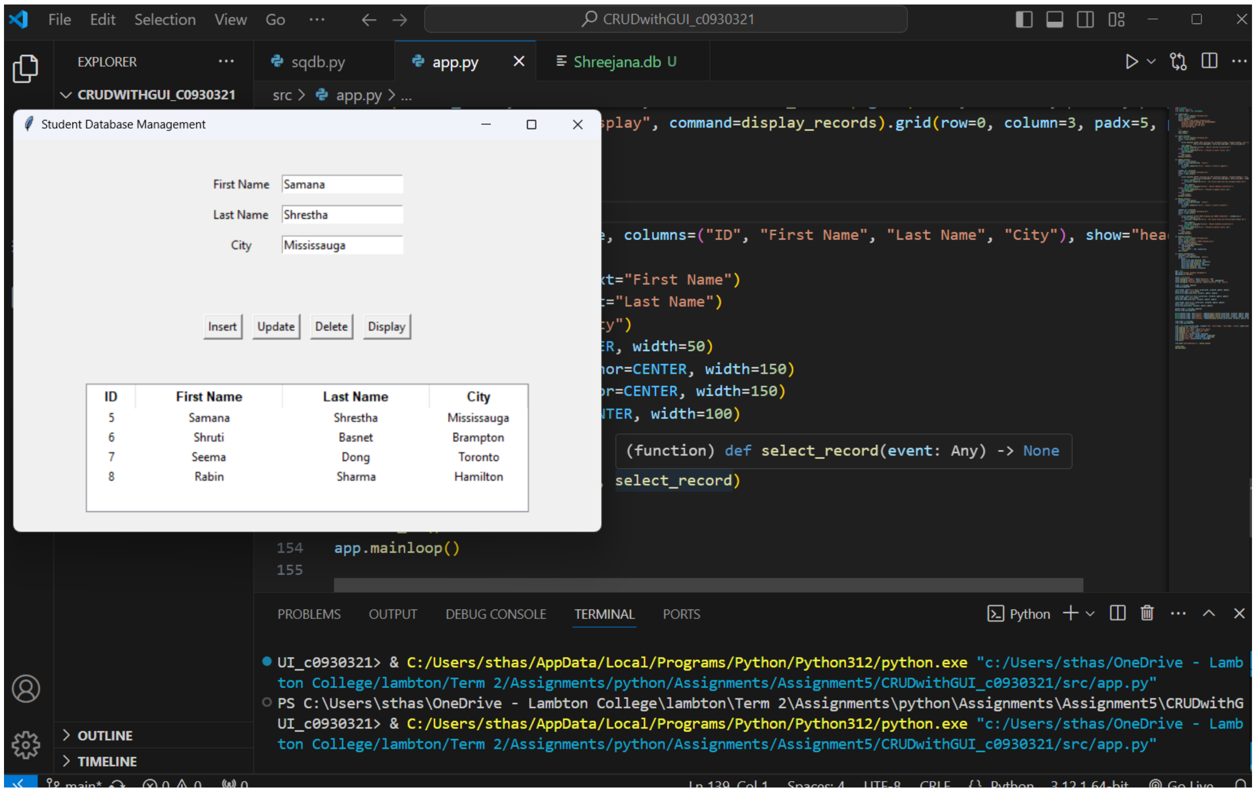
Success

Record updated successfully!

```
splay", command=display_records).grid(row=0, column=3, padx=5,
, columns=("ID", "First Name", "Last Name", "City"), show="hea
t="First Name")
="Last Name")
y")
R, width=50)
width=150)
idth=150)
100)
ord)
app.mainloop()
```

```
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
PS C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithG
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
```

