

Course title: Web Development Bootcamp

Duration: 30 Weeks

1. Course Overview

This intensive course covers the fundamentals of front-end web development, focusing on HTML, CSS, and JavaScript. The program includes practical exercises, projects, and sessions conducted three times per week.

Schedule Structure:

- **Monday:** Core Concepts & Theory
 - **Wednesday:** Practical Implementation
 - **Friday:** Project Work & Review
-

2. Weekly Breakdown

Week 1: HTML Foundations

Session 1: Introduction to Web Development

- Course overview and setup
- Introduction to web browsers and how they work
- Basic development tools installation (VS Code, Chrome DevTools)
- HTML document structure (DOCTYPE, html, head, body)

Session 2: HTML Elements & Semantics

- Essential HTML tags
- Semantic HTML5 elements
- Text formatting and lists
- Links and navigation
- Images and media

Session 3: HTML Forms & Tables

- Form elements and attributes
- Input types and validation
- Table structure and formatting
- Accessibility basics

- **Project:** Personal Profile Page

Week 2: CSS Fundamentals

Session 1: CSS Basics

- CSS syntax and selectors
- Cascading and specificity
- Box model
- Colors and typography

Session 2: Layout Fundamentals

- Display properties
- Positioning (relative, absolute, fixed)
- Margin and padding
- Units (px, em, rem, %)

Session 3: Responsive Design Basics

- Media queries introduction
- Viewport concepts
- Flexible layouts
- **Project:** Styling the Profile Page

Week 3: Advanced CSS

Session 1: Flexbox

- Flex container properties
- Flex item properties
- Building flexible layouts
- Common flexbox patterns

Session 2: CSS Grid

- Grid container properties
- Grid item properties
- Grid areas and templates
- Combining flexbox and grid

Session 3: CSS Transformations & Animations

- Transforms (2D and 3D)
- Transitions
- Keyframe animations
- **Project:** Interactive Landing Page with Animations

Week 4: JavaScript Basics

Session 1: Introduction to JavaScript

- Variables and data types
- Operators
- Control structures (if/else/switch)
- Functions basics

Session 2: Arrays and Objects

- Array methods
- Object creation and manipulation
- Loops and iterations
- Basic algorithms

Session 3: DOM Manipulation

- Selecting elements
 - Modifying content and attributes
 - Event handling basics
 - **Project:** Interactive Form Validation
-

3. Advanced Topics

Week 5: Advanced JavaScript

Session 1: ES6+ Features

- Arrow functions
- Template literals
- Destructuring
- Spread/rest operators

Session 2: Asynchronous JavaScript

- Callbacks
- Promises
- Async/await
- Error handling

Session 3: Event Handling & DOM

- Event bubbling and capturing
- Event delegation
- Custom events
- **Project:** Task Manager Application

Week 6: JavaScript Patterns & Best Practices

Session 1: Design Patterns

- Module pattern
- Factory pattern
- Observer pattern
- Singleton pattern

Session 2: Code Quality

- Clean code principles
- Error handling
- Debugging techniques
- Performance optimization

Session 3: Testing Basics

- Unit testing introduction
- Jest basics
- Testing DOM manipulation
- **Project:** Refactoring and Testing

Week 7: Modern Web Development

Session 1: Build Tools

- npm basics
- Webpack introduction
- Babel overview
- Development workflow

Session 2: Modern CSS

- CSS preprocessors (SCSS)
- CSS-in-JS introduction
- CSS Modules
- Tailwind CSS basics

Session 3: API Integration

- RESTful APIs
- Fetch API
- AJAX
- **Project:** Weather Dashboard

Week 8: State Management & Advanced JavaScript

Session 1: Client-Side State Management

- Local Storage and Session Storage
- Cookies and their use cases
- State management patterns
- Application state vs UI state
- Memory management and performance

Session 2: Advanced JavaScript Patterns

- Publish/Subscribe pattern
- Prototype pattern
- Command pattern
- State pattern
- Implementing patterns in real applications

Session 3: Project Implementation

- Building a shopping cart with state management
- Implementing persistent storage
- User session handling
- **Project:** E-commerce Product Page with Cart Functionality

Week 9: Performance Optimization & Security

Session 1: Web Performance

- Critical rendering path
- JavaScript performance optimization
- Asset optimization
- Lazy loading
- Code splitting strategies

