JUST LOGIC

JUST-University management system

Team names	Email GIt Hub	University major
Yanal Abed	mahmoodabed24@gmail.com	CS
omar Jalboush	osabujalbosh24@cit.just.edu.jo	CS
Mohammad hanandeh	iljoahhanandeh06@gmail.com	SE

Project date: September 4, 2025
Academic year: First year

Introduction

📖 Project Overview (Summary)

The **JUST University** Management System is a **C++** project designed as a centralized platform to manage students, teachers, and administrators. It streamlines data management, course registration, grade handling, and provides a communication channel between users and administration.

The system uses **OOP** concepts such as **Inheritance** and Encapsulation, along with **Data Templates** and text files for permanent **data storage**.

Project objectives

1. Demonstrate Programming Skills
2. Apply OOP Principles
3. Develop Logical Thinking
4. Showcase Project Management
5. Enhance Teamwork
6. Build an Integrated System
7. Design a Good User Experience
8. Manage Data Effectively
9. Ensure Data Persistence
10. Simulate a Real-World System

Project Description

📖 Project Description

📂 Folder Structure

- src/ → Contains all source code files (.cpp, .h).
- docs/ → Documentation files such as PDF reports and diagrams.
- assets/ → Screenshots, images, and GIFs used in the README.
- data/ → Text files for storing system data (students, teachers, courses).

🏷️ Classes Overview

- Student → Manages student information, courses, grades, and profile updates.
- Teacher → Handles teacher information, assigned courses, students in their classes, and profile editing.
- Course → Represents course details, enrolled students, grades, and course management functions.
- Admin → Provides system administration tools, such as managing students, teachers, courses, and generating reports.

⚡ Core Functionalities

- Student Management → Create, edit, delete, and view student information.
- Teacher Management → Create accounts, edit profiles, and view assigned courses.
- Course Management → Add, update, and manage courses.
- Registration System → Register/unregister students from courses.
- Grades Management → Enter, update, and display student grades.
- Reports → Generate enrollment statistics, transcripts, complaints reports, and performance summaries.
- File Handling → Load and save data from .txt files.
- Interactive Console UI → Colorful, user-friendly menu-driven interface.

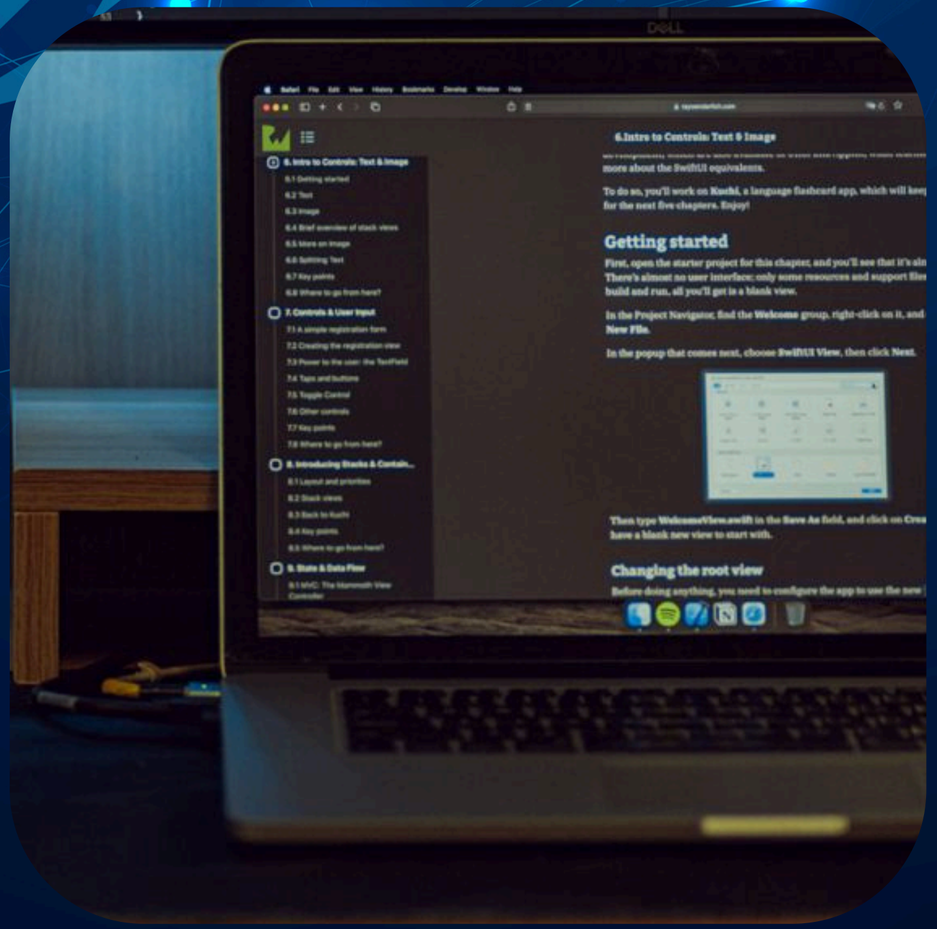
How to Run

01 **Open the src/ folder**
This folder contains all the C++ source code files.

02 **Compile the files**
Use any C++ compiler (e.g., g++) to build the project.

