



LeadSoft Placement Academy

Grow with us...



HTML, CSS, and JavaScript Training Syllabus (Chapter-Wise)

Chapter 1: Introduction to Web Development

- How websites work: Client vs Server
 - Introduction to browsers, servers, and developer tools
 - Overview of HTML, CSS, and JavaScript
 - Setting up the development environment (VS Code, Live Server)
-

Chapter 2: HTML – Structuring Web Pages

- Basic HTML Document Structure
 - Headings, Paragraphs, Lists, Images
 - Links and Anchor Tags
 - Tables and Semantic Tags
 - Forms and Input Types
 - Embedding Media (Audio, Video, iframe)
 - HTML5 Semantic Elements (<article>, <nav>, <section>, etc.)
-

Chapter 3: CSS – Styling Web Pages

- Introduction to CSS
 - Types of CSS: Inline, Internal, External
 - CSS Syntax and Selectors
 - Colors, Fonts, Text Styling
 - Backgrounds and Borders
 - Box Model (Margin, Padding, Border)
 - Display Types: block, inline, inline-block
-

Chapter 4: CSS Layout and Positioning

- Flexbox: One-dimensional layout
 - display: flex, justify-content, align-items, etc.
 - CSS Grid: Two-dimensional layout
 - grid-template-rows, grid-template-columns, gap, etc.
 - Float and Clear
 - Positioning: static, relative, absolute, fixed, sticky
 - Z-index and Stacking Context
 - Media Queries for Responsive Design
-

Chapter 5: CSS Transitions and Animations

- CSS Transitions
 - CSS Transform: rotate, scale, translate
 - Keyframe Animations
 - Hover and Active Effects
 - Animation Timing and Delays
-

Chapter 6: JavaScript Basics

- What is JavaScript?
 - Linking JavaScript to HTML
 - Variables: var, let, const
 - Data Types and Type Conversion
 - Operators and Expressions
 - Functions and Scope
 - Control Structures: if, else, switch, loops
-

Chapter 7: Arrays and Objects

- Declaring Arrays and Accessing Elements

- Array Methods: push(), pop(), shift(), map(), filter(), forEach()
 - Objects: Key-Value Pairs
 - Nested Objects and Arrays
 - Looping through Arrays and Objects
-

Chapter 8: DOM Manipulation

- Understanding the DOM
 - Selecting Elements: getElementById, querySelector, etc.
 - Changing Text, HTML, and CSS with JavaScript
 - Creating, Appending, Removing Elements
 - DOM Traversal
-

Chapter 9: Events and Event Handling

- Understanding Events
 - Event Listeners: addEventListener()
 - Mouse Events: click, mouseover, mouseout
 - Keyboard Events
 - Form Validation (on submit)
 - Event Bubbling and Delegation
-

Chapter 10: Local Storage and Browser APIs

- Understanding localStorage and sessionStorage
 - Setting, Getting, and Removing Items
 - JSON Parse and Stringify
 - Introduction to simple fetch API (optional)
-

Chapter 11: Mini Projects (Hands-On Implementation)

- **Form Validation Project (HTML + CSS + JS)**

- **To-Do List Project** (with Local Storage)
 - **Basic Calculator** (Event Handling + DOM)
 - **Responsive Portfolio Website** (HTML + CSS + JS)
-

Chapter 12: Deployment and Best Practices

- Folder structure and asset organization
- Hosting on GitHub Pages or Netlify
- Intro to SEO tags and Accessibility (a11y)
- Best practices in HTML, CSS, and JS
- Final Q&A and Portfolio Guidance



Advanced Web Development Syllabus (Chapter 13 Onwards)

Chapter 13: GSAP (GreenSock Animation Platform) with JavaScript

- Introduction to GSAP and its benefits over CSS animations
 - Setting up GSAP in projects (CDN and npm)
 - Basic Tween animations (`gsap.to()`, `gsap.from()`, `gsap.fromTo()`)
 - Timeline creation and control (`gsap.timeline()`)
 - Easing functions and stagger effects
 - Scroll-based animations with `ScrollTrigger` plugin
 - Animating SVGs and complex shapes
 - Performance optimization and best practices with GSAP
-

Chapter 14: Version Control with Git & GitHub

- Introduction to Version Control and Git fundamentals
- Installing and configuring Git
- Basic Git commands: `init`, `clone`, `add`, `commit`, `status`, `log`
- Branching and merging workflows
- Resolving merge conflicts
- Remote repositories and GitHub basics
- Pull requests, forks, and collaboration workflows

