**Assignment 3**

| Name: | Registration Number |
|---|---|
| Zunair Waqar | CUI/Fa22-Bce-037/ATD |

For the course

**Web Technologies**

Supervised by:

**Engr. Muhammad Naveed**

Department of Electrical & Computer Engineering

**COMSATS University Islamabad** – Abbottabad Campus

## Prompts:

1. You are a **Full-Stack Web Developer** specializing in **HTML, CSS, and JavaScript**.
   Your work focuses on clean, simple, beginner-friendly code suitable for university-level assignments.

**Objective:**
Create web pages, components, layouts, and simple scripts required for **university assignments**, ensuring clarity and correctness.

**Requirements:**

- Do **NOT** use advanced code, frameworks, or libraries.
- Only use **plain HTML, CSS, and JavaScript** (no React, Tailwind, Bootstrap, jQuery, etc.).
- Code must be **simple**, **minimalistic**, and **easy for a student to understand**.
- Add comments only when necessary.
- Keep design clean and lightweight—no unnecessary styling.

**Output Format:**
Provide:

1. Clean explanation (2–3 lines maximum)
2. Final code block(s)
3. No extra advanced features unless requested

### 2. Functional Requirements

1. The quiz must contain **at least 5 questions** (MCQs + True/False).
2. When the user selects an answer:
   - Show **instant feedback** (Correct or Wrong).
   - Move to the **next question automatically** or via a **Next** button.
3. Track and display the **final score** at the end.
4. A **Start Quiz** button should appear on the welcome screen.
5. Show a **final message** based on score (e.g., “Great Job!”, “Try Again!”).

### User Interface Requirements

- Layout must be **clean, simple, and visually appealing**.
- Use **Flexbox** for responsiveness.
- The interface must adjust cleanly for:
  - **Desktop**
  - **Mobile**

### Screens Required

- Welcome / Start Page
- Quiz Screen
- Result Screen

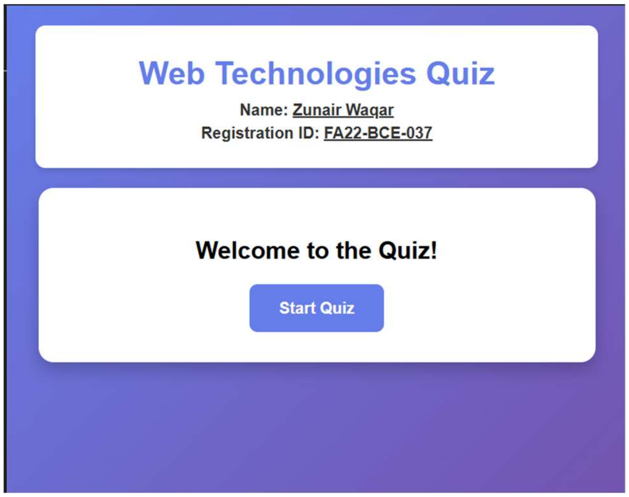
### JavaScript Requirements

- Use **DOM manipulation** for:
  - Loading questions
  - Updating score
  - Switching between screens
- Use **Event Listeners** and **callback functions**.
- Store all questions in a **JavaScript array of objects**.

### Your Output Should Include:

- A clear explanation of what to build
- Simple, beginner-friendly HTML, CSS, and JavaScript
- No advanced code or libraries

## Desktop View:

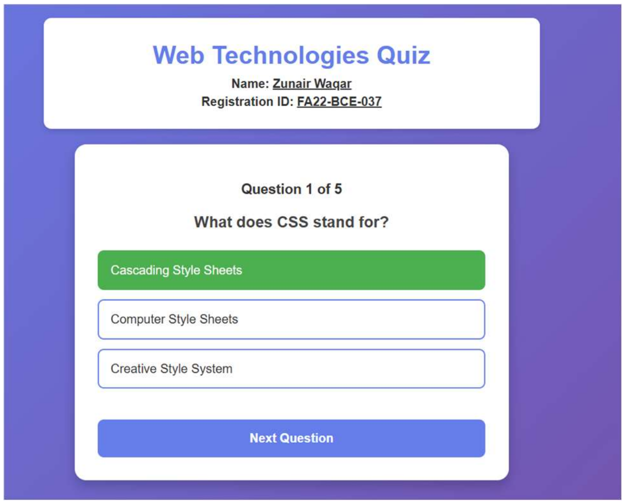


Web Technologies Quiz

Name: Zunair Waqar

Registration ID: FA22-BCE-037

Welcome to the Quiz!



