



Prepared by
Eng. Mohamed Saied Hassan

Web Development (part one)

Introductory session



Introduction to Web Development

Lecture 00

Prepared by
Eng. Mohamed Saied Hassan

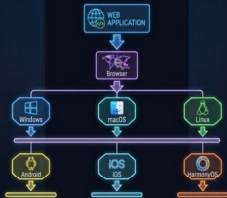
From the Web to Modern Web Applications

- Why Web Applications?
- What is the Web?
- What is a Website?
- What is Web Development?
- Frontend vs Backend
- Course Roadmap

Why Web Applications?

Build Once, Reach Everywhere

- **One web application** can serve users across many operating systems.
- Users usually don't need to install a dedicated application.
- Users can access it through common browsers such as Chrome, Edge, Safari
- Updates can be deployed to the server without requiring every user to download a new version.
- Easier to share — send a **URL**.
- The same application can potentially reach desktop, tablet, and mobile users.



Windows
Application



macOS
Application



Linux
Application



Android
Application



iOS
Application

What is the Internet?

The Internet = A global network of connected devices

- Computers
- Servers
- Smartphones
- Routers

The Internet is the infrastructure that allows devices to communicate.

What is the World Wide Web?

The Web (WWW) is a system of linked documents and resources invented by Tim Berners-Lee in 1989 and opened to the public in 1991.

Resources Examples

- Websites
- Web pages
- Web applications
- Online services

Internet $\neq$ Web

The Web is one service that runs on the Internet.



What is a Website?

website is a collection of related web pages available through the Web.

Examples:

- University website
- Company website
- Portfolio
- News website
- Restaurant website

A website can contain:

- Text
- Images
- Videos
- Links
- Forms
- Interactive elements

What is a Web Page?

A **web page** is a single document that you can open in a web browser..

Examples

A Home page

An About Us page

A Login page

A Contact page

Each one can be a separate web page.

When you visit:

<https://example.com/about>

The `/about` page is a web page.

A web page is usually built using:

`</>`

HTML → structure and
and content

`🎨`

CSS → design and
appearance

`{ }`

JavaScript → behavior
and interactivity

What is Web Development?

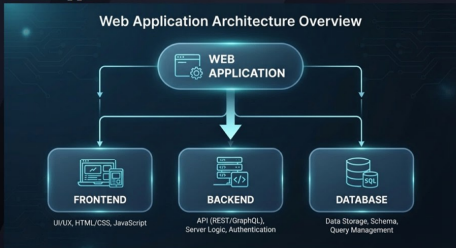
Web Development is the process of designing, building, testing, deploying, and maintaining websites

Software Development Life Cycle



The Three Main Parts

A typical web application consists of



The Three Main Parts



FRONTEND:

What the user sees and interacts with.



BACKEND:

The server-side logic behind the application.



DATABASE:

Where application data is stored.

DATABASE: Where application data is stored.

Frontend Development

Frontend = User Interface + User Interaction

It runs primarily in the user's browser.

- Buttons
- Menus
- Forms
- Cards
- Tables
- Animations
- Pages

Main technologies:

- HTML
- CSS
- JavaScript

Modern tools:

- Bootstrap
- React
- Angular
- Vue

HTML, CSS & JavaScript

Think of a web page as a building:

HTML:

What is on the page?

CSS:

How does it look?

JavaScript:

How does it behave?

THE WEB DEVELOPMENT TRIAD



Backend Development

Backend = Server-side logic

The backend can handle:

- Business logic
- Authentication
- Authorization
- Data processing
- APIs
- Database communication
- File processing
- Notifications

Popular technologies:

- ASP.NET Core
- Node.js
- Django
- Spring Boot
- Laravel

Database

The database stores application data

Examples:

- Users
- Products
- Orders
- Courses
- Students
- Messages
- Payments

Popular databases:

- SQL Server
- PostgreSQL
- MySQL
- MongoDB
- Oracle

Lecture 1 — Web & HTML Foundations

- How the web works (client/server, browsers, VS Code, Live Server, Dev Tools basics)
- HTML document structure, semantic tags (header, nav, main, section, footer)
- Text elements, links, images, lists
- Forms and form inputs (intro)
- Mini project: static personal profile page “pure un styled HTML”

Lecture 2 — HTML + CSS

- Recap
- Tables, forms in depth (validation attributes, input types) •
- Accessibility basics (alt text, labels, semantic structure)
 - CSS syntax, selectors, specificity
 - Box model (margin, border, padding, content)
 - Flexbox
 - Live demo (nav, cards, basic responsive)
 - Task “fully style their project pages”

Lecture 3 — HTML + CSS + Bootstrap

- Positioning (static, relative, absolute, fixed, sticky)
- Animation
- Bootstrap setup ,grid system
- Utility classes
- Core components
- Wrap-up
- Task rebuild their own project with Bootstrap

Lecture 4 — JavaScript Fundamentals I

- Variables
- Data types
- Operators
- Conditions
- Loops
- Functions
- Scope

Lecture 5 — JavaScript + DOM

- DOM
- Selecting elements
- Manipulation
- Events

Lecture 6 — JavaScript + OOP + BOM

- Objects
- Classes
- OOP
- BOM
- localStorage
- sessionStorage
- Project

Lecture 7 — Modern JavaScript (ES6+) + API

- Recap
- Arrow functions, template literals, destructuring, spread/rest
- Array methods (map, filter, reduce, forEach)
- Modules, promises/async-await, fetch API
- Live build — small app using fetch + storage
- Wrap-up + Project Task

Lecture 8 — React I — Foundations

- Components
- JSX
- Props
- Composition
- Vite

Lecture 9 — React II — State & Effects

- useState
- useEffect
- useContext
- API integration

Lecture 10 — React III — Routing

- React Router
- Routes
- Navigation
- Parameters
- Final Project

What We Will Build?

Final Project:

small real-world web application containing:

- Multiple pages
- Responsive UI
- Reusable React components
- Forms
- User interaction
- API communication
- State management
- Routing

By the end of the course
You will understand how the pieces fit together
rather than learning each technology independently.

**In the next lecture, we start building the
frontend from the ground up — beginning with
HTML**

Thank You