Assignment0.md

Assignment 0 - Basics of Databases and GUI

Description 1 Description 2

Introduction

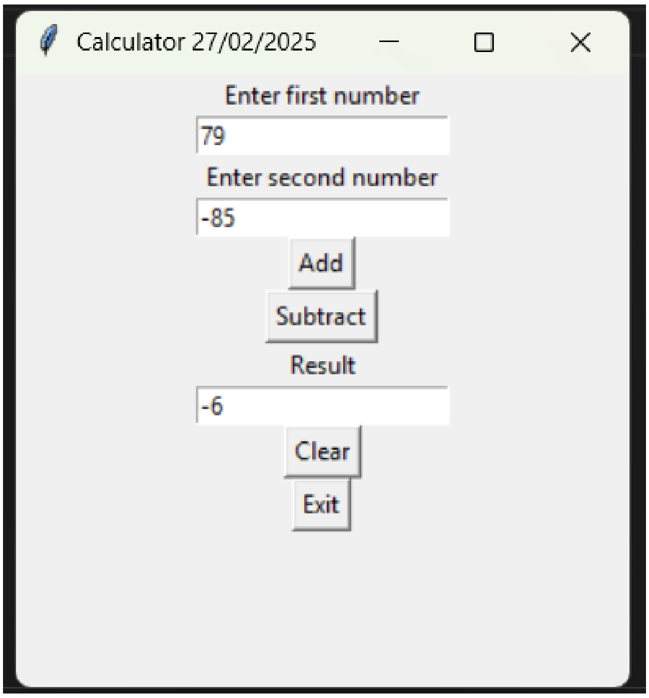
This is the first assignment of the course. The purpose of this assignment is to get familiar with the basics of databases and Graphic User Interface that we will be using throughout the course by the help of simple problems.

We have used two of the most common storage files for this assignment:

1. .txt files
2. .json files

1. Basic Calculator

In this problem, we have built the graphic interface with the help of the Python `tkinter` library. In this interface, we have added entries to take the user input and buttons to ask for addition or subtraction. The result is displayed in the result box, and each operation's result is stored in a .txt file for future reference.



