CHAPTER 1

INTRODUCTION TO WEB DESIGN

Introduction of internet

<u>1.History of the Internet</u>

1. Early Beginnings

- 1950s-60s:
 - Concept of networking computers began.
 - U.S. military wanted a way to communicate securely even during a war.
- ARPANET (1969): Created by the U.S. Department of Defense (ARPA). First network to connect computers at universities and research centers. First message sent between UCLA and Stanford.

2. Important Milestones

- **1971**:
 - **First email** sent by Ray Tomlinson.
 - "@" symbol introduced in email addresses.
- **1973**: **First international connection**: University College London joins ARPANET.
- **1974**: Term "Internet" introduced in a paper by Vint Cerf and Bob Kahn.
- **1983**: **TCP/IP** protocol introduced the "language" of the Internet.
 - ARPANET officially switched to TCP/IP. **Domain Name System (DNS)** created (e.g., .com, .org, .net).

3. Growth and Public Use

- Late 1980s:
 - Networks expanded beyond ARPANET.
 Introduction of commercial internet service providers (ISPs).
- **1990**:
 - **ARPANET shut down** Internet continues growing.
 - Tim Berners-Lee invents the **World Wide Web (WWW)** while working at CERN.
- 1991:

• WWW made public. • Introduction of **web browsers** like Mosaic (1993), later Netscape.

4. Expansion and Modern Internet

• 1990s-2000s:

- Massive growth in websites and users.
 Search engines (like Google, 1998) made finding information easier.
- Social media platforms started (e.g., Facebook in 2004, YouTube in 2005).
- **Today**: Internet connects billions of people. Used for communication, education, entertainment, business, etc.

Key Figures

- Vint Cerf and Bob Kahn invented TCP/IP.
- Ray Tomlinson sent first email.
- **Tim Berners-Lee** invented the World Wide Web.

Quick Timeline

Year Event 1969 ARPANET launched

- 1971 First email sent
- 1983 TCP/IP adopted, DNS created
- 1990 World Wide Web invented
- 1991 WWW made public
- 1993 First web browser (Mosaic)
- 1998 Google founded

2.Features of the Internet

1. Global Connectivity

- Connects millions of computers worldwide.
- Allows people to communicate from anywhere on the planet.

2. Wide Range of Information

- Provides access to information on any topic.
- Includes text, images, audio, video, and more.

3. Communication

- Supports email, instant messaging, video calls, and social media.
- Enables fast and easy sharing of information.

4. Services Available

- E-commerce (online shopping), online banking, e-learning, and entertainment.
- Cloud services for storage and computing.

5. Accessibility

- Available 24/7 (anytime, anywhere) with an internet connection.
- Can be accessed via computers, smartphones, tablets, etc.

6. Cost-Effective Communication

• Sending emails, making voice or video calls over the internet is often cheaper than traditional methods.

7. Interactivity

- Users can interact with websites, social media, games, and online services.
- Enables real-time communication and feedback.

8. Multimedia Support

• Allows sharing and viewing of multimedia content (videos, music, podcasts).

9. Hyperlinking

• Web pages are linked through hyperlinks, allowing easy navigation from one page to another.

10. Decentralization

- No single person, company, or government controls the whole Internet.
- Operates through a network of servers and systems around the world.

11. Security Features

- Supports encryption (for secure transactions).
- Firewalls, antivirus software, and other tools protect users online.

Quick Keywords to Remember

Feature Meaning

ConnectivityConnects people worldwideCommunicationEmail, chat, video callsAccessibility24/7 accessMultimediaText, video, imagesInteractivityReal-time user actionsDecentralizationNo single control

3. Applications of the Internet

1. Communication

- **Email** Sending and receiving messages quickly.
- Instant Messaging Real-time text communication (e.g., WhatsApp, Messenger).
- Video Conferencing Virtual meetings (e.g., Zoom, Google Meet).
- **Social Networking** Connecting with friends and sharing updates (e.g., Facebook, Instagram).

2. Information Retrieval

- Searching for information using search engines (e.g., Google, Bing).
- Access to **online libraries**, encyclopedias, news websites.

3. E-Commerce (Electronic Commerce)

- Buying and selling products online (e.g., Amazon, eBay).
- Online banking and digital payments.

4. Entertainment

- Streaming videos (e.g., YouTube, Netflix).
- Online gaming.
- Listening to music and podcasts.

5. Education (E-Learning)

- Online courses, virtual classrooms (e.g., Coursera, Khan Academy).
- Access to educational resources and research papers.

6. E-Government

- Government services available online (e.g., tax filing, applying for IDs).
- Information portals and citizen services.

