**🎯 Problem Statement**

Design and implement a **console-based University Course Registration System** using **JDBC** and **Java 8 Streams/Lambdas** with **MVC architecture**.

The system should manage **students** and **courses**, and allow students to register for multiple courses. All data is stored in a **database**, but business logic (filtering, sorting, aggregations) should be performed in **Java 8 Streams** in the **service layer**.

**🔹 Database Schema (university\_db)**

1. **students**
   * student\_id (INT, PK, Auto Increment)
   * name (VARCHAR)
   * year (INT)
2. **courses**
   * course\_id (INT, PK, Auto Increment)
   * title (VARCHAR)
   * credits (INT)
3. **registrations**
   * student\_id (FK → students)
   * course\_id (FK → courses)

**🔹 Operations (Console Menu)**

1. **Add Student** (insert into DB)
2. **Add Course** (insert into DB)
3. **Register Student for Course** (insert into registrations)
4. **View All Students with Registered Courses**
   * Use JOIN in DAO + **Java 8 groupingBy** in Service
5. **Search Courses by Minimum Credit Requirement**
   * Fetch courses from DB → **Stream filter()** in Service
6. **Get Students Registered in a Particular Course**
   * Fetch all → **Stream filter() + map()**
7. **Sort Students by Year and then by Name**
   * Use Comparator.comparing() with Streams
8. **Calculate Total Credits Per Student**
   * Fetch join data → **Collectors.groupingBy() + summingInt()**

**🔹 Complexity Highlights**

* **DAO Layer**: Performs JDBC CRUD on students, courses, registrations.
* **Service Layer**: Uses **Java 8 Streams** for filtering, mapping, grouping, and sorting.
* **Controller Layer**: Console-driven interaction.
* **Collections**:
  + Use List<Student>, List<Course>, Map<Student, List<Course>>.
* **Intermediate Complexity**: Multiple tables + Many-to-Many mapping + Java 8 logic.