**ITO Online Exam Portal Documentation**

**1**. Project Overview

* Title: ITO Online Exam Portal
* Description: This portal aims to streamline the assessment process by providing an online platform for creating and evaluating exams.
* Purpose: To facilitate a smooth and efficient examination process for both administrators and candidates.
* Technologies Used: Spring Boot, Java, Maven, JPA, REST API, SQL

**Setup Instructions**

* **Prerequisites**:
  + JDK 17+
  + Maven
  + IDE (e.g., IntelliJ IDEA)
  + MySQL Workbench

**Running the Project**:

* Start the application: mvn spring-boot: run

**Configuration**

* **Application Properties**:
  + src/main/resources/application.properties

**properties**

spring.application.name=ITO\_Quiz  
#Database Configuration  
spring.datasource.url=jdbc:mysql://localhost:3306/itoquiz  
spring.datasource.username=root  
spring.datasource.password=root  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.jpa.hibernate.ddl-auto=update

**Package Organization**:

com.example

├── controller

├── entity

├── repository

├─ service

**Key Classes and Interfaces**:

 AdminController: Handles CRUD operations for questions and evaluation.

 AssessmentController: Manages candidate assessment processes.

 CandidateController: Manages candidate operations.

 QuestionService: Business logic for questions.

 CandidateService: Business logic for candidates.

 EvaluationService: Business logic for evaluating candidates.

 QuestionRepository: JPA repository for questions.

 AnswerRepository: JPA repository for answers.

 CandidateRepository: JPA repository for candidates.

 Question: Entity representing a question.

 Answer: Entity representing an answer.

 Candidate: Entity representing a candidate.

**Endpoints and APIs**

** Admin Module:**

* Create Question: POST /api/admin/question
* Read Questions List: GET /api/admin/question
* Read Question by ID: GET /api/admin/question/{id}
* Update Question: PUT /api/admin/question/{id}
* Delete Question: DELETE /api/admin/question/{id}
* Evaluate: POST /api/admin/evaluate/{candidateId}
* Submit Answer Sheet: POST /api/admin/submitAnswerSheet

** User Module:**

* Generate Exam ID: POST /api/candidates
* Get Question Set: GET /api/assessment/questions/{candidateId}
* Submit Answer Sheet: POST /api/assessment/submit

**Database Schema**

* **Entity Relationship Diagram (ERD):**
  + Tables: Question, Answer, Candidate
* **Schemas:**
  + Question: /itoquiz/question/{question\_id}
  + Answer: /itoquiz/answer/{id}
  + Candidate: /itoquiz/candidate/{candidate\_id}

**POSTMAN Test**

[**http://localhost:8080/api/admin/questions**](http://localhost:8080/api/admin/questions) **(POST)**

**A screenshot of a computer

Description automatically generated**

[**http://localhost:8080/api/admin/question**](http://localhost:8080/api/admin/question) **(POST)**

A screenshot of a computer

Description automatically generated

[**http://localhost:8080/api/admin/question(GET)**](http://localhost:8080/api/admin/question(GET))

A screenshot of a computer

Description automatically generated

[**http://localhost:8080/api/admin/question/2(GET)**](http://localhost:8080/api/admin/question/2(GET))

**A screenshot of a computer

Description automatically generated**

[**http://localhost:8080/api/admin/question/6**](http://localhost:8080/api/admin/question/6) **(PUT)**

A screenshot of a computer

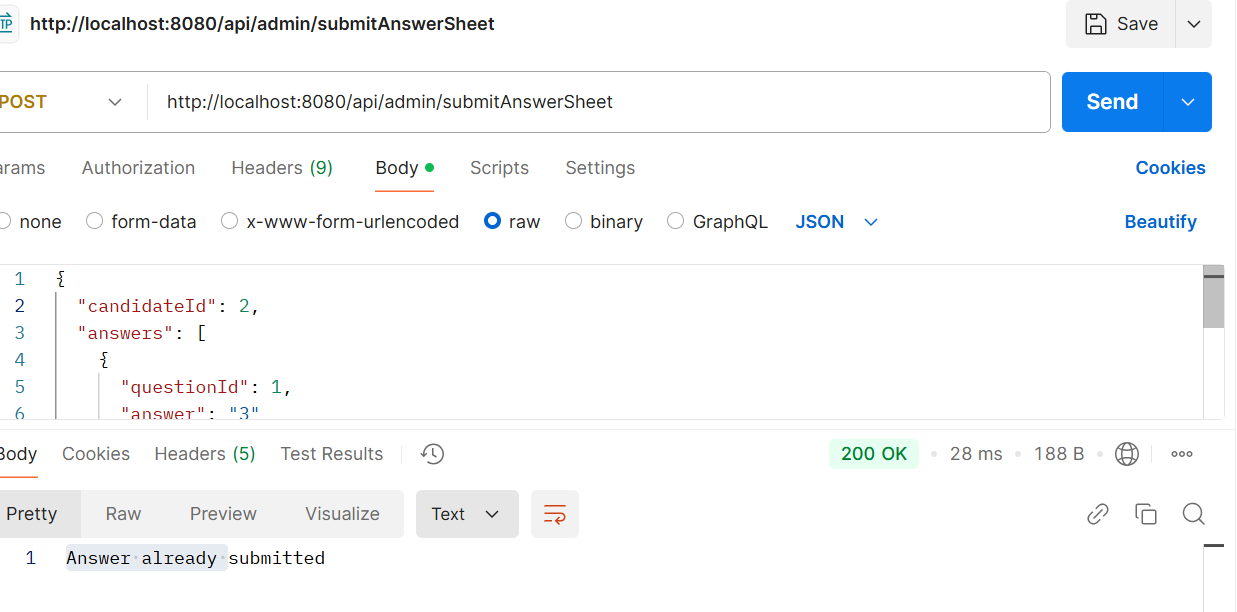
Description automatically generated

[**http://localhost:8080/api/admin/question/18(DELETE)**](http://localhost:8080/api/admin/question/18(DELETE))

A screenshot of a computer

Description automatically generated

[**http://localhost:8080/api/admin/submitAnswerSheet(POST)**](http://localhost:8080/api/admin/submitAnswerSheet(POST))

****

[http://localhost:8080/api/admin/ evaluate/1(POST)A screenshot of a computer

Description automatically generated](http://localhost:8080/api/admin/evaluate/1(POST))

<http://localhost:8080/api/candidates(POST)>

A screenshot of a computer

Description automatically generated

<http://localhost:8080/api/candidates(GET)>

A screenshot of a computer

Description automatically generated

<http://localhost:8080/api/candidates/2(GET)>

A screenshot of a computer

Description automatically generated

<http://localhost:8080/api/candidates/2(PUT)>

http://localhost:8080/api/candidates/2A screenshot of a computer

Description automatically generated

<http://localhost:8080/api/candidates/2(DELETE)>

A screenshot of a computer

Description automatically generated

<http://localhost:8080/api/assessment/questions/1(GET)>

A screenshot of a computer

Description automatically generated