

Presentation: Quiz Application using Flask

Slide 1: Title Slide

- ✦ **Title:** *"Building a Quiz Application with Flask"*
- ✦ **Subtitle:** *A Beginner-Friendly Web Development Project*
- ✦ **Presented by:** Sarvothayan S

Slide 2: Introduction to the Project

What is this project?

- This is a **web-based quiz application** built using **Flask (Python)**.
- It allows users to **select difficulty levels**, **answer questions**, and **view their scores**.
- It provides an **interactive experience** using **HTML, CSS, and JavaScript**.

Slide 3: Technologies Used




- **Flask (Python)** → Backend framework to run the web application.
- **HTML & CSS** → Frontend for structuring and styling the pages.
- **JavaScript** → Adds functionality like **timers** and **animations**.

Slide 4: Features of the Quiz App

- ✓ Select **Easy, Medium, or Hard** difficulty levels.
- ✓ Questions load dynamically from a **Python dictionary** (instead of a database).
- ✓ **Timer countdown** to make the quiz more interactive.
- ✓ **Shows correct answers** after submission.
- ✓ Uses **separate background images** for each page.
- ✓ Fully **responsive design** (works on PC & Mobile).

Slide 5: Project File Structure

quiz_app/

```
| —  app.py      # Main Flask application
| —  templates/    # HTML files
|   | — index.html    # Homepage
|   | — quiz.html     # Quiz page
|   | — result.html   # Results page
| —  static/      # Images
|   | — background.png
|   | — luffy.png
|   | — luffy1.jpg
```

◆ **Templates Folder** → Stores **HTML files** that Flask renders.

◆ **Static Folder** → Stores **images and other static files**.

Slide 6: How the Program Works?

- 1 ☐ User opens the homepage (`index.html`).
- 2 ☐ Selects a **difficulty level** (Easy, Medium, Hard).
- 3 ☐ The quiz page (`quiz.html`) loads with **random questions**.
- 4 ☐ User **answers the questions** and submits the quiz.
- 5 ☐ The result page (`result.html`) **displays the score and correct answers**.

Slide 7: Common Mistakes People Make

1. Forgetting to run Flask app

python app.py

Fix: Always start the server before testing the project.

2. Incorrect Flask Routing

```
@app.route("/quiz/<difficulty>")
```

Fix: Make sure the URL is correctly passed in the HTML files.

3. Not linking CSS & JS properly

```
<link rel="stylesheet" href="style.css">
```

Fix: Use **internal CSS** (since we removed `style.css`).

4. Background images not loading

Fix: Ensure images are placed in `/static/Images/` and paths are correct.

5. Timer not working

Fix: Ensure **JavaScript** is included inside `<script>` tags at the right place

Slide 8: How Hard is This Project?

✦ **Difficulty Level:** ☐ Beginner-Friendly

✦ **Why?**

- ✓ Uses **simple Python concepts** (lists, dictionaries).
- ✓ No **complex database setup** (questions stored in a file).
- ✓ Basic **HTML, CSS, and JavaScript** knowledge is enough.

💡 **Who Can Build This?**

- **School Students (10th grade and above)**
- **Beginners learning Flask & Web Development**

Slide 9: Why is This Project Useful?

- ✓ **Teaches Flask basics** → Routes, templates, rendering.
- ✓ **Improves problem-solving** → Handling user inputs, displaying results.
- ✓ **Enhances UI/UX skills** → CSS styling, animations, backgrounds.
- ✓ **Good Portfolio Project** → Showcasing skills for beginners.

Slide 10: Conclusion & Final Thoughts

👉 This project **teaches Flask, HTML, CSS, and JavaScript** in a simple way.

👉 By completing it, you have built a **fully functional web app**.

👉 **Next Steps?**

- ◆ Add a **database** to store quiz history.
- ◆ Improve UI with **more animations**.
- ◆ Make it a **multiplayer quiz**!