CS 301 - Introduction to Database Systems
Course Project Specification Phase A
Academic Portal (Total Weightage: 18%)

General Instructions:

- **Allowed team size:** 2 members or 3 members. Projects from teams of size 3 would be more complicated for the same score and same deadline. Note that given the open-ended nature of the project specification, it's not possible to have a precisely equitable distribution of work across the team sizes.
- Students working solo would have to do the same work as that of team size 2.
- Students should use the given idea and develop it into a full-scale project proposal. We are just giving a just high level idea of the project and by no means a full specification.
- Periodic reviews (via live sessions) are mandatory to give a proper direction to the projects.
- Do not disappear after taking a topic and show up only in the week before the final deadline. In such cases, instructors would simply proceed with their discretion on what should have been done!

Project Description:

You are expected to develop an academic portal (similar to AIMS at IIT Ropar) for an academic university. For this project, you are expected to develop the application using only database system functionalities such as stored procedures and triggers. In other words, there would not be any UI component for this project.

Application Description (Compulsory section for both size-2 and size-3 teams):

In this course project, you are required to design and implement a multi-user database application for managing the academic portal of an academic institute. For sake of convenience, we would be focusing on creating something which is based on the academic policies of our institute. As you are already aware, our academic eco-system for UG program basically consists of the following stakeholders.

- Students
- Faculty
- Batch Advisor
- Dean Academics Office

Key concepts for the academic system:

1. Course Catalogue: This contains all the list of courses which can be offered in IIT Ropar. For each course, we have information on its credit structure (L-T-P-S-C) and list of prerequisites (if any).
2. Course Offerings: Each semester, a faculty offers one or multiple courses. These courses should be present in the course catalogue. With each course offering, the instructor provides a timetable slot (from a list already defined by the academic office at the start of semesters), a list of batches which are allowed to register in the course. In addition, instructors may also define additional constraints (e.g., CGPA > 7.0).
3. Student Registration: A student registers for one or more courses. However, the number of credits he/she is allowed is governed by the scheme governed by the institute (1.25 times the average of the credits earned in the previous two semesters. Additionally, a student is not allowed to register for more than one course in a single timetable slot.
4. Ticket Generation: If a student wishes to enroll for courses which are beyond what is allowed by the credit-limit and the course-offering-restrictions, then he/she raises a ticket. This ticket first goes to the instructor of the course,

followed by the academic advisor and finally the Dean Academics office. Each of them can either accept and forward OR reject. Final decision is made only by the dean academic office. You should store the history of decisions made by various people.

5. Report Generation: Staff in the Dean's office need to generate transcripts of students.
6. Grade entry by Course Instructors: Grades are to be imported from a csv file.

Various Checks, Constraints and Privileges to be implemented:

1. A student is not allowed to register for courses which are scheduled in the same time-slot.
2. A student is not allowed to register for a course without clearing the pre-req. Also he/she not allowed to register for more than the allowed credit limit (unless his tickets are approved).
3. **The course catalogue can be edited only by the Dean Academics office.**
4. **A student can see only his/her grades. But a faculty or Dean Academics office can see all the grades.**

Key Stored Procedures to be developed:

1. Develop stored procedures for uploading the time-table slots for a semester (by academic staff), offering a course (by faculty), registering a course (by student), ticket generation and propagation, generating the transcript (for a given studentid) and grade uploading for a course (importing from a csv file).
2. Implement a stored procedure to compute the current CGPA of the student.

Extra Work for Size-3 Teams:

1. Implement the concept of UG curriculum into the application.
2. You should maintain a list of Program Cores, Program Electives, Science Core, Open Electives for each batch.
3. Note that information in item 2 may change with time. For e.g., a course which was a program code for a batch may no longer be a program code for their junior batch.
4. A key function to be implemented by size-3 teams is the following: "is the student ready to graduate(<studentID>)." This stored procedure should check if the student has completed all the required credits in program core, program elective, science core and open elective. Additionally, he should have a minimum CGPA of 5.0.

Important Things to be noted:

1. It should have the least amount of redundancy while keeping an eye on the performance.
2. **Note that you would have to create individual database logins for each user and assign appropriate permissions that need to be given for each table. Suitable authorization should be given to database tables (use GRANT and REVOKE commands wisely).**
3. Pay more attention to underlying database design, for e.g,, triggers, constraints, stored procedures and authorizations.