Login system

It is the entry point for users. A dictionary named users' predefined usernames and passwords. For simplicity, passwords are stored in plain text (although in real-world applications, you should use encryption). The login() function takes user input for a username and password.,checks if the input matches an entry in the dictionary, and identifies whether the user is an admin or student. If the credentials are correct it returns the username to decide the role; otherwise, it prints an error message and exits.

Admin features

The admin menu allows the administrator to manage student profiles. It includes the following, add_student() adds a new student ID and name to the student's dictionary. Before adding, it checks for duplicates to prevent overwriting existing data. View_students(): loops through the student's dictionary and displays all the stored IDs and names .delete_students(): Deletes a student's profile from the dictionary using the student if. If the ID doesn't exist, it alerts the admin.

Student feature

Students have limited access compared to admins. View_profile(): uses the username from the login to look up the student ID and name of the user whose profile is not available *

Data persistence

save _data_to_file() converts the student's dictionary to JSON for format and writes it to a file. JSON is a popular format for storing structured data because it's easy to write and read. Load _data_from_file() checks if a file already exists and reads the stored student data into the student dictionary when the program starts, if the file isn't found, it initializes an empty dictionary.

Main programflow ,

```
import json
```

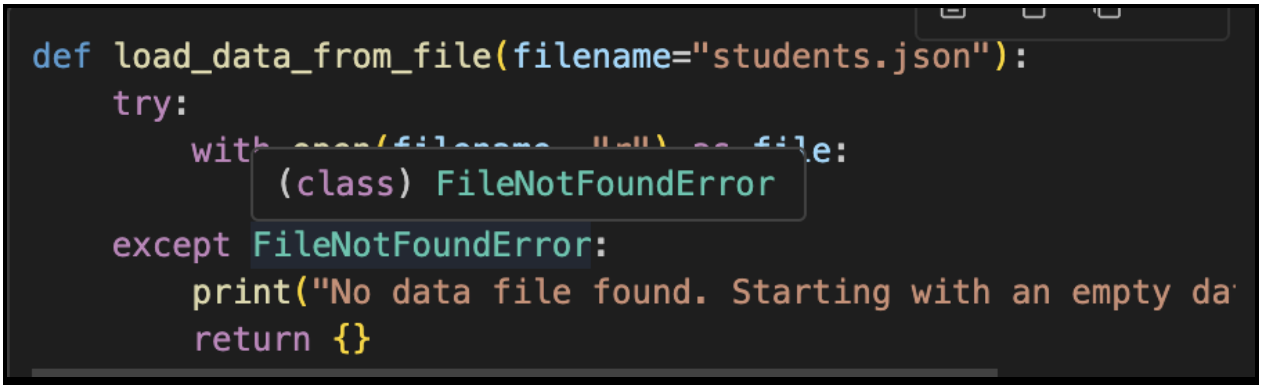
import json , this module is used to read and write data to file in JSON format. This allows you to save and load student data persistently.

2 users dictionary

```
users = {"admin": "admin123", "student1": "pass123"}
```

This dictionary stores predefined usernames and passwords. The admin user has special privileges to manage students, student 1 is a regular user who can only view their profile

3 loading data from file



this function reads students' data from a file named students.json. If the file doesn't exist it prints messages and returns an empty dictionary. Purpose,ensures student data is loaded when the program starts,

4 saving data to file

```
def save_data_to_file(data, filename="students.json"):
    with open(filename, "w") as file:
        json.dump(data, file)
        print("Data saved successfully!")
```

This function saves the students dictionary to a file in JSON format .purpose ensures that changes to student data are saved persistently

5 login function

```
def login(users):
    username = input("Enter username: ")
    password = input("Enter password: ")
    if username in users and users[username] == password:
        print("Login successful!")
        return username
    else:
        print("Invalid username or password.")
        return None
```

This function handles user login .It checks if the entered username and password match the users dictionary,if the login is successful,it returns the username, otherwise, it returns none

6 Adding a student,

```
def add_student():
    student_id = input("Enter student ID: ")
    student_name = input("Enter student name: ")
    if student_id in students:
        print("Student ID already exists!")
    else:
        students[student_id] = student_name
        print("Student added successfully!")
```

this function allows the admin to add a new student .It checks if the student_id already exists in students dictionary ,if not ,it adds students to the dictionary

7 Viewing all students

```python
def view_students():
    if not students:
        print("No students found.")
    else:
        print("\nStudent List:")
        for student_id, student_name in students.items():
            print(f"ID: {student_id}, Name: {student_name}")
```

This function displays all students in the students dictionary,if no students are found ,it prints the message .

8 deleting a students ,

```
def delete_student():
    student_id = input("Enter the student ID to delete: ")
    if student_id in students:
        del students[student_id]
        print("Student deleted successfully!")
    else:
        print("Student not found.")
```

This function allows the admin to delete a student by their ID. If the student_id exists ,it deletes the student from the dictionary.

9 viewing the students profile

```
def view_profile(username):
    if username in students:
        print(f"\nYour Profile:\nID: {username}, Name: {stud
    else:
        print("Profile not found.")
```

This function allows a student to view their profile .It checks if the username exists in the student dictionary and displays their details

10 Global Variable

```
students = load_data_from_file()
```

This initializes students dictionary by loading data from the file ,if the file doesn't exist it starts with an empty dictionary .

.11Main program flow

```
11. Main Program Flow

username = login(users)
```

The program starts by asking the user to log in . If the login is successful ,the program checks if the user is an admin or regular student