

LECTURE 1

Introduction to Web Development

From what the web actually is, to building full applications anyone can open in a browser.

CSC336 Web Technologies



In This Lecture

- 1 The difference between the Internet and the Web
- 2 The client–server request–response model behind every website
- 3 Core vocabulary: URL/URI, HTTP/HTTPS, DNS, hosting, web server, browser
- 4 The organizations that create and maintain web standards
- 5 Types of web apps: static, dynamic, MPA, SPA, and PWA
- 6 The full technology landscape — where HTML, CSS, JS, React, Node and MongoDB fit

The Internet vs. the Web

They are not the same thing.

THE INTERNET

- A global network of interconnected computers that exchange data.
- Exists since the late 1960s (ARPANET).
- Carries far more than websites: email, video calls, games, file transfer.
- The roads, cables and traffic rules of the digital world.

THE WEB (WORLD WIDE WEB)

- One service that runs on top of the Internet.
- Invented in 1989 by Tim Berners-Lee.
- Built from three ideas: HTML documents, URLs, and the HTTP protocol.
- Just one kind of “mail” sent through the Internet's postal system.

The Web = Three Simple Ideas

1

Documents

HTML

Pages written in HyperText Markup Language that can link to one another.

2

Addresses

URLs

A unique address that identifies where each document lives.

3

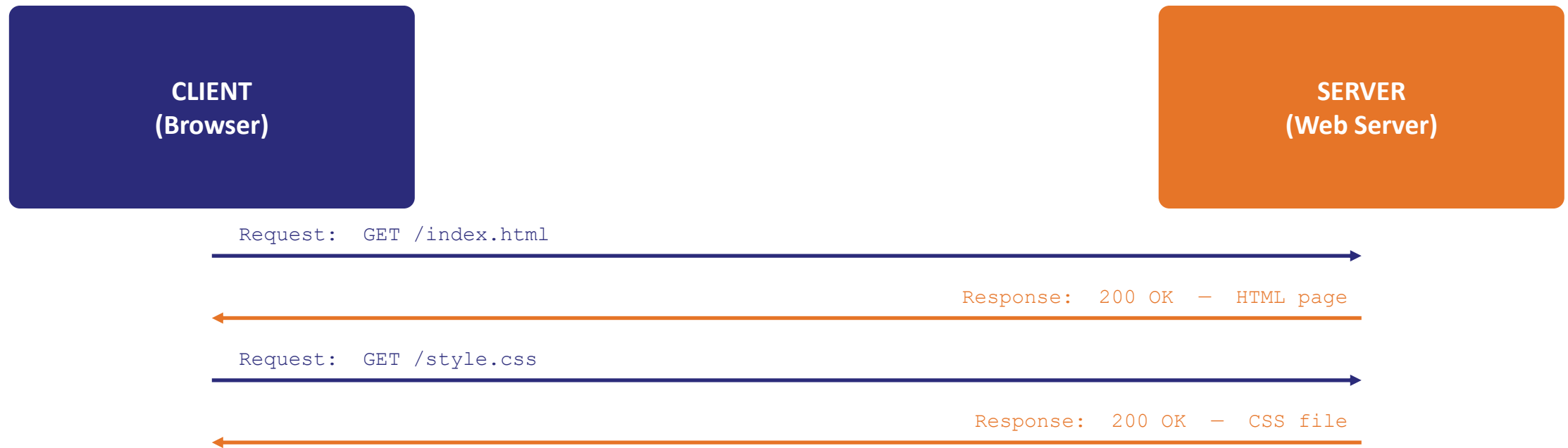
A Protocol

HTTP

The rules for requesting a document and receiving it back.

The Client–Server Request–Response Model

A client asks for something. A server answers. The server does nothing until asked.



One web page can take many request–response exchanges — one for the HTML, then the CSS, images, and more.

Anatomy of a URL

A URI identifies a resource. A URL is a URI that also says where it is and how to fetch it.

```
https://www.example.com:443/courses/web-tech?semester=fall#lecture1
```

<code>https://</code>	scheme	Which protocol to use (here, secure HTTP)
<code>www.example.com</code>	host	Which server to contact
<code>:443</code>	port	Often omitted — 443 is the default for HTTPS
<code>/courses/web-tech</code>	path	Which resource on that server
<code>?semester=fall</code>	query	Extra parameters, as key=value pairs
<code>#lecture1</code>	fragment	A specific spot within the page

HTTP and HTTPS

HTTP

- The rules for how a browser and server exchange requests and responses.
- Defines methods: GET (fetch data), POST (send data), and more.
- Defines status codes: 200 OK, 404 Not Found, 500 Server Error.

