

# Online Quiz System (OQS)

## 1. Functional Requirements

### 1.1. User Authentication & Authorization

#### 1.1.1. User Registration

The system Should Allow Instructors and Students To create an account using their email, and password.

#### 1.1.2. User Login

After Ensuring The Data The User Entered Is Correct We allow The User The Login To The system

#### 1.1.3. Role Based Access

Different Roles With Different Permissions For Example Student Can Solve Quizzes But Cannot Create Them

#### 1.1.4. TokensGeneration

After The User Logged in The System Generates Access Token (Short Time Expiration) And Refresh Token(Long Time Expiration) For Him

### 1.2. Instructor: Quiz & Question Management

#### 1.2.1. The system should allow instructors to create a new quiz with metadata: title, description, duration, start/end time, total marks, and visibility.

#### 1.2.2. The system support multiple-choice questions (MCQs) with one correct answers

### 1.3. Student: Attempting Quizzes

#### 1.3.1. The system should allow a student to start a quiz only during its availability time.

#### 1.3.2. Student Can See all Quizzes He Entered And His Answers, Graded And Passed Or Non Passed Quizzes

### 1.4. Grading Mechanism

#### 1.4.1. After Student Sends Submit The Graded Calculated And The Answers Stored In The DataBase

#### 1.4.2. Student Can't Edit Any Answer After He Submit And Can't Start The Same Quiz Again

### 1.5. UI & Accessibility

- 1.5.1. Separated DashBoard For Each Role Compliant With His Function He Can Do In The System
  - 1.5.2. Responsive UI Due To Using Asynchronous Programming The Mean No Freezing To The Threads
- 

## 2. Non-Functional Requirements

### 2.1. Security

- 2.1.1. All Password Saved Encrypted In The DataBase Using
- 2.1.2. Using Strong Secret Key With The JWT Token

### 2.2. Performance

- 2.2.1. Using Caching To The Quizzes Currently Available So We Reducing Number Of Times We Need To Access The Quizzes of The Database Due To We Know Most Of Students Will ask For This Quiz Right Now So We Cash it

### 2.3. Compatibility

- 2.3.1. The system **shall** support modern browsers (Chrome, Firefox, Edge, Safari) and degrade gracefully on older browsers.

### 2.4. Maintainability

- 2.4.1. The codebase should be modular (separation of concerns)

### 2.5. Reliability & Availability

- 2.5.1. Applying Rate Limiting Due To Preventing The User From Sending Infinite Stream Of Requests