



Quiz Application API Endpoints Summary

Name: Hazrat Hanzala

Student ID: B00957159

Title: API Endpoints Summary

(Quiz Management System)

Overview

The Quiz Application is a comprehensive RESTful API built for managing online quizzes with role-based access control. The system supports multiple user roles (users and admins) and provides complete CRUD functionality for quiz management, user attempts tracking, and leaderboard features. Built with Flask and utilizing JWT token-based authentication, the platform ensures secure access to resources while maintaining scalability and modularity. The API separates concerns between guest

access, authenticated users, and administrative operations, making it suitable for educational platforms and quiz-based applications

1. Authentication Endpoints (/auth)

The authentication module handles user identity management, registration, and session control through token-based authentication.

POST /register

Allows new users to create an account by providing their details. The system assigns appropriate roles during registration.

Request Body:

```
{ "name": "John Doe", "email": "john@example.com", "password": "password123",  
  "role": "user" }
```

Headers:

Content-Type: application/json

Response Codes:

201 – User account created successfully

400 – Missing required fields or invalid data

409 – Email already exists in the system

POST /auth/login

Authenticates registered users and issues a JWT token for accessing protected routes. Uses Basic Authentication for credential submission.

Authentication:

Basic Auth Username:

user's email

Password: user's password

Response Codes:

- 200 – Authentication successful, token returned
- 401 – Invalid credentials or user not found

Response:

Returns a JWT token to be used as x-access-token in subsequent requests.

POST /logout

Invalidates the current user session by blacklisting the active token.

Headers:

- x-access-token: {{user_token}}

Response Codes:

- 200 – Successfully logged out
 - 401 – Token missing or invalid
-

2. User Management Endpoints

These endpoints allow users to manage their profiles, view their quiz attempt history, and handle account operations.

GET /profile

Retrieves the authenticated user's profile information including name, email, and role.

Headers:

- x-access-token: {{user_token}}

Response Codes:

- 200 – Profile data retrieved successfully
 - 401 – Authentication required
 - 404 – User profile not found
-

GET /USR-201/attempts

Fetches all quiz attempts made by a specific user. Useful for tracking performance history and progress.

Headers:

- x-access-token: {{user_token}}

Response Codes:

- 200 – Attempts retrieved successfully
 - 401 – Unauthorized access
 - 404 – User not found or no attempts available
-

DELETE /delete

Allows users to permanently delete their own account from the system.

Headers:

- x-access-token: {{user_token}}

Response Codes:

- 200 – Account deleted successfully
 - 401 – Authentication required
 - 403 – Insufficient permissions
-

PUT /update-role (Admin Only)

Enables administrators to modify user roles (e.g., promoting a user to admin).

Request Body: { "email": "john@example.com", "role": "admin" }

Headers:

- Content-Type: application/json
- x-access-token: {{admin_token}}

Response Codes:

- 200 – Role updated successfully
- 400 – Invalid role or missing fields
- 401 – Admin authentication required
- 403 – Insufficient permissions
- 404 – User not found

3. Guest Access Endpoints (Public)

These endpoints are publicly accessible without authentication, allowing visitors to browse available quizzes and view public leaderboards.

GET /public

Returns a list of all publicly available quizzes that don't require authentication to view.

Response Codes:

- 200 – Public quizzes retrieved successfully
- 500 – Server error

GET /QZ-001/preview

Provides a preview of a specific quiz including its title, difficulty level, and question structure (without revealing correct answers).

Response Codes:

- 200 – Preview retrieved successfully
- 404 – Quiz not found or not public

GET /QZ-001/leaderboard

Displays the public leaderboard for a specific quiz showing top performers and their scores.

Response Codes:

- 200 – Leaderboard retrieved successfully
- 404 – Quiz not found or leaderboard unavailable

4. User Quiz Operations

Authenticated users can interact with quizzes through these endpoints, including viewing, attempting, and tracking their performance.

GET /?pn=1&ps=10

Retrieves all available quizzes with pagination support for efficient data loading.

Headers:

- x-access-token: {{user_token}}

Query Parameters:

- pn – Page number (default: 1)
- ps – Page size (default: 10)

Response Codes:

- 200 – Quizzes retrieved successfully
- 401 – Authentication required
- 500 – Server error

GET /QZ-001

Fetches detailed information about a specific quiz including all questions, options, and metadata.

Headers:

- x-access-token: {{user_token}}

Response Codes:

- 200 – Quiz details retrieved successfully
 - 401 – Authentication required
 - 404 – Quiz not found
-

POST /QZ-001/attempts

Submits a quiz attempt with user answers for evaluation. The system automatically scores the attempt based on correct answers.

