Random Quiz Generator Documentation

Table of Contents

1. Introduction

This documentation guides you through the setup and usage of a Random Quiz Generator built with Django and React.

2. Prerequisites

Ensure you have the following prerequisites installed:

Python (3.8 or higher)

Django (3.2)

Node.js

npm (Node Package Manager)

3. Installation

Install the Django

pip install django==3.2

Create Django Project (Backend)

```
django-admin startproject Simplequizsystem.
```

After install Django let's create the django Project

Run the Project

python manage.py runserver

4. Configuration

Install MySQL:

pip install pymysql

Upgrade:

pip install pymysql –upgrade

Create a virtual environment:

python -m veny Env

Activate:

cd Env\Scripts
activate
cd ../..

Install Django in the virtual environment:

pip install django==3.2

5. Usage

Angular Connected Port

Add the following CORS settings to settings.py":

python

CORS_ALLOWED_ORIGINS = [

"http://localhost:3000",
]

Create the Model
python

from django.db import models

class Exam (models.Model):
Question = models.CharField(max_length=100)
Option1 = models.CharField(max_length=100)
Option2 = models.CharField(max_length=100)
Option3 = models.CharField(max_length=100)
Option4 = models.CharField(max_length=100)

6. API Endpoints

The following API endpoint is used in this Random Quiz Generator:

Base URL: http://localhost:8000

7. Frontend Components
React (Front-end)

Installing React

npx create-react-app front-end

After installation, run the project:

npm start

Open the React project in your preferred code editor. Adding Bootstrap Styles

Add the Bootstrap styles inside the index.html file, within the head tag:

html

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.

Installing Axios

Install Axios for handling API requests:

npm install axios

Create a **'Component'** folder and a new component, **e.g., "Quiz.js".**

import React, { useState, useEffect } from 'react'; import axios from 'axios';

const QuizApp = ({ exam }) => {

// ... (Refer to the previous Quiz.js code) };

export default QuizApp;

8. Testing

No specific testing instructions provided in the documentation. Ensure to test each component thoroughly to identify and fix any issues.

9. Troubleshooting

If you encounter issues during setup or usage, consider the following troubleshooting steps:

- Verify that all dependencies are correctly installed.
- Check for typos or errors in configuration files
- Review error messages in the console for more information.
- Consult the documentation for known Issues and solution.

10. Deployment

Deploy the Django backend and React frontend as per your deployment platform requirements.

11. Security and Authentication

No specific security and authentication details provided. Implement secure coding practices and consider user authentication if needed.

12. Scaling and Optimization

Optimize the application for performance based on usage patterns. Consider caching, code splitting, and other optimization technique.

13. Maintenance

Regularly update dependencies, address security vulnerabilities, and perform maintenance tasks as needed.

14. Contributing

If you'd like to contribute to the project, follow the guidelines outlined in the documentation.

15. License

Include information about the license under which the Random Quiz Generator is released.

FINAL OUTPUT:

Random Question Generation

Time Remaining: 0:00

1) What is the maximum possible length of an identifier??

○ 16
○ 32
○ 64
○ None of these above

2) Who developed the Python language??

○ Zim Den
◉ Guido van Rossum
○ Niene Stom
○ Wick van Rossum

Submit

Your Score: 3/5

Keep practicing! You can do better next time.

1) What is the maximum possible length of an identifier??

◉ 16
○ 64
○ None of these above

Correct Answer: None of these above

2) Who developed the Python language??

○ Zim Den
○ Niene Stom
○ Wick van Rossum

Correct Answer: Guido van Rossum