7. Remote Work (Telecommuting)

- Working from home using internet-based tools.
- Sharing files, attending meetings online.

8. Cloud Computing

- Storing data on the internet (e.g., Google Drive, Dropbox).
- Accessing software and services without installing them locally.

9. Internet of Things (IoT)

• Connecting everyday devices (e.g., smart homes, smartwatches) to the Internet for automation and control.

Quick Keywords to Remember

Example
Email, chat apps
Google search
Online shopping
Netflix, YouTube
Online courses
Tax filing online
Virtual meetings
Example
Google Drive
Smart homes

World Wide Web (WWW)

1. What is the World Wide Web?

- A system of interlinked hypertext documents accessed via the Internet.
- Allows users to view web pages that contain text, images, videos, and links.
- Invented by Tim Berners-Lee in 1989-1990 at CERN.

2. Difference Between Internet and World Wide Web

Internet

World Wide Web

Physical network of computers and Collection of websites and web pages accessed via the servers. Internet.

Includes email, FTP, VoIP, etc. Only websites and web content. Internet existed before the WWW. WWW started in 1990.

3. Key Components of the WWW

- Web Browser
 - Software to access web pages (e.g., Chrome, Firefox, Safari).
- Web Server
 - A computer that stores and serves web pages.
- Web Pages O Documents written in HTML (HyperText Markup Language).
- URL (Uniform Resource Locator) Address used to access a specific web page.
- HTTP/HTTPS (HyperText Transfer Protocol / Secure) Rules used to transfer web pages from servers to browsers.

4. Features of the World Wide Web

- Hypertext links allow easy navigation between pages.
- Multimedia support (text, images, audio, video).
- Interactive content through forms, games, and online tools.
- Global accessibility: anyone with an Internet connection can access it.
- **Dynamic content**: web pages can update automatically.

5. How WWW Works (Simple Steps)

- 1. User enters a URL into a web browser.
- 2. Browser sends a request to the **web server**.
- 3. Server sends back the **web page** data.
- 4. Browser displays the web page for the user.

Quick Keywords to Remember

Term	Meaning
Web Browser S	Software to view websites
Web Server	Stores and delivers web pages
URL	Web address
HTTP/HTTPS	Protocols for transferring web pages
HTML	Language for creating web pages

Evolution of the World Wide Web (WWW)

- 1. Web (The Static Web)
 - **Time Period**: Early 1990s to early 2000s.
 - Features:
 - Mostly static (unchanging) web pages. O Content created and controlled by website owners. O Users could only read information, no interaction.
 - Very basic design: mostly text and a few images.
 - **Examples**: o Early websites, online encyclopedias.

2. Web (The Social Web)

- **Time Period**: Early 2000s to present.
- Features:
 - Web became **interactive** and **dynamic**. Users can **create content** (upload videos, write blogs, comment).
 - Focus on **social networking**, **sharing**, and **collaboration**.
 - Use of **multimedia** (videos, animations).
- Examples:
 - Facebook, YouTube, Wikipedia, Instagram, blogs.

3. Web (The Semantic Web)

- **Time Period**: Emerging from 2010 onwards (still developing).
- Features:
 - Internet becomes more intelligent.

 Focus on personalization (content tailored to users).
 Machines understand data better using AI (Artificial Intelligence).
 - More use of voice assistants (e.g., Siri, Alexa).
- Examples:
 - Smart searches (Google Search predicting your needs), recommendation systems (Netflix, Amazon).

4. (Future) Web (The Intelligent Web) (Optional for extra knowledge)

• Expected Features:

- Fully integrated AI and smart devices.
- Highly personalized experiences.

 Internet of Things (IoT) more widespread.
 Augmented Reality (AR) and Virtual Reality (VR) common.

What is a Website?

1. Definition

- A website is a collection of related web pages linked together and available on the Internet.
- Each website has a unique address called a URL (Uniform Resource Locator).
- Websites are stored on **web servers** and viewed using a **web browser** (e.g., Chrome, Firefox).

2. Components of a Website

- Web Pages: Individual documents (like "Home," "About Us," "Contact").
- Homepage: The main page; usually the first page visitors see.
- Hyperlinks: Links that connect different pages within the site or to other websites.
- Multimedia: Websites may include text, images, videos, animations, and sound.

3. Types of Websites

- **Personal Websites**: Made by individuals (e.g., blogs).
- Commercial Websites: For businesses (e.g., online shops like Amazon).
- Government Websites: Provide public services (e.g., passport application sites).
- Educational Websites: For learning and teaching (e.g., online courses).
- Entertainment Websites: For fun and relaxation (e.g., YouTube, Netflix).

