**Quiz App using django**

**Description**

This project is a Django application designed for multi-level authentication security using JSON Web Tokens (JWT). It features role-based logins to provide granular access control within the system. The application leverages the Django framework for backend development, incorporating Python for logic implementation. It utilizes React for frontend development, ensuring a dynamic and interactive user interface. MySQL serves as the database management system to store and manage application data efficiently. JavaScript is employed for frontend scripting, enhancing user experience and interactivity.

**Technology Stack**

* Python
* Django
* React
* MySQL
* JavaScript

**Installation**

1. Clone the repository:

git clone https://github.com/Narendra-lella/quiz-app.git

1. Install dependencies:

pip install -r requirements.txt

**Docker Setup**

1. Install Docker Desktop from Docker's official website.
2. Build the Docker image:

docker build -t quiz-app .

1. Run the Docker container:

docker run -p 8000:8000 quiz-app

1. Access the application in your web browser at **http://localhost:8000/**.

**Usage**

1. Run the development server:

python manage.py runserver

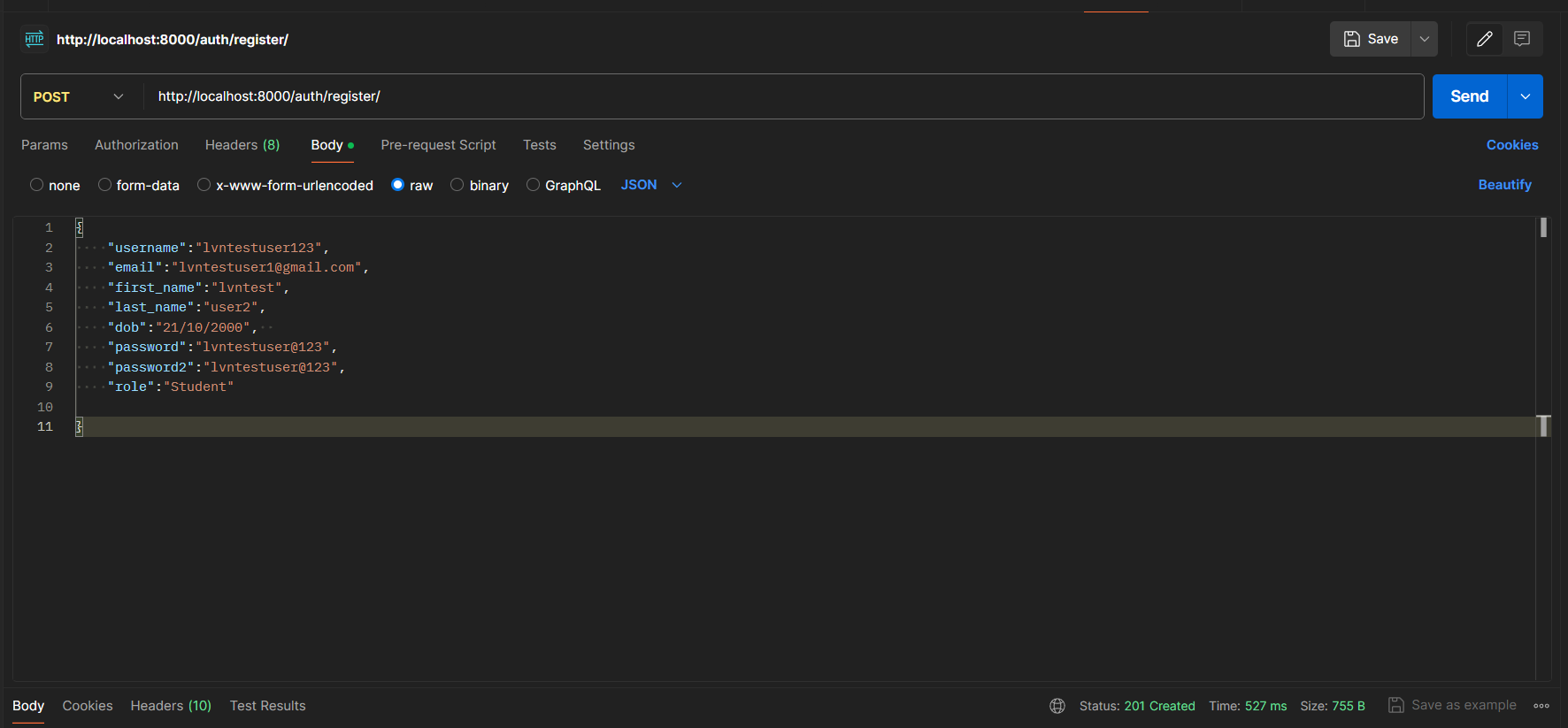
1. Access the application in your web browser at **http://localhost:8000/**.