# After update of record that is selected above



Student Database Management

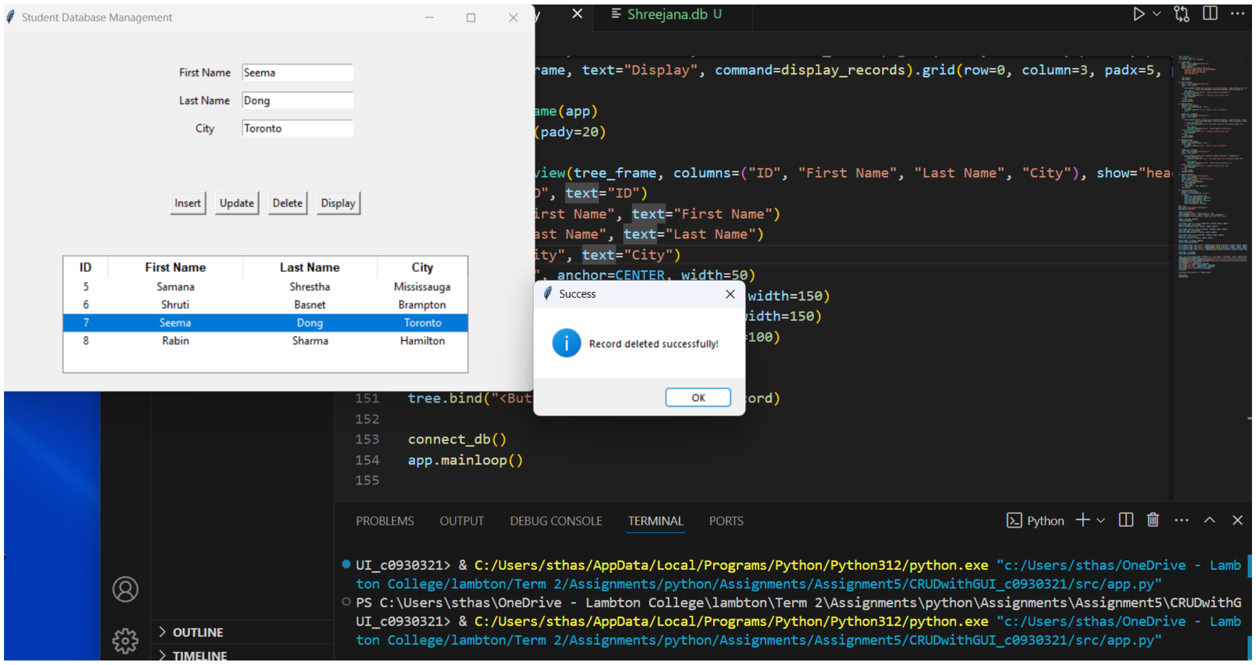
First Name: Samana
Last Name: Shrestha
City: Mississauga

ID	First Name	Last Name	City
5	Samana	Shrestha	Mississauga
6	Shruti	Basnet	Brampton
7	Seema	Dong	Toronto
8	Rabin	Sharma	Hamilton

```
splay", command=display_records).grid(row=0, column=3, padx=5,
, columns=("ID", "First Name", "Last Name", "City"), show="hea
t="First Name")
="Last Name")
y")
R, width=50)
or=CENTER, width=150)
r=CENTER, width=150)
TER, width=100)
(function) def select_record(event: Any) -> None
select_record)
app.mainloop()
```

```
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
PS C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithG
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
```

# Deleting the record by selecting the row from the listed table



First Name: Seema
Last Name: Dong
City: Toronto

ID	First Name	Last Name	City
5	Samana	Shrestha	Mississauga
6	Shruti	Basnet	Brampton
7	Seema	Dong	Toronto
8	Rabin	Sharma	Hamilton