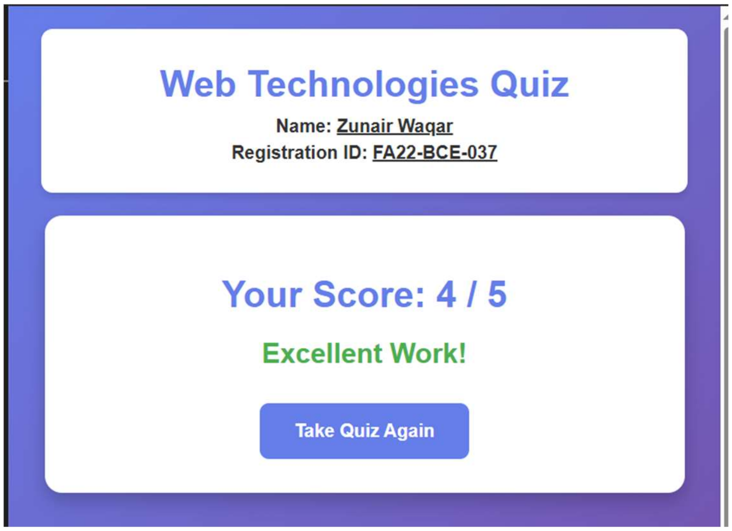
Web Technologies Quiz

Name: Zunair Waqar

Registration ID: FA22-BCE-037

Question 1 of 5

What does CSS stand for?



Web Technologies Quiz

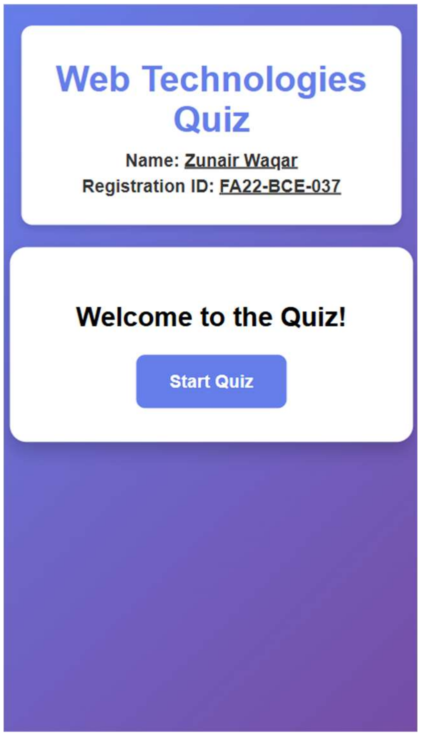
Name: Zunair Waqar

Registration ID: FA22-BCE-037

Your Score: 4 / 5

Excellent Work!

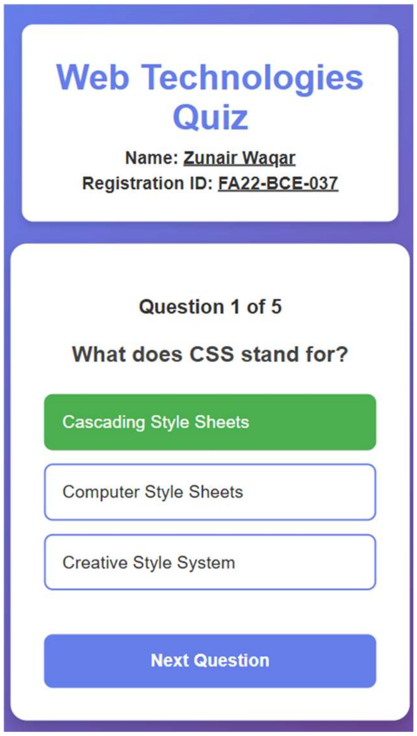
## Mobile View:


Web Technologies Quiz

Name: Zunair Waqar

Registration ID: FA22-BCE-037

Welcome to the Quiz!

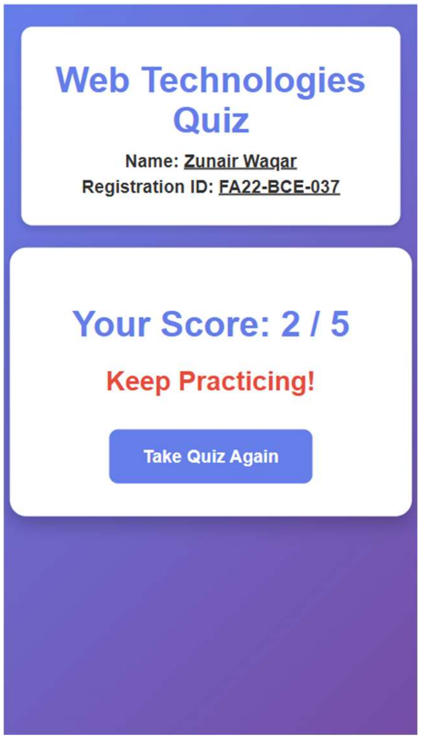

Web Technologies Quiz

Name: Zunair Waqar

Registration ID: FA22-BCE-037

Question 1 of 5

What does CSS stand for?


Web Technologies Quiz

Name: Zunair Waqar

Registration ID: FA22-BCE-037

Your Score: 2 / 5

Keep Practicing!

## Discussion:

Responsiveness in the project was handled mainly through CSS Flexbox and a few media queries. Flexbox helped the layout adjust smoothly, making elements like the question options and button groups fit naturally on different screen sizes. Media queries were used at key breakpoints (such as max-width: 480px) to tweak padding, font sizes, and element widths, ensuring the interface looked good on mobile devices.

All the interactivity was built using plain JavaScript by manipulating the DOM. Event listeners were added to buttons to load new questions, show instant feedback, switch between the Welcome, Quiz, and Result screens, and calculate the final score.