Week 1: Introduction

1. **Introduction to Web Engineering**
   Web Engineering is a discipline focused on systematic, quantifiable approaches to develop high-quality web-based systems and applications.
   - **Example:** Developing a user-friendly e-commerce website like Amazon.
2. **A Brief Introduction to the Internet**
   The Internet is a global network connecting millions of private, public, academic, business, and government networks.
   - **Example:** Accessing Google through a web browser.
3. **The World Wide Web (WWW)**
   The WWW is a service provided on the internet that uses HTTP to transmit web pages.
   - **Example:** Viewing a blog on WordPress.
4. **Web vs Internet**
   - **Web:** A collection of web pages hosted on servers (WWW).
   - **Internet:** The infrastructure connecting devices worldwide.
   - **Example:** Email is an internet service, while a website like Wikipedia is a web service.
5. **Web Browsers and Web Servers**
   - **Web Browser:** Software for accessing web pages (e.g., Chrome).
   - **Web Server:** Stores and serves web content (e.g., Apache).
   - **Example:** A browser requests a webpage from a server via HTTP.
6. **Uniform Resource Locators (URLs)**
   A URL identifies the address of a resource on the web.
   - **Example:** https://www.example.com/about.
7. **Hypertext and HTTP**
   - **Hypertext:** Text with links to other texts or resources.
   - **HTTP:** A protocol to transfer hypertext between browsers and servers.

    - **Example:** Clicking on a Wikipedia link.

8. **IP Address: IPv4 vs IPv6**
    - **IPv4:** 32-bit address format (e.g., 192.168.1.1).
    - **IPv6:** 128-bit address format (e.g., 2001:0db8::1).
    - **Example:** IPv6 provides more unique addresses than IPv4.

9. **Website vs Web Application**
    - **Website:** Static information pages.
    - **Web Application:** Interactive systems with dynamic content.
    - **Example:** Wikipedia (website), Gmail (web application).

10. **Domain Name Structure and Working**

- **Structure:** www.example.com (subdomain.domain.TLD).
- **Working:** Resolves names to IP addresses using DNS.
- **Example:** www.google.com resolves to 142.250.64.78.

11. **Web Request-Response Cycle**
    A browser sends a request to a server, and the server sends back the response (HTML).
    - **Example:** Typing facebook.com in the browser initiates this cycle.

12. **Categories of Web Applications**
    - Document-Centric: Static content (e.g., Wikipedia).
    - Interactive: Forms and inputs (e.g., Google Forms).
    - Transactional: E-commerce (e.g., Amazon).
    - Workflow-Based: Project management tools (e.g., Trello).
    - Collaborative: Google Docs.
    - Portal-Oriented: Yahoo.
    - Ubiquitous: Smart IoT apps.
    - Knowledge-Based: AI platforms (e.g., WolframAlpha).

13. **Web Application Architecture**

- **Single Tier:** Everything runs on one machine.
- **Two Tier (Client-Server):** Data logic and UI separate (e.g., MySQL + PHP).
- **Three Tier:** Frontend, backend, and database (e.g., React + Node.js + MongoDB).
- **Example:** Netflix uses three-tier architecture.

Week 1-2: HTML

1. **Introduction to HTML**
   HTML (Hypertext Markup Language) structures web content using elements.
   - **Example:** <h1>Hello World</h1> creates a heading.
2. **Creating HTML Documents**
   Start with <!DOCTYPE html> and include <html>, <head>, and <body> tags.
   - **Example:**

html

Copy code

```
<!DOCTYPE html>
<html>
 <head><title>Page Title</title></head>
 <body><h1>Welcome</h1></body>
</html>
```

3. **Nesting HTML Elements**
   Elements inside others must follow a logical hierarchy.
   - **Example:**

html

Copy code

```
<div>
 <p>This is a paragraph inside a div.</p>
```

```
</div>
```

4. **Embedding Audios & Videos in HTML**
   Use <audio> and <video> tags.
   - **Example:**

html

Copy code

```html
<audio controls>
  <source src="audio.mp3" type="audio/mpeg">
</audio>
<video controls>
  <source src="video.mp4" type="video/mp4">
</video>
```

5. **HTML Tables, Forms & Inputs**
   Use <table>, <form>, and <input> for structured data and user inputs.
   - **Example:**

html

Copy code

```html
<form>
  <input type="text" name="username">
</form>
```

Week 2-3: CSS

1. **CSS Selectors, Properties & Values**
   Select HTML elements and apply styles.
   - **Example:**

css

Copy code

```
p { color: red; }
```

2. **Flexbox & Grid**
   CSS tools for layout management.
   - **Example:**

css

Copy code

```
display: flex; justify-content: center;
```

3. **Responsive Design**
   Adapt layouts for screen sizes using breakpoints.
   - **Example:**

css

Copy code

```
@media (max-width: 600px) { body { font-size: 12px; } }
```

Week 4-6: JavaScript

1. **Variables, Functions, Loops**
   JavaScript handles logic with these tools.
   - **Example:**

javascript

Copy code

```
let name = "John";

console.log(name);
```

2. **DOM Manipulation**
   Change the web structure dynamically.
   - **Example:**

javascript

Copy code

```
document.querySelector("h1").textContent = "Updated!";
```

3. **Events & Local Storage**
   Capture user actions and save data locally.
   - **Example:**

javascript

Copy code

```
localStorage.setItem("key", "value");
```