Success

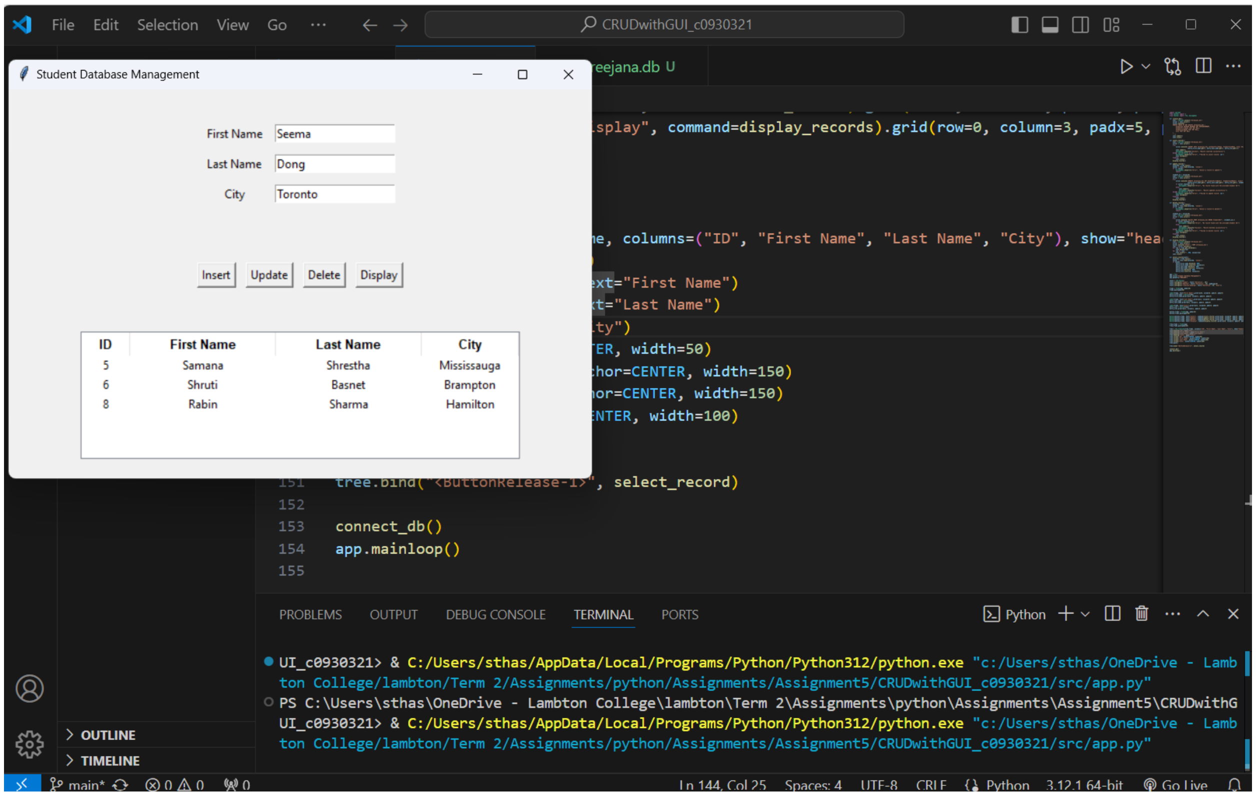
Record deleted successfully!

OK

```
rame, text="Display", command=display_records).grid(row=0, column=3, padx=5,

ame(app)
(pady=20)

view(tree_frame, columns=("ID", "First Name", "Last Name", "City"), show="hea
D", text="ID")
irst Name", text="First Name")
ast Name", text="Last Name")
ity", text="City")
", anchor=CENTER, width=50)
                         width=150)
                         idth=150)
                         100)

tree.bind("<But                ord)

connect_db()
app.mainloop()
```

```
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
PS C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithG
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
```

