**Project Report: Student Management System in Python**

**Project Title**

**Project 9: Student Management System**

**Abstract**

The Student Management System is a basic Python application designed to help educational institutions manage student records efficiently. The system allows users to perform essential tasks such as adding, displaying, searching, updating, and deleting student information. This beginner-friendly project demonstrates how to build a simple menu-driven application using Python's core programming features such as functions, file handling, and conditional logic.

**Objective**

* To store and manage student information.
* To enable operations like adding, updating, deleting, and viewing records.
* To simplify record management for schools or training centers.
* To practice Python programming using real-life use cases.

**Tools & Technologies Used**

| **Technology** | **Purpose** |
| --- | --- |
| Python 3.x | Core programming language |
| File Handling | Store and retrieve student records |
| Command-Line Interface | Simple user interaction |

**System Requirements**

* Python 3.x installed
* No external libraries required

**Methodology**

1. **Input**: User chooses an operation (Add, View, Search, Update, Delete).
2. **Processing**: Perform the selected operation using file-based storage.
3. **Output**: Display success messages or data based on operation type.

**Code Implementation (Simple Menu-Driven Console App)**

import os

def add\_student():

with open("students.txt", "a") as file:

name = input("Enter student name: ")

roll = input("Enter roll number: ")

grade = input("Enter grade: ")

file.write(f"{roll},{name},{grade}\n")

print("Student added successfully!\n")

def view\_students():

print("\n--- Student Records ---")

try:

with open("students.txt", "r") as file:

for line in file:

roll, name, grade = line.strip().split(",")

print(f"Roll No: {roll}, Name: {name}, Grade: {grade}")

except FileNotFoundError:

print("No records found.\n")

def search\_student():

roll\_search = input("Enter roll number to search: ")

found = False

try:

with open("students.txt", "r") as file:

for line in file:

roll, name, grade = line.strip().split(",")

if roll == roll\_search:

print(f"Found: Roll No: {roll}, Name: {name}, Grade: {grade}")

found = True

break

if not found:

print("Student not found.\n")

except FileNotFoundError:

print("No records found.\n")

def delete\_student():

roll\_delete = input("Enter roll number to delete: ")

updated\_lines = []

deleted = False

try:

with open("students.txt", "r") as file:

for line in file:

roll, name, grade = line.strip().split(",")

if roll != roll\_delete:

updated\_lines.append(line)

else:

deleted = True

with open("students.txt", "w") as file:

file.writelines(updated\_lines)

if deleted:

print("Student deleted successfully.\n")

else:

print("Student not found.\n")

except FileNotFoundError:

print("No records found.\n")

def update\_student():

roll\_update = input("Enter roll number to update: ")

updated\_lines = []

updated = False

try:

with open("students.txt", "r") as file:

for line in file:

roll, name, grade = line.strip().split(",")

if roll == roll\_update:

name = input("Enter new name: ")

grade = input("Enter new grade: ")

updated\_lines.append(f"{roll},{name},{grade}\n")

updated = True

else:

updated\_lines.append(line)

with open("students.txt", "w") as file:

file.writelines(updated\_lines)

if updated:

print("Student updated successfully.\n")

else:

print("Student not found.\n")

except FileNotFoundError:

print("No records found.\n")

def main():

while True:

print("\n=== Student Management System ===")

print("1. Add Student")

print("2. View All Students")

print("3. Search Student")

print("4. Update Student")

print("5. Delete Student")

print("6. Exit")

choice = input("Enter your choice: ")

if choice == "1":

add\_student()

elif choice == "2":

view\_students()

elif choice == "3":

search\_student()

elif choice == "4":

update\_student()

elif choice == "5":

delete\_student()

elif choice == "6":

print("Exiting...")

break

else:

print("Invalid choice. Try again.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Sample Output**

=== Student Management System ===

1. Add Student

2. View All Students

3. Search Student

4. Update Student

5. Delete Student

6. Exit

Enter your choice: 1

Enter student name: Alice

Enter roll number: 101

Enter grade: A

Student added successfully!

**Advantages**

* Simple and beginner-friendly
* Covers key CRUD operations
* No database required
* Runs in any Python environment

**Limitations**

* Data stored in plain text (not secure)
* Not suitable for large-scale systems
* No GUI or web interface

**Future Enhancements**

* Add GUI using Tkinter or PyQt
* Store data in SQLite or MySQL instead of text files
* Add authentication for admin access
* Add attendance and marks management modules

**References**

1. [Python File Handling Docs](https://docs.python.org/3/tutorial/inputoutput.html)
2. [GeeksforGeeks – Student Management System in Python](https://www.geeksforgeeks.org/)
3. [Python.org – Official Documentation](https://www.python.org/doc/)
4. [W3Schools – Python File Handling](https://www.w3schools.com/python/python_file_handling.asp)
5. [TutorialsPoint – Python Projects](https://www.tutorialspoint.com/python/index.htm)

**Code execution with pdf link:**

<https://www.online-python.com/503nkC94Gb>