Request Body: { "userId": "USR-201", "name": "Emma Williams", "email": "emma.williams@yahoo.com", "answers": ["HTML", "Alexander Graham Bell", "Pacific"] }

Headers:

- Content-Type: application/json
- x-access-token: {{user_token}}

Response Codes:

- 201 – Attempt submitted and scored successfully
 - 400 – Invalid answers format or missing fields
 - 401 – Authentication required
 - 404 – Quiz not found
-

5. Admin Operations

Administrative endpoints provide full control over quiz management, including creation, modification, deletion, and monitoring of all platform activities.

POST /admin/quizzes

Creates a new quiz with questions, options, and correct answers. Only accessible to administrators.

Request Body:

```
{ "title": "Python Programming Quiz", "difficulty": "Medium", "creator": { "name": "Admin User", "email": "admin@quizapp.com" }, "questions": [ { "questionId": "Q1", "text": "What is Python?", "options": ["Snake", "Programming Language", "Bird", "Car"], "correct_answer": "Programming Language", "marks": 5 } ] }
```

Headers:

- Content-Type: application/json
- x-access-token: {{admin_token}}

Response Codes:

- 201 – Quiz created successfully
 - 400 – Invalid quiz structure or missing fields
 - 401 – Admin authentication required
 - 403 – Insufficient permissions
-

GET /admin/quizzes

Retrieves all quizzes in the system regardless of creator or visibility status. Provides admins with a complete overview.

Headers:

- x-access-token: {{admin_token}}

Response Codes:

- 200 – All quizzes retrieved successfully
 - 401 – Admin authentication required
 - 403 – Insufficient permissions
-

GET /admin/quizzes/QZ-001

Fetches complete details of a specific quiz including all metadata, questions, and statistics.

Headers:

- x-access-token: {{admin_token}}

Response Codes:

- 200 – Quiz details retrieved successful
 - y • 401 – Admin authentication required
 - 403 – Insufficient permissions
 - 404 – Quiz not found
-

PUT /admin/quizzes/QZ-001

Updates quiz information such as title, difficulty, questions, or any other metadata.

Request Body: { "title": "Updated Quiz Title", "difficulty": "Hard" }

Headers:

- Content-Type: application/json
- x-access-token: {{admin_token}}

Response Codes:

- 200 – Quiz updated successfully
 - 400 – Invalid update data
 - 401 – Admin authentication required
 - 403 – Insufficient permissions • 404 – Quiz not found
-

DELETE /admin/quizzes/QZ-001

Permanently removes a quiz from the system including all associated attempts and data.

Headers:

- x-access-token: {{admin_token}}

Response Codes:

- 200 – Quiz deleted successfully
 - 401 – Admin authentication required
 - 403 – Insufficient permissions
 - 404 – Quiz not found
-

GET /admin/quizzes/QZ-001/attempts

Retrieves all attempts made on a specific quiz by all users. Useful for monitoring quiz performance and user engagement.

Headers:

- x-access-token: {{admin_token}}

Response Codes:

- 200 – Attempts retrieved successfully
 - 401 – Admin authentication required
 - 403 – Insufficient permissions
 - 404 – Quiz not found
-

GET /admin/quizzes/QZ-001/leaderboard

Displays the complete leaderboard for a specific quiz showing all participants and their scores.

Headers:

- x-access-token: {{admin_token}}

Response Codes:

- 200 – Leaderboard retrieved successfully
 - 401 – Admin authentication required
 - 403 – Insufficient permissions
 - 404 – Quiz not found
-

GET /admin/stats

Provides comprehensive platform statistics including total users, quizzes, attempts, and performance metrics.

Headers:

- x-access-token: {{admin_token}}

Response Codes:

- 200 – Statistics retrieved successfully
 - 401 – Admin authentication required
 - 403 – Insufficient permissions
 - 500 – Server error
-

Security & Authentication

The Quiz Application implements robust security measures:

- **JWT Token-Based Authentication:** All protected routes require a valid JWT token passed via the x-access-token header
 - **Role-Based Access Control (RBAC):** Separate permissions for users and administrators
 - **Password Security:** User passwords are hashed before storage
 - **Token Blacklisting:** Logout functionality invalidates tokens to prevent reuse
 - **Public Access Control:** Guest endpoints are clearly separated from authenticated routes
-

Pagination Standards

Endpoints that return lists of data support pagination using the following query parameters:

- **pn (Page Number)** – Specifies which page to retrieve (default: 1)
- **ps (Page Size)** – Defines the number of items per page (default: 10)

Example: `/?pn=2&ps=20` retrieves the second page with 20 items per page.

Overall Summary

The Quiz Application API delivers a complete and secure backend solution for managing online quizzes with sophisticated user management and role-based access control. By implementing RESTful principles, JWT authentication, and comprehensive CRUD operations across all entities (Users, Quizzes, Attempts, Leaderboards), the system ensures both security and scalability. The clear separation between guest access, user operations, and administrative functions makes it ideal for educational platforms, training systems, and competitive quiz applications. The API's modular design allows for easy integration with frontend applications while maintaining data integrity and user privacy throughout the entire quiz lifecycle.

