**OOP LAB**

**Project Report**



**Submitted to:**

Mam. Ammara Yaseen

**Submitted by:**

Noor-ul-Huda Imran (230201002)

Hafsa Shahzad (230201091)

BS-CS-04B

**Student Complaint System :**

**Purpose:**

To enable students to register, submit complaints, and provide feedback efficiently.

**Table of Contents**

* Introduction
* Methodology
* Functionality
* Sample Outputs
* Limitations & Future Enhancements

**1. Introduction**

The Student Complaint System is designed to streamline the process of complaint registration and feedback collection for students. This system allows students to register, log in, submit complaints, and provide feedback on the resolution of their complaints. The system is implemented in C++ with a focus on object-oriented principles to ensure modularity and reusability.

**2. Methodology**

**Inheritance:** Used to create a hierarchical relationship between classes, ensuring code reusability and logical structure. **For example** the `User` class is a base class from which the `Student` class is derived.

**Templates**: Used to create generic functions for reading and writing items, enabling code reuse for different types of objects.

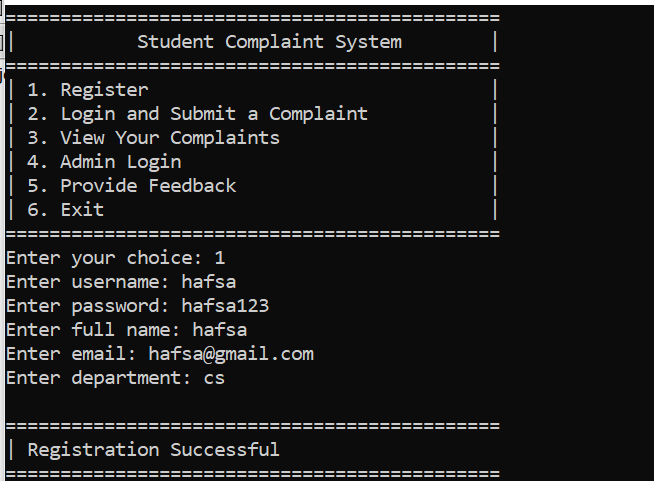
**Polymorphism**: Implemented to allow different user types to handle file I/O operations in their specific ways through virtual functions.

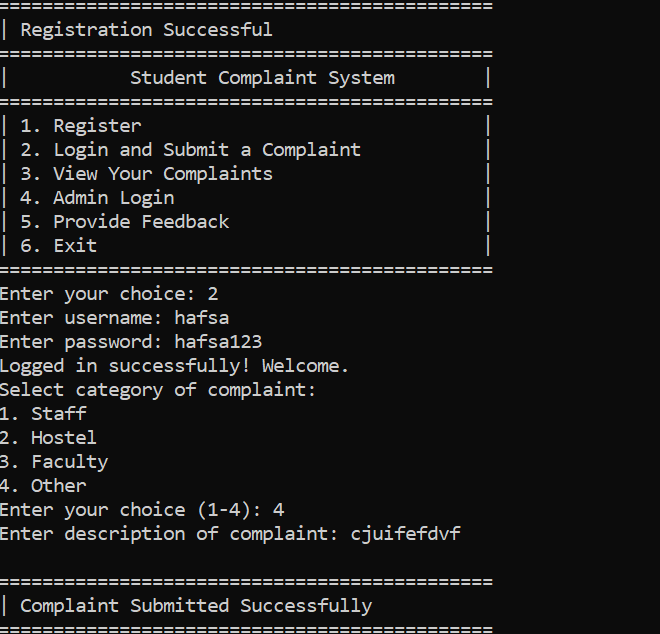
**File Handling:** Data is stored persistently using file handling, ensuring that user registrations and complaints are saved and retrieved accurately across sessions.