4. Purpose of Websites

- Share information (news websites, blogs).
- Sell products or services (e-commerce).
- **Provide entertainment** (games, videos).
- Offer communication (social media, forums).
- Support education (e-learning platforms).

5. How Websites Work

- 1. User enters the URL in a web browser.
- 2. Browser sends a request to the **web server**.
- 3. Server sends back the required web pages.
- 4. Browser displays the pages to the user.

Quick Keywords to Remember

Term	Meaning
Website	Collection of related web pages
Homepage	Main entry page of a website
URL	Address of a website
Web Server	Stores and delivers website data
Web Browser	Software to view websites

How the Website Works

1. Basic Steps

- 1. User Requests a Website o The user types a URL (e.g., <u>www.google.com</u>) into a web browser (e.g., Chrome, Safari).
- 2. Browser Sends a Request
 - The browser sends a **request** over the **Internet** to find the web server where the website is stored. **DNS (Domain Name System)** helps translate the website name into an IP address to locate the server.
- 3. Web Server Responds The web server receives the request. It finds the requested web page (HTML file) and sends it back to the user's browser.
- 4. **Browser Displays the Page** The browser receives the web page. It **interprets** the code (HTML, CSS, JavaScript) and **displays** the page to the user.

2. Key Terms

Term	Meaning
URL	Address of a website (e.g., <u>www.example.com</u>)
Web Browser	Software used to access and view websites
DNS	System that converts website names into IP addresses
Web Server	A computer that stores and serves web pages
HTML	Language used to create web pages

3. Simple Flow of How a Website Works

pgsql CopyEdit User \rightarrow Browser \rightarrow DNS \rightarrow Web Server \rightarrow Browser \rightarrow Display Website Or in simple steps:

• Type address \rightarrow Browser finds server \rightarrow Server sends page \rightarrow Browser shows page

4. Quick Example

If you type <u>www.bbc.com</u> into Chrome:

- Chrome asks DNS: "What is the IP address of bbc.com?"
- DNS replies with the IP address.
- Chrome sends a request to the BBC web server.
- Server sends back the homepage.
- Chrome displays the BBC homepage on your screen.

Front-End and Back-End Applications

1. What is Front-End?

- Front-End is the part of the application that users see and interact with.
- Also called the "client-side" of an application.
- Focuses on the **design**, **layout**, and **user experience**.
- Examples: buttons, forms, menus, images on a website.

Technologies Used:

- **HTML** (HyperText Markup Language) for page structure.
- CSS (Cascading Style Sheets) for styling and design.
- JavaScript for making the page interactive (e.g., clicking buttons, animations).

2. What is Back-End?

- **Back-End** is the **part of the application** that users **cannot see**.
- Also called the "server-side" of an application.
- Manages the database, server, and application logic.
- It processes user requests and sends the correct information back to the front-end.

Technologies Used:

- **Programming Languages** like PHP, Python, Java, Node.js.
- Databases like MySQL, MongoDB to store data.
- Servers to host the application (e.g., Apache, Nginx).

3. How Front-End and Back-End Work Together

- The user interacts with the **front-end** (e.g., filling out a form).
- The **front-end** sends a **request** to the **back-end** (e.g., submit the form).

- The **back-end** processes the request (e.g., saves the form data in the database).
- The **back-end** sends a **response** back to the **front-end** (e.g., show "Form Submitted Successfully").

Front-End Development

1. What is Front-End Development?

- Front-End Development is about building the parts of a website or application that users see and interact with.
- It is also called **Client-Side Development**.
- Focuses on design, layout, usability, and user experience.

2. Key Responsibilities of a Front-End Developer

- Creating the **structure** of web pages.
- Designing the look and feel (colors, fonts, buttons, menus).
- Making websites **responsive** (work well on computers, tablets, and phones).
- Adding interactivity (e.g., animations, forms, navigation menus).

3. Main Technologies Used	
Technology	Purpose
HTML (HyperText Markup	Builds the structure of web pages (headings, paragraphs,
Language)	links).
CSS (Cascading Style Sheets)	Styles and designs the web pages (colors, layouts, fonts).
JavaScript	Adds interactivity to web pages (animations, pop-ups,
L.	sliders).

4. Tools Commonly Used in Front-End Development

- Text Editors (e.g., VS Code, Sublime Text) for writing code.
- Frameworks (e.g., Bootstrap) for easier and faster web design.
- Libraries (e.g., jQuery) for simpler JavaScript coding.
- Version Control (e.g., Git) to track changes in code.