Session 2: Web Security

- Common security vulnerabilities (XSS, CSRF)
- Content Security Policy
- CORS in depth
- Secure data storage
- Input validation and sanitization

Session 3: Implementation & Testing

- Performance auditing with Chrome DevTools
- Implementing security best practices
- Load testing

- **Project:** Optimizing and Securing the E-commerce Application

Week 10: Modern Web Features

Session 1: Progressive Web Apps

- Service Workers
- Web App Manifest
- Offline functionality
- Push notifications
- Cache strategies

Session 2: Modern Web APIs

- Intersection Observer
- ResizeObserver
- Web Storage API
- Geolocation API
- Web Workers

Session 3: Implementation

- Converting existing application to PWA
- Implementing offline functionality
- Adding push notifications
- **Project:** PWA Weather Dashboard with Offline Support

Week 11: Build Tools & Deployment

Session 1: Build Systems

- Advanced npm concepts
- Webpack configuration
- Babel setup and configuration
- Source maps
- Tree shaking and code splitting

Session 2: Deployment & CI/CD

- Build optimization
- Environment configuration
- Deployment strategies
- Basic CI/CD setup
- Version control best practices

Session 3: Implementation

- Setting up build pipeline
- Configuring development and production environments
- Deploying to hosting platforms
- **Project:** Setting up CI/CD for Previous Projects

Week 12: Final Project & TypeScript Introduction

Session 1: TypeScript Basics

- TypeScript setup
- Basic types and interfaces
- Classes and inheritance
- Generics
- Decorators
- This session begins bridging to Angular

Session 2: Final Project Development

- Project architecture review
- Code quality assessment
- Performance optimization
- Implementation of remaining features

Session 3: Final Project Presentation

- Project demonstrations
- Code review
- Performance analysis
- Discussion of potential improvements
- Preparation for Angular course

Final Project Requirements

Build a full-stack web application incorporating:

- State management
- PWA features
- Performance optimization
- Security best practices
- Modern Web APIs
- TypeScript implementation

Project Options:

1. **E-commerce Platform**

- Product catalog
- Shopping cart with state management
- User authentication
- Offline capability

2. **Social Media Dashboard**

- User authentication
- Real-time updates
- Data visualization
- User interactions
- Push notifications
- Offline support

3. **Project Management Tool**

- User authentication
- Task management
- Real-time updates
- Data persistence
- Progressive Web App features

Technical Requirements for Final Project:

- Clean, modular code structure
- Proper error handling
- Performance optimizations
- Security implementation
- Responsive design
- Progressive enhancement
- Offline functionality
- Type safety (TypeScript)
- Documentation
- Testing coverage

Assessment Metrics:

- **Code Quality (25%)**
 - Clean code principles
 - Modern JavaScript features
 - TypeScript implementation
 - Error handling

- **Functionality (25%)**
 - Feature completeness
 - Bug-free operation
 - Performance
 - Security
 - **Technical Implementation (25%)**
 - State management
 - API integration
 - PWA features
 - Build configuration
 - **Project Documentation (15%)**
 - Code documentation
 - README
 - API documentation
 - Setup instructions
 - **Presentation (10%)**
 - Demo
 - Code walkthrough
 - Technical communication
 - Q&A handling
-

This final phase of the course serves as a bridge to the Angular portion, introducing TypeScript and more advanced architectural concepts that will be crucial for Angular development

Week 1: Angular Fundamentals

Class 1:

- Overview of Angular: Features, Use Cases, and Benefits
- Setting up Angular Development Environment
- Introduction to Angular CLI and creating your first Angular application

Class 2:

- Understanding Angular Project Structure
- Angular Modules and Components: Overview and Creation
- Angular App Lifecycle

Class 3:

- TypeScript Basics for Angular: Variables, Classes, Functions, and Interfaces
- Data Binding: Interpolation, Property, Event, and Two-Way Binding

Quiz:

- Multiple-choice and coding questions on Angular setup, project structure, and TypeScript basics.

Weekly Project:

- Create a basic Angular app with two components and demonstrate data binding.

