# Assessment 02 – Quiz Application Backend

Build a **backend-only** quiz application in **Spring Boot**.  
No frontend is needed — only REST APIs with **JWT auth, CRUD, pagination, and filtering**.

**Time Limit: 2 hours**

**Functional Requirements**

**1. Authentication (JWT)**

* **POST /auth/register → Register (username, password, role: USER/ADMIN).**
* **POST /auth/login → Login and return JWT token.**

**2. Quiz Management (Admin only)**

* **POST /quizzes → Create a quiz (title, category, difficulty).**
* **POST /quizzes/{quizId}/questions → Add question(s) to a quiz.**

**3. Quiz Access (User)**

* **GET /quizzes?page=0&size=5&category=Programming&difficulty=EASY**
  + **Supports pagination & simple filtering.**
* **GET /quizzes/{quizId} → Get quiz details with questions.**

**4. Quiz Attempt & Results (User)**

* **POST /quizzes/{quizId}/attempt → Submit answers, backend evaluates and returns score.**
* **GET /results/{attemptId} → Get quiz attempt result.**

**Data Model (Simplified)**

* **User: id, username, password, role**
* **Quiz: id, title, category, difficulty, createdDate**
* **Question: id, quizId, text, options[], correctAnswerIndex**
* **Attempt: id, userId, quizId, score, submittedAt**

**Technical Requirements**

* **Spring Boot 3+, Spring Security (JWT), Spring Data JPA, H2 DB**
* **REST APIs return JSON**
* **Use Pageable for pagination**
* **Use JWT in all secured APIs**

**Deliverables**

* **A working Spring Boot app exposing the above APIs.**
* **JWT-secured endpoints.**
* **H2 in-memory database (no external DB setup required).**