Quiz Application API Documentation

1. Project Overview

This document provides an overview of the Quiz Application API built using Flask and MySQL, containerized for Kubernetes (MicroK8s). The application supports user registration, login, quiz creation, question management, and answer submission. It includes automated scripts for deployment, testing, and cleanup on a local Kubernetes environment.

2. API Codebase Structure

The codebase is structured for modularity and maintainability. Major directories include: - **api/** — Contains Flask backend source code (routes, models, controllers, and utils). - **k8s/** — Contains Kubernetes manifests, Helm chart templates, and deployment scripts. - **scripts/** — Contains automation scripts like deploy_api.sh, undeploy.sh, and api_test.sh. - **requirements.txt** — Lists all Python dependencies required for the API. - **Dockerfile** — Defines container image for the Flask backend.

3. Features and Functionality

Feature	Description
User Management	User registration and authentication using MySQL database.
Quiz Management	Admin users can create, update, and delete quizzes.
Question Handling	Supports dynamic question creation, retrieval, and answer submission.
Result Evaluation	Automatically calculates scores and provides feedback.
API Integration	Endpoints structured with RESTful design for easy frontend integration.

4. Deployment & Running

To set up and run the Quiz Application API locally: 1. Clone the repository and navigate into it. 2. Run **deploy_api.sh** with root or sudo permissions to deploy backend and MySQL on MicroK8s. 3. Wait for pods to reach the **Running** state. 4. Run **./api_test.sh** to verify all endpoints and functionality. 5. To clean up, execute **./undeploy.sh**.

5. API Endpoints Summary

Method	Endpoint	Description
POST	/api/register	Register a new user.
POST	/api/login	Authenticate a user.
GET	/api/quizzes	Fetch list of available quizzes.
POST	/api/quizzes	Create a new quiz (admin only).
GET	/api/questions/<quiz_id>	Retrieve questions for a quiz.
POST	/api/submit	Submit answers and get results.