# List of records after deletion



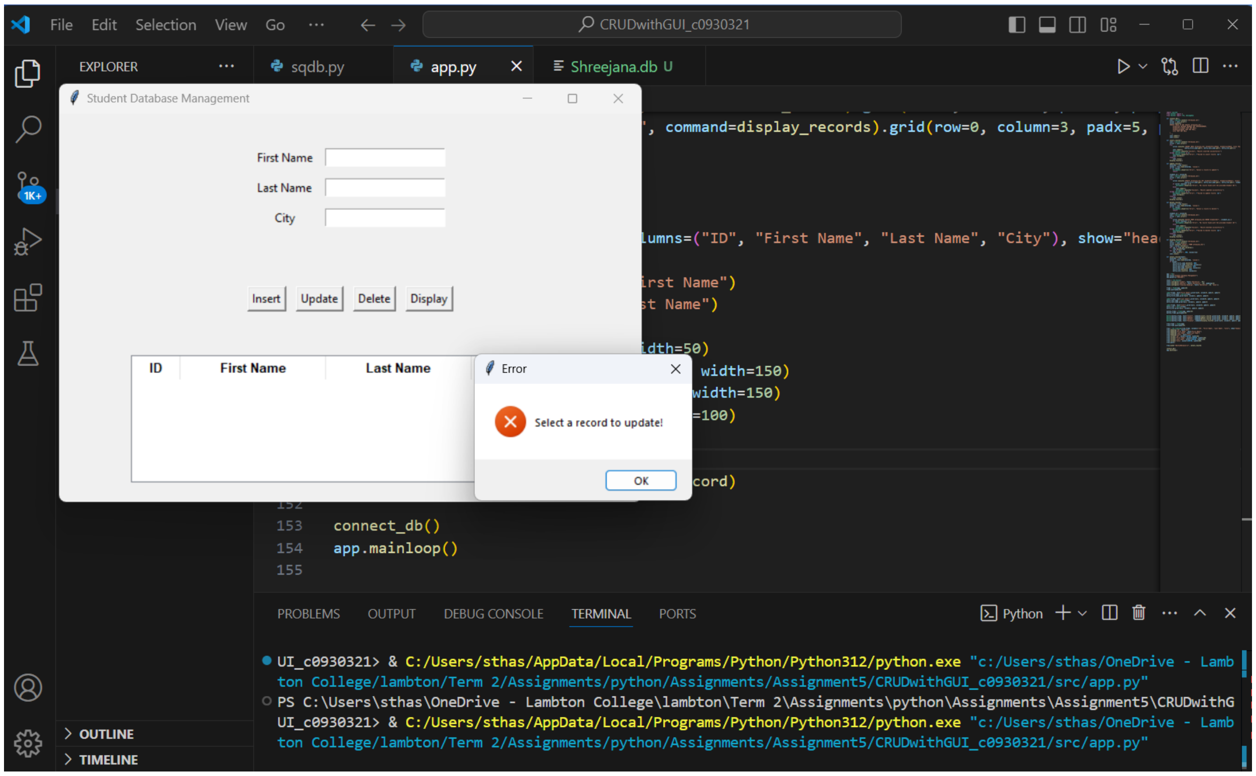
First Name: Seema
Last Name: Dong
City: Toronto

ID	First Name	Last Name	City
5	Samana	Shrestha	Mississauga
6	Shruti	Basnet	Brampton
8	Rabin	Sharma	Hamilton

```
splay", command=display_records).grid(row=0, column=3, padx=5,

e, columns=("ID", "First Name", "Last Name", "City"), show="hea

xt="First Name")
t="Last Name")
ty")
ER, width=50)
hor=CENTER, width=150)
or=CENTER, width=150)
NTER, width=100)

tree.bind("<ButtonRelease-1>", select_record)

connect_db()
app.mainloop()
```

```
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
PS C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithG
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
```

