

Git & Version Control

1. Git Basics

- a) Git Init, Clone, Add, Commit, Push, Pull
- b) Branching & Merging
- c) Merging vs Rebasing
- d) Git Stash & Cherry-Pick
- e) Undoing Changes (Reset, Revert)
- f) Git Log & Blame
- g) Git Tags & Releases

2. Collaboration

- a) Git Pull Requests (PRs) & Code Review Process
- b) Resolving Merge Conflicts
- c) Forking & Upstream Workflow
- d) Tools: GitHub, GitLab, Bitbucket

HTML

1. HTML Fundamentals

- a) HTML Syntax & Structure
- b) DOCTYPE Declaration
- c) Semantic Tags (<article>, <section>, <header>, etc.)
- d) Forms and Input Types
- e) Validation Attributes
- f) HTML5 APIs:
 - g) - Web Storage (localStorage, sessionStorage)
 - h) - Geolocation API

CSS

1. Core Concepts

- a) Box Model
- b) Selectors & Specificity
- c) Positioning (static, relative, absolute, fixed, sticky)

- d) Display: Block, Inline, Flex, Grid

2. Layout and Styling

- a) Flexbox & Grid Layouts
- b) Media Queries (Responsive Design)
- c) Pseudo-classes/elements
- d) Transitions, Animations, Keyframes

3. Advanced

- a) CSS Variables
- b) Preprocessors: SASS/SCSS
- c) Utility-first CSS (e.g., Tailwind Basics)

JavaScript (JS)

1. Core Concepts

- a) Data Types & Variables (var, let, const)
- b) Operators, Control Flow (if, switch, loops)
- c) Functions (Regular, Arrow)
- d) Closures, Hoisting, Lexical Scope
- e) The 'this' keyword (context resolution)
- f) call(), apply(), bind()

2. Objects & Prototypes

- a) Object Methods & Object.defineProperty()
- b) Prototypes & Prototype Chain
- c) Extending Objects
- d) Inheritance (ES5 & ES6)
- e) OOP in JS

3. Asynchronous JS

- a) Event Loop
- b) Callbacks, Promises, Async/Await

4. DOM & Events

- a) DOM Traversal & Manipulation
- b) Event Binding
- c) Event Capturing & Bubbling
- d) Forms & Input Handling
- e) DOM Parsing

TypeScript (TS)

1. Basics

- a) Type Annotations & Inference
- b) Primitive Types, Union, Intersection
- c) Arrays, Tuples, Enums, any, unknown, void, never

2. Advanced

- a) Interfaces vs Types
- b) Generics
- c) Type Assertions
- d) Classes, Inheritance, Access Modifiers
- e) Utility Types (Partial, Pick, etc.)
- f) Namespaces & Modules

React

1. Core Concepts

- a) JSX
- b) Functional vs Class Components
- c) Props and State
- d) Event Handling
- e) Conditional & List Rendering
- f) Controlled vs Uncontrolled Components

2. React Hooks

- a) useState, useEffect, useRef
- b) useCallback, useMemo
- c) useContext, useReducer
- d) Custom Hooks

3. Advanced Topics

- a) Context API
- b) Error Boundaries
- c) Lazy Loading & Code Splitting
- d) Ref Forwarding
- e) Portals
- f) HOCs & Render Props

4. Routing & State Management

- a) React Router (v6+)
- b) Redux (Toolkit), Zustand, Context for State
- c) Forms (Formik, React Hook Form)
- d) E. Testing in React
- e) React Testing Library
- f) Jest Mocks & Coverage

Node.js & Express

1. Node.js Core

- a) Node Architecture
- b) Modules (CommonJS & ES Modules)
- c) File System (fs)
- d) EventEmitter
- e) Buffers & Streams
- f) Environment Variables

2. Express.js

- a) Express Setup
- b) Routes (GET, POST, PUT, DELETE)
- c) Middleware (built-in, custom, third-party)
- d) Serving Static Files
- e) Error Handling
- f) REST API Design

3. Advanced

- a) Authentication (JWT, OAuth)
- b) Authorization & Role-Based Access
- c) MongoDB with Mongoose
- d) Input Validation (Joi, express-validator)
- e) Async/Await & Error Handling
- f) Rate Limiting & Security (helmet, rate-limiter)
- g) Socket.IO (WebSockets)
- h) Deployment (Heroku, Render, Railway)

Cloud Services

1. AWS Essentials

- a) S3 (Static File Hosting)
- b) EC2 (Virtual Servers)
- c) Lambda (Serverless Functions)
- d) DynamoDB (NoSQL DB)
- e) Cognito (User Authentication)
- f) SQS/SNS (Message Queuing)

2. DevOps Integration

- a) CI/CD Pipelines with GitHub Actions, GitLab, Jenkins
- b) Environment Variables & Secrets Management
- c) Docker (Containerization Basics)

Unit Testing

1. JavaScript & Node Testing

- a) Jest (Unit, Mocks, Spies)
- b) Mocha + Chai
- c) Jasmine Syntax (Basics)

2. React Testing

- a) React Testing Library
- b) Snapshot Testing
- c) Integration & E2E Testing (Cypress / Playwright)

System Design

1. Design Principles

- a) SOLID
- b) DRY, KISS, YAGNI

2. Design Patterns

- a) Singleton, Factory, Observer, Strategy, Decorator
- b) MVC / MVVM Patterns

3. Scalable Architecture

- a) Monolith vs Microservices
- b) Caching (Redis, CDN)
- c) Load Balancing
- d) API Rate Limiting
- e) Database Scaling (Sharding, Replication)

4. SDLC & Agile

- a) SDLC Phases (Requirement, Design, Dev, Test, Deploy)
- b) Agile (Scrum/Kanban), Waterfall
- c) Tools: Jira, Azure DevOps, Rally
- d) Versioning, Release Pipelines
- e) EngX Bootcamp, Clean Code/Design Practices

Soft Skills & Mindset

- a) Communication (Technical & Non-Technical)
- b) Team Collaboration
- c) Time & Task Management
- d) Problem Solving Approach
- e) Ownership & Accountability
- f) Results-Driven Mindset
- g) Learning Agility
- h) Scalable & Startup-Oriented Thinking