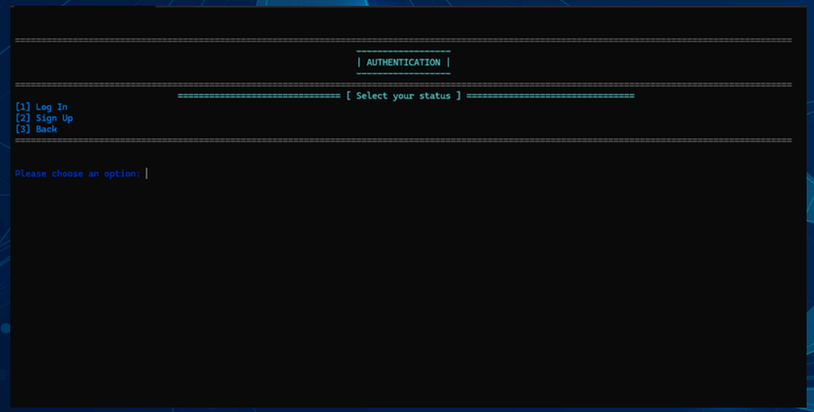
03 **Run main.cpp**
Execute the compiled program from your terminal or IDE.

04 **Follow the on-screen menus**
Navigate through the interactive console interface to use the system.

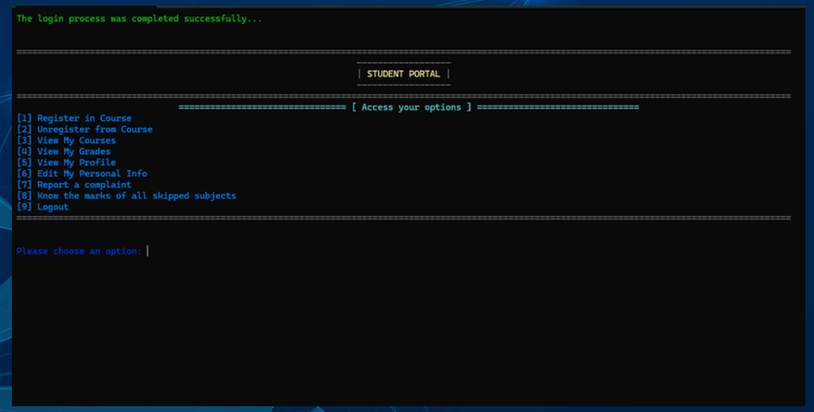


Screenshots / UI

01



02



03

These screenshots demonstrate the core interface of the system. The first image shows the main program window, providing the entry point to the application. The second illustrates the account creation and login process, ensuring secure access for students, teachers, and administrators. The third highlights the system's main features and options, giving users access to course management, grades, and administrative tools in a clear and organized way.

Diagrams

Class Diagram → Provides a clear overview of the main classes **(Student, Teacher, Course, Admin)** and how they are related through inheritance and composition. This helps explain the overall structure of the system and how different parts interact with each other.

- **Flow / Sequence Diagram** → Explains the program's main flow:
  a. User selects login type **(Student / Teacher / Admin)**.
  b. System authenticates credentials.
  c. User navigates through a role-specific portal **(register, manage courses, generate reports)**.
  d. Data is **saved/loaded** from files at the end of each session.

Challenges & Solutions

File Handling Issues

Challenge: Difficulty reading and writing structured data consistently.

Solution: Implemented ifstream and ofstream with a clear format for saving/loading.

LargeCodebase Organization

Challenge: Managing a project with over 5000 lines of code became complex

Solution: Applied Object-Oriented Programming (OOP) principles and separated logic into multiple classes (Student, Teacher, Course, Admin).

Team Collaboration

Challenge: Coordinating code among three first-year students without prior experience

Solution: Divided responsibilities (student, teacher, course, admin modules) and merged code gradually into a final system.

Data Consistency Across Classes

Challenge: Synchronizing updates (e.g., when a student registers in a course, both Student and Course objects needed updates).

Solution: Created dedicated functions for registration/unregistration that update all related lists consistently.

🏆 Results / Achievements

- ✅ Developed a fully functional university management system in C++ with ~5000 lines of code.
- ✅ Implemented organized data management for students, teachers, courses, and administrators.
- ✅ Created interactive and colorful console menus for a better user experience.
- ✅ Added file persistence: all data (students, teachers, courses) can be saved and reloaded across sessions.
- ✅ Generated accurate reports such as enrollment statistics, transcripts, and complaints reports.
- ✅ Applied advanced OOP concepts (inheritance, polymorphism, encapsulation) at an early academic stage.
- ✅ Demonstrated teamwork, dedication, and problem-solving by building a professional-level project as first-year students.

🌱 **Learning a programming language is like planting a tree; the more you nurture it, the stronger it grows.**

💡 **Every bug you fix is not just an error solved, but a lesson that makes you sharper.**

🚀 **Mastery comes step by step — consistency beats speed in the journey of coding.**

Our Team

The project was developed by a dedicated team of students committed to building a comprehensive university management system. Each member contributed to different aspects, including C++ programming, object-oriented design, data management, and user interface design. The team collaborated closely to ensure a high-quality, functional, and user-friendly system.

omar Jalboush

Mohammad hanandeh

Yanal Abed

Conclusion

This project represents a complete University Management System developed in C++. It demonstrates how core programming concepts, such as Object-Oriented Programming, file handling, and data structures, can be combined to build a practical and functional application. The system successfully manages students, teachers, courses, grades, and reports, providing a structured solution for academic administration.

While the current version runs entirely in the console, it opens the door for exciting future improvements — such as adding a graphical user interface (GUI) for better usability, integrating databases for more scalable data storage, or enabling network support to allow multi-user access.

In conclusion, this project reflects not only technical skills but also teamwork, persistence, and problem-solving. It is a strong foundation that can be expanded into a professional-grade system with more advanced features in the future. 🚀

Thank You

Thank you for taking the time to review our project. We hope this system demonstrates our commitment, skills, and passion for creating effective solutions.

0788023665

iljoahhanandeh06@gmail.com

m_hanandeh_

0789148345

mahmoodabed24@gmail.com

yx__up

0781738389

osabujalbosh24@cit.just.edu.jo

wh_r3_