5. Importance of Front-End Development

- Provides a good user experience.
- Makes websites **easy to use** and **attractive**.
- Helps in building **responsive** and **accessible** websites for all users.

Quick Keywords to Remember

Term	Meaning
HTML	Structure of a web page
CSS	Styling of a web page
JavaScript	Interactivity on a web page
Term	Meaning
Responsive Design	Website adapts to different screen sizes
User Experience (UX)	How easy and enjoyable the website is to use

Back-End Development

1. What is Back-End Development?

- Back-End Development is about building the parts of a website or application that users do NOT see.
- It is also called **Server-Side Development**.
- Focuses on managing databases, servers, and application logic.
- Responsible for storing, organizing, and processing data.

2. Key Responsibilities of a Back-End Developer

- Handling databases (storing user information, products, content).
- **Processing user requests** (e.g., login forms, buying products).
- Ensuring security (protecting user data).
- Managing servers and making sure the website runs smoothly.

3. Main Technologies Used	
Technology	Purpose
Programming Languages	PHP, Python, Java, Node.js (write server-side logic).
Databases	MySQL, MongoDB, PostgreSQL (store and manage data).
Servers	Apache, Nginx (host and deliver websites to users).
APIs (Application Programming Interfaces)	Allow communication between front-end and backend.

4. How Back-End Works (Simple Steps)

1. User interacts with the **front-end** (e.g., clicks "Login").

- 2. Front-end sends **request** to the **server**.
- 3. Back-end processes the request:
 - Checks username and password in the **database**.
- 4. Server sends a **response** (success or error) back to the front-end.

5. Importance of Back-End Development

- Keeps websites and apps **functional** and **secure**.
- Manages **user data** safely.
- Supports dynamic websites (websites that change based on user input).

Client and Server Scripting Languages

1. What is Scripting?

- Scripting refers to writing small programs or scripts that perform specific tasks.
- Scripts are executed by the **client** (browser) or **server** (server-side machine).

2. Client-Side Scripting Languages

- Client-Side Scripting refers to scripts that run on the user's device (client), typically in the web browser.
- **Purpose**: Interactivity, dynamic content on websites, and immediate response to user actions.

Popular Client-Side Scripting Languages:

- JavaScript:
 - Main language for client-side scripting.
 - Allows interactivity (e.g., validating forms, animations).
 - HTML/CSS (though not "scripts," they are part of client-side web development):
 - \circ HTML structures the webpage. \circ CSS styles the webpage.

Features:

- Runs on the client's browser.
- Does not require the server after the page is loaded.
- Makes websites interactive and dynamic (e.g., drop-down menus, form validation).

3. Server-Side Scripting Languages

• Server-Side Scripting refers to scripts that run on the server to process and manage data before sending it to the client.

• **Purpose**: Process user input, interact with databases, and manage requests from the client.

Popular Server-Side Scripting Languages:

- **PHP**: 0 Widely used for dynamic websites.
 - Can connect to databases, process forms, send emails, etc.
- Python:
 - Used for web development with frameworks like Django or Flask.
 - \circ Easy to read and write.
- Node.js (JavaScript on the server-side):
 - Allows using JavaScript for both client-side and server-side scripting.

Features:

- Runs on the server.
- Processes user input and requests.
- Interacts with **databases** and returns the data to the client.
- Requires a web server to execute the code (e.g., Apache, Nginx).

4. Difference Between Client and Server-Side Scripting

Feature	Client-Side Scripting	Server-Side Scripting
Location	Runs on the user's browser	Runs on the web server
Purpose	Interactivity, user interface	Processing data, database access
Example Languages	JavaScript, HTML, CSS	PHP, Python, Node.js
Execution	Faster (no need to contact the server)	Slower (requires server request/response)
Security	Less secure (user can see the coo	More secure (code is hidden on the le) server)

Responsive Web Designing

1. What is Responsive Web Design?

- **Responsive Web Design (RWD)** is a design approach aimed at creating websites that **work well on all devices** (desktops, tablets, and smartphones).
- The website layout adjusts automatically to different screen sizes and orientations.

2. Importance of Responsive Design

• With increasing use of **mobile devices**, it is important for websites to look good and function well on any screen size.

Enhances user experience (UX) by providing a seamless experience across devices.

• **Improves SEO (Search Engine Optimization)** since search engines like Google prioritize mobile-friendly websites.

