



---

# ONLINE QUIZ SYSTEM

## (QUIZZO)

---

TRACK: PHP Web Developer  
DEPI Graduation

**Team Members :**

- Mennatuallah Nabil**
- Anton Rizk**
- Engy Mohamed**
- Mohamed Ahmed Mousa**
- Ghada Ahmed**

**Instructor : Amr Mohamed Eissa**

The Online Quiz System is a web-based platform developed using PHP, Laravel, and MySQL. It allows users to take quizzes on various topics, track their scores, and view their performance history

# **1. Introduction**

## **1.1 Project Overview**

The Online Quiz System is a web-based application developed using PHP and MySQL, leveraging the Laravel framework. This system allows users to take quizzes on various topics, track their scores, and view their performance history. Administrators can create quizzes, set time limits, view participant results, and generate reports.

## **1.2 Objectives**

- Develop a user-friendly platform for online quiz-taking.
- Implement a robust system for quiz creation and management.
- Provide score tracking and performance history for users.
- Create an admin dashboard for quiz management and result analysis.
- Ensure secure user authentication and role-based access control.

## **1.3 Technologies Used**

- PHP 8.0
- Laravel 9.0
- MySQL
- HTML5, CSS3, JavaScript
- Bootstrap5 (for responsive design)
- Laravel Breeze (for user authentication)

# **2. System Architecture**

## **2.1 High-Level Architecture**

The Online Quiz System follows typical Laravel MVC (Model-View-Controller) architecture:

- Models: Represent the data structures and business logic.
- Views: Handle the presentation layer.
- Controllers: Manage the flow between models and views.

The system also utilizes Laravel's built-in features such as Eloquent ORM for database interactions, Blade templating engine for views, and Laravel Mix for asset compilation.

## **2.2 Database Schema**

The database consists of the following main tables:

1. users: Stores user information and authentication details.
2. Topics: Contains all available topics.
3. quizzes: Contains quiz metadata (title, description, time limit, type, topic).
4. questions: Stores individual questions linked to quizzes.
5. options: Contains answer options for each question.
6. answers: Stores user responses to individual questions.
7. performance\_histories: Records user attempts with the score at quizzes.

## **2.3 Laravel Framework Structure**

The project follows Laravel standard directory structure:

- app/: Contains the core code of the application
  - Http/Controllers/: Holds all the controllers
  - Models/: Contains Eloquent model classes
  - Providers/: Service providers for the application
- config/: Configuration files
- database/: Database migrations and seeders
- public/: Publicly accessible files (CSS, JavaScript, images)
- resources/: Views, language files, and uncompiled assets
- routes/: Route definitions

## **3. Setup and Installation**

### **3.1 Prerequisites**

Ensure you have the following installed on your system:

- PHP 8.0 or higher
- Composer
- MySQL 8.0 or higher
- Node.js and npm (for frontend asset compilation)
- XAMPP for cross-platforms (WAMPP for MS windows, MAMPP for MacOS)

### 3.2 Environment Setup

1. Clone the repository:

```
git clone  
https://github.com/your-repo/online-quiz-system.git  
  
cd online-quiz-system
```

2. Install PHP dependencies:

```
composer install
```

3. Install JavaScript dependencies:

```
npm install
```

4. Copy the .env.example file to .env and configure your environment variables:

```
cp .env.example .env
```

5. Generate an application key:

```
php artisan key:generate
```

### 3.3 Database Configuration

1. Create a new MySQL database for the project.
2. Update the .env file with your database credentials:

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=your_database_name  
DB_USERNAME=your_username  
DB_PASSWORD=your_password
```

3. Run database migrations and seeders:

```
php artisan migrate --seed --class=SuperAdminSeed
```

### 3.4 Laravel Installation and Configuration

1. Compile and optimize assets:

```
npm run build
```

2. Install all the required packages:

```
composer require realrashid/sweet-alert  
composer require barryvdh/laravel-dompdf
```

3. Start the development server:

