Modern Frontend Web Development

Comprehensive Course Slides

Module 1: Introduction to Web Development Fundamentals

Course Overview

Modern Frontend Web Development

What You'll Learn:

- HTML5 semantic markup and modern standards
  - CSS3 with Grid, Flexbox, and responsive design
    - JavaScript ES6+ with modern programming patterns
      - DOM manipulation and event handling
        - API integration and asynchronous programming
          - Modern development tools and workflows
            - React fundamentals and component architecture
              - Testing, debugging, and deployment strategies

              Course Structure: 8 modules, 17 hands-on labs, 1 final project

              ---

The Modern Web Platform

Evolution of Web Development:

Then (Early 2000s):

- Static HTML pages
  - Table-based layouts
    - Inline styles and scripts
      - Browser compatibility nightmares

      Now (2024):

      - Component-based architectures
        - Mobile-first responsive design
          - Modern JavaScript with ES6+ features
            - Build tools and development workflows
              - Progressive Web Applications (PWAs)

              Key Principles: Semantic HTML, Separation of concerns, Progressive enhancement

              ---

Client-Server Architecture

How Web Applications Work:

```
Client (Browser)          Server
     |                      |
     |-- HTTP Request ----→ |
     |                      |
     |← HTTP Response ------ |
     |                      |
```

Client-Side (Frontend):

- HTML structure and content
  - CSS styling and layout
    - JavaScript interactivity and logic
      - User interface and experience

      Server-Side (Backend):

      - Data processing and storage
        - Business logic and APIs
          - Authentication and security
            - Database management

Understanding Web Protocols

Internet Protocols enable web communication:

TCP/IP - Transmission Control Protocol/Internet Protocol

- The Internet is a packet-switched network
  - TCP collects and reassembles packets
    - IP ensures packets reach the right destination

  DNS - Domain Name System

  - Converts between IP addresses and Domain Names
    - Example: google.com → 142.250.191.14

    HTTP/HTTPS - Hypertext Transfer Protocol

    - Application-level protocol for web communication
      - HTTPS adds security with SSL/TLS encryption

Modern Development Environment

Essential Tools for Front-End Development:

Code Editor: Visual Studio Code

- Syntax highlighting, IntelliSense, extensions
  - Integrated terminal and Git support

  Runtime: Node.js and npm

  - JavaScript runtime outside the browser
    - Package manager for dependencies

    Build Tool: Vite

    - Fast development server with Hot Module Replacement
      - Optimized production builds

      Version Control: Git

      - Track changes and collaborate effectively

Lab 01 - Working with the Command Line in VSCode

Learning Objectives:

- Open and use integrated terminal in VSCode
  - Practice basic command line navigation
    - Build confidence with commands for Git, npm, and project setup

    Key Commands:

    - `pwd` - Show current directory
      - `ls` - List files and folders
        - `cd` - Change directory
          - `mkdir` - Create directory
            - `touch` - Create file

            ## Module 2: Tools and Workflows

Version Control with Git

Why Version Control Matters:

- Track changes over time
  - Collaborate with team members
    - Revert to previous versions
      - Branch and merge features

      Git Workflow:

      1. `git init` - Initialize repository
      2. `git add` - Stage changes
      3. `git commit` - Save changes
      4. `git push` - Upload to remote repository

Package Management with npm

What is npm?

- Node Package Manager
  - Manages dependencies for JavaScript projects
    - Provides scripts for common tasks

    Key npm Commands:

    - `npm init` - Initialize project
      - `npm install` - Install dependencies
        - `npm run` - Execute scripts
          - `npm update` - Update packages

          package.json - Project configuration file

Browser Developer Tools

Chrome DevTools Features:

Elements Panel:

- Inspect and modify HTML/CSS live
  - Debug layout issues

Console Panel:

- View JavaScript errors and logs
  - Test code interactively

Sources Panel:

- Set breakpoints and debug JavaScript
  - Step through code execution

Network Panel:

- Monitor HTTP requests and responses

Lab 02 - Using Visual Studio Code Basics

Learning Objectives:

- Master VSCode features and extensions
  - Learn Markdown for documentation
    - Use Emmet for faster HTML writing
      - Customize development environment

      Key VSCode Features:

      - Command Palette (Cmd/Ctrl + Shift + P)
        - Multi-cursor editing
          - Live Server extension
            - Markdown preview

            ## Module 3: HTML Fundamentals

HTML5 Semantic Elements

Modern HTML5 provides meaningful structure:

Document Structure:

- `<header>` - Page or section header
  - `<nav>` - Navigation links
    - `<main>` - Primary content
      - `<footer>` - Page or section footer

        Content Organization:

        - `<article>` - Self-contained content
          - `<section>` - Logical document divisions
            - `<aside>` - Sidebar or tangential content
              - `<figure>` & `<figcaption>` - Images with captions

                Benefits: Better SEO, accessibility, and maintainability

                ---

HTML Forms and User Input

Essential Form Elements:

Input Types:

- `<input type="text">` - Text fields
  - `<input type="email">` - Email validation
    - `<input type="password">` - Hidden input
      - `<input type="number">` - Numeric input
        - `<input type="date">` - Date picker

Form Structure:

```
<form action="/submit" method="POST"
  <label for="name">Name:</label>
  <input type="text" id="name" name=
  <button type="submit">Submit</butt
</form>
```

Lab 03 - HTML Forms

Learning Objectives:

- Create interactive forms with proper validation
  - Use semantic form elements and attributes
    - Implement accessibility best practices
      - Handle form data with JavaScript

      Key Concepts:

      - Form validation attributes (`required`, `pattern`)
        - Label-input relationships
          - Form accessibility
            - Modern input types

Lab 04 - Semantic HTML

Learning Objectives:

- Structure content with HTML5 semantic elements
  - Improve accessibility and SEO
    - Create meaningful document outlines
      - Build foundation