Project : Student Details Management System Using Pandas

1. Introduction

This project is a simple command-line Student Details Management System built using Python and the Pandas library. It allows users to view, add, update, delete, search, and save student records efficiently. The data is stored in a CSV file and loaded into a Pandas DataFrame for easy manipulation.

The student records contain the following fields:

Student ID

Name

Age

Email

CGPA

Department

Graduation Year

The system provides an interactive menu to perform various operations on the student data.

2. Objectives

To develop a basic system that can manage student records stored in a CSV file.

To use Pandas for efficient data handling and manipulation.

To provide an interactive command-line interface for ease of use.

To ensure data persistence by saving changes back to the CSV file.

3. Features and Functionalities

View Students: Display the full list of student records.

Add Student: Input new student details and append them to the data.

Update Student: Modify any field of a student's record using their Student ID.

Delete Student: Remove a student record based on Student ID.

Search Student: Retrieve details of a specific student by Student ID.

Save Data: Save all changes made to the student data back to the CSV file.

Exit: Close the application.

4.Program Code and Workflow

```
import pandas as pd

# Loading Data: The CSV file is loaded into a DataFrame at the start.

df = pd.read_csv("studentp.csv")

# Viewing Students: Prints the entire DataFrame.

def view_students():
    print(df)
```

- Adding Students: Takes user input, creates a new dictionary, and appends it to the DataFrame.

```
def add_student():
    global df
    sid = input("Enter Student ID: ")
    name = input("Enter Name: ")
    age = int(input("Enter Age: "))
    email = input("Enter Email: ")
    cgpa = float(input("Enter CGPA: "))
    dept = input("Enter Department: ")
    grad_year = int(input("Enter Graduation Year: "))

    new_student = {
        "Student ID": sid, "Name": name, "Age": age, "Email": email,
        "CGPA": cgpa, "Department": dept, "Graduation Year": grad_year
    }

    df = pd.concat([df, pd.DataFrame([new_student])], ignore_index=True)
    print("successfully add the data")
```

- Updating Students: Locates student by ID and updates specified field after validating input

```python
def update_student():
    global df

    sid = input("Enter Student ID to update: ")

    # Check if ID exists
    if sid in df["Student ID"].astype(str).values:
        print("Fields available to update:", list(df.columns))
        field = input("Enter field to update (e.g., Name, Age, CGPA): ")

        # Check if field is valid
        if field in df.columns:
            new_value = input("Enter new value: ")

            # Type conversion based on the field
            if field in ["Age", "Graduation Year"]:
                try:
                    new_value = int(new_value)
                except ValueError:
                    print("Invalid input: please enter an integer value.")
                    return
            elif field == "CGPA":
                try:
                    new_value = float(new_value)
                except ValueError:
                    print("Invalid input: please enter a float value.")
                    return

            # Update the DataFrame
            df.loc[df["Student ID"].astype(str) == sid, field] = new_value

            # Save to CSV (optional for Jupyter; remove or comment if not needed)
            df.to_csv(r'C:\Users\PRITHVIRAJ\Desktop\studentP.csv', index=False)

            print(" Student information updated.")
        else:
            print(" Invalid field name.")
    else:
        print(" Student ID not found.")
```

- Deleting Students: Filters out the student record by ID.

```python
def delete_student():
    global df
    sid = int(input("Enter ID to delete: "))
    df = df[df["Student ID"] != sid]
    print("deleted the data")
    print(df)
```

- Searching Students: Filters and prints the record matching the ID.

```
def search_student():
    global df
    student_id = input("Enter Student ID: ").strip()

    # Match types (if SIDs are stored as strings)
    df["Student ID"] = df["Student ID"].astype(str)

    # Search for the student
    student_data = df[df["Student ID"] == student_id]
    if not student_data.empty:
        print("\n Student Details:\n")
        print(student_data.to_string(index=False))
    else:
        print(" Student ID not found.")
```

- Saving Data: Writes the current DataFrame back to the CSV file.

```
def save_data():
    df.to_csv(r'C:\Users\PRITHVIRAJ\Desktop\studentP.csv', index=False)
    print("Data saved.")
```

