PROJECT IMPLEMENTATION

OF

STUDENT MANAGEMENT SYSTEM

A CLI INTERFACE

SUMMER BOOTCAMP REPORT 2025

SUBMITTED BY: SANJEEV KUMAR.

SYSTEM ID- 2024203327.

In partial fulfilment of the requirement of the degree of

Masters in Computer Applications



<COMPUTER SCIENCE AND APPLICATION DEPARTMENT>

SHARDA UNIVERSITY GREATER NOIDA, U.P., INDIA

MAY 2025-JUNE 2025

**CONTENTS-**

> Acknowledgement.

> Objective.

> Description.

> Process.

> Technologies used.

> Screenshots / code snippets.

> Outcome / result.

> Conclusion.

> References.

**Acknowledgement-**

I would like to express my sincere gratitude to everyone who contributed to the success of my Summer Bootcamp learning experience.

First and foremost, I extend my heartfelt thanks to the **organizers of the Bootcamp**, for providing a valuable platform to enhance my technical skills and practical knowledge. The well-structured curriculum, interactive sessions, and hands-on projects greatly enriched my understanding of [**Bootcamp Topic].**

I am especially thankful to all the **mentors and instructors** who shared their expertise and guided me patiently throughout the bootcamp. Their commitment, timely feedback, and encouragement played a significant role in my learning journey.

I would also like to thank my **college/institute**, and my faculty members for their continuous support and for encouraging me to participate in such meaningful learning opportunities.

Lastly, I am grateful to my peers and teammates for their collaboration, discussions, and motivation, which made this bootcamp a truly enjoyable and enriching experience.

Objectives-

* To create a simple and user-friendly **Graphical User Interface (GUI)** for managing student information.
* To allow **secure login authentication** for administrators.
* To provide functionality for **registering student details** such as name, roll number, and degree.
* To enable students or admins to **select courses** from predefined options.
* To record **exam marks** and automatically calculate **total, percentage, and grade**.
* To generate a **certificate-like summary** of student academic records.
* To maintain a **persistent and formatted record** of student data using file storage.
* To allow viewing of all saved student information in a structured **tabular format**.
* To demonstrate practical usage of **Java Swing, event handling, and file I/O** concepts.
* To develop a **modular, extendable system** that serves as a base for more advanced student management solutions.

**DESCRIPTION-**

The **Student Management System** is a Java-based desktop application developed using the Swing framework. It is designed to streamline the administrative tasks involved in managing student data. This system provides a graphical user interface (GUI) that allows administrators to perform operations such as student registration, course selection, exam marks entry, and certificate generation. The application starts with a secure login screen, ensuring that only authorized users can access the system. Upon successful login, users are directed to a dashboard where they can navigate to different sections of the application. Student details, including name, roll number, and degree, can be entered and saved. Users can then select courses and input marks for four subjects, after which the system calculates the total, percentage, and grade automatically. All data is saved to a file (students.txt) in a structured, tabular format for readability and easy access. The application also includes features to view saved data and generate a certificate-style summary of student records. This project demonstrates key programming concepts such as GUI development, event handling, and file handling in Java, making it a practical and educational tool for learning and applying Java in real-world applications.

**PROCESS-**

The development of the Student Management System began with identifying the core functionalities needed for managing student data, such as login authentication, registration, course selection, marks entry, and record viewing. The project was structured using Java’s Swing framework to build a user-friendly GUI. The first step involved creating the LoginPage class with proper layout using GridBagLayout to handle user authentication. Once the login system was validated, the Dashboard was developed to serve as the central navigation hub for all features.

Subsequently, each major module—RegistrationPage, CoursePage, ExamPage, CertPage, and ViewPage—was implemented as a separate JFrame class to maintain modularity. Layout managers like GridLayout and BorderLayout were used to organize components logically. Data input components such as JTextField, JComboBox, and JTextArea were integrated, and event handling was added through ActionListener to perform actions like saving data, calculating results, and navigating between windows.

For persistent storage, Java’s FileWriter was used to save data into a text file (students.txt). A significant enhancement was formatting the saved data into tabular structure using String.format() to ensure readability. Error handling and validations were introduced where necessary to ensure robustness. After all features were functional, final UI adjustments were made to maintain a consistent color scheme, add emojis for visual clarity, and ensure a smooth user experience.

**TECHNOLOGIES USED-**

* **Java (JDK 8 or above)** – Core programming language used to build the application.
* **Java Swing** – For designing the graphical user interface (GUI).
* **AWT (Abstract Window Toolkit)** – For layout management and event handling (e.g., buttons, panels).
* **Java File I/O (FileWriter, Scanner)** – For saving and reading student data from files.
* **GridLayout, BorderLayout, GridBagLayout** – Layout managers used to structure GUI components.
* **JTextField, JComboBox, JTextArea, JButton, JLabel, JFrame** – Swing components used for interaction.
* **Formatted Output (String.format)** – Used to align data in tabular format within the text file.

**OUTCOMES AND RESULT-**

The Student Management System project successfully met its intended objectives by delivering a fully functional desktop application developed using Java Swing and AWT. The system features a secure login module that ensures only authorized access to student records. Through a user-friendly interface, it allows for smooth student registration, course selection, and entry of examination marks. One of the key outcomes is the automated calculation of total marks, percentage, and the corresponding grade based on entered scores. All student information is stored persistently in a structured and readable table format within a text file, enabling easy access and review. The system also includes options to view saved data and generate certificate-style summaries, making it a comprehensive tool for basic academic data management. The project demonstrates practical implementation of Java GUI programming, event-driven interactions, and file handling, and serves as a solid foundation for building more advanced student information systems in the future.

**CONCLUSION**

In conclusion, the Student Management System project has effectively demonstrated the development of a functional and interactive desktop application using Java. By incorporating core concepts such as GUI design with Swing, event handling, and file management, the project provides a complete solution for managing basic student data. It enables administrators to register students, manage course selections, enter exam marks, and generate certificates, all through an intuitive graphical interface. The use of structured file storage ensures that student records are maintained in a readable and organized format. This project not only enhances user experience through a clean and consistent UI but also reinforces essential programming skills. It lays a strong foundation for future enhancements, such as integrating databases, search features, and printing capabilities, making it a scalable and extendable educational tool.

**REFERENCES**

* **Oracle Java Documentation** – Official documentation for Java programming language.  
   🔗 <https://docs.oracle.com/javase/8/docs/>
* **Java Swing Tutorial – GeeksforGeeks** – Helpful for understanding Swing components and layout managers.  
   🔗 <https://www.geeksforgeeks.org/java-swing/>
* **Java File Handling – W3Schools** – Basic guide to working with files in Java (FileWriter, Scanner, etc.).  
   🔗 <https://www.w3schools.com/java/java_files.asp>
* **Java AWT & Event Handling – TutorialsPoint** – Useful for implementing user interaction and UI layout.  
   🔗 <https://www.tutorialspoint.com/java/java_event_handling.htm>
* **Stack Overflow** – Community discussions and troubleshooting help during development.  
   🔗 <https://stackoverflow.com/>
* **Oracle Swing Layout Managers Guide** – Detailed explanation of GridLayout, BorderLayout, GridBagLayout.  
   🔗 <https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>