urbancode
Training and Solutions

Fullstack Web Development

M E R N

www.urbancode.in

Web Development Fundamentals

- Understand how websites work and the roles of HTML, CSS, and
- JavaScript in web development.
- Learn how the internet functions, including how websites are hosted and served.
- Gain hands-on experience with Git, GitHub, and version control.
- Develop key troubleshooting and debugging skills to apply to your projects.

HTML 5

- HTML Expedition: Learn HTML syntax to structure your digital world.
- Base Camp Setup: Use HTML boilerplate and understand doctypes.
- Nesting Mastery: Apply indentation and nesting for clean code.
- Tag Treasures: Use HTML tags for headings and paragraphs.
- Conquer Lists: Create ordered and unordered lists.
- Capture Moments: Insert images with the <img> tag.
- Link Builder: Create page links with anchor tags.
- Multi-Page Adventure: Design interconnected web pages.
- Navigate HTML: Use best practices for effective coding.

CSS

- CSS Basics: Discover how cascading style sheets bring your website to life.
- Selector Skills: Master CSS selectors and properties for styling.
- Inline to External: Learn to apply CSS using inline, internal, and external methods.
- Best Practices: Follow CSS coding best practices for clean styles.
- Specificity Secrets: Understand CSS specificity and inheritance rules.
- Box Model Blueprint: Dive into the CSS Box Model for layout control.
- Inspect & Diagnose: Use developer tools to troubleshoot CSS.
- Positioning & Display: Explore CSS positioning and display techniques.
- Font Styling: Style text with CSS and web-safe fonts.
- Responsive Magic: Implement responsiveness using media queries.
- Float & Clear: Utilize float and clear properties for layout.
- Selector Mastery: Combine CSS selectors and grasp their priorities.

FLEXBOX

- Flexbox Origins: Learn why Flexbox was introduced to CSS.
- Display Flex: Understand display: flex for flexible layouts.
- Direction Control: Master flex-direction to control layout flow.
- Align & Justify: Position child items using alignment and justification.
- Sizing & Distribution: Use the flex shorthand to control child item sizing.

GRIDS

- Grid vs. Flexbox: Explore differences and strengths of Grid layout.
- Grid Display: Use display: grid for structured designs.
- Grid Sizing: Create and arrange grids with tracks and gaps.
- Positioning: Place items across rows and columns.
- Combine Forces: Mix Grid with Flexbox for complex layouts.

BOOTSTRAP

- Native vs. Frameworks: Explore the differences between CSS tooling and external frameworks.
- Bootstrap Installation: Learn to add Bootstrap to your website for rapid styling.
- 12-Column Layout: Master the Bootstrap grid system for flexible layouts.
- Components Galore: Use Bootstrap's buttons, carousels, cards, and navbars.
- Icon Access: Incorporate designer icons from Bootstrap’s collection.
- Quick Builds: Use code snippets, examples, and templates to create stunning, mobile-first websites.

WEB DESIGN

- Design Principles: Master the fundamentals of web design, UI, and UX.
- Color Theory: Explore modern color palettes for a professional website look.
- Typography: Choose stylish, free-for-commercial-use fonts for a user-friendly design.
- UI Focus: Learn to manage user attention using key UI design principles.
- UX Mastery: Design user-friendly websites by understanding core UX concepts.

DOCUMENT OBJECT MODEL(DOM)

- HTML Tree: Explore the tree structure of HTML-based websites.
- DOM Navigation: Traverse the document using object notation.
- Separation of Concerns: Apply coding best practices for clean, maintainable code.
- Element Manipulation: Change HTML elements using your DOM mastery.

JAVASCRIPT ES6

- Language Showdown: Compare JavaScript with scripting, markup, and stylesheet languages.
- JavaScript Overview: Explore JS and its versatile uses in web development.
- Coding Tools: Learn to work with code editors and IDEs.
- Syntax & Data Types: Understand variables (let, const), data types, and operators.
- Control Structures: Master if/else, loops (for, while, for...of, for...in) for code flow.
- Functions: Explore function declarations, scope, closures, arrow functions, and higher-order functions.
- Array Mastery: Create, manipulate, and use methods (map, filter, reduce, slice, concat).
- Spread Syntax: Utilize ... for array and function call expansion.
- OOP in JS: Create objects using literals, constructors, and classes; understand this and super.
- Object Handling: Use Object.keys, Object.values, and Object.assign for manipulation.