- Menu: Runs an interactive loop for user to select options.

```
def menu():
    while True:
        print("\n1. View\n2. Add\n3. Update\n4. Delete\n5. Search\n6. Save\n7. Exit")
        choice = input("Choose an option: ")

        if choice == "1":
            view_students()
        elif choice == "2":
            add_student()
        elif choice == "3":
            update_student()
        elif choice == "4":
            delete_student()
        elif choice == "5":
            search_student()
        elif choice == "6":
            save_data()
        elif choice == "7":
            break
        else:
            print("Invalid choice!")

if __name__ == "__main__":
    menu()
```

- Result:

```
1. View
2. Add
3. Update
4. Delete
5. Search
6. Save
7. Exit
Choose an option:  1
```

	Student ID	Name	Age	Email	CGPA	Department
0	111	Karthika	23	Karthika@gmail.com	3.330	Maths
1	112	Arun	24	Arun@gmail.com	2.500	Physics
2	113	Ann	25	Ann@gmail.com	3.000	Chemistry
3	114	Melbins	26	Melbins@gmail.com	3.990	Data Science
4	115	Keerthi	27	Keerthi@gmail.com	1.200	Chemistry
5	116	Giya	28	Giya@gmail.com	2.200	Maths
6	117	Anju	29	Anju@gmail.com	3.200	Chemistry
7	118	Athul	22	Athul@gmail.com	4.200	Data Science
8	119	Ciya	31	Ciya@gmail.com	4.500	Physics
9	120	Ammu	32	Ammu@gmail.com	3.700	Physics
10	121	Ponnu	33	Ponnu@gmail.com	2.600	Physics
11	122	VK	34	VK@gmail.com	3.990	Maths
12	123	Geethu	21	Geethu@gmail.com	3.800	Chemistry
13	124	Abitta	22	Abitta@gmail.com	4.600	Chemistry
14	125	Mary	19	Mary@gmail.com	4.300	Maths
15	126	Nithin	20	Nithin@gmail.com	3.800	Physics
16	127	Thomas	21	Thomas@gmail.com	3.400	Data Science
17	128	Alberto	22	Alberto@gmail.com	4.700	Maths
18	129	Nimmy	23	Nimmy@gmail.com	4.440	Maths
19	130	Melbin	24	Melbin@gmail.com	4.520	Physics
20	131	Arjun	25	Arjun@gmail.com	4.880	Physics
21	132	Deepak	26	Deepak@gmail.com	2.990	Chemistry
22	133	Arun	22	Arun@gmail.com	3.400	Data Science
23	134	Vishnu	28	Vishnu@gmail.com	3.610	Data Science
24	135	Yadli	29	Yadli@gmail.com	3.770	Maths
25	136	Babu	30	Babu@gmail.com	4.210	Physics
26	137	Dona	25	Dona@gmail.com	4.560	Chemistry
27	138	Dinta	32	Dinta@gmail.com	3.998	Data Science
28	139	Rahul	21	Rahul@gmail.com	3.330	Physics

	Graduation Year
0	2021
1	2022
2	2021
3	2023
4	2023
5	2022
6	2023
7	2023
8	2023
9	2021
10	2021
11	2021
12	2021
13	2023
14	2021
15	2021
16	2022
17	2022
18	2023
19	2022
20	2022
21	2021
22	2022
23	2021
24	2022
25	2021
26	2022
27	2021
28	2023

```
1. View
2. Add
3. Update
4. Delete
5. Search
6. Save
7. Exit
Choose an option:  2
Enter Student ID:  140
Enter Name:  preethi
Enter Age:  23
Enter Email:  preethi@gmail.com
Enter CGPA:  4.52
Enter Department:  chemistry
Enter Graduation Year:  2023
successfully add the data
```

```
1. View
2. Add
3. Update
4. Delete
5. Search
6. Save
7. Exit
Choose an option:  3
Enter Student ID to update:  111
Fields available to update: ['Student ID', 'Name', 'Age', 'Email', 'CGPA', 'Department', 'Graduation Year']
Enter field to update (e.g., Name, Age, CGPA):  Name
Enter new value:  Ancy
 Student information updated.
```

```
1. View
2. Add
3. Update
4. Delete
5. Search
6. Save
7. Exit
Choose an option:  4
Enter ID to delete:  139
deleted the data
```

