SOFT TECH SOLUTION

WEB DEVELOPMENT COURSE — OFFICIAL SYLLABUS

Complete Syllabus, Learning Roadmap & Industrial Training Strategy

Mentored by MCA Software Engineers (Ex-Tata Power & Railworld)

- **Duration: 6 Months (24 Weeks)**
- **Schedule: Mon–Fri, 8:00 PM – 9:00 PM**
- **Sundays: Weekly Interview Prep & Assessment**
- **Approach: Fundamentals -> Exercises -> Projects -> Deploy**
- **Portal: softtechsolution.in/training**
- **Mode: Live Online + Mentor Code Reviews**

COURSE OBJECTIVE

To develop practical skills in frontend web development, database management, version control, AI-assisted coding, website deployment, and interview preparation. By the end of the course, learners should be able to design responsive websites, build interactive React applications, manage databases, use GitHub collaboratively, and deploy a project online.

MODULE 1: HTML3 FUNDAMENTALS & HTML5 — Duration: Weeks 1–2

Topics Covered:

- Introduction to the web, browsers, servers, and websites
- HTML structure, tags, elements, and attributes
- HTML3: legacy concepts and historical overview
- HTML5 document structure and modern standards
- Headings, paragraphs, links, images, and lists
- Tables, forms, inputs, buttons, and validation
- Semantic elements: header, nav, main, section, article, footer
- Audio, video, iframe, and embedded content
- HTML entities, metadata, favicon, and page titles
- Accessibility basics, SEO-friendly structure, and best practices

Practical Tasks:

- Create a personal profile webpage
- Build a registration form
- Develop a multi-page portfolio using HTML

Learning Outcome: Build structured, accessible web pages using modern HTML5.

MODULE 2: CSS — STYLING & RESPONSIVE DESIGN — Duration: Weeks 3–5

Topics Covered:

- CSS syntax, selectors, properties, and values (inline, internal, external)
- Colors, backgrounds, borders, shadows, and gradients
- Units: px, %, em, rem, vh, and vw | Typography, fonts, text alignment, and icons
- Box model, margin, padding, borders, display, position, z-index, overflow
- Flexbox and CSS Grid layout systems
- Pseudo-classes, pseudo-elements, combinators
- Transitions, transforms, animations, and keyframes
- Responsive design, media queries, and mobile-first layouts
- CSS variables, reusable styles, and browser developer tools

Practical Tasks:

- Design a responsive landing page
- Create a navigation bar and card layouts
- Build a responsive portfolio website
- Reproduce a sample webpage from a reference design

Learning Outcome: Design attractive, responsive websites for mobile, tablet, and desktop.

MODULE 3: JAVASCRIPT — PROGRAMMING & DOM

Duration: Weeks 6–10

Topics Covered:

• JavaScript setup, browser console, variables (var, let, const), data types
• Operators, type conversion, conditional statements, and switch
• Loops: for, while, do-while, for...of, and for...in
• Functions, parameters, return values, arrow functions, scope, closures, hoisting
• Arrays, objects, destructuring, and spread/rest operators
• Array methods: map, filter, find, reduce, and forEach
• DOM selection, manipulation, event handling, and form validation
• ES6+ modules, error handling (try/catch), JSON, localStorage, sessionStorage
• Callbacks, promises, async/await, event loop, Fetch API & REST fundamentals

Practical Tasks:

- Calculator application
- To-do list with localStorage persistence
- Interactive form validation system
- Search and filter interface
- Live weather application using a public REST API

Learning Outcome: Write interactive JavaScript applications and master asynchronous programming.

MODULE 4: GIT & GITHUB — VERSION CONTROL

Duration: Week 11

Topics Covered:

• Git vs GitHub: core differences and industry use cases
• Repository initialization (git init, status, add, commit, log)
• Branching: git branch, checkout/switch, and merge
• Remote repositories, origin, git push, pull, fetch, and clone
• Pull requests, code review basics, merge conflicts and resolution
• .gitignore, README.md, branching workflows, and commit conventions

Practical Tasks:

- Push all previous assignments to GitHub
- Create feature branches and merge changes
- Resolve a simulated merge conflict
- Maintain a clean project repository with documentation

Learning Outcome: Manage project code safely and collaboratively using Git and GitHub.

MODULE 5: MYSQL, MYSQL WORKBENCH & XAMPP

Duration: Weeks 12–14

Topics Covered:

• Database fundamentals: relational schemas, tables, rows, columns
• MySQL installation, Workbench setup, and XAMPP local environment
• Starting/stopping services, database & table creation, constraints, keys
• CRUD queries: INSERT, SELECT, UPDATE, DELETE with WHERE, ORDER BY, GROUP BY, HAVING
• Aggregate functions, aliases, Joins (INNER, LEFT, RIGHT, Self joins)
• Subqueries, views, indexes, normalization, transactions, and ACID basics
• Database design, ER diagrams, import, export, backup, and restore

Practical Tasks:

- Create a student management database schema
- Design an employee management database
- Write complex CRUD queries and multi-table joins
- Build an ER diagram and database schema

Learning Outcome: Design relational databases and write practical SQL queries using MySQL tools.

MODULE 6: CHATGPT & EFFECTIVE PROMPTING

Duration: Week 15

Topics Covered:

• Understanding AI assistants, strengths, and limitations
• Writing clear, specific prompts with context, requirements, and examples
• Prompting for HTML, CSS, JavaScript, React, and SQL
• Asking ChatGPT to explain code line by line, debugging errors with logs
• Generating test cases, edge cases, project documentation, and code reviews
• Security practices: avoiding sharing passwords, API keys, or private data

Practical Tasks:

- Convert a business requirement into a structured coding prompt
- Debug a JavaScript runtime error using AI assistance
- Generate SQL queries and validate their results
- Use ChatGPT to plan and document a mini-project

Learning Outcome: Use AI productively while understanding and verifying code rather than blindly copying.

