

## HTML

- <!DOCTYPE html>
- <html lang="en">
- <head><meta charset="UTF-8"><title>Title</title></head>
- <body>...</body></html>
- Common Tags: <h1>-<h6>, <p>, <a>, <img>, <form>, <ul>/<ol>/<li>
- Semantic Elements: <header>, <nav>, <section>, <article>, <footer>

## CSS

- Selectors: .class, #id, tag
- Box Model: width, padding, border, margin
- Flexbox: display:flex, justify-content, align-items
- Grid: display:grid, grid-template-columns, gap

## Bootstrap

- Containers: .container, .container-fluid
- Grid: .row > .col-md-4, .col-sm-6
- Components: .btn.btn-primary, navbar, cards, modals

## JavaScript

- ES6: const add = (a,b) => a + b
- DOM: document.querySelector, addEventListener
- Fetch: fetch('/api').then().then()
- Array Methods: .map(), .filter(), .reduce(), .includes()

## Angular

- @Component({ selector: 'app-root', templateUrl })
- Two-way binding: [(ngModel)]="title"
- HTTPClient: this.http.get('/api')

## **React**

- Function Component: `const App = () => <h1>Hello</h1>;`
- JSX: `return (<div>{value}</div>)`
- Hooks: `useState`, `useEffect`
- Props and State

## **Next.js**

- Pages Routing: `pages/index.js`
- Data Fetching: `getStaticProps`, `getServerSideProps`
- API Routes: `pages/api/hello.js`

## **Node.js**

- Create Server: `require('http').createServer()`
- Read File: `fs.readFile()`
- `npm init`, `require` modules

## **Express.js**

- Setup: `const express = require('express');`
- Routing: `app.get('/')`, `app.post('/')`
- Middleware: `app.use()`, `next()`

## **MongoDB**

- Connect: `MongoClient.connect(url)`
- CRUD: `insertOne`, `find`, `updateOne`, `deleteOne`
- Mongoose Schema and Model

## **MySQL**

- Connect: `mysql.createConnection()`
- Query: `SELECT`, `INSERT`, `UPDATE`, `DELETE`

- CRUD with Node.js using mysql package