**Features**

* Admin or manger can add the questions
* Teacher or Head of the Department can access all the questions listed and can modify
* Students can get the question to take the test

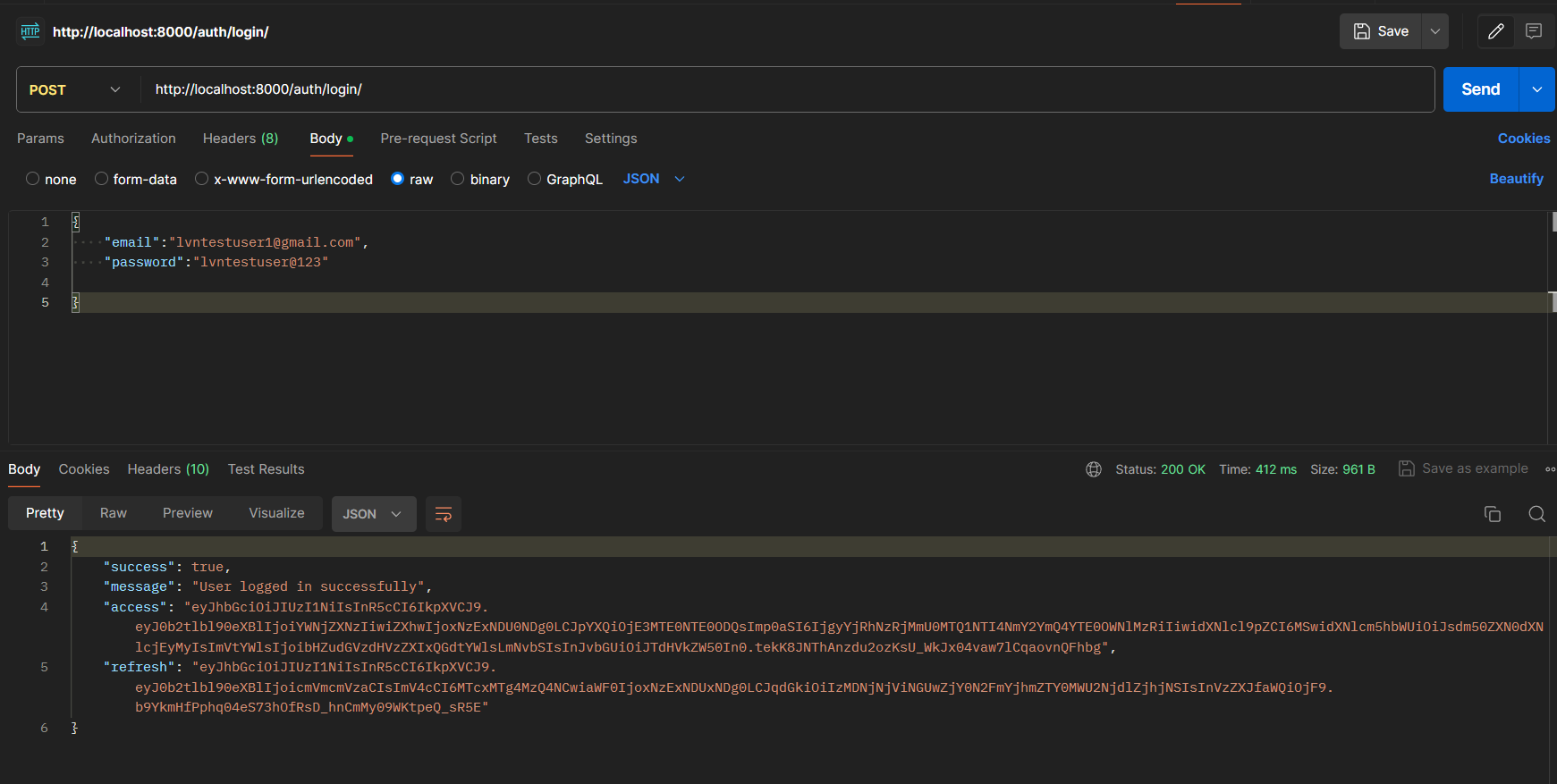
**End Points**

* ../auth/register/ -- POST

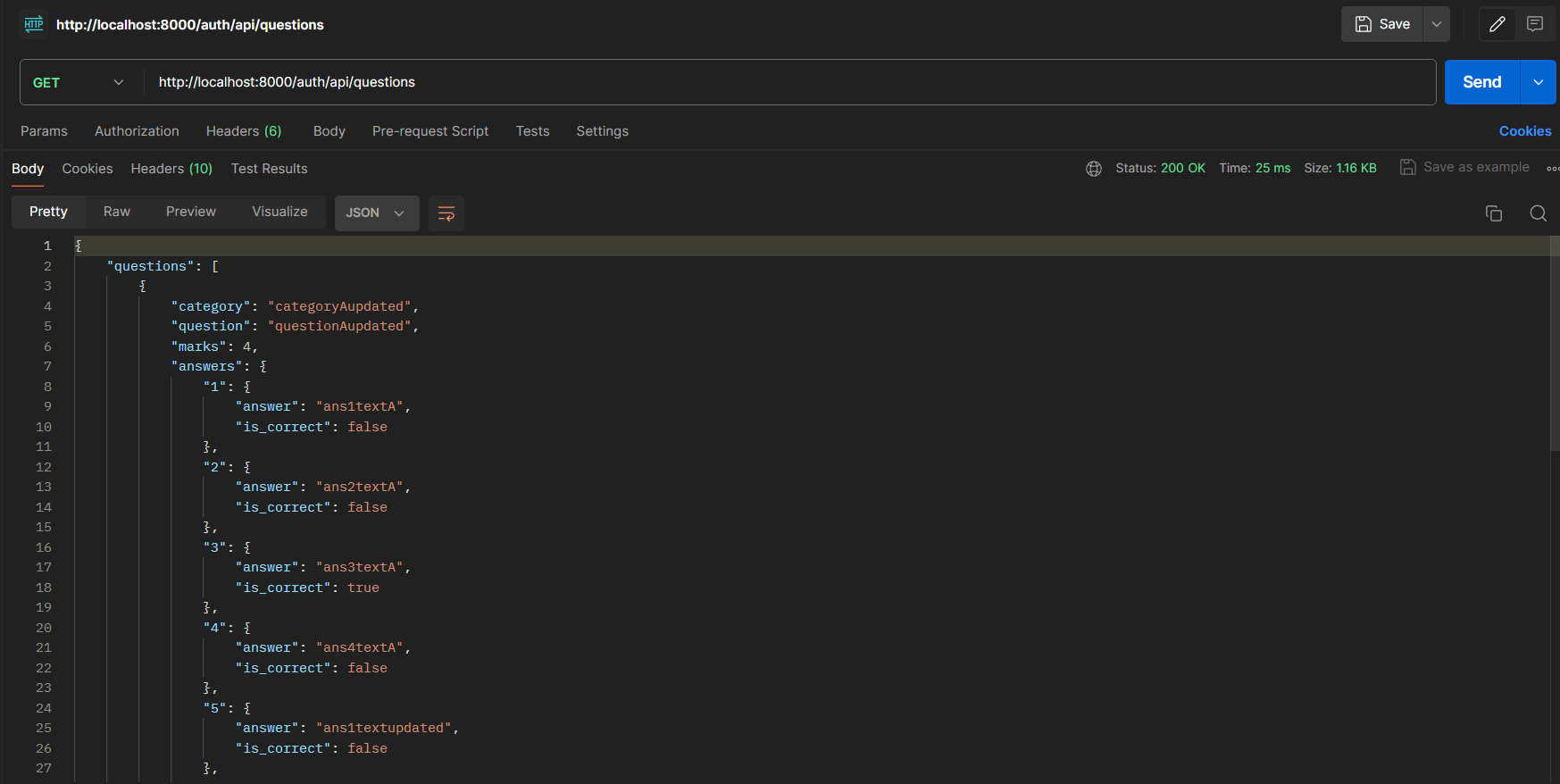


../auth/login – POST