```
php artisan serve
```

Your Online Quiz System should now be accessible at  
**http://127.0.0.1:8000**

## 4. Features and Functionality

### 4.1 User Management

#### 4.1.1 User Registration

Users can register for an account by providing:

- Name
- Email address
- Password

The system validates input and ensures email uniqueness before creating a new user account.

#### 4.1.2 User Authentication

The system uses Laravel's built-in authentication features, including:

- Secure password hashing
- Remember me functionality
- Password reset via email
- Email verification

#### 4.1.3 Role-Based Access Control

Two main roles are implemented using middlewares:

1. Admin: Full access to quiz creation, management, and user data.
2. User: Can take quizzes and view personal performance history.

## **4.2 Admin Dashboard**

### **4.2.1 Quiz Management**

Admins can:

- Add, delete, or edit topics
- View and filter quizzes based on topic
- Create new quizzes
- Edit existing quizzes or delete them (only if they are the creator)
- Set time limits for quizzes
- Add, edit, or remove questions and answer options (only if they are the creator of the quiz)
- View the performance history for each quiz and filter them based on the month

### **4.2.2 User Management**

Admins can:

- View all registered users
- Search for user

### **4.2.3 Reporting**

The admin dashboard provides various reports, including:

- Overall quiz performance statistics
- Individual user performance metrics
- Most popular quizzes
- Difficulty analysis of questions

## **4.3 Quiz System**

### **4.3.1 Quiz Creation**

Admins can create quizzes by:

- Setting a title, description, topic, and type
- Defining a time limit (if applicable)
- Adding multiple-choice or true/false questions
- Specifying correct answers

#### **4.3.2 Quiz Display**

Users see a list of available quizzes with:

- Title and brief description
- Quiz image (if applicable)
- Time limit label (if applicable)
- Quiz type label (if it's multiple attempts)

#### **4.3.3 Quiz Taking**

When a user starts a quiz:

- Questions are presented all at once
- A countdown timer is displayed (for timed quizzes)

#### **4.3.4 Scoring System**

- Each question has a point value of 1
- The system calculates the total score based on correct answers
- Score is displayed immediately upon completion

### **4.4 User Dashboard**

#### **4.4.1 Performance History**

Users can view:

- A list of all quizzes taken
- Scores for each quiz attempt

#### **4.4.2 Performance Trends**

- A graphical representation of the user's performance over time is displayed, allowing users and admins to spot trends in performance.
- This could include line graphs or bar charts that show how scores have changed month-by-month or quiz-by-quiz.
- Trend Insights:
- Users can gain insights into their learning progress. For instance, they can see if their quiz scores are improving, staying consistent, or declining over time.

## **5. User Guide**

### **5.1 For Admins**

1. Login to the admin dashboard using your admin credentials.
2. Creating a Quiz:
  - Click on "Create New Quiz" in the dashboard.
  - Fill in the quiz details (title, description, time limit).
  - Add questions and their respective answer options.
  - Set the correct answer for each question.
  - Click "Save" to publish the quiz.
3. Managing Users:
  - Navigate to the "User Management" section.
  - View user details, edit information, or deactivate accounts as needed.
4. Viewing Reports:
  - Access the "Reports" section in the dashboard.
  - Select the type of report you want to view (e.g., quiz performance, user statistics).
  - Use filters to refine the data as needed.



- Admins can use these performance trends to assess the effectiveness of quizzes and identify common areas where students may be struggling.

## **5.2 For Quiz Takers**

1. Register for an account or log in if you already have one.
2. Feature: Shuffling Quiz Questions this feature introduces the ability to shuffle the order of quiz questions to provide a unique experience for each user attempting the quiz. The shuffled order will be stored in the database to ensure consistency across multiple quiz sessions for the same user.
3. Shuffling Process When a user begins a quiz, the system automatically shuffles the questions associated with that quiz. The shuffled order is saved. This allows for consistent delivery of the shuffled quiz during the quiz attempt.
4. This ensures that the same order of questions is shown to the user if they return to the quiz later.
5. Taking a Quiz:
  - Navigate to topic list in the navigation bar.
  - Select topic to browse available quizzes.
  - Hover on the quiz to view the type.
  - Click "Start Quiz" when ready.
  - Answer questions within the given time limit (if applicable).
  - Click "Submit" when finished.
6. Viewing Results:
  - After submitting, you'll see your score.
  - Visit your performance history to view your results.

## **6. Development Process**

### **6.1 Week 1: Setup and Quiz Creation**

- Set up the Laravel project and configured the database.

- Implemented the database schema for quizzes, questions, users, and scores.
- Created the admin dashboard with basic CRUD operations for quiz management.
- Developed the quiz creation form with support for multiple-choice and true/false questions.

## **6.2 Week 2: Quiz Display and User Interaction**

- Designed and implemented the basic layout using HTML, CSS, and Bootstrap.
- Created the quiz display functionality with a countdown timer for timed quizzes.
- Implemented user authentication and role-based access control.
- Developed the answer submission system and score calculation logic.

## **6.3 Week 3: Score Tracking and History**

- Implemented the score tracking system to store user scores in the database.
- Created the user dashboard to display quiz history and performance trends.
- Developed admin reporting features for quiz results and participant performance.
- Conducted thorough testing of the score tracking and reporting systems.

## **6.4 Week 4: Final Testing, UI Enhancements, and Deployment**

- Enhanced the user interface with additional CSS styling and JavaScript interactions.
- Conducted comprehensive testing of all system features.
- Finalized the project documentation and user manual.

# **7. Maintenance and Support**

## **7.1 Troubleshooting Common Issues**

- Login Issues: Check database connectivity and user table integrity.
- Slow Quiz Loading: Optimize database queries and implement caching if necessary.
- Score Calculation Errors: Verify the scoring algorithm and check for data consistency.

## 7.2 Updating the System

To update the system:

1. Pull the latest changes from the repository.
2. Install any new dependencies:

```
composer install
```

```
npm install
```

3. Run database migrations:

```
php artisan migrate
```

4. Clear caches:

```
php artisan cache:clear
```

```
php artisan config:clear
```

```
php artisan view:clear
```

5. Recompile assets if necessary:

```
npm run build
```

## 8. Future Enhancements

### 8.1 Planned Features

- Integration with learning management systems (LMS)
- Support for more question types (e.g., short answer, matching)
- Social sharing of quiz results

### 8.2 Scalability Considerations

- Implement caching mechanisms for frequently accessed data
- Consider moving to a cloud-based infrastructure for easier scaling
- Optimize database queries and indexes for large datasets

## 9. Conclusion

The Online Quiz System provides a robust platform for creating, managing, and taking quizzes. With its user-friendly interface and comprehensive feature set, it meets the needs of both quiz administrators and participants. As the system evolves, the planned enhancements will further improve its functionality and user experience.

## Appendices

### A. Glossary of Terms

- Quiz: A set of questions on a particular topic
- User: A person who takes quizzes on the platform
- Admin: A user with privileges to create and manage quizzes
- Score: The number of points earned by a user for correct answers in a quiz
- Time Limit: The maximum duration allowed for completing a quiz

### B. Database Diagram

