

# Report: Online Quiz Platform

Aneesh Aithal 15C0107  
Sagar Bharadwaj 15C0141

---

## Introduction

The proposed project is to create an online platform to make taking and conducting quizzes an easier task. This document will go about defining the problem space and the requirements for the platform.

**Brief :** Students can simply log into their respective accounts and take tests. These tests are uploaded onto the server by the test setters. These tests and responses are stored in the database and the DBMS facilitates easy access.

## Requirements

### Student Requirements

- Each student would require his own credentials in order to login and participate in a test.
  - Students will be presented with a schedule of all the tests they are eligible for.
  - They should be able to start a test as and when the test is active.
  - The platform must enable submissions of both objective and subjective answers for each test.
  - The students must also be able to query the correct/expected answers for each question after the test.
  - Students will also be able to view the score allotted to them in test/quiz by the faculty after the evaluation is complete.
  - Students can also visit the leaderboard to analyse their relative performance in the test if allowed by the test setter.
-

---

### Test setters' requirements

- The test setter (eg. a professor in a college setting) can add new tests. Adding new tests will involve adding questions along with expected answers
- The test setter should be able to mention a list of eligible students who can attempt the test.
- The test setter can also make the test results public and give permission to generate a leaderboard for the test to encourage healthy competition.
- After the test, the test setter can use the evaluation engine to evaluate students' responses by comparing them with expected answers.
- The test setter should be able to evaluate descriptive answers and assign marks to responses.

### Preliminary Database requirements

The platform relies on a database to store and query questions/answers. The login capability also relies on the database to store credentials. The database consists of several tables. The list of tables are (This list is non exhaustive):

- A table to store student details. It'll consist of personal information about the student like Name, Sex, Class, etc. The table also consists of all the tests the student has taken and will be linked to the tests table. The table will also contain an attribute with a list of tests the student is eligible for.
- A similar table like above to store details about professors/Test setters. It'll also consist of a list of the tests given by the Test setter.
- A table to store the login credentials and permissions given to the user.
- A universal test table containing information about all the tests conducted or to be conducted on the platform. The table will contain information about the test schedule, test setter, max test score etc
- Test tables : Each test will be associated with two tables
  - A table updated by the test setter consisting of questions with the corresponding expected answers for each question. It'll also store the marks allotted to each question.

- 
- A table of responses with rows corresponding to the response given by each student who participated in the test. The table will also contain a field to store marks allotted to that particular student for his responses.
  - A leaderboard table if allowed by the test setter. Visitors can query for a subset of the leaderboard entries by using filters.

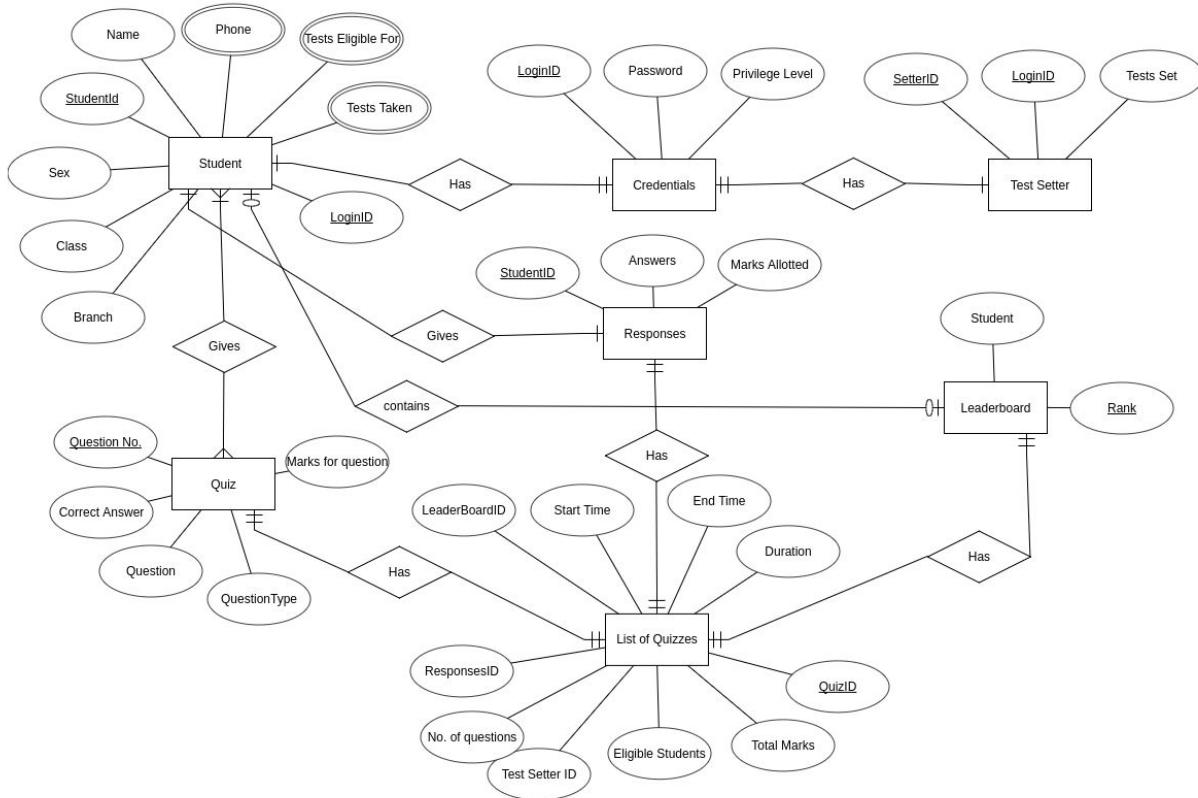
## Website Requirements

- The front page is a login screen.
- After login, student and test setters will be directed to their respective dashboards.
- On the student dashboard, students can:
  - View the list of tests they're eligible for.
  - Take an active test.
  - View results of previously taken tests.
  - Visit the leaderboard.
  - View expected answers for tests they've already taken.
  - Can change student information/login credentials.
- On the test setter's dashboard, test setters can:
  - Add a new test. The questions and answers are also uploaded.
  - Select all students eligible for a test.
  - Evaluate a test after all responses have been submitted.

## Technologies to be used

- Database Management System : **MySQL**
- Server backend : **Node.js**
- Front end framework : **Angularjs**
- Languages used : HTML, CSS, javascript, SQL
- Other packages : Express, passportjs etc

# Entity Relationship Model



## Legend

Crow's foot notation has been used in the Entity-Relationship model.

Symbol	Meaning
—	One—Mandatory
— <=	Many—Mandatory
—○	One—Optional
—○<=	Many—Optional