HTTPS

- HTTP plus an encryption layer (TLS/SSL).
- Scrambles data so eavesdroppers on the network cannot read it.
- Now used by almost every site; browsers warn when a site is not on HTTPS.

Never send passwords, card numbers or personal data over plain HTTP — it can be intercepted. Always use HTTPS.

DNS — The Phone Book of the Internet

Computers connect using numeric IP addresses. DNS translates a name into that number.

1

You type **example.com** into the browser's address bar

2

The DNS resolver **looks up the name** in the Internet's distributed directory

3

DNS returns **93.184.216.34** the IP address of the server that hosts the site

4

The browser connects **to that IP** and the HTTP request–response begins

Hosting, Web Servers, and Browsers

Hosting

Storing your site's files on a computer connected to the Internet 24/7, so anyone can reach it any time.

Vercel • Netlify • AWS • DigitalOcean

Web Server

The software that listens for HTTP requests and sends back responses.

Apache • Nginx • Node.js + Express

Browser

The client the user interacts with: sends requests, receives HTML/CSS/JS, renders the page.

Chrome • Firefox • Safari • Edge

Web Standards Bodies

Standards let every browser interpret HTML, CSS and JavaScript the same way.

W3C

World Wide Web Consortium

Standards for HTML, CSS, and accessibility (WCAG). Founded by Tim Berners-Lee.

WHATWG

Web Hypertext Application Technology Working Group

Maintains the HTML Living Standard, updated continuously by browser vendors.

ECMA

Ecma International

Standardizes ECMAScript — the specification that JavaScript implements (ES2015, ES2020...).

IETF

Internet Engineering Task Force

Core Internet protocols: HTTP, TLS, DNS. Publishes specs called RFCs.

Types of Web Applications

Static

Serves the same HTML file to everyone. Nothing changes per user.

Portfolios, documentation

Dynamic

Builds or changes content per request, using data from a database.

Most real applications

MPA

Multi-Page App: every navigation loads a brand-new page from the server.

News sites, classic e-commerce

SPA

Single-Page App: loads once, then JavaScript swaps content in place.

Dashboards, social apps

PWA

An app-like SPA: works offline, installable, can send push notifications.

