**AI QGen Project**

This project is a full-stack application that includes a Python backend and a React frontend. The backend uses FastAPI to serve various endpoints, and the frontend is built with React to provide a user interface for interacting with the backend.

**Setup Frontend:**

1. Navigate to the react-app directory:

cd react-frontend

1. Install the required dependencies:

npm install

1. Start the React development server:

npm start

**About Backend**

The backend is in the **python-backend** directory and uses FastAPI to serve various endpoints. The main entry point is server.py, which includes routers for different functionalities:

1. **askquestion.py**: Endpoint for generating interview questions.
2. **descriptive.p**y: Endpoint for generating and submitting
3. **mcq.py**: Endpoint for generating and submitting multiple-choice questions.
4. **rag.py**: Endpoint for uploading and querying documents.
5. **chat.py:** Endpoint for chat functionality descriptive questions.

**Steps to Setup and run the application:**

1. Navigate to the python-backend directory:

cd python-backend

Code .

1. **Add .env to project**

OPENAI\_API\_KEY=TO-BE-GENERATED-BY-UTIL

TIME\_GENERATED=2025-01-13 16:16:11.547398

TOKEN\_URL=https://sso-dev.johndeere.com/oauth2/ausx8znnnrQpD7x0f0h7/v1/token

CLIENT\_ID=<CLIENT ID GOES HERE>

CLIENT\_SECRET=<CLIENT SECRET GOES HERE>

SCOPE=mlops.deere.com/model-deployments.llm.region-restricted-invocations

OPENAI\_BASE\_URL=https://ai-gateway.deere.com/openai

GATEWAY\_HEADER\_NAME=deere-ai-gateway-registration-id

GATEWAY\_HEADER\_VALUE=vilt-jan-27-batch-training

1. Create a virtual environment:

python -m venv venv

1. Activate the virtual environment from Terminal of VS Code (Ctrl + ~):

On Windows: **venv\Scripts\activate**

1. Install the required dependencies:

pip install -r requirements.txt

1. Run the FastAPI server:

python server.py

**Backend Setup for NodeJS**

The backend is in the **nodejs-backend** directory and uses **Express.js** to serve various endpoints. The main entry point is server.js, which includes routers for different functionalities in folder **routes**:

1. **askquestion.js**: Endpoint for generating interview questions.
2. **descriptive.js**: Endpoint for generating and submitting
3. **mcq.js**: Endpoint for generating and submitting multiple-choice questions.
4. **rag.js**: Endpoint for uploading and querying documents.
5. **chat.js:** Endpoint for chat functionality descriptive questions.

**Steps to Setup and Run the Application:**

1. cd nodejs-backend
2. npm install
3. code .
4. **Add .env to project**

OPENAI\_API\_KEY=TO-BE-GENERATED-BY-UTIL

TIME\_GENERATED=2024-12-30T12:47:15.150Z

TOKEN\_URL=https://sso-dev.johndeere.com/oauth2/ausx8znnnrQpD7x0f0h7/v1/token

CLIENT\_ID=<CLIENT ID GOES HERE>

CLIENT\_SECRET=<CLIENT SECRET GOES HERE>

SCOPE=mlops.deere.com/model-deployments.llm.region-restricted-invocations

OPENAI\_BASE\_URL=https://ai-gateway.deere.com/openai

GATEWAY\_HEADER\_NAME=deere-ai-gateway-registration-id

GATEWAY\_HEADER\_VALUE=vilt-jan-27-batch-training