# CS 122 Assignment 6

**Due Date: Nov 15**

**Total Points: 100**

**Topics: GUI Programming (Tkinter) and Databases (SQLAlchemy + SQLite)**

## Objective

Build a simple desktop application that uses a Tkinter GUI and a SQLite database (through SQLAlchemy ORM) to manage student records.

## Tasks and Requirements

## Task 1 (50 points) – Database & ORM Model

1. Create a Python file named database.py.
2. Use SQLAlchemy to define a database model for a Student table:
    - id – Integer, Primary Key, auto-increment.
    - name – String, not null.
    - major – String, not null.
3. Use SQLite for storage (e.g., sqlite:///students.db).
4. Ensure the table is created automatically using Base.metadata.create_all().

## Task 2 (50 points) – GUI Application with Tkinter

1. Create a second Python file named app.py.
2. Design a Tkinter window titled “Student Record Manager” that includes:
    - Two Entry fields for Name and Major.
    - Three buttons:
        - Add Student → Inserts a new student into the database.
        - View Students → Displays all records in a ttk.Treeview table.
        - Delete Selected → Deletes the selected student.
3. Display data in columns ID, Name, Major.
4. Use message boxes to show success or error messages when appropriate.

No update/edit feature required. No ID input field needed. ID auto-increments.

**Project Structure (Must Match Exactly)**

HW6_Solution/
├── app.py
├── database.py
└── students.db ← optional: include if you want to show working data

## Running Instructions

1. Install SQLAlchemy:
   *pip install sqlalchemy*
2. Run the application:
   *python app.py*
3. The GUI should open with fields for Name and Major.

## Submission Instructions

Submit a ZIP file named HW6_Solution.zip containing:
app.py
database.py
students.db (optional)

Do NOT include venv/, __pycache__/, or any system files.

## Grading Rubric

| Criteria | Points | Description |
|---|---|---|
| Database Model (ORM setup) | 20 | Student class, table creation, SQLAlchemy configured correctly |
| Database Operations (Add/View/Delete) | 30 | Functions implemented and working without errors |
| GUI Design (Tkinter Layout & Treeview) | 30 | GUI loads properly with labels, entries, buttons, and table |
| Code Organization & Comments | 10 | Clear code structure and inline comments explaining logic |
| Total | 100 | |

**Example Output**

Name: [_____]
Major: [_____]
[ Add Student ] [ View Students ] [ Delete Selected ]

---
| ID | Name | Major |
---
| 1 | Alice | Computer Science |
| 2 | Bob | Data Analytics |

***Note: This assignment will be checked manually; therefore, no test file is provided.***
*Ensure your program runs correctly and the GUI functions as expected.*