4-Week Web Development Visual Course

Course Overview

This comprehensive visual course introduces beginners to web development fundamentals through hands-on learning and practical examples. Over 4 weeks, students will master HTML, CSS, JavaScript, and deployment basics.

Week-by-Week Breakdown

Week 1: Introduction to Web Development & HTML

Learning Objectives: - Understand how the web works (client-server model) - Learn HTML document structure and semantic markup - Create basic web pages using common HTML elements

Key Topics: - Client-server architecture and HTTP protocols - HTML document structure (DOCTYPE, html, head, body) - Essential HTML tags: headings, paragraphs, links, images, lists - Semantic HTML for better structure and accessibility

Practical Exercises: - Build a personal profile page - Create a simple blog post with proper HTML structure - Practice using different HTML elements and attributes

Week 2: Styling with CSS

Learning Objectives: - Understand the role of CSS in web design - Learn CSS selectors and basic styling properties - Master the CSS box model and layout principles

Key Topics: - CSS selectors (element, class, ID) and specificity - CSS box model (margin, border, padding, content) - Basic styling properties (color, typography, backgrounds) - Introduction to Flexbox for modern layouts

Practical Exercises: - Style the personal profile page from Week 1 - Create a responsive navigation menu - Build a card-based layout using Flexbox

Week 3: Interactivity with JavaScript

Learning Objectives: - Learn JavaScript programming fundamentals - Understand DOM manipulation and event handling - Add interactive features to web pages

Key Topics: - JavaScript variables, data types, and functions - Control flow (conditionals and loops) - DOM selection and manipulation - Event handling (clicks, form submissions, keyboard events)

Practical Exercises: - Create an interactive to-do list - Build a simple calculator - Add form validation to a contact form

Week 4: Responsive Design & Deployment

Learning Objectives: - Create responsive websites that work on all devices - Learn version control basics with Git - Deploy websites to the internet

Key Topics: - Media queries and responsive breakpoints - Mobile-first design approach - Git fundamentals and GitHub workflow - Deployment options (GitHub Pages, Netlify, Vercel)

Practical Exercises: - Make previous projects responsive - Set up a GitHub repository - Deploy a complete website to GitHub Pages

Course Features

Visual Learning Approach

- **Interactive Slides:** Modern, visually appealing presentation slides with live code examples
- **Diagrams and Illustrations:** Clear visual representations of concepts like the client-server model, CSS box model, and responsive breakpoints
- **Code Examples:** Syntax-highlighted code snippets with explanations
- **Before/After Comparisons:** Visual demonstrations of styling and functionality improvements

Hands-On Projects

Each week includes practical exercises that build upon previous learning: 1. **Personal Profile Page** (Week 1-2): HTML structure with CSS styling 2. **Interactive Components** (Week 3): Adding JavaScript functionality 3. **Complete Website** (Week 4): Responsive design and deployment

Progressive Learning

The course follows a logical progression: - **Foundation:** HTML structure and semantics - **Presentation:** CSS styling and layout - **Behavior:** JavaScript interactivity - **Deployment:** Making websites live and accessible

Technical Requirements

- Modern web browser (Chrome, Firefox, Safari, Edge)
- Text editor (VS Code recommended)
- Git installed on local machine
- GitHub account for version control and deployment

Learning Outcomes

By the end of this course, students will be able to: - Create well-structured HTML documents using semantic markup - Style web pages with CSS, including responsive layouts - Add interactivity using JavaScript and DOM manipulation - Use Git for version control and deploy websites to the internet - Understand web development best practices and modern workflows

Next Steps

After completing this course, students can continue their web development journey by: - Learning JavaScript frameworks (React, Vue.js, Angular) - Exploring backend development (Node.js, Python, PHP) - Studying databases and API integration - Building more complex, full-stack applications - Contributing to open-source projects

Course Materials Included

- 10 interactive presentation slides
- Visual diagrams and illustrations
- Code examples and templates
- Project starter files
- Resource links and references
- Step-by-step deployment guides

This course provides a solid foundation for anyone looking to start their web development journey with a focus on practical, hands-on learning and visual understanding of core concepts.