**Student Management System (CUI – Java)**

**Anshika Kumari**

**Abstract:**

**The Student Management System (SMS) is a simple console-based application developed in Java to streamline the interaction between students and administrators. The application provides separate modules for Admin and Student users, enabling efficient management of academic courses, subjects, and exam records. The system ensures user-friendly functionalities and maintains academic data integrity through a structured interface.**

**2. Problem Statement:**

**Managing student records, course structures, and exam results manually in educational institutions can be time-consuming and error-prone. Lack of a digital solution often leads to inefficient handling of academic processes. This project aims to automate these tasks by providing a Java-based console application that helps administrators manage courses, subjects, and student records while allowing students to register, select subjects, and take exams.**

**3. Methodologies:**

* **Programming Language: Java (Core)**
* **Development Environment: Any IDE supporting Java (e.g., IntelliJ IDEA, Eclipse)**
* **Design Approach: Object-Oriented Programming**
* **Key Classes and Components:**
  + **Main: Entry point of the program**
  + **Admin: Handles admin functionalities**
  + **Student: Handles student operations**
  + **Course, Subject, Exam, Result: Models representing core entities**
* **Storage: In-memory storage using Java collections (ArrayList, HashMap)**
* **Testing: Manual testing via console inputs**

**4. Use Case Scenarios:**

**Admin Use Case:**

* **Admin logs in → Adds courses and subjects → Views student registrations → Checks student exam results**

**Student Use Case:**

* **Student registers → Views courses → Selects subjects → Takes an exam → Views result**

**5. Future Enhancements:**

* **Integration with databases for persistent storage**
* **GUI version using JavaFX or Swing**
* **Email notifications for results**
* **Role-based login system with password authentication**
* **Question bank and random exam generation**

**6. Conclusion:**

**This Student Management System efficiently addresses the gap in academic record management using a simple Java console interface. By separating functionalities based on user roles, it enhances usability and improves institutional workflow. With scope for future upgrades, this system can serve as a foundation for more robust academic ERP systems.**