Student ID	Name	Age	Email	CGPA	Department	\
0	111	Ancy	23	Karthika@gmail.com	3.330	Maths
1	112	Arun	24	Arun@gmail.com	2.500	Physics
2	113	Ann	25	Ann@gmail.com	3.000	Chemistry
3	114	Melbins	26	Melbins@gmail.com	3.990	Data Science
4	115	Keerthi	27	Keerthi@gmail.com	1.200	Chemistry
5	116	Giya	28	Giya@gmail.com	2.200	Maths
6	117	Anju	29	Anju@gmail.com	3.200	Chemistry
7	118	Athul	22	Athul@gmail.com	4.200	Data Science
8	119	Ciya	31	Ciya@gmail.com	4.500	Physics
9	120	Ammu	32	Ammu@gmail.com	3.700	Physics
10	121	Ponnu	33	Ponnu@gmail.com	2.600	Physics
11	122	VK	34	VK@gmail.com	3.990	Maths
12	123	Geethu	21	Geethu@gmail.com	3.800	Chemistry
13	124	Abitta	22	Abitta@gmail.com	4.600	Chemistry
14	125	Mary	19	Mary@gmail.com	4.300	Maths
15	126	Nithin	20	Nithin@gmail.com	3.800	Physics
16	127	Thomas	21	Thomas@gmail.com	3.400	Data Science
17	128	Alberto	22	Alberto@gmail.com	4.700	Maths
18	129	Nimmy	23	Nimmy@gmail.com	4.440	Maths
19	130	Melbin	24	Melbin@gmail.com	4.520	Physics
20	131	Arjun	25	Arjun@gmail.com	4.880	Physics
21	132	Deepak	26	Deepak@gmail.com	2.990	Chemistry
22	133	Arun	22	Arun@gmail.com	3.400	Data Science
23	134	Vishnu	28	Vishnu@gmail.com	3.610	Data Science
24	135	Yadli	29	Yadli@gmail.com	3.770	Maths
25	136	Babu	30	Babu@gmail.com	4.210	Physics
26	137	Dona	25	Dona@gmail.com	4.560	Chemistry
27	138	Dinta	32	Dinta@gmail.com	3.998	Data Science
29	140	preethi	23	preethi@gmail.com	4.520	chemistry

```
    Graduation Year
0               2021
1               2022
2               2021
3               2023
4               2023
5               2022
6               2023
7               2023
8               2023
9               2021
10              2021
11              2021
12              2021
13              2023
14              2021
15              2021
16              2022
17              2022
18              2023
19              2022
20              2022
21              2021
22              2022
23              2021
24              2022
25              2021
26              2022
27              2021
29              2023


1. View
2. Add
3. Update
4. Delete
5. Search
6. Save
7. Exit
Choose an option:  5
Enter Student ID:  111

 Student Details:

Student ID Name  Age              Email  CGPA Department  Graduation Year
       111 Ancy   23 Karthika@gmail.com  3.33      Maths             2021
```

```
1. View
2. Add
3. Update
4. Delete
5. Search
6. Save
7. Exit
Choose an option:  6
Data saved.
```

```
 1. View
 2. Add
 3. Update
 4. Delete
 5. Search
 6. Save
 7. Exit
 Choose an option:  7
```

conclusion

This project allowed us to create a basic student data management system using Python and Pandas. We were able to read student data from a CSV file, update specific student records, and save the changes back to the file. The use of Pandas made it easy to work with tabular data and perform updates based on conditions like Student ID.

Overall, this project helped us understand how to manipulate real-world data using Python and provided a useful foundation for building more advanced data-driven applications.