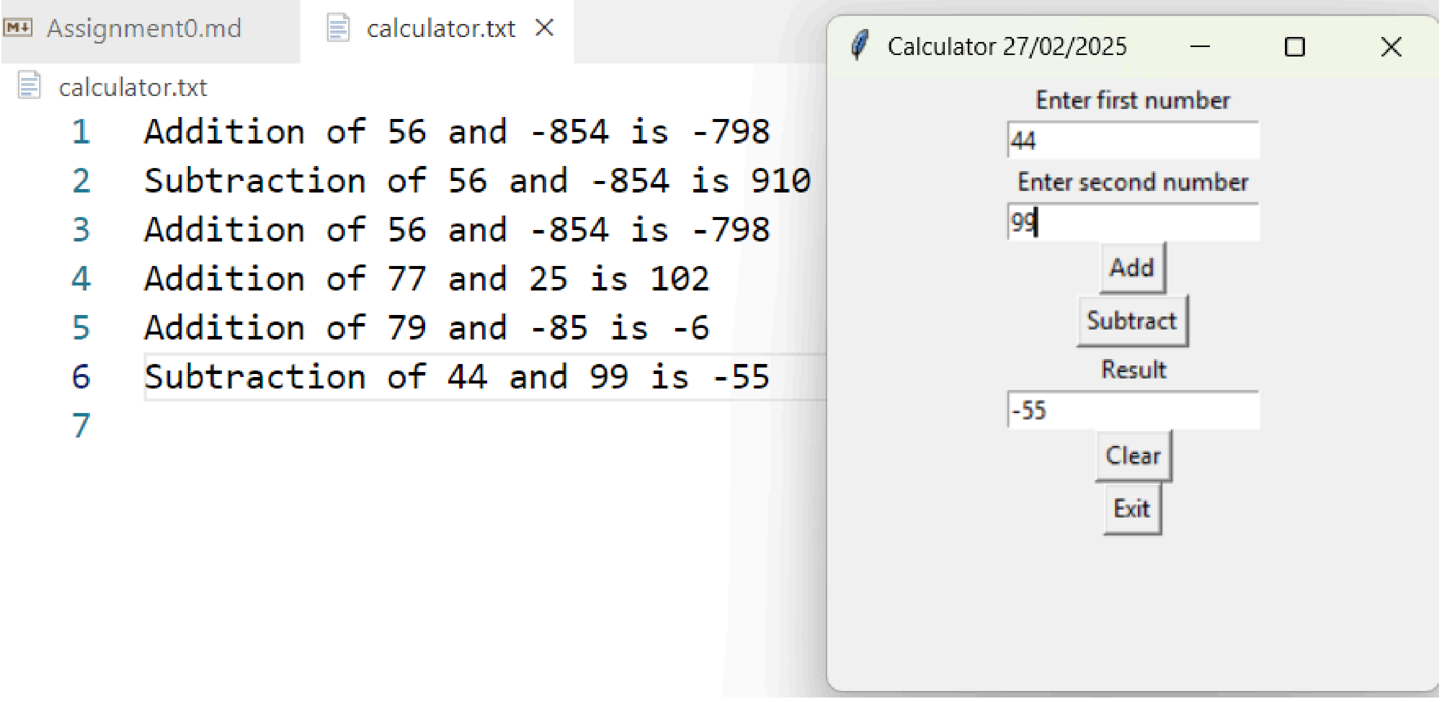
Calculator 27/02/2025

Enter first number: 79

Enter second number: -85

Result: -6

Storage of the results obtain previously in the text file.



```
Addition of 56 and -854 is -798
Subtraction of 56 and -854 is 910
Addition of 56 and -854 is -798
Addition of 77 and 25 is 102
Addition of 79 and -85 is -6
Subtraction of 44 and 99 is -55
```

Enter first number: 44

Enter second number: 99

Result: -55

2. Student Database Management

Student Management System Using Tkinter and JSON

This Python program is a Student Management System built using the `tkinter` library for the graphical user interface (GUI) and a JSON file for data storage. The system allows users to perform the following operations:

- Add Student
- Search Student
- Edit Student
- Delete Student
- Display All Students

The JSON file plays a crucial role in this system as it stores all the data, including the university name, department details, and student records. Below is a detailed explanation of each functionality and the importance of the JSON file.

Importance of the JSON File

The JSON file (`records.json`) acts as the database for this application. It stores the following data:

- **University Name:** The name of the university.
- **Departments:** A list of departments with their names and codes.
- **Students:** A list of student records, each containing a roll number, name, and department code.

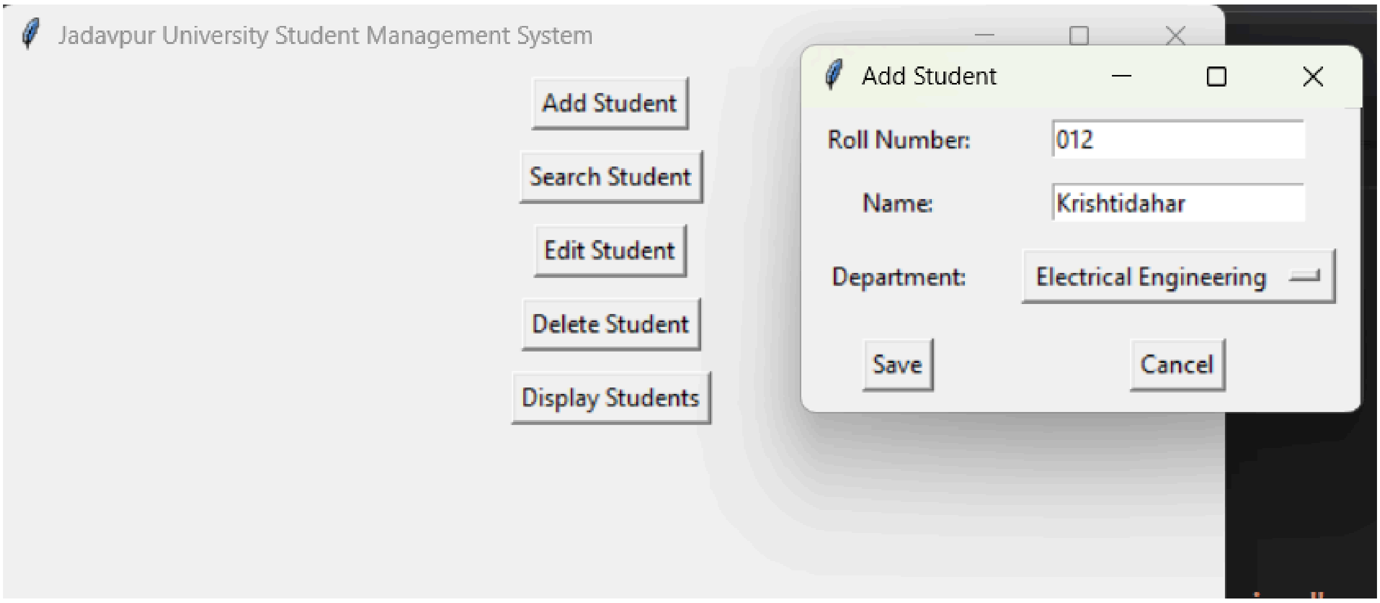
The JSON file is loaded at the start of the program, and any changes made to the data (e.g., adding, editing, or deleting students) are saved back to the file. This ensures that the data persists even after the program is closed.