MODULE 7: REACT.JS — FRONTEND DEVELOPMENT

Duration: Weeks 16–20

Topics Covered:

- React component-based architecture, Node.js, npm, and dev server
- JSX syntax, functional components, props, and composition
- State management with useState, event handling, and conditional rendering
- List rendering with keys, forms, and controlled components
- useEffect and component lifecycle concepts, Fetch API & integration
- Loading, error, and empty states; React Router & multi-page navigation
- Context API (useContext), custom hooks, project structure, and performance

Practical Tasks:

- Build reusable component design library
- Create a product listing and live search interface
- Build a multi-page React web application
- Connect React frontends to external REST APIs

Learning Outcome: Build maintainable, component-based frontend applications using React.js.

MODULE 8: POSTGRESQL & NEON DATABASE

Duration: Week 21

Topics Covered:

- PostgreSQL fundamentals, comparison with MySQL, and data types
- Neon serverless PostgreSQL introduction and project creation
- Connecting using connection strings and pooling concepts
- Tables, constraints, relationships, CRUD queries, joins, and aggregation
- Environment variables, database credentials, security, and backups

Practical Tasks:

- Create a Neon serverless PostgreSQL database
- Design tables for the final capstone project
- Run CRUD queries and verify results
- Configure database credentials securely

Learning Outcome: Create and manage cloud-hosted PostgreSQL databases.

MODULE 9: VERCEL — WEBSITE HOSTING & DEPLOYMENT

Duration: Week 22

Topics Covered:

- Hosting, domains, DNS, and HTTPS fundamentals
- Static hosting vs serverless execution; Connecting GitHub to Vercel
- Deploying React applications, build commands, and output directories
- Environment variables and production configuration
- Preview vs Production deployments, auto-deploy on git push, logs & debugging

Practical Tasks:

- Deploy a React project to Vercel
- Configure environment variables in Vercel project settings
- Push to GitHub and verify automatic redeployment
- Test live website responsiveness on mobile, tablet, and desktop

Learning Outcome: Publish websites to production and understand modern CI/CD deployment workflows.

MODULE 10: FINAL PROJECT & PORTFOLIO

Duration: Week 23

Topics Covered:

- Capstone Project: Employee Management System / Job Portal
- Responsive interface using HTML, CSS, and React.js
- Reusable React components, page navigation, forms, validation, and filtering
- Database design using PostgreSQL on Neon with CRUD access layer
- GitHub repository documentation and live Vercel deployment

Practical Tasks:

- Requirement analysis and UI planning
- Database schema and component architecture design
- Frontend implementation and API integration
- Live deployment on Vercel with presentation demonstration

Learning Outcome: Complete and present a practical project demonstrating end-to-end full stack skills.

MODULE 11: REVISION, TESTING & FINAL ASSESSMENT

Duration: Week 24

Topics Covered:

- Comprehensive revision of HTML, CSS, JavaScript, and React
- SQL query practice, database design review, and GitHub workflow revision
- AI prompt-writing and code review exercises
- Final practical assessment, code improvements, and portfolio review
- Resume and LinkedIn project presentation discussion

Practical Tasks:

- Final practical assessment execution
- Complete code review and performance optimization
- Resume and portfolio project walkthrough

Learning Outcome: Consolidate all knowledge and prepare to explain and demonstrate projects with confidence.

WEEKLY INTERVIEW PREPARATION — EVERY SUNDAY (24 WEEKS)

- Sunday 1: HTML fundamentals and common interview questions
- Sunday 2: HTML5, forms, semantic elements, and accessibility
- Sunday 3: CSS selectors, box model, and specificity
- Sunday 4: Flexbox, Grid, and responsive design
- Sunday 5: CSS practical tasks and layout debugging
- Sunday 6: JavaScript variables, data types, and operators
- Sunday 7: Conditions, loops, and coding problems
- Sunday 8: Functions, scope, and closures
- Sunday 9: Arrays, objects, and array methods
- Sunday 10: DOM, events, and form validation
- Sunday 11: Promises, async/await, and API questions
- Sunday 12: Git commands, branching, push/pull, and conflicts
- Sunday 13: SQL fundamentals and CRUD queries
- Sunday 14: Joins, subqueries, indexes, and database design
- Sunday 15: ChatGPT prompting, debugging, and code review
- Sunday 16: React components, JSX, and props
- Sunday 17: State, events, and conditional rendering
- Sunday 18: Hooks and lifecycle concepts
- Sunday 19: Routing, forms, and API integration
- Sunday 20: React coding test and debugging
- Sunday 21: PostgreSQL and Neon database questions
- Sunday 22: Vercel deployment and GitHub integration
- Sunday 23: Final project review and project-based questions
- Sunday 24: Mock interview, technical assessment, and feedback

DAILY LEARNING STRATEGY (Monday–Friday | 8:00 PM – 9:00 PM)

- 8:00–8:10 PM: Revise previous lesson | 8:10–8:35 PM: Learn new concept
- 8:35–8:55 PM: Hands-on coding exercises | 8:55–9:00 PM: Summarize & note doubts
- Saturday: Self-practice & assignment work | Sunday: Dedicated interview preparation