CSC-336
Web Technologies

Lecture 1:

Topics:

- Course Overview & Web Fundamentals
- HTML (vscode session) : Headings,paragraph,formatting, br, hr tags

Muhammad Naveed Shaikh
Department of Computer Engineering
naveedshaikh@cuiatd.edu.pk

COMSATS University Islamabad | Abbottabad Campus

Course Expectations & Commitments

- This is an Elective Course 📌
- High potential & scope in industry (web apps, freelancing, startups)

Requires:

- Basic programming skills (logic, problem-solving)
- Interest & motivation to learn new tools

- What we will learn:
  - **HTML, CSS, JS basics, PHP**
  - Small Web Projects

- Tools:

Browser, VS Code, Online editors (CodePen, GitHub Pages)

Advice:

This course requires consistent coding practice and active participation.

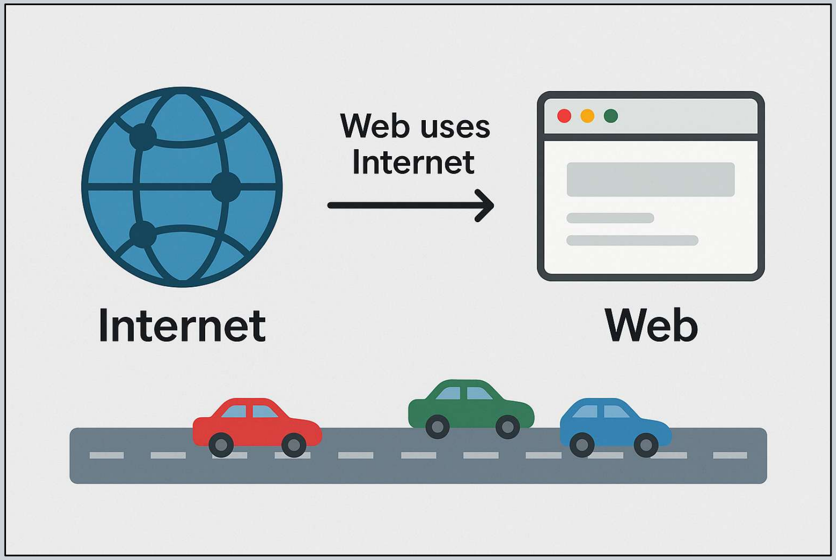
If you feel :

your interest or availability is limited or
it is not aligned with your current priorities,

Please make a thoughtful decision.

How internet and Web Works? ! (Internet vs WWW)

- Internet 🌐
  - Global network of computers/devices.
  - Infrastructure: cables, satellites, routers, servers.
  - Supports multiple services (Email, FTP, Video calls, Web).
- World Wide Web 💻
  - Service that runs on top of the Internet.
  - Uses HTTP/HTTPS to deliver content.
  - Accessed using browsers (Chrome, Firefox, Edge).
- Key Difference
  - 👉 Internet = “Highway”
  - 👉 Web = “Vehicles” running on it



How the Web Works: Client–Server Model

🧠 Basic Concept

The client-server model is a distributed architecture where:

- Client: Requests services or data (usually your browser).
- Server: Provides services or data (like a web server hosting a website)

Step-by-Step: How a Website Loads ! Client–Server Model

1. **User Enters a URL**
   - You type something like www.example.com into your browser.
2. **DNS Lookup**
   - The browser asks a DNS server to translate the domain name into an IP address.
3. **Browser Sends Request**
   - Using HTTP or HTTPS, your browser sends a request to the server at that IP address asking for the website's content.
4. **Server Processes Request**
   - The server receives the request,
   - fetches the Web contents(HTML, CSS, JS, images)
   - And, sends them back to the client.

Step-by-Step: How a Website Loads ! Client–Server Model

1. **User Enters a URL**
   - You type something like www.example.com into your browser.
2. **DNS Lookup**
   - The browser asks a DNS server to translate the domain name into an IP address.
3. **Browser Sends Request**
   - Using HTTP or HTTPS, your browser sends a request to the server at that IP address asking for the website's content.
4. **Server Processes Request**
   - The server receives the request,
   - fetches the Web contents(HTML, CSS, JS, images)
   - And, sends them back to the client.
5. **Browser Renders the Page**
   - Your browser interprets the files:
     - HTML defines structure.
     - CSS styles the page.
     - JavaScript adds interactivity.
6. **User Interacts**
   - Any clicks, form submissions, or actions send new requests to the server, which responds accordingly

What Is Web Content?

Web content is the visible and interactive part of a website, made up of three core technologies:

Technology	Purpose	Example
HTML (Hypertext Markup Language)	Structure and content	Headings, paragraphs, images
CSS (Cascading Style Sheets)	Styling and layout	Colors, fonts, spacing
JavaScript	Interactivity and logic	Buttons, animations, form validation

What Is Web Development?

Web development is the process of building and maintaining websites.
It includes:

- Designing user interfaces
- Writing code for functionality
- Managing data and servers

It's divided into two main areas:

1. Front-End
2. Back-End.

Front-End Development (Client-Side)

Definition:

The part of the website users see and interact with.

Responsibilities:

- Designing layout and visuals
- Ensuring responsiveness across devices
- Adding interactivity with JavaScript

Technologies:

- HTML, CSS, JavaScript
- Frameworks: React, Angular, Vue.js

Tools:

VS Code, Chrome DevTools, Figma

Back-End Development (Server-Side)

Definition:

The part of the website that processes data and handles logic behind the scenes.

Responsibilities:

- Managing databases
- Handling user authentication
- Processing form submissions

Technologies:

- Languages: Node.js, Python, PHP,
- JavaDatabases: MySQL, MongoDB, PostgreSQL
- Frameworks: Express.js, Django, Laravel

Tools:

Postman, GitHub, Heroku

Full-Stack Development

Definition:
A developer who works on **both front-end and back-end**.

Benefits:

- End-to-end control
- Versatility in projects
- Ideal for startups and freelance work