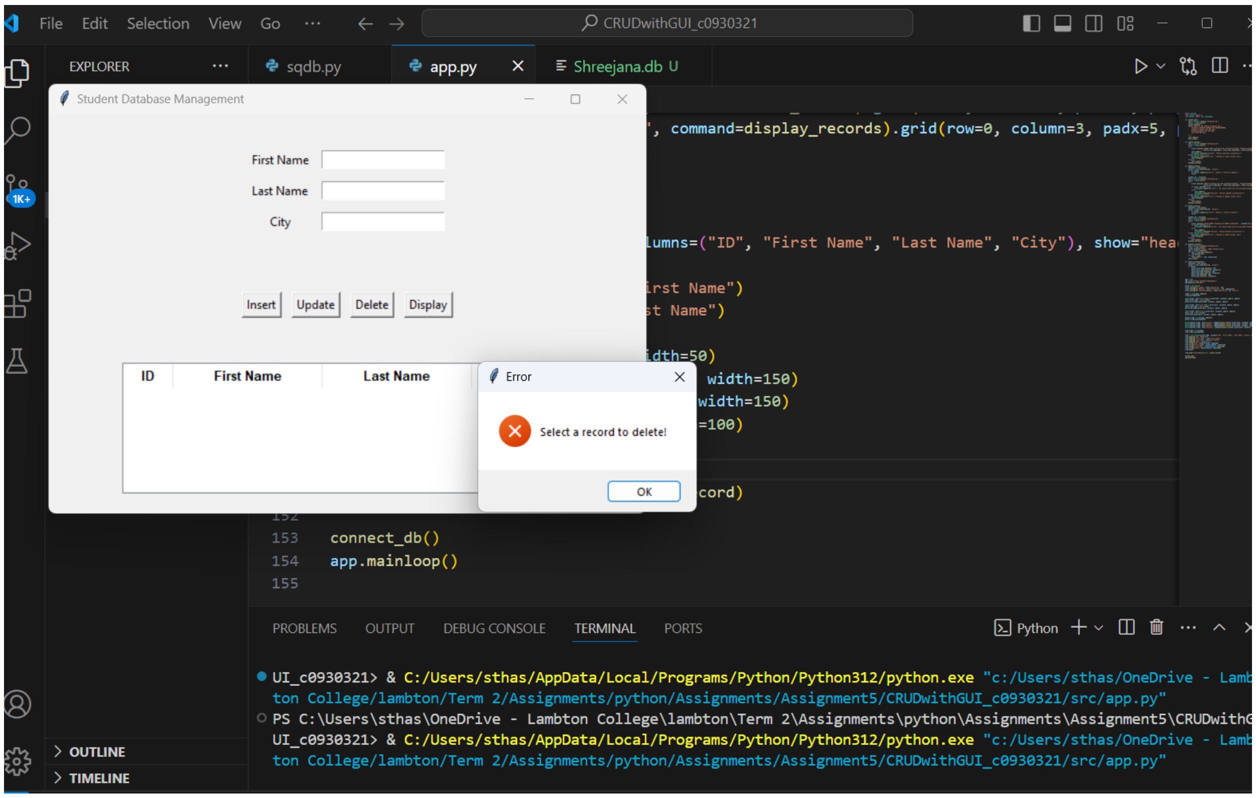
Error handling and showing message while trying to update / delete without selecting any record



Select a record to update!

```
", command=display_records).grid(row=0, column=3, padx=5,
lumns=("ID", "First Name", "Last Name", "City"), show="hea
irst Name")
st Name")
idth=50)
 width=150)
width=150)
=100)
cord)
connect_db()
app.mainloop()
```

```
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
PS C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithG
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
```

