**Query Management System**

**Overview**

The Query Management System is a full-stack application designed to handle and manage queries efficiently. It comprises a backend built with Django and a frontend developed using React. The backend provides a RESTful API for managing queries, while the frontend offers a user-friendly interface for interacting with the system.

**Features**

* User authentication and authorization
* Query management (create, read, update)
* Custom exception handling
* CORS support for frontend-backend communication

**Technologies**

* Backend: Django, Django REST Framework, MySQL
* Frontend: React, CSS/SCSS
* Containerization: Docker

**Prerequisites**

* Python 3.9
* Node.js and npm
* Docker and Docker Compose
* MySQL

**Backend Setup**

**Configuration**

The backend configuration is managed in the query\_management/settings.py file. Ensure you have the correct database configurations:

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.mysql',

        'NAME': 'query\_management',

        'USER': 'queryMgtUser',

        'PASSWORD': 'queryMgtPass01#',

        'HOST': 'localhost',

        'PORT': '3306',

    }

}

**Database Setup**

To create the database and user in MySQL, use the following script:

CREATE DATABASE query\_management;

CREATE USER 'queryMgtUser'@'%' IDENTIFIED BY 'queryMgtPass01#';

GRANT ALL PRIVILEGES ON query\_management.\* TO 'queryMgtUser'@'%'*;*

FLUSH PRIVILEGES;

You can execute this script in the MySQL shell:

*mysql -u root -p*

Then paste the script and execute it.

**Installation**

1. **Clone the repository**:

git clone https://github.com/mazinakundai/query\_management.git

cd query-management

1. **Create a virtual environment and activate it**:

python3 -m venv env

source env/bin/activate

1. **Install dependencies**:

pip install -r requirements.txt

1. **Run database migrations**:

python manage.py migrate

1. **Create a superuser**:

python manage.py createsuperuser

1. **Run the development server**:

python manage.py runserver

**Frontend Setup**

**Installation**

1. **Navigate to the frontend directory**:

cd query\_management\_frontend

cd queries

1. **Install dependencies**:

npm install

1. **Run the development server**:

npm start

**Building and Running with Docker**

Ensure you have the correct database configurations in settings.py:

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': os.environ.get('DATABASE\_NAME'),

'USER': os.environ.get('DATABASE\_USER'),

'PASSWORD': os.environ.get('DATABASE\_PASSWORD'),

'HOST': os.environ.get('DATABASE\_HOST'),

'PORT': os.environ.get('DATABASE\_PORT', '3306'),

}

}

1. **Navigate to the project root**:
2. **Build the Docker images**:

Docker-compose build

1. **Run the Docker containers**:

docker-compose up

**Running the Application**

1. **Ensure the backend is running** on http://localhost:8000.
2. **Ensure the frontend is running** on http://localhost:3000.

You should now be able to access the Query Management System at http://localhost:3000.

**License**

No licencing requires

**Contributing**

Contributions are welcome! Please open an issue or submit a pull request.