PROJECT REPORT

Simple Console-Based Student Management System (SMS)

1. Introduction

This report documents the design, implementation, and functionality of the Student Management System (SMS), a simple Python application designed to manage student records via a command-line interface. The project was developed with a focus on simplicity and ease of use, leveraging only the Python standard library.

2. System Architecture

The system utilizes a simple, file-based architecture.

- **Frontend:** A console application handles all user interactions (input/output).
- **Backend Logic:** Python functions manage the data flow and operations.
- **Data Storage:** A single JSON file (`students.json`) acts as the persistent "database."

This structure is highly portable and runs on any system with Python installed (Windows, macOS, Linux).

3. Features Implemented

The system provides the following core functionalities:

- **Add New Student:** Allows a user to input a unique student ID, name, and grade level.
- **View All Students:** Displays a formatted list of all currently saved student records.
- **Search for a Student:** Locates and displays a specific student's details using their unique ID.

- **Delete a Student:** Removes a student's record after a user confirmation prompt.
- **Exit Program:** Terminates the application safely, ensuring all data is saved.

4. Technical Details

- **Programming Language:** Python 3.x
- **Libraries Used:**
    - `json`: Used for serializing Python dictionaries into JSON format for disk storage.
    - `os`: Used for basic operating system interaction (e.g., checking file existence, clearing the screen).
- **Data Structure:** Records are managed in a dictionary (`students_data`), where the student ID is the key for fast lookups.

5. User Guide (How to Run)

1. Ensure you have Python installed on your computer.
2. Save the code provided as a file named `student_management_system.py`.
3. Open your terminal or command prompt (or use the built-in terminal in VS Code).
4. Navigate to the directory where you saved the file.
5. Run the application using the following command:

   bash

   ```bash
   python student_management_system.py
   ```

   Use code with caution.
6. Follow the on-screen numbered menu prompts to interact with the system.

6. Conclusion and Future Enhancements

The Simple Student Management System successfully meets its objective of providing a basic, easy-to-understand tool for managing student records. The use of a command-line interface and standard Python libraries makes it highly accessible for beginners.

Potential future enhancements could include:

- Adding more student data fields (e.g., contact info, attendance).
- Implementing data validation for input types (e.g., ensuring an ID is a number).
- Migrating to a simple GUI using `Tkinter` or a database using `sqlite3`.