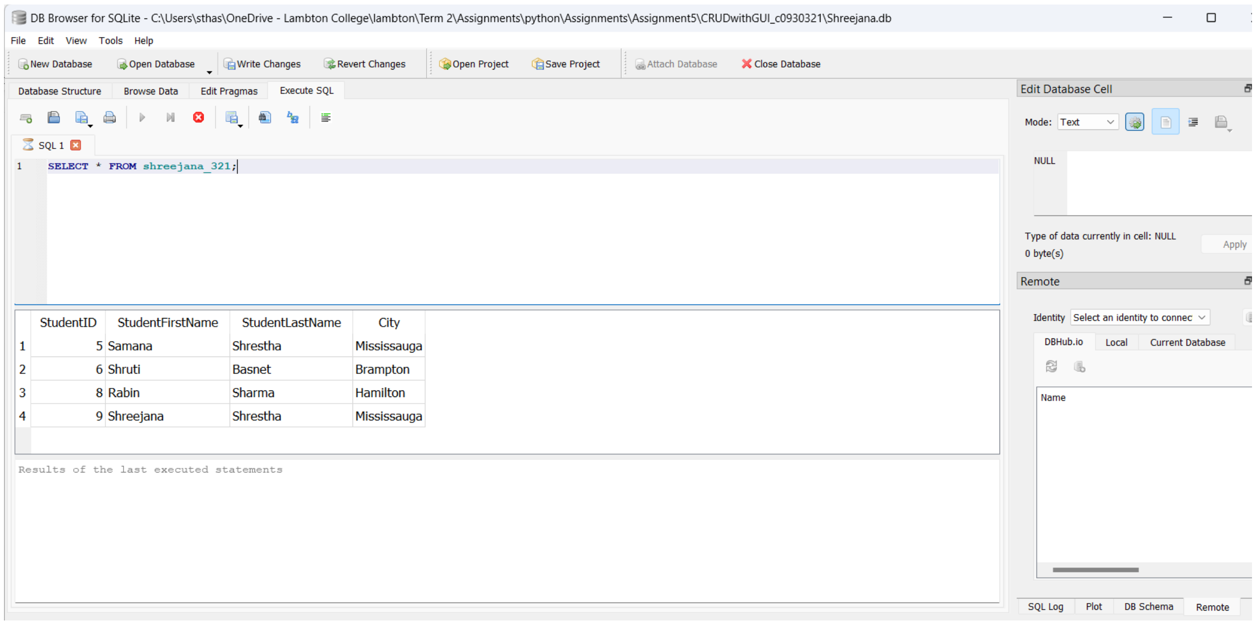


Select a record to delete!

```
", command=display_records).grid(row=0, column=3, padx=5,
lumns=("ID", "First Name", "Last Name", "City"), show="hea
irst Name")
st Name")
idth=50)
 width=150)
width=150)
=100)
cord)
connect_db()
app.mainloop()
```

```
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
PS C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithG
UI_c0930321> & C:/Users/sthas/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sthas/OneDrive - Lamb
ton College/lambton/Term 2/Assignments/python/Assignments/Assignment5/CRUDwithGUI_c0930321/src/app.py"
```

# All the list of records after CRUD from GUI


DB Browser for SQLite - C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\python\Assignments\Assignment5\CRUDwithGUI_c0930321\Shreejana.db

```sql
SELECT * FROM shreejana_321;
```

	StudentID	StudentFirstName	StudentLastName	City
1	5	Samana	Shrestha	Mississauga
2	6	Shruti	Basnet	Brampton
3	8	Rabin	Sharma	Hamilton
4	9	Shreejana	Shrestha	Mississauga