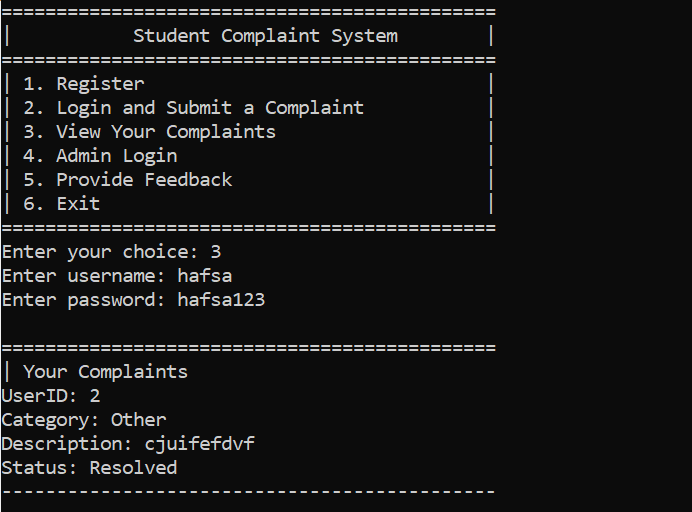
**3. Functionality**

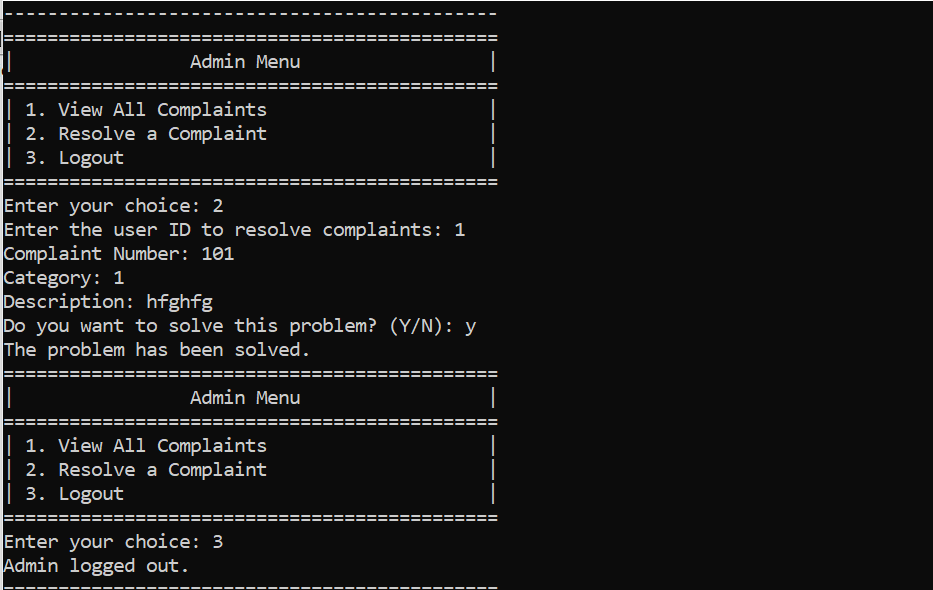


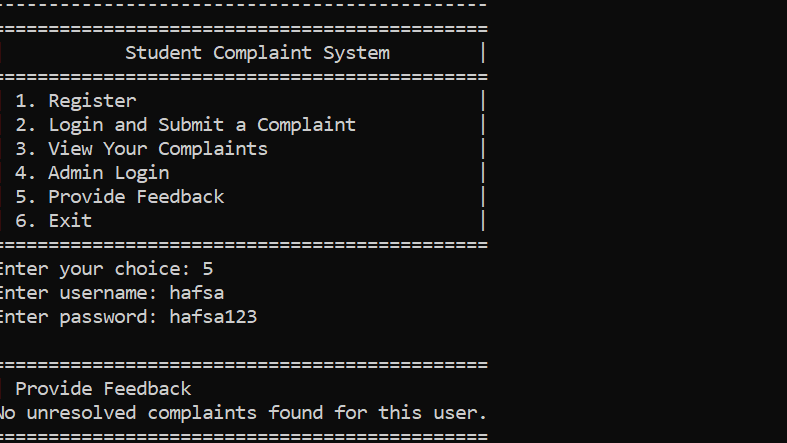
**4. Sample outputs:**











**5.Limitations**

 Passwords are stored in plaintext without encryption

 Limited to viewing and resolving complaints

 Basic input validation.

 No support for multiple simultaneous user

**Future Enhancements** Implement password encryption

 Add user management and detailed reporting

 Use databases instead of text files for better scalability

 Develop a graphical or web-based user interface

 Enable handling of multiple simultaneous users