Functionality Breakdown

1. Add Student

This feature allows the user to add a new student record. The roll number must be unique, and the user selects a department from a dropdown list.

Add Student Screenshot



Roll Number: 012

Name: Krishtidahar

Department: Electrical Engineering

Steps:

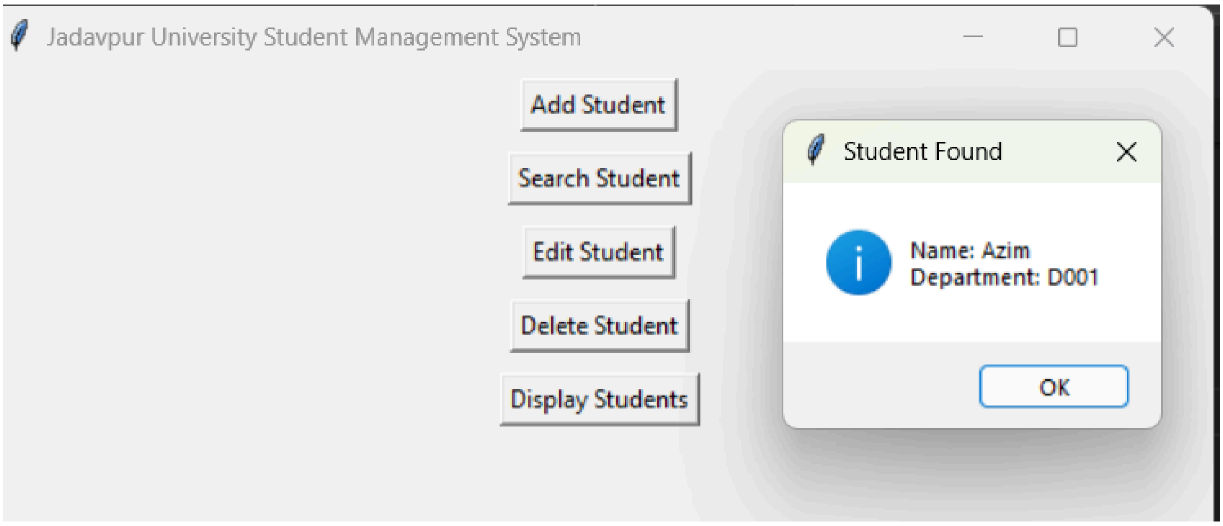
1. Enter the roll number, name, and select a department.
2. Click Save to add the student or Cancel to discard the input.

JSON Role: The new student is appended to the students list in the JSON file.

2. Search Student

This feature allows the user to search for a student by their roll number. If the student exists, their details are displayed; otherwise, a “Not Found” message is shown.