Week 2: Components, Templates, and Angular Material

Class 4:

- Components Deep Dive: Input and Output Decorators
- Parent-Child Communication in Angular

Class 5:

- Introduction to Angular Material: Installation and Setup
- Material Components: Buttons, Toolbars, and Icons
- Styling Angular Apps with Angular Material

Class 6:

- Structural Directives (*ngIf, *ngFor) and Template Expressions
- Building Custom Components with Angular Material

Quiz:

- Questions on components, Angular Material basics, and structural directives.

Weekly Project:

- Build a responsive UI using Angular Material components and structural directives.

Week 3: Forms, RxJS Basics, and Angular Material Components

Class 7:

- Template-Driven Forms: Form Controls and Basic Validation
- Material Form Components: Inputs, Selects, and Radio Buttons

Class 8:

- Reactive Forms: Overview, Setup, and Validation
- RxJS Basics: Observables, Operators, and Subscriptions

Class 9:

- Advanced Angular Material Components: Dialogs, Snackbars, Tooltips, and Steppers
- Building a Form with Reactive Programming using RxJS

Quiz:

- Focus on forms (template-driven and reactive) and basic RxJS concepts.

Weekly Project:

- Develop a form-driven application using Angular Material and RxJS to handle dynamic input validation.

Week 4: Services, Dependency Injection, and Navigation

Class 10:

- Angular Services and Dependency Injection (DI)
- Using Angular HttpClient with RxJS to Fetch Data
- Introduction to Subjects and BehaviorSubjects

Class 11:

- RxJS Operators for Data Streams: map, filter, switchMap, and mergeMap
- Using Angular Material Table with Sorting, Filtering, and Pagination

Class 12:

- Angular Router Basics: Configuring Routes and Navigation
- Angular Material Navigation Components: Side Nav, Menus, and Tabs

Quiz:

- Questions on services, RxJS operators, and Angular Router basics.

Weekly Project:

- Build a multi-page application with navigation and a table displaying paginated data fetched from a public API.

Week 5: Routing, RxJS Advanced, and State Management

Class 13:

- Route Guards: Protecting and Managing Routes
- Lazy Loading Modules for Optimization

Class 14:

- Advanced RxJS Concepts: Combining Observables with combineLatest, forkJoin, and zip
- Managing HTTP Errors and Retry Mechanisms using RxJS

Class 15:

- Introduction to State Management with NgRx
- Setting Up Store, Actions, and Reducers

Quiz:

- Advanced RxJS concepts and state management basics.

Weekly Project:

- Implement route guards and lazy loading for a secured, modular Angular app.

Week 6: Pipes, Testing, and Animations

Class 16:

- Angular Pipes: Built-in and Custom Pipes
- Material DatePickers with Formatting and RxJS Observables

Class 17:

- Unit Testing Angular Components and Services
- Debugging Angular Applications

Class 18:

- Adding Angular Animations: Basic Setup and Integration
- Custom Animations for Angular Material Components

Quiz:

- Questions on pipes, testing, and animations.

Weekly Project:

- Build a dashboard with custom pipes, date pickers, and animations.

Week 7: Advanced Topics and Optimization**Class 19:**

- Advanced Angular Material Components: Expansion Panels, Cards, and Grid Lists
- Optimizing Angular Applications for Performance

Class 20:

- Integrating RxJS with State Management (NgRx)
- Practical Session: Building a Dashboard with Angular Material and RxJS

Class 21:

- Error Handling with Angular and RxJS: Using CatchError and Retry Strategies

Quiz:

- Focus on optimization, RxJS with NgRx, and advanced Angular Material components.

Weekly Project:

- Build a state-managed application using NgRx and Angular Material.

Week 8: Deployment and Final Project**Class 22:**

- Preparing Angular Apps for Production
- Building and Deploying Angular Applications (Firebase, Netlify, GitHub Pages)

Class 23:

- Final Project Kickoff: Building a Complete Angular Application with Angular Material and RxJS

Class 24:

- Final Project Presentation and Review
- Best Practices for Angular Development with RxJS and Angular Material

Final Project:

- Develop a fully functional Angular application with routing, state management (NgRx), Angular Material components, RxJS, and API integration.
-

Assessment Structure

- **Weekly Quizzes:** Assess key topics through theory and coding exercises (15–20 questions each).
- **Weekly Projects:** Hands-on assignments to implement weekly concepts in real-world scenarios.
- **Final Project:** A capstone project integrating all the skills learned in the course.