PROJECT PROPOSAL

Simple Console-Based Student Management System (SMS)

Project Aspect	*Detail*
Proposed By:	*SYEDA FAIZA ASLAM*
Date:	*December 26, 2025*
Platform:	*Command-Line Interface (CLI)*
Language:	*Python 3.x (Standard Library only: json)*

1.1 Executive Summary

This project proposes the development of a straightforward, user-friendly Student Management System (SMS) using Python 3.x. The primary goal is to automate the basic administrative tasks of managing student records in a simple and accessible manner. The system will operate via a command-line interface, making it universally accessible without the need for complex graphical environments or external databases. Data will be stored permanently using a simple `JSON` file.

1.2 Problem Statement & Objectives

Problem: Manual methods of tracking student information (e.g., spreadsheets or paper records) are inefficient, prone to data entry errors, and time-consuming to search and update.

Objectives:

- To create a system that allows users to add, view, search, update, and delete student records (CRUD operations).
- To ensure data persistence so records are not lost upon program termination.
- To design an intuitive menu-driven interface that requires minimal user training.
- To keep the codebase simple and modular for easy future maintenance and learning.

1.3 Methodology

The project will be developed using a modular programming approach, where each major feature (add, view, search, delete) is implemented as a separate function.

1. **Data Structure:** A Python list of dictionaries will be used to manage data in memory during runtime.
2. **Persistence:** The standard `json` library will be used for file handling, serializing the list of dictionaries to a `students.json` file for permanent storage.
3. **User Interface:** A continuous `while` loop will present a clear, numbered menu of options, accepting user input to navigate the system.

1.4 Timeline

The system is designed for rapid development.

- **Week 1:** Setup environment, define data structure, implement "Add" and "View" functions.
- **Week 2:** Implement "Search," "Update," "Delete" functions, add JSON file handling.
- **Week 3:** Testing, refinement of error handling, and final documentation (Project Report).