AI Quizzer API - README

This document provides the steps to set up, run, and test the AI Quizzer API. The API allows users to register, log in, create quizzes, generate questions, submit answers, and retrieve hints using AI-powered content generation.

Prerequisites

Before you begin, ensure you have the following installed:

- Python 3.8 or higher
- Django 3.x or higher
- Django REST Framework
- djangorestframework-simplejwt
- google-generativeai
- PostgreSQL or another database setup
- Postman (for API testing)

Setup Steps

1. Clone the Project

Clone the repository to your local machine:

```
bash
Copy code
git clone <repository_url>
```

2. Install Dependencies

Navigate to your project directory and install the required packages from the `requirements.txt` file:

```
bash
Copy code
cd <project_directory>
pip install -r requirements.txt
```

3. Setup Database

Make sure to configure your database settings in the `settings.py` file. If you are using PostgreSQL, the default database configurations will work, but you may need to adjust accordingly if you're using a different database.

4. Apply Migrations

Run the migrations to set up the database schema:

```
bash
Copy code
python manage.py migrate
```

5. Create a Superuser (Optional)

If you want to access the Django Admin interface, create a superuser:

```
bash
Copy code
python manage.py createsuperuser
```

Follow the prompts to create the admin account.

API Endpoints

1. User Registration (Sign Up)

Endpoint: `POST /api/register/`

Request Body:

```
json
Copy code
{
    "username": "user_name",
    "password": "password"
}
```

Response:

```
json
Copy code
{
    "message": "User created successfully",
    "username": "user_name"
}
```

2. User Login (JWT Token Generation)

Endpoint: `POST /api/login/`

Request Body:

```
json
Copy code
{
    "username": "user_name",
    "password": "password"
}
```

Response:

```
json
Copy code
{
    "token": "<access_token>",
    "refresh": "<refresh_token>"
}
```

---

3. Create a Quiz (Admin/Authenticated User)

Endpoint: `POST /api/create_quiz/`

Request Body:

```
json
Copy code
{
    "subject": "General Knowledge",
    "quiz_title": "Sample Quiz",
    "duration": 30
}
```

Response:

```
json
Copy code
{
    "message": "Quiz created successfully",
    "quiz": {
        "subject": "General Knowledge",
        "quiz_title": "Sample Quiz",
        "duration": 30
    }
}
```

---

4. Generate Hint for a Question

Endpoint: `POST /api/generate_hint/`

Request Body:

```
json
Copy code
{
    "question_text": "What is the capital of France?"
}
```

Response:

```
json
```

```
Copy code
{
    "hint": "The capital of France is known for the Eiffel Tower."
}
```

Testing the APIs in Postman

1. **Open Postman.**
2. **Create a New Request**:
   - Choose `POST` as the HTTP method.
   - Enter the API endpoint (e.g., `http://localhost:8000/api/register/`).
   - In the "Body" tab, select `raw` and `JSON` format.
   - Add the request data in JSON format (refer to each endpoint for the expected data).
3. **Authentication**:
   - For endpoints that require authentication (like `Create Quiz`, `Submit Quiz`), use the token obtained from the Login endpoint:
     - In Postman, go to the "Authorization" tab.
     - Select "Bearer Token" and paste the JWT access token.
4. **Send the Request**:
   - Click on the "Send" button.
   - View the response from the API in the "Response" section.
5. **Check Responses**:
   - Verify that the responses are as expected (check status codes, messages, and response data).

Additional Notes

- **Caching**: The API uses caching to store quizzes for 1 hour after creation to optimize performance.
- **Error Handling**: If there are any issues such as missing fields or invalid credentials, the API will return an appropriate error message.

Final Steps

Once you’ve tested all the endpoints in Postman, your API should be ready for use in production.

If you face any issues during setup or testing, please refer to the error messages in the responses, and feel free to ask for assistance.