3. Key Principles of Responsive Web Design

- Fluid Grid Layout:
 - Instead of using fixed-width layouts, the width of elements is set in percentages.
 - \circ The layout changes in relation to the size of the screen.
- Flexible Images:
 - Images are also set to scale with the browser width.
 - Use CSS properties like max-width: 100% to ensure images resize properly.
- Media Queries:
 - Special CSS rules are applied based on the device's screen size and resolution.
 Media queries allow you to define specific styles for different devices (e.g., smartphones, tablets, desktops).

Example of Media Query:

```
css
CopyEdit
@media only screen and (max-width: 600px)
{ body {
    background-color: lightblue;
  }
}
```

This code changes the background color to light blue on screens that are 600px wide or smaller (like smartphones).

4. Techniques Used in Responsive Web Design

- Viewport Meta Tag:
 - Ensures the website scales correctly on mobile devices.
 - Typically placed in the HTML <head> section.

Example:

```
html
CopyEdit
<meta name="viewport" content="width=device-width,
initial-scale=1">
```

Flexbox and Grid Layouts:

- CSS layout systems that make it easier to create flexible and adaptive designs.
 Flexbox allows for items to adjust and align in a flexible manner.
 Grid Layout divides the page into rows and columns, making it easier to manage layout on different screen sizes.
- **Responsive Navigation**: On smaller screens, a **hamburger menu** (a three-line menu icon) is often used to collapse the navigation links into a drop-down menu.

5. Advantages of Responsive Web Design

- **One Website for All Devices**: No need to create separate websites for desktop and mobile.
- Improved User Experience: Consistent layout and navigation across devices.
- Faster Development & Maintenance: A single codebase for all devices, reducing work and complexity.
- Better SEO: Google rewards responsive websites in search rankings.

6. Challenges of Responsive Web Design

- **Complexity**: It can be difficult to design for all devices, especially if they have very different screen sizes.
- **Performance**: Making websites responsive might require more CSS and JavaScript, which could affect load times.

Downloading Free Editors

1. What is a Text/Code Editor?

- A **Text/Code Editor** is a program used for writing and editing source code for web development (HTML, CSS, JavaScript, etc.).
- It is an essential tool for front-end and back-end development.
- Some popular code editors offer advanced features like **syntax highlighting**, **autocompletion**, and **error checking** to help developers write code more easily and efficiently.

2. Why Download Free Editors?

- Free editors allow beginners and experienced developers to start coding without spending money on expensive software.
- Most free editors come with powerful features, plugins, and community support.
- They are usually lightweight and easy to set up.

3. Popular Free Editors

1. VS Code (Visual Studio Code)

Developed by Microsoft.

- Features include:
 - Syntax highlighting for many programming languages. Extensions for adding features (e.g., linters, debuggers).
 - Integrated Git control and debugging tools. **Platform**: Windows, macOS, Linux.

Download Link: VS Code Official Download

2. Sublime Text

- Lightweight and fast text editor.
- Features include:
 - "Goto Anything" to quickly jump to files, symbols, or lines.
 - Plugin support through **Package Control**.
 - **Platform**: Windows, macOS, Linux.

Download Link: Sublime Text Official Download

3. Atom

- Developed by GitHub.
- Features include:
 - Easy-to-use interface with many community-driven packages.
 - Supports collaborative coding with **Teletype**. **Platform**: Windows, macOS, Linux.

Download Link: Atom Official Download

4. Notepad++

- Lightweight text editor for coding, especially for **Windows**.
- Features include:
 - Syntax highlighting and code folding. Supports plugins for additional features.
 - **Platform**: Windows.

Download Link: Notepad++ Official Download

5. Brackets

- Focused on web development (HTML, CSS, JavaScript).
- Features include:
 - Live Preview (see changes in real-time).
 - Inline Editors for CSS and JavaScript.
 Platform: Windows, macOS, Linux.

Download Link: Brackets Official Download

4. How to Download and Install Free Editors

- Visit the official website of the editor.
- Look for the **Download** button (usually clearly visible).
- Choose the appropriate version for your **operating system** (Windows, macOS, or Linux).
- Follow the installation instructions provided by the editor's website or installer.

5. Additional Tools to Enhance Your Editor Experience

- **Extensions/Plugins**: Most free editors support extensions (e.g., themes, code snippets, debugging tools) that improve productivity.
- Version Control Integration: Many editors allow integration with Git for version control, making it easier to track code changes.

Key Points to Remember

Editor	Platform	Key Feature
VS Code	Windows, macOS, Linux	Extensions, Git integration, debugging
Sublime Text	Windows, macOS, Linux	Fast, customizable, plugin support
Atom	Windows, macOS, Linux	Open-source, collaborative coding
Notepad++	Windows	Lightweight, syntax highlighting
Brackets	Windows, macOS, Linux	Live preview for web development