STUDENT MANAGEMENT SYSTEM – JAVA

PROJECT REPORT

1. INTRODUCTION

The Student Management System is a simple Java application that performs the basic functions required by an educational institution: managing student records, courses, enrollments, and results.

This project uses Core Java concepts such as Classes, Objects, Inheritance, File Handling, Collections, and Serialization.

2. OBJECTIVES

- To build a functional Java-based Student Management System.
- To apply Object-Oriented Programming principles.
- To implement file handling and persistent data storage.
- To use modular code with packages and multiple classes.

3. SOFTWARE REQUIREMENTS

- Operating System: Windows / macOS / Linux
- JDK 8 or above
- Any text editor (VS Code, Notepad++, IntelliJ, Eclipse)
- Command Prompt / Terminal

4. SYSTEM DESIGN

The system is divided into the following modules:

- Admin Module
- Student Module
- Course Module
- Enrollment Module
- Result Module
- Data Storage Module

4.1 CLASS DIAGRAM (TEXT FORMAT)

Classes Used:

- Student

- Course

- Enrollment

- Result

- Store

- App

- Main

- StudentCLI

- CourseCLI

- ResultCLI

Each class handles a specific responsibility, following modular and clean coding practices.

5. FEATURES IMPLEMENTED

Admin Features:

- Add Student
- Update Student
- Delete Student
- Add Course
- Update Course
- Delete Course
- Enroll Student
- Add & Update Results
- View all enrollments and results

Student Features:

- Login using Student ID
- View Profile
- View Enrolled Courses
- View Results

6. WORKING OF THE SYSTEM

- The Admin logs into the system using a password.
- Admin can manage students, courses, enrollments, and results.
- Data is stored permanently using Java Serialization.
- Students can log in to view their own information.

7. OUTPUT (SAMPLE)

- Student added successfully
- Course added successfully
- Student enrolled successfully
- Result added successfully
- Showing profile, enrollments, and results

8. CONCLUSION

This Student Management System fulfills the basic requirements of an educational institution.

It demonstrates Java OOP concepts, file handling, collections, and modular coding.

The project is simple, effective, and easy to expand in the future.

9. FUTURE SCOPE

- GUI using JavaFX or Swing
- Adding database (MySQL)
- Attendance Management
- Timetable Module

10. REFERENCES

- Oracle Java Documentation
- Java OOP Tutorials
- Classroom materials