CLOUD SERVER PROJECT DOCUMENTATION

Student Details

Name: Rima Muhammed Muneer

Student ID: 35498008

Project title: Quiz Website Deployment on Cloud Server

Global IP Address: 64.227.188.30

1. **Introduction**

   This document outlines the setup, configuration, and deployment of an online quiz website on a cloud server. The project is hosted on DigitalOcean and serves as an interactive platform where users can take quizzes. The objective is to demonstrate proficiency in cloud infrastructure, server setup, and web deployment.

2. **Setup Overview**

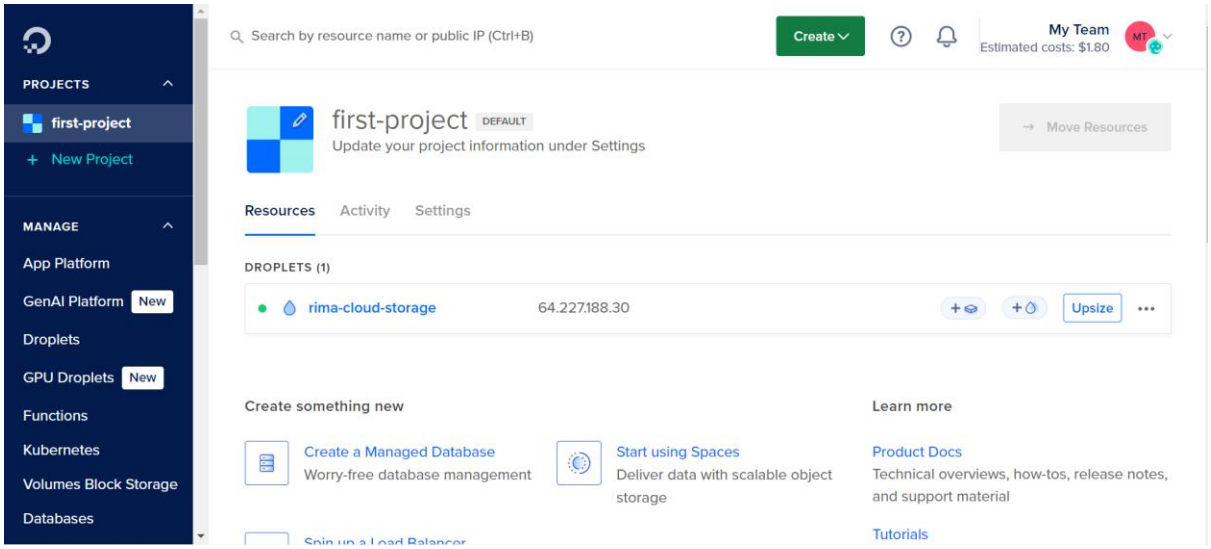
   2.1 Cloud Server Provider

- Platform: DigitalOcean
- Server Type: Virtual Machine (Droplet)
- Operating System: Ubuntu 22.04 LTS
- Web Technologies: HTML, CSS, JavaScript (Static files)
- Specifications: 1 vCPU, 1GB RAM, 25GB SSD, 1TB Bandwidth
- Domain: No custom domain configured (Accessed via IP address)

3. **Server Configuration**

   Step 1: Creating a DigitalOcean Droplet

   1. Log into a DigitalOcean Account.
   2. Create a new droplet with **Ubuntu 20.04.**
   3. Choose the size of the droplet based on the project needs (the smallest size is typically sufficient for basic setups).
   4. For SSH key authentication, either create a new SSH key or use an existing one for secure access.
   5. Complete the droplet creation and note the public IP address of the droplet. This IP will be used to access the quiz website.



Step 2: Accessing the Server

After the droplet is created, the server can be accessed via SSH using the following command:

```
C:\Users\rimam>ssh root@64.227.188.30
```

Step 3: Installing Apache Web Server

To install **Apache**, run the following commands:

```
apt update
apt install apache2 -y
```

Step 4: Visiting the Server

Opening the browser and go to:
http://64.227.188.30

Opens us a page “Apache2 Ubuntu Default Page”

Step 5: Cleaning up the web folder:

Now delete the default Apache page:

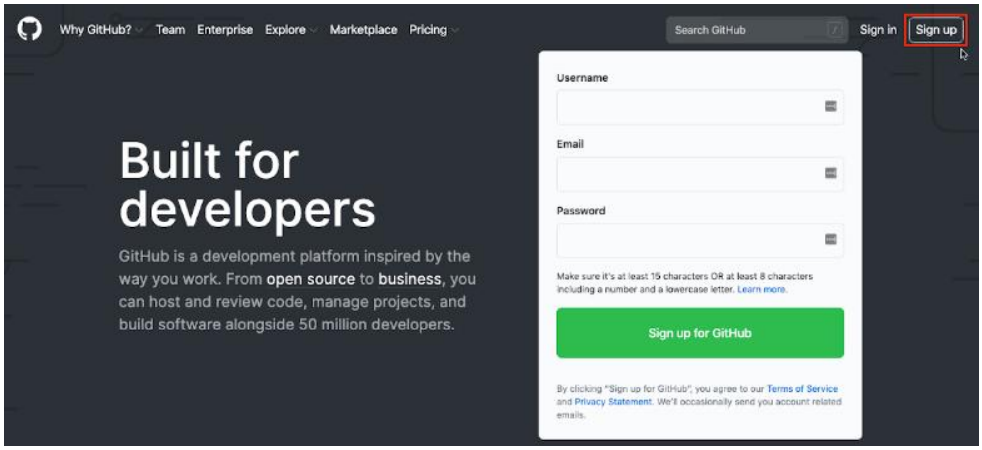
```
root@rima-cloud-storage:~# rm /var/www/html/index.html
root@rima-cloud-storage:~# rm /var/www/html/quiz.js
root@rima-cloud-storage:~# rm /var/www/html/main.js
root@rima-cloud-storage:~# rm /var/www/html/style.css
```

Step 6: Create Quiz page :

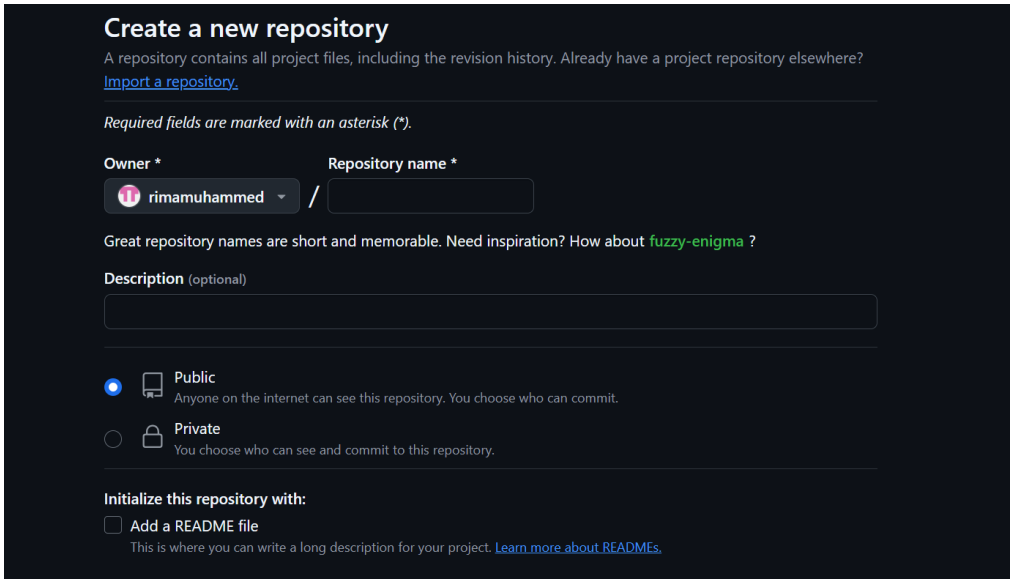
```
root@rima-cloud-storage:~# nano /var/www/html/index.html
root@rima-cloud-storage:~# nano /var/www/html/style.css
root@rima-cloud-storage:~# nano /var/www/html/quiz.js
root@rima-cloud-storage:~# nano /var/www/html/main.js
```

4. GitHub Setup

Step 1: Go to https://github.com/ and log in.



Step 2 : Create a New Repository



Set a name , Choose Public (pr private).

Click **Create Repository**

5. Connect Server to GitHub

Step 1: SSH into server

```
ssh root@64.227.188.30
```

Step 2: Install Git

```
apt update && apt install git -y
```

Step 3 : Navigate to the project directory:

```
cd /var/www/html/quiz
```

Step 4: Initialize Git in this directory

```
git init
```

Step 5: Link the GitHub repository to the local Git repo:

```
git remote set-url origin https://github.com/rimamuhammed/quiz-website.git
```

Step 6:Add new files/modified files

```
git add .
```

Step 7: Commit the changes

```
git commit -m "Initial commit of quiz website"
```