- Using GitHub Issues and Projects for task management
 - Best practices for commit messages and branching strategies
-

Chapter 15: Deployment Platforms — Vercel & Render

- Introduction to modern deployment and hosting services
 - Overview: Vercel and Render features & differences
 - Deploying static sites and React apps on Vercel
 - Deploying Node.js backend on Render
 - Environment variables and secrets management
 - Continuous deployment with GitHub integration
 - Custom domains and HTTPS setup
 - Monitoring deployments and logs
 - Rollbacks and troubleshooting deployment issues
-

Chapter 16: React.js – Core Concepts and Component-Based Architecture

- Introduction to React and Single Page Applications (SPA)
 - Setting up React project with Create React App or Vite
 - JSX syntax and rendering elements
 - Components and props
 - State and lifecycle methods (using useState, useEffect)
 - Handling events and forms in React
 - Conditional rendering and list rendering
 - Styling components (CSS Modules, Styled Components, inline styles)
-

Chapter 17: Advanced React.js Features

- React Router for client-side routing
- Context API for state management
- Custom hooks and hook rules
- Error boundaries and Suspense for lazy loading
- Performance optimization (memoization, React Profiler)
- Handling API calls and asynchronous data (fetch, axios)
- Testing React components with Jest and React Testing Library
- Integrating GSAP animations with React components

Chapter 18: Node.js and Backend Development

- Introduction to Node.js and runtime environment
 - Setting up Node.js project and package management (npm/yarn)
 - Creating a simple HTTP server with native Node.js
 - Express.js framework basics (routing, middleware)
 - RESTful API design and implementation
 - Handling requests and responses, parsing JSON and URL-encoded data
 - Working with databases (MongoDB or any SQL DB) – basic CRUD operations
 - Environment variables and configuration management
 - Error handling and logging
-

Chapter 19: Fullstack Integration and Project Workflow

- Connecting React frontend with Node.js backend
 - Using Axios or Fetch for API calls
 - Authentication basics (JWT or session-based)
 - Handling CORS and security best practices
 - Deployment of fullstack apps on Vercel and Render
 - Setting up CI/CD pipelines with GitHub Actions (basic overview)
 - Debugging and performance tuning
-

Chapter 20: Capstone Project

- Planning and designing a fullstack web app with animations (GSAP)
- Implementing frontend UI with React and animations
- Building backend APIs with Node.js and Express
- Version control workflow with Git & GitHub collaboration
- Deploying frontend and backend on Vercel and Render
- Presentation, code review, and iterative improvements

Database Syllabus: MySQL & MongoDB (Chapter 21 Onwards)

Chapter 21: Introduction to Databases

- What is a Database?

- Types of Databases: Relational vs NoSQL
 - Overview of MySQL and MongoDB
 - Use cases for SQL and NoSQL databases
 - Installing MySQL and MongoDB locally or using cloud services
-

Chapter 22: MySQL Fundamentals

- Introduction to Relational Database Concepts
 - MySQL Architecture and Components
 - Creating Databases and Tables
 - Data Types in MySQL
 - Primary Keys, Foreign Keys, and Constraints
 - Basic SQL Queries: SELECT, INSERT, UPDATE, DELETE
-

Chapter 23: Advanced MySQL Queries

- Filtering Data with WHERE, LIKE, IN, BETWEEN
 - Sorting Data with ORDER BY
 - Aggregate Functions: COUNT, SUM, AVG, MAX, MIN
 - Grouping Data with GROUP BY and HAVING
 - Joining Tables: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN
 - Subqueries and Nested Queries
-

Chapter 24: MySQL Database Design and Optimization

- Normalization and Denormalization
 - Indexes and their types
 - Views and Stored Procedures
 - Transactions and ACID properties
 - Backup and Restore Strategies
 - User Management and Security
-

Chapter 25: Introduction to MongoDB

- What is MongoDB and NoSQL?
- Document-oriented database basics
- BSON format and JSON-like documents

- MongoDB Architecture
 - Installing and running MongoDB
 - MongoDB Shell and Compass UI
-

Chapter 26: CRUD Operations in MongoDB

- Creating Databases and Collections
 - Insert operations (insertOne(), insertMany())
 - Querying Documents (find(), findOne())
 - Updating Documents (updateOne(), updateMany(), \$set)
 - Deleting Documents (deleteOne(), deleteMany())
-

Chapter 27: MongoDB Advanced Queries

- Filtering with Comparison and Logical Operators
 - Querying Nested Documents and Arrays
 - Projection to include/exclude fields
 - Aggregation Framework: \$match, \$group, \$project, \$sort
 - Indexes in MongoDB
 - Text Search and Geospatial Queries
-

Chapter 28: Data Modeling and Schema Design in MongoDB

- Schema Design Principles for NoSQL
 - Embedded vs Referenced Data Models
 - Using Mongoose ODM (Overview)
 - Validation and Constraints in MongoDB
 - Working with Relationships and Data Integrity
-

Chapter 29: Integration and Use Cases

- Connecting MySQL with Backend (Node.js / Other)
 - Connecting MongoDB with Backend (Node.js / Other)
 - When to choose MySQL vs MongoDB
 - Hybrid database usage and syncing data
 - Example projects using MySQL and MongoDB
-

Chapter 30: Backup, Security, and Performance

- Backup and Restore in MongoDB
 - Security Best Practices for MySQL and MongoDB
 - User Roles and Authentication
 - Performance Tuning and Monitoring
 - Scaling Databases: Replication and Sharding in MongoDB
-

Chapter 31: Final Project & Practical Exercises

- Design and implement a sample application with MySQL backend
- Design and implement a sample application with MongoDB backend
- Compare query performance and design trade-offs
- Code review, optimization, and presentation

AngularJS & Node.js Syllabus (Chapter 32 Onwards)

Chapter 32: Introduction to AngularJS

- What is AngularJS? Overview and History
 - MVC Architecture in AngularJS
 - Setting up AngularJS Environment
 - Understanding Directives and Expressions
 - Data Binding: One-way and Two-way Binding
 - AngularJS Modules and Controllers
-

Chapter 33: AngularJS Core Concepts

- Scope and \$scope Object
 - Built-in Directives (ng-model, ng-repeat, ng-if, ng-show, ng-hide)
 - Filters and Custom Filters
 - Form Handling and Validation
 - Services and Factories
 - Dependency Injection in AngularJS
-

Chapter 34: AngularJS Routing and SPA Development

- Introduction to Single Page Applications (SPA)

- Setting up Routing with ngRoute Module
 - Configuring Routes and Templates
 - Nested Views and Multiple Views
 - Using \$routeParams for Dynamic URLs
 - Handling Route Change Events
-

Chapter 35: Advanced AngularJS Concepts

- Custom Directives and Reusable Components
 - Promises and \$http Service for AJAX Calls
 - Two-way Data Binding Deep Dive
 - Working with Events and Watchers
 - Animation with AngularJS
 - Debugging and Testing AngularJS Applications
-

Chapter 36: Introduction to Node.js

- What is Node.js? Overview and Features
 - Installing Node.js and NPM
 - Node.js Architecture: Event Loop and Non-blocking I/O
 - Creating a Simple HTTP Server
 - Understanding Modules and require()
-

Chapter 37: Node.js Core Modules

- File System (fs) Module Basics
 - Path Module
 - Events Module and EventEmitter
 - Working with Buffers and Streams
 - Timers and Process Objects
-

Chapter 38: Building Web Applications with Express.js

- Introduction to Express Framework
- Setting up Express Project
- Routing and Middleware
- Handling GET and POST Requests

- Serving Static Files
 - Using Template Engines (EJS, Pug)
-

Chapter 39: Working with Databases in Node.js

- Connecting Node.js to Databases (MySQL, MongoDB overview)
 - Performing CRUD Operations
 - Using Mongoose for MongoDB (if applicable)
 - Data Validation and Schema Design
 - Handling Database Errors
-

Chapter 40: RESTful API Development with Node.js

- REST API Principles and Best Practices
 - Building REST Endpoints with Express
 - Route Parameters and Query Strings
 - JSON Handling and Response Formatting
 - Authentication Basics (JWT / Sessions)
 - API Testing with Postman
-

Chapter 41: Real-time Applications with Socket.io

- Introduction to WebSockets and Socket.io
 - Setting up Socket.io in Node.js
 - Building Real-time Chat Application
 - Handling Events and Broadcasting
 - Managing Rooms and Namespaces
-

Chapter 42: Deployment and Best Practices

- Preparing AngularJS and Node.js apps for Production
 - Environment Variables and Configuration
 - Deployment on Platforms (Heroku, Vercel, Render)
 - Logging, Monitoring, and Error Handling
 - Security Best Practices for AngularJS and Node.js apps
-

Chapter 43: Final Project

- Designing a Fullstack Application using AngularJS frontend and Node.js backend
- Implementing Routing, REST APIs, and Real-time Features
- Testing and Debugging
- Deployment and Presentation

Leadsoft