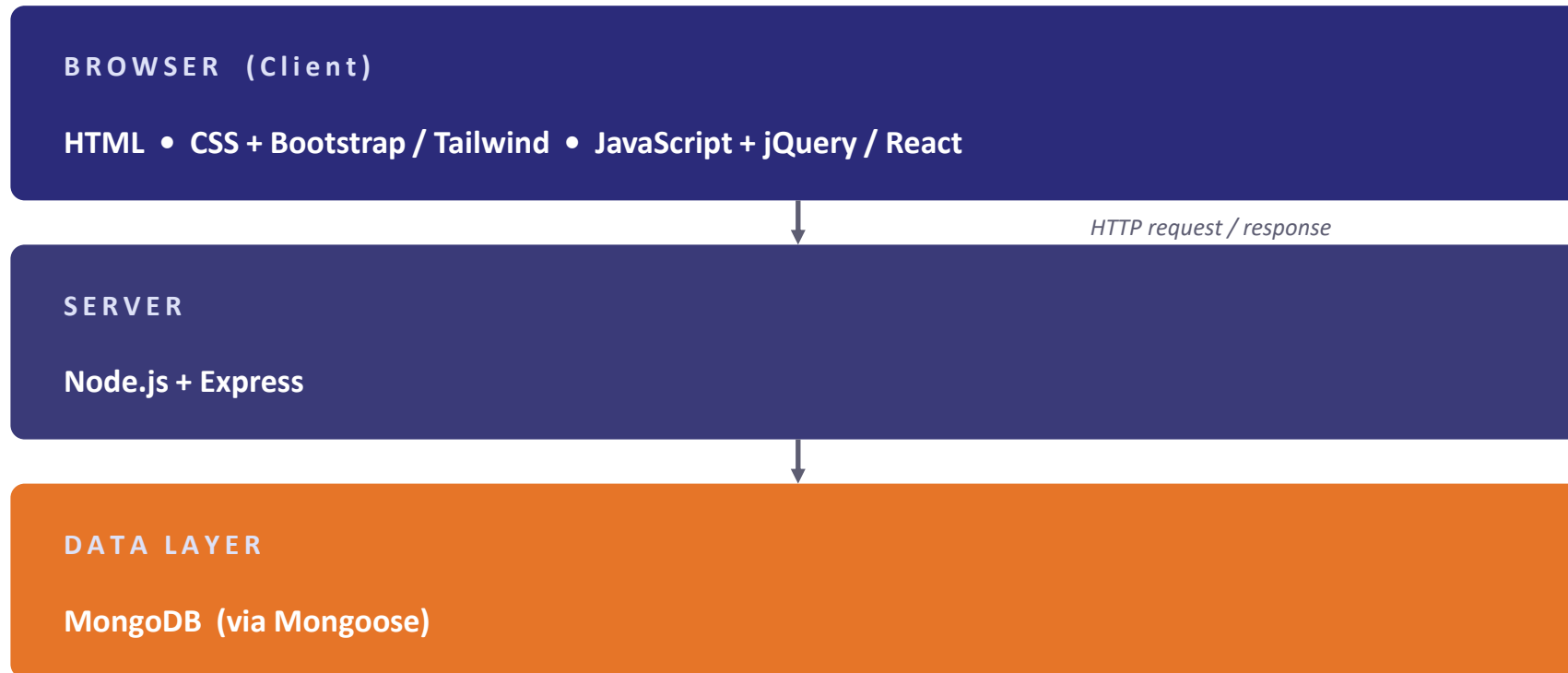
Apps needing offline access

Comparing the Types

Type	Content changes per user?	Full page reloads?	Typical use case
Static	No	Yes (content never changes)	Portfolio, documentation
Dynamic (MPA)	Yes	Yes, on every navigation	News sites, e-commerce
SPA	Yes	No, after the first load	Dashboards, social apps
PWA	Yes	No	Offline / installable apps

These categories overlap: an SPA is almost always dynamic, and a PWA is usually an SPA with extra powers.

The Technology Landscape



Nicknamed the MERN stack: MongoDB • Express • React • Node

This Course Picks One Path Per Layer

Markup

HTML

alt: — (no real alternative in the browser)

Styling

CSS + Bootstrap or Tailwind

alt: Sass, Bulma, UnoCSS, ...

Scripting

JavaScript (ES6+), plus jQuery

alt: TypeScript

Front-end framework

React

alt: Vue, Angular, Svelte, SolidJS

Server runtime

Node.js + Express

alt: Django, Rails, Laravel, Spring Boot

Database

MongoDB + Mongoose

alt: PostgreSQL, MySQL, SQLite, Firebase

REALITY CHECK

What You Actually Need to Know

You are not expected to learn everything on the previous slide. Go deep on this stack; recognize the rest.

Solid — non-negotiable

HTML, CSS, JavaScript (ES6+), Git & GitHub, REST API design

Solid — the MERN core

React, Node.js + Express, MongoDB + Mongoose

One, deeply — not both

A CSS framework: Bootstrap OR Tailwind

Awareness only

jQuery — enough to read and maintain older codebases

Nice to have

TypeScript; basic SQL/PostgreSQL; a second front-end framework

The pick-one pattern: depth in one option per category + awareness of the rest is what makes you employable.

KEY TAKEAWAYS

Lecture 1 in Six Points

- 1 The Internet is the global network; the Web is one service (pages, links, HTTP) running on top of it.
- 2 The web works through a client–server, request–response model — browsers ask, servers answer.
- 3 URL/URI identify resources, HTTP/HTTPS transfer them, DNS resolves names to IP addresses, hosting keeps a server reachable.
- 4 W3C, WHATWG, ECMA and IETF write the standards that keep browsers and developers speaking one language.
- 5 Web apps range from static sites to dynamic ones, built as MPAs, SPAs, or installable PWAs.
- 6 This course teaches the MERN stack (+ Bootstrap/Tailwind, jQuery) — depth in one path, awareness of the alternatives.