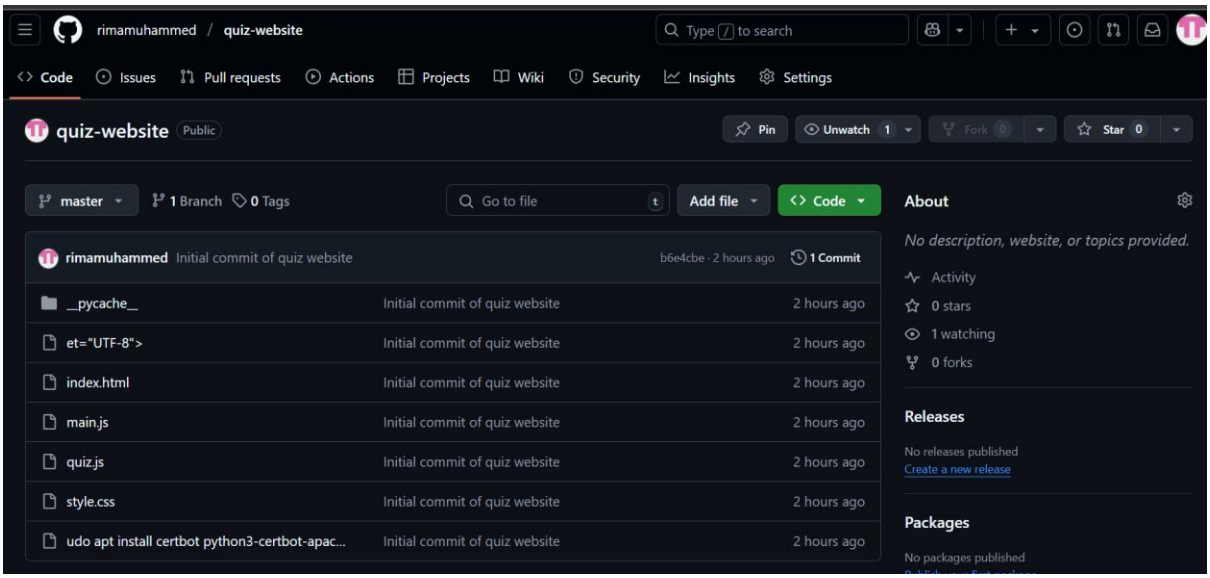
Step 8: Push the changes to GitHub:

```
/var/www/html# git push -u origin master
```

Final Step: Verify on GitHub

Go to the GitHub repository in a browser:

https://github.com/rimamuhammed/quiz-website



6. Testing the Website

Once everything is set up:

- Open a Browser and visit the server's IP address:
  http:// 64.227.188.30

7. Overview of the quiz:

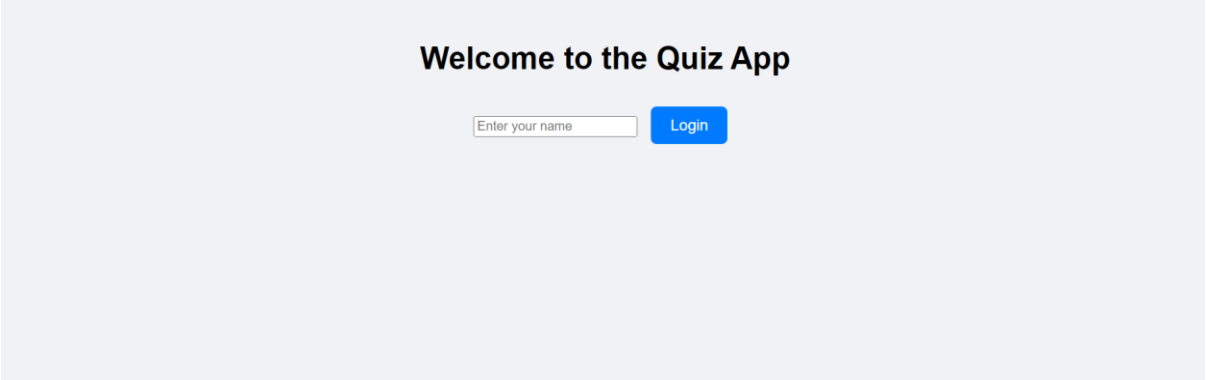
This is a web-based quiz application where users can log in, select a quiz category, and answer 10 questions. At the end, the application displays the user's score based on their performance.

Features:

