

# **CS 432: Databases**

## Assignment 4: DEPLOYING THE DBMS

<b>Total marks: 100M</b>	<b>Submission deadline: 23:59:59 Hrs, April 15, 2023</b>
--------------------------	--

### **1. Assignment Instructions**

Please refer to the following assignment instructions:

1. Regarding the late submission, we will be following the penalty as per the table:

<b>Late Submission</b>	<b>Penalty (Out of 100)</b>
Till 1-hour past deadline	5 points
1 to 5 hours past deadline	10 points
5 to 12 hours past deadline	20 points
12 to 24 hours past deadline	40 points
24+ hours past deadline	100 points

2. No assignment-related queries will be answered after April 10, 2023, 23:59:59.
3. We will follow the zero plagiarism policy, and any act of plagiarism will result in a zero for the assignment.
4. Please cite and mention others' work and give credit wherever possible.
5. If you seek help and discuss it with the stakeholders or individuals, please ask their permission to mention it in the report/submission.

### **2. Problem Statement & Requirements**

1. All the relations and their constraints should be reflected from Assignment 1.
2. Login page with authentication of users and stakeholders.
3. The system should be safe from 4 attacks (2 on the front end and 2 on the back end), 1 additional attack by G1 or G2 will lead to 20 bonus points per G1/G2 subgroup.
4. Show the systems to respective stakeholders, and get feedback on how to improve and upgrade their systems.
5. Push it to GitHub and share the link for submission.

### 3. Tasks

#### 3.1 Responsibility of G1:

**40 Pts.**

1. The G1 takes two feedbacks from the stakeholders, [one initial feedback](#) (on or before 30th March 11:59 PM), and then makes relevant changes as suggested per the first feedback, then [final feedback](#) (on or before 10th April 11:59 PM) post changes. The write-up/documentation should have screenshots before the first feedback, after the first feedback, and after the second feedback. If a team discusses with multiple stakeholders, please fill out the forms again (Initial and final feedback forms). **(30 Points)**
2. Attach screenshots of different views [along with a write-up on their privileges] of the database as seen by different classes of users. **(10 Points)**

#### 3.2 Responsibility of G2:

**40 Pts.**

1. Concurrent multi-user access: Multiple users with different roles can access and update the database concurrently. In such a scenario, the same item can not be updated by two different users. For example, locks can be applied to tables in MySQL. **(10 Points)**
2. Implement the changes in the database as per the feedback received from stakeholders. **(30 Points)**

#### 3.3 Responsibility of G1 & G2:

**20 Pts**

1. Documentation and screenshots of a total of 2 attacks [SQL Injection and XSS] performed and the defenses against those attacks. **(15 points)**
  - a. Additional attack and defense will lead to 10 **bonus** points for the team. *(Maximum 20 bonus points)*
2. Show that all the relations and their constraints, finalized after the second feedback, are present and valid as per the ER diagram constructed in Assignment 1. **(5 points)**

### 4. Submission

1. The submission pdf (or README) must have four sections with screenshots of the operations of the web app:
  - a. Responsibility of G1 (Answers to the above questions)
  - b. Responsibility of G2 (Answers to the above questions).
  - c. Responsibility for both G1 and G2.
  - d. Contributions with a list of members in G1 and G2.
2. Combine all the code & files, push it to Github, and submit the link in [this](#) Google form. Make sure to add all the files (HTML/CSS/JS/.py files), as there will be a pilot test of the web app (Ensure the path of files are relative).
3. While compiling the final PDF or README, please make sure that all the responsibilities and contributions are mentioned clearly (Also, please justify the individual contributions). The contributions can be added in the last.

4. PLEASE ADD THE SQL DUMB OR THE **.sql** FILE WITH QUERIES.
5. Working code of the implementation (HTML/CSS/JS/.py files).

*Note: By submitting this assignment solution, you confirm to follow the IITGN's honour code. We shall strictly penalise the submissions containing plagiarised text/code.*

## **5. References:**

If required, please feel free to take help from the following references:

1. User authentication for flask [[here](#)].
2. Tutorial & recording on SQL Injection [[here](#)].