UE17CS252: DATABASE MANAGEMENT SYSTEMS (4-0-0-0-4)

DBMS course evaluation scheme

Sl. No.	Details	Marks	Reduced to	Final Marks
1	ISA/CBT-1	40	**20**	
2	ISA/CBT-2	40	**20**	
3	**In Semester Assessment (ISA)**			**40**
4	**DBMS Project**	40	**20**	
5	ESA – pen and paper		**40**	
6	**End Semester Assessment (ESA)**			**60**
7	**Total**			**100**

As part of the course, students are expected to do a project.
The details of how this project must be done and evaluated is described below.

The DBMS Project

The project must be a traditional database application. The first two paragraphs of the textbook clearly mentions traditional database applications and new types of database systems. (NoSQL database applications are not allowed in this project)

The following databases are recommended/permitted to be used:

- **PostgreSQL** (Mandatory for creation of tables, constraints, and running of queries for *first and second evaluation*)
- **MySQL** (In the final evaluation, for development of user interfaces PHP-mysql is allowed, even though PostgreSQL+ PHP is the preferred development platform.)

Only PHP is allowed for development of user interfaces.

Project Teams

The project will be done by a *group of 3 students (3 and no more or no less)*. One or two teams in a class may be an exception (one/two teams of two members each) with prior approval of class teacher. *All teams must be among students belonging to the same section. (No teams are allowed to span multiple sections)*

Project Deliverables and Evaluation Details

Sl. No.	Deliverable	Marks Breaakup	Marks	Due Date
1	ER Diagram/Schema	8		
2	CREATE statements	2		
3	**First Evaluation**		**10**	**Feb 9**
4	CRUD (Insert, Simple Select, Update, Delete) Operations and HTML/PHP forms.	5		
5	SQL Queries (Complex Queries)	5		
6	**Second Evaluation**		**10**	**Mar 23**

7	**Final Evaluation**		**20**	**Apr 18**
8	Demo (Group)	5		
9	Report (Group)	5		
10	Viva (Individual)	10		

Evaluation Criteria:

1. For ER Diagram, must incorporate and explain as much of the features as possible as described in Chpater – 3 (Data Modeling Using ER Model)
2. For CREATE statements, must incorporate as much of the features as possible as described in Chapter 6.1 and 6.2
3. For CRUD (Insert, Simple Select, Update, Delete) Operations, must incorporate as much of the details as possible in Chapters 6.3 and 6.4
4. For HTML/PHP forms, must be incorporating the ideal input/output (queries) scenarios. *Even though there are no chapters for UI design, students must use their knowledge of PHP and HTML forms from the Web-Technology-1 course of previous semester and their own creativity.*
5. SQL Queries (Complex Queries), must include as man of the types of complex queries as described in Chapter 7.
6. Final Evaluation
    1. Demo:
        1. Must demonstrate the working of the application including inserting, updating, deleting and retrieving records using PHP forms.
        2. Must be able to explain code used.
        3. Must be able to modify the feature/behavior of a form/user interface.
        4. Must be able to show exceptions/errors.
    2. Report: Must include a section for each of the following:
        1. Introduction and problem definition: what purpose does your database application serve and how does it help?
        2. The *miniworld*: a brief description of the data that will be represented in the database.
        3. Requirements: A list of “things” the user can do with the system. For example, “The user must be able to insert, update, delete a student.” - this requirement results in a form to insert/update/delete a student record. “The user could query a student by name, department and semester and section” - this requirement results in a form to query a student record. …Each requirement results in a coding unit (form, query or a batch update program.)
        4. ER Diagram
        5. Schema Diagram
        6. A section on CREATE/ALTER table scripts and Constraints.
        7. PHP forms/UI Screens. Also called wireframe diagrams.
        8. SQL Queries.
        9. Test cases for testing exception/error conditions that are handled in the design.
        10. Discussion and Conclusion.
    3. Viva:
        1. *Must be able to explain each and every component. No excuses of division of work.*
        2. *Must explain code.*
        3. *Must be able to modify code.*
        4. *Must be able to write new SQL queries based on requirement given by teacher.*
        5. *Must include some prepare-and-execute SQL (dynamic and embedded SQL)*