Student Management System (Python)

Suraj Gupta

Python Student Management System (Beginner Level)

This project is designed for beginners to understand the basic concepts of file handling and user interaction in Python.
It allows users to add, view, search, and remove student records, storing data in a text file for persistence.

```
import os

# File to store student data
FILE_NAME = "students.txt"

# Function to load student data from file
def load_students():
    if not os.path.exists(FILE_NAME):  # Check if file exists
        return []  # Return empty list if file is not found
    with open(FILE_NAME, "r") as file:  # Open file in read mode
         return [line.strip() for line in file.readlines()]  # Read and return students list

# Function to save student data to file
def save_students(students):
    with open(FILE_NAME, "w") as file:  # Open file in write mode
        for student in students:
            file.write(student + "\n")  # Write each student name in a new line

# Load student records when the program starts
students = load_students()
```

Student Management System (Python)

```
# Function to display student list
def view_list():
    if not students:
        print("\nNo students found!")  # Message if list is empty
    else:
        print("\nStudent List:")
        for student in students:  # Loop through student names and print them
            print(student)

# Function to add a new student
def add_data():
    name = input("\nEnter the name: ").strip()
    if name in students:
        print("Student already exists!")  # Check for duplicate entry
    else:
        students.append(name)  # Add student to list
        save_students(students)  # Save updated list to file
        print("Name added successfully!")

# Function to remove a student
def remove_data():
    name = input("\nEnter the name to remove: ").strip()
    if name in students:
        students.remove(name)  # Remove student from list
        save_students(students)  # Save updated list to file
        print("Record deleted successfully!")
    else:
        print("Record not found!")  # Message if name is not found

# Function to search for a student
def search_data():
```

Student Management System (Python)

```
    name = input("\nEnter the name to search: ").strip()
    if name in students:
        print("Name found!")  # Display if student exists
    else:
        print("Record not found!")  # Message if student is not found

# Main program loop
while True:
    print("\n" + "-" * 50)
    print("\nPlease choose an option:")
    print("1. View student list")
    print("2. Add new student")
    print("3. Remove student")
    print("4. Search student")
    print("5. Exit")

    try:
        choice = int(input("\nEnter your choice: "))
        if choice == 1:
            view_list()
        elif choice == 2:
            add_data()
        elif choice == 3:
            remove_data()
        elif choice == 4:
            search_data()
        elif choice == 5:
            print("Exiting the program. Goodbye!")
            break
        else:
            print("Invalid input! Please enter a valid option.")
```

Student Management System (Python)

```
except ValueError:
    print("Please enter a number between 1 and 5.")


cont = input("\nDo you want to continue? (y/n): ").lower()
if cont != 'y':
    break
```

Student Management System (Python)

Suraj Gupta

Sample Output

```
-------------------------------------------------

Please choose an option:

1. View student list

2. Add new student

3. Remove student

4. Search student

5. Exit


Enter your choice: 2

Enter the name: Rohan

Name added successfully!


Do you want to continue? (y/n): y


-------------------------------------------------

Please choose an option:

1. View student list

2. Add new student

3. Remove student

4. Search student

5. Exit


Enter your choice: 1

Student List:

Rohan
```

Student Management System (Python)

```
Do you want to continue? (y/n): y


------------------------------------------------
Please choose an option:
1. View student list
2. Add new student
3. Remove student
4. Search student
5. Exit

Enter your choice: 4
Enter the name to search: Rohan
Name found!

Do you want to continue? (y/n): n
```