Here you will get the access token and refresh token for authentication

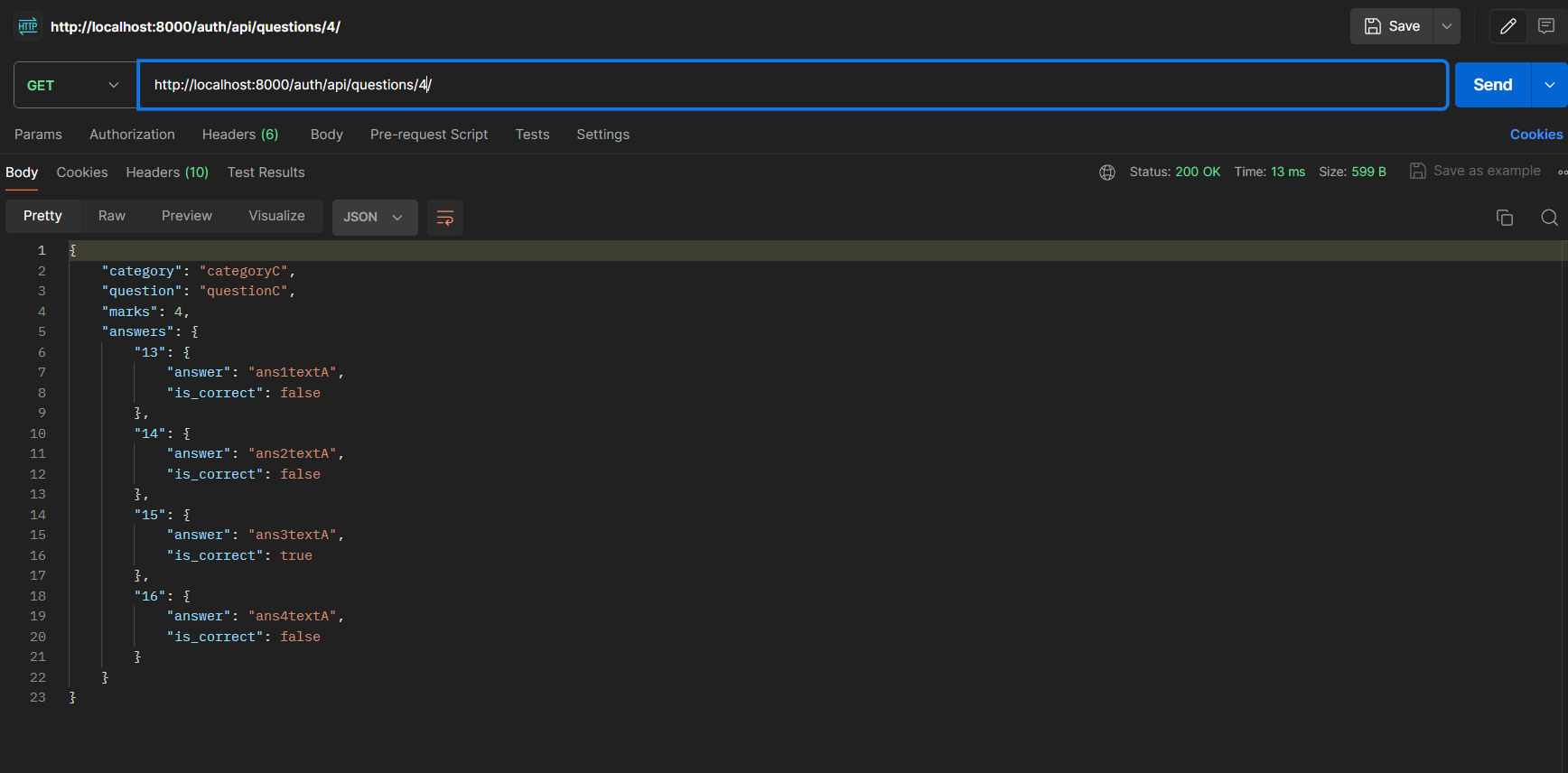


..api/auth/questions/ -- GET

Get list of all questions  


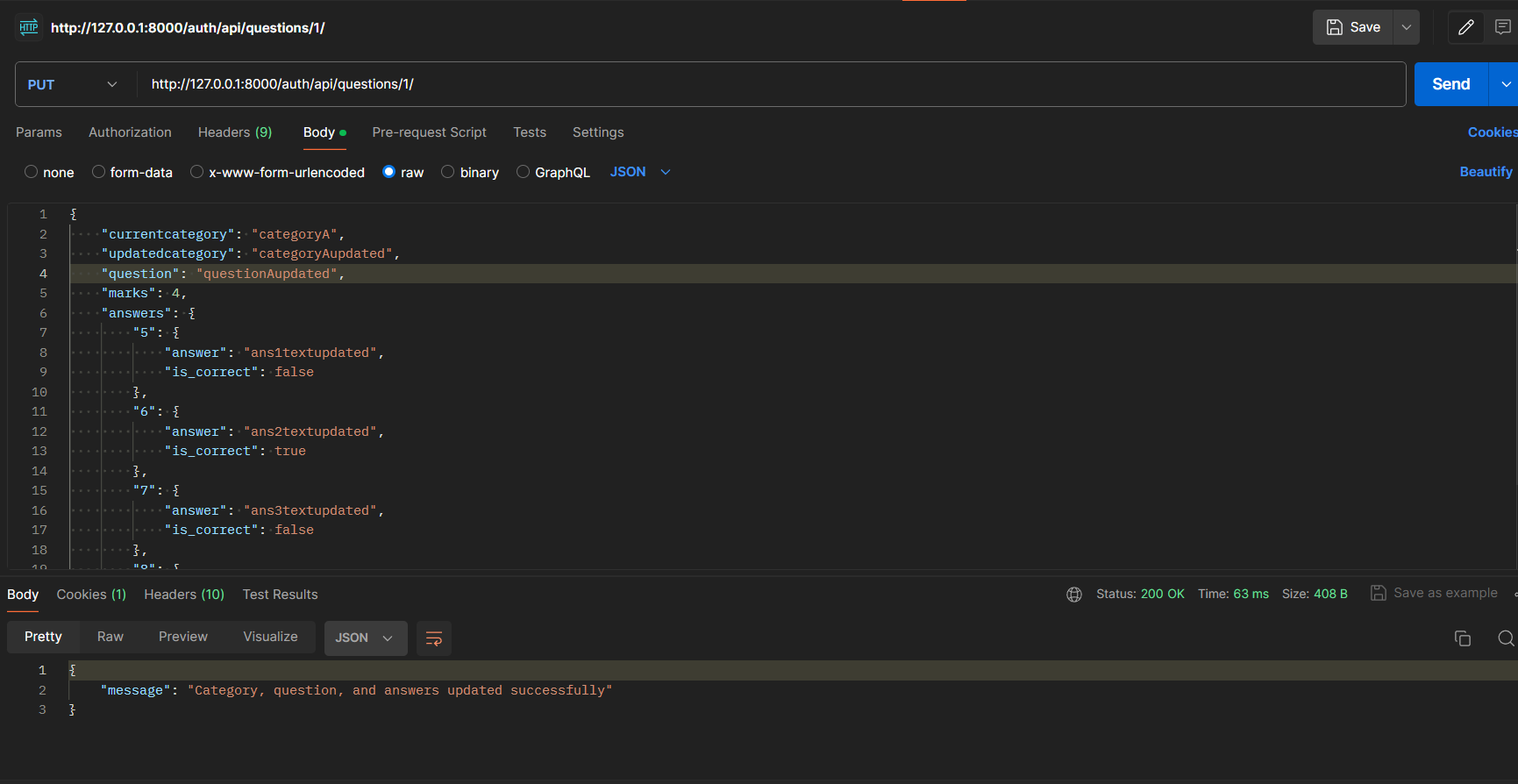
..api/auth/questions/{question\_id} -- GET

Get particular question



..api/auth/question/{question\_id} -- PUT

Update the questions



..api/auth/delete/question -- DELETE