Search Student Screenshot



Name: Azim
Department: D001

Steps:

1. Enter the roll number in the dialog box.
2. If the student exists, their name and department are displayed.

JSON Role: The students list in the JSON file is searched for the provided roll number.

3. Edit Student

This feature allows the user to edit an existing student’s details (name and department). The roll number cannot be edited.

Edit Student Screenshot

Steps:

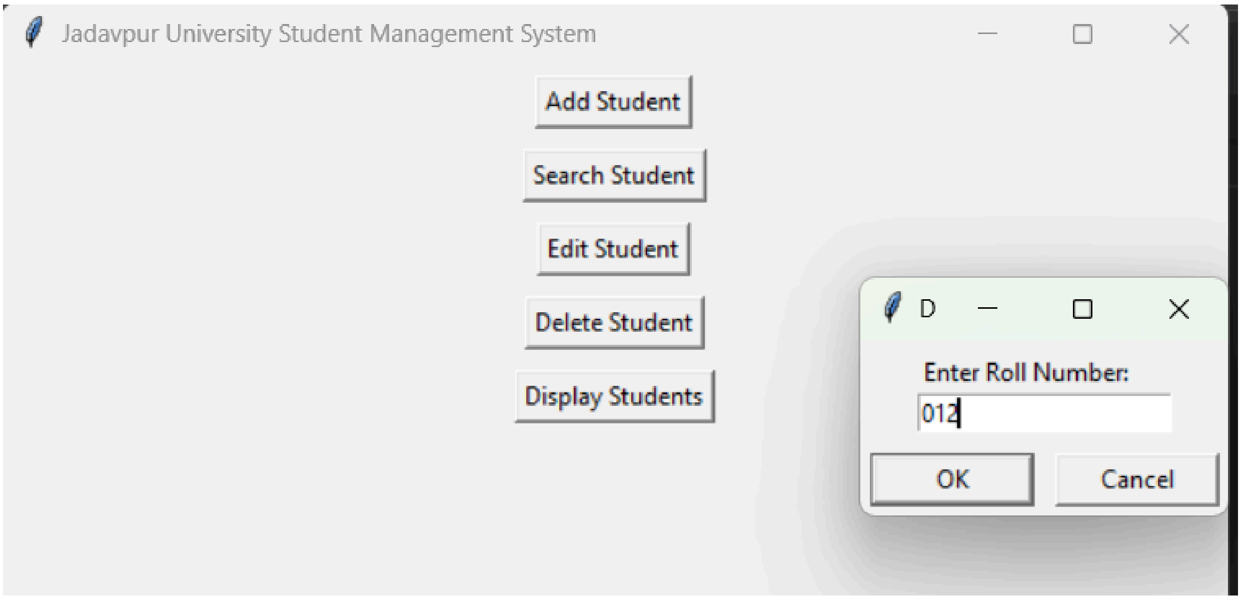
1. Enter the roll number of the student to edit.
2. Modify the name or department.
3. Click Save to update the record or Cancel to discard changes.

JSON Role: The corresponding student record in the students list is updated in the JSON file.

4. Delete Student

This feature allows the user to delete a student record by providing the roll number.

Delete Student Screenshot



Jadavpur University Student Management System

Enter Roll Number: 012

Steps:

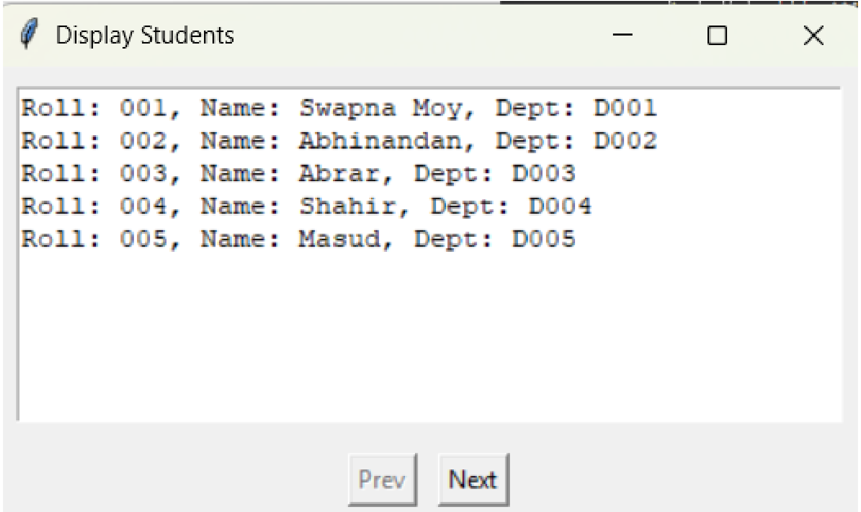
1. Enter the roll number of the student to delete.
2. If the student exists, they are removed from the records.

JSON Role: The corresponding student record is removed from the students list in the JSON file.

5. Display All Students

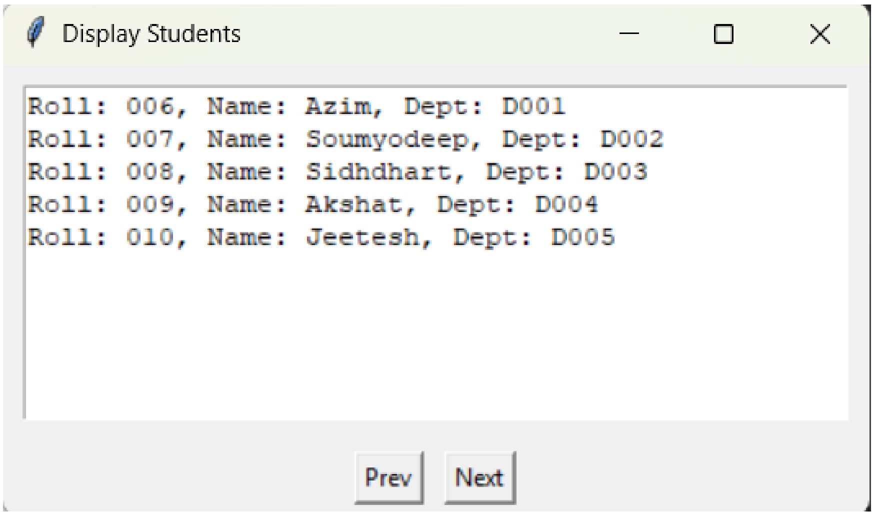
This feature displays all student records in sets of five. The user can navigate through the records using Prev and Next buttons.

Display Students Screenshot



```
Roll: 001, Name: Swapna Moy, Dept: D001
Roll: 002, Name: Abhinandan, Dept: D002
Roll: 003, Name: Abrar, Dept: D003
Roll: 004, Name: Shahir, Dept: D004
Roll: 005, Name: Masud, Dept: D005
```

First 5 students



```
Roll: 006, Name: Azim, Dept: D001
Roll: 007, Name: Soumyodeep, Dept: D002
Roll: 008, Name: Sidhdhart, Dept: D003
Roll: 009, Name: Akshat, Dept: D004
Roll: 010, Name: Jeetesh, Dept: D005
```

Next 5 students **Steps:**

1. Open the “Display Students” window.
2. Use the Prev and Next buttons to navigate through the records.

JSON Role: The students list in the JSON file is used to populate the display.

How the JSON File is Used

Loading Data:

- At the start of the program, the JSON file is loaded into memory.
- The data is stored in variables (`university_name`, `departments`, `students`).

Saving Data:

- Whenever a change is made (e.g., adding, editing, or deleting a student), the updated data is saved back to the JSON file using the `save_data()` function.

Data Persistence:

- The JSON file ensures that all changes are saved permanently, allowing the program to retain data even after it is closed and reopened.

Conclusion

The JSON file is the backbone of this application, providing a simple and effective way to store and manage data. The `tkinter` GUI makes it easy for users to interact with the system, while the JSON file ensures that all data is persisted and accessible across sessions.

By combining these two technologies, the Student Management System provides a user-friendly and efficient way to manage student records.