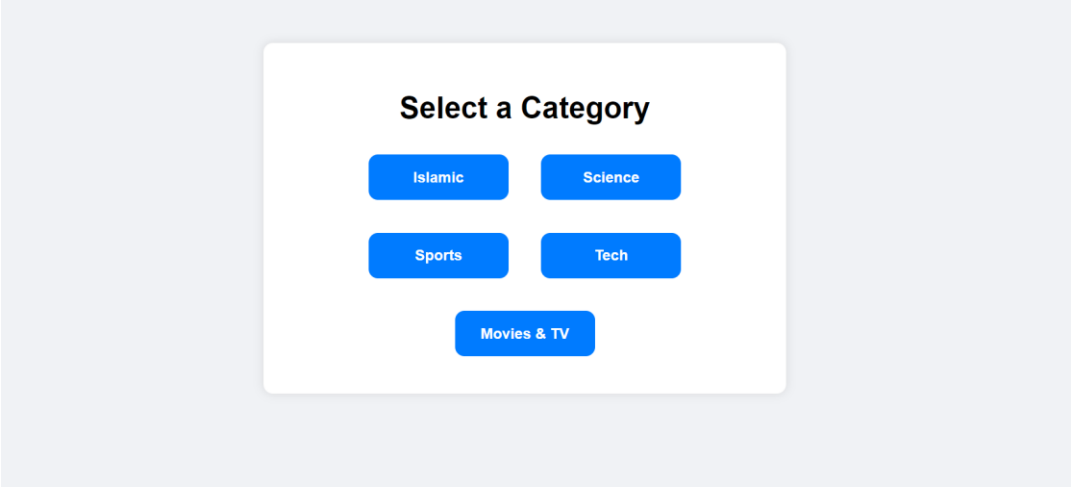
- **Login Screen:** Users must login before starting the quiz
- **Category Selection**: After logging in, users can select the category of their choice for the quiz, which are science, tech, Islamic, sports, movie & tv shows.
- **Quiz with 10 Questions**: The quiz contains 10 multiple-choice questions based on the selected category.
- **Score Calculation**: After completing the quiz, users are shown their score based on the number of correct answers.

How it works:

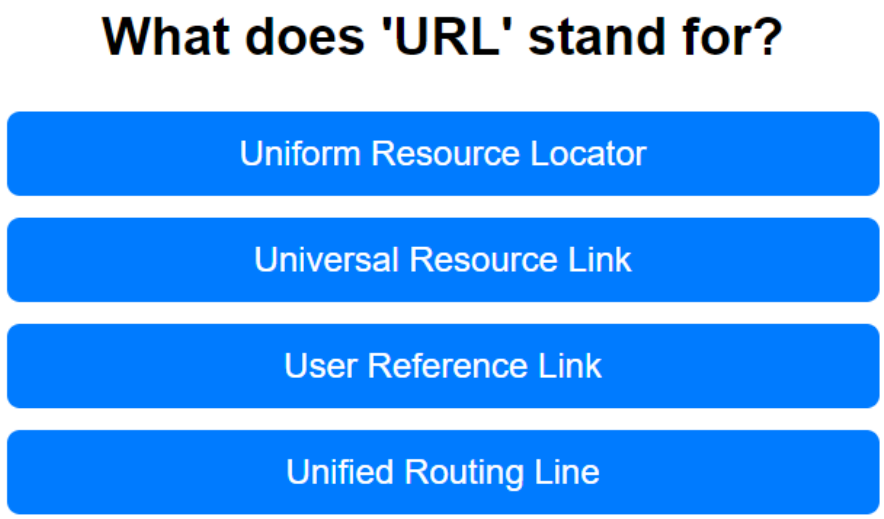
1. Login:
   Users are prompted to log in with their credentials.



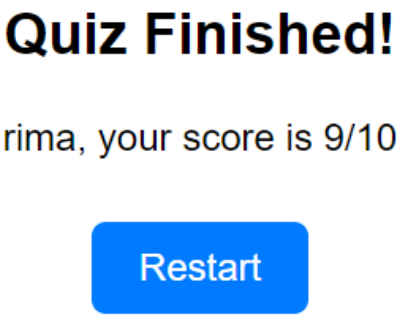
2. Category Selection:
   After a successful login, the user selects a quiz category.



3. Answering Questions:
   The user is presented with 10 questions from the chosen category. Each question has multiple options to choose from.



4. Score Display:
   After answering all the questions, the user sees their total score, calculated by the number of correct answers.



Technical Details:

- Frontend: The quiz interface is built using HTML, CSS, and JavaScript.
- Logic:
  The quiz logic is managed using JavaScript, which handles user interactions, question flow, and score tracking.
  The login and category selection are handled in separate pages (or components), ensuring a smooth transition for the user.

Future Improvements:

- Adding a database to store user data and quiz scores.
- Allowing users to review their answers after completing the quiz.

8. References:

https://docs.digitalocean.com/products/

https://httpd.apache.org/docs/2.4/vhosts/

https://docs.github.com/en/get-started/start-your-journey/hello-world

9. Conclusion:

This project successfully implements a frontend-based quiz website hosted on a DigitalOcean cloud server using Apache. The quiz includes user login, category selection, and a scoring system, all handled through HTML, CSS, and JavaScript without a backend.

The server setup was manually configured, ensuring hands-on experience with deploying a web application in a cloud environment. While this implementation is functional, it has some limitations, such as the lack of a database for persistent storage and a backend for user authentication. Future improvements could include integrating a backend (Node.js, Python, or PHP) and a database (MySQL, MongoDB, or Firebase) to enhance security and data management.

Overall, this project demonstrates proficiency in cloud deployment, web development, and server management, aligning with the key learning objectives of this assignment.