# Project Description

## Project Overview

1. This project is a Python application designed to help you manage student case data. You can use it to keep track of details like the date of the case, the student's name, the module code, the module leader, the allegation, and the outcome of the case. You have the flexibility to interact with the application either through a text-based interface (TUI) or a graphical interface (GUI).

## Key Features

### Database Setup

2. The application sets up a SQLite database called `all\_data.db` with a table named `All\_Data` to store all the case details.

3. The table includes columns for serial number, date of case, student name, module code, module leader, allegation, and outcome of the case.

### Data Entry Functions

4. Insert Data: You can input case details either through the TUI or GUI and save them to the database.

5. Save Button Function: In the GUI, you can enter data into entry fields and save it using a button.

### User Interface Options

6. Text User Interface (TUI): You can choose to display, insert, change, or delete data through text-based inputs.

7. Display Data: Fetches and prints all records from the database.

8. Insert Data: Allows continuous data entry until you decide to stop.

9. Change Data: Updates existing records based on the serial number.

10. Delete Data: Deletes records based on the serial number.

11. Graphical User Interface (GUI): Provides a more interactive way to manage data.

12. Data Entry: You can enter case details through text fields and save them using a button.

13. Display Data: Shows all records in a scrollable canvas.

14. Delete Data: Deletes records based on the serial number entered in a text field.

15. Change Data: Updates records based on the serial number entered in a text field.

## GUI Components

16. Entry Fields: For entering case details (date, student name, module code, module leader, allegation, outcome).

17. Buttons: For saving, displaying, updating, and deleting data.

18. Labels: For guiding users on what information to enter.

19. Canvas and Scrollbars: For displaying data in a scrollable format.

## Functionality Flow

20. Initialization: The database and table are set up.

21. User Choice: You choose between TUI and GUI.

22. TUI Operations: Based on your input, the application performs display, insert, change, or delete operations.

23. GUI Operations: You interact with entry fields, buttons, and other GUI components to manage data.

## Libraries Used

24. tkinter: For creating the GUI.

25. sqlite3: For database operations.

26. PIL (Python Imaging Library): For handling images in the GUI.