Results of the last executed statements

# Source code with brief description

```python
import sqlite3
from tkinter import *
from tkinter import ttk, messagebox


# database connection and table creation
def connect_db():
    conn = sqlite3.connect('Shreejana.db')
    cursor = conn.cursor()
    cursor.execute('''
    CREATE TABLE IF NOT EXISTS shreejana_321 (
        StudentID INTEGER PRIMARY KEY AUTOINCREMENT,
        StudentFirstName TEXT NOT NULL,
        StudentLastName TEXT NOT NULL,
        City TEXT NOT NULL
    )
    ''')
    conn.commit()
    conn.close()

def insert_record():
    '''
    Takes input from the user (First Name, Last Name, City) and inserts a new record into
the database.
    If the insertion is successful, a success message is displayed.
    If there is an error, an error message is displayed, and the transaction is rolled back
    '''

    conn = sqlite3.connect('Shreejana.db')
    cursor = conn.cursor()
    try:
                cursor.execute('INSERT INTO shreejana_321 (StudentFirstName,
StudentLastName, City) VALUES (?, ?, ?)',
                (entry_first_name.get(), entry_last_name.get(), entry_city.get()))
        conn.commit()
        messagebox.showinfo("Success", "Record inserted successfully!")
    except sqlite3.Error as e:
        messagebox.showerror("Error", f"Failed to insert record: {e}")
        conn.rollback()
    finally:
        conn.close()
    display_records()


def update_record():
    '''
    Updates an existing record in the database. It retrieves the selected record from the Treeview,
    takes new input values, and updates the corresponding record in the database.
    It handles errors such as no selection and database errors, displaying appropriate messages.
    '''

    selected = tree.focus()
    values = tree.item(selected, 'values')
    if not values:
        messagebox.showerror("Error", "Select a record to update!")
        return

    student_id = values[0]
    conn = sqlite3.connect('Shreejana.db')
    cursor = conn.cursor()
    try:
        cursor.execute('UPDATE shreejana_321 SET StudentFirstName=?, StudentLastName=?, City=? WHERE StudentID=?',
                (entry_first_name.get(), entry_last_name.get(), entry_city.get(), student_id))
        if cursor.rowcount == 0:
            messagebox.showerror("Error", "No record found with the provided Student ID!")
        else:
            conn.commit()
        messagebox.showinfo("Success", "Record updated successfully!")
    except sqlite3.Error as e:
        messagebox.showerror("Error", f"Failed to update record: {e}")
        conn.rollback()
    finally:
        conn.close()
    display_records()


def delete_record():
    '''
    Deletes a selected record from the database.
    It retrieves the selected record from the Treeview and deletes it from the database.
     Appropriate success or error messages are displayed based on the operation's
outcome.
    '''


    selected = tree.focus()
    values = tree.item(selected, 'values')
    if not values:
        messagebox.showerror("Error", "Select a record to delete!")
        return


    student_id = values[0]
    conn = sqlite3.connect('Shreejana.db')
    cursor = conn.cursor()
    try:
            cursor.execute('DELETE FROM shreejana_321 WHERE StudentID=?',
(student_id,))
```

```
        if cursor.rowcount == 0:
            messagebox.showerror("Error", "No record found with the provided Student ID!")
        else:
            conn.commit()
            messagebox.showinfo("Success", "Record deleted successfully!")
    except sqlite3.Error as e:
        messagebox.showerror("Error", f"Failed to delete record: {e}")
        conn.rollback()
    finally:
        conn.close()
    display_records()

def display_records():
    '''
    Fetches all records from the database and displays them in the Listbox.
    The list is updated each time a record is inserted, updated, or deleted to
    reflect the current state of the database.
    '''

    conn = sqlite3.connect('Shreejana.db')
    cursor = conn.cursor()
    cursor.execute('SELECT * FROM shreejana_321')
    rows = cursor.fetchall()
    for row in tree.get_children():
        tree.delete(row)
    for row in rows:
        tree.insert("", END, values=row)
    conn.close()


def select_record(event):
    '''
    Handle the selection of a record from the Treeview. When a row is selected,
    it populates the entry fields with the selected record's data for easy updating or deletion.
    '''

    selected = tree.focus()
    values = tree.item(selected, 'values')
    if values:
        entry_first_name.delete(0, END)
        entry_first_name.insert(0, values[1])
        entry_last_name.delete(0, END)
        entry_last_name.insert(0, values[2])
        entry_city.delete(0, END)
        entry_city.insert(0, values[3])

# GUI section logic

app = Tk()
app.title("Student Database Management")
app.geometry("600x400")

style = ttk.Style()
style.configure("TLabel", font=('Helvetica', 10))
style.configure("TButton", font=('Helvetica', 10), padding=10)
style.configure("Treeview.Heading", font=('Helvetica', 10, 'bold'))


frame = Frame(app, pady=10)
frame.pack(pady=20)


#  input for StudentID, First Name, Last Name, and City using label and entry widgets
Label(frame, text="First Name").grid(row=0, column=0, padx=5, pady=5)
entry_first_name = Entry(frame)
entry_first_name.grid(row=0, column=1, padx=5, pady=5)


Label(frame, text="Last Name").grid(row=1, column=0, padx=5, pady=5)
entry_last_name = Entry(frame)
entry_last_name.grid(row=1, column=1, padx=5, pady=5)


Label(frame, text="City").grid(row=2, column=0, padx=5, pady=5)
entry_city = Entry(frame)
entry_city.grid(row=2, column=1, padx=5, pady=5)


button_frame = Frame(app, pady=10)
button_frame.pack(pady=10)


Button(button_frame, text="Insert", command=insert_record).grid(row=0, column=0,
padx=5, pady=5)
Button(button_frame, text="Update", command=update_record).grid(row=0, column=1,
padx=5, pady=5)
Button(button_frame, text="Delete", command=delete_record).grid(row=0, column=2,
padx=5, pady=5)
```

```
Button(button_frame, text="Display", command=display_records).grid(row=0, column=3,
padx=5, pady=5)


# using treeview widget to display the records in tabular format with column headings
tree_frame = Frame(app)
tree_frame.pack(pady=20)


tree = ttk.Treeview(tree_frame, columns=("ID", "First Name", "Last Name", "City"),
show="headings")
tree.heading("ID", text="ID")
tree.heading("First Name", text="First Name")
tree.heading("Last Name", text="Last Name")
tree.heading("City", text="City")
tree.column("ID", anchor=CENTER, width=50)
tree.column("First Name", anchor=CENTER, width=150)
tree.column("Last Name", anchor=CENTER, width=150)
tree.column("City", anchor=CENTER, width=100)
tree.pack()


tree.bind("<ButtonRelease-1>", select_record)


# establish a connection to the database
connect_db()
app.mainloop()
```