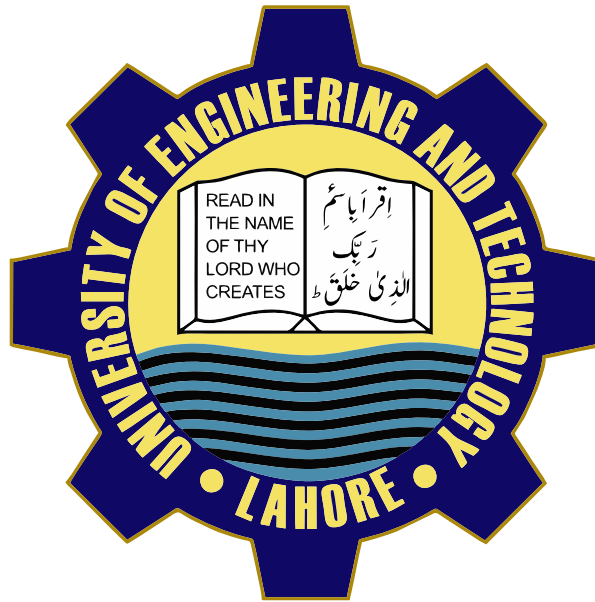


# Semester Project Proposal

## Database System for Version Control Platform



### ***Submitted By***

AbaidUllah    2023-CS-713

Abdul Rahman    2023-CS-725

Sohaib Shaukat    2023-CS-740

### ***Submitted To***

Ms. Rimsha Noreen

### ***Submission Date***

April 16, 2024

Department of Computer Science University of Engineering and  
Technology Lahore, New Campus

## Scenario Description

The project aims to deliver a robust and scalable database system that effectively supports the functionalities of a version control platform while ensuring data integrity, security, and optimal performance. The platform will enable users to create repositories, track changes made to source code, manage issues, collaborate with other users, and engage in discussions through comments. The system will also facilitate version control by recording commits, managing branches, and allowing users to star repositories and follow other users.

## Requirements and Constraints

- **User Management:** Track user details including username, email, password, and login activity.
- **Repository Management:** Store information about repositories such as name, description, and creation date. Associate repositories with their respective owners.
- **Issue Tracking:** Manage issues related to repositories including title, description, status, and timestamps for creation, update, and closure. Associate issues with their respective repositories and users.
- **Comment System:** Enable users to comment on issues with timestamps for creation and updates.
- **Version Control:** Record commits made to repositories including commit message, commit date, and associated user.
- **Branching:** Manage branches within repositories including name, creation date, and association with repositories.
- **Collaboration:** Allow users to collaborate on repositories by adding collaborators.
- **Star System:** Enable users to star repositories they find interesting.
- **Follower System:** Facilitate user following functionality.

## Rationale Behind Design Decisions

- **Data Integrity:** Utilize primary keys and foreign keys to maintain data integrity and enforce referential integrity constraints.
- **Normalization:** Employ normalization techniques to minimize redundancy and ensure efficient storage and retrieval of data.
- **Scalability:** Design the database with scalability in mind, allowing for the potential growth of users, repositories, and associated activities.

## Potential Challenges and Strategies

- **Data Consistency:** Implement robust transaction management to ensure data consistency, especially in scenarios involving concurrent access.
- **Performance Optimization:** Employ indexing and query optimization techniques to enhance performance, particularly when dealing with large datasets.
- **Security:** Implement stringent security measures to protect user data and prevent unauthorized access or malicious activities.
- **User Experience:** Prioritize user experience by optimizing database queries.

## Entity-Relationship Diagram