REACT.JS

- React Adventure: Dive into front-end development with React.
- Component Mastery: Learn when and how to use React components.
- Prop Passing: Discover how to pass and use props effectively.
- JSX Magic: Write and understand JSX syntax for dynamic UI.
- React DOM: Explore the React DOM for rendering components.
- State Management: Master state handling in React.
- Hook Exploration: Learn about React Hooks for enhanced functionality.
- Conditional Rendering: Implement conditional logic in your components.
- Component Types: Understand the difference between class and functional components.

GIT, GITHUB, AND VERSION CONTROL

- Version Control Basics: Use Git for version control and seamless collaboration.
- Fork, Branch & Clone: Master forking, branching, and cloning for efficient workflows.
- Remote Repos: Utilize GitHub as your remote code repository.
- Change Management: Learn to checkout and roll back changes using Git.
- Git with Xcode: Integrate Git and GitHub into your Xcode projects.

NODE.JS

- Back-End Basics: Explore the components of back-end development using an MVC framework.
- OOP in Node: Apply data types, objects, methods, and classes to back-end development.
- Server-Side JavaScript: Dive into JavaScript for server-side programming.
- Node Command Line: Use Node.js commands for back-end tasks.
- NPM Power: Manage packages with Node Package Manager (NPM).
- Build Processes: Learn JavaScript build processes for efficient development.
- Event Loop: Understand Node's event loop and emitters for asynchronous operations.
- File System: Interact with the file system using Node.js.
- Modular Magic: Work with modules for organized, reusable code.
- Native Drivers: Utilize native Node.js drivers for various database interactions.

EXPRESS.JS

- Express Setup: Learn how to install and use Express in Node applications.
- Server Creation: Build servers using Node and Express.
- RESTful Routing: Master routing to handle requests with Express.
- Middleware Magic: Implement middleware for enhanced Node applications.

APPLICATION PROGRAMMING INTERFACE (API)

- API Basics: Explore what APIs are and how they function.
- Deep Dive into HTTP: Understand the role of HTTP in API communication.
- API Calls: Learn how to make API requests effectively.
- Read the Docs: Navigate and understand API documentation.
- API Authentication: Grasp basic authentication methods for secure access.
- Server Communication: Implement server-to-server interactions.
- Data Formats: Compare JSON and XML for data exchange.

MONGO DB

- Introduction to MongoDB: Discover the world of NoSQL databases with MongoDB.
- Setting Up: Learn how to install and set up MongoDB for your projects.
- Database Structure: Explore collections and documents as the building blocks of MongoDB.
- CRUD Operations: Master Create, Read, Update, and Delete operations in MongoDB.
- Querying the Database: Use queries to search and filter data effectively.
- Schema Design: Understand schema-less design and data modeling for flexibility.
- Aggregation: Harness aggregation pipelines to process and analyze data.
- Indexes: Learn to create indexes for faster search and query performance.
- Relationships: Implement relationships between data using references and embedding.
- Integration with Node.js: Connect your MongoDB database to Node.js applications.

PROJECTS

- Personal Portfolio Website
- QR Code Generator
- Building a React Advice App with API Integration
- Weather App with React
- BMI Calculator App with React
- Currency Converter App with React
- Digital Clock with React
- Strong Password Generator in React
- Quiz App with React
- CRUD Application with React.js Frontend and Node.js Backend

CHALLENGES AND INTERACTIVE LEARNING

Each topic features engaging challenges to provide hands-on, real-world learning experiences. These activities start easy, building up in difficulty to reinforce concepts. Students receive instant feedback, collaborate in groups, and participate in friendly competitions to boost skills and confidence. Advanced learners can "challenge a mentor" for deeper insights, ensuring a dynamic, enjoyable, and practical learning journey. This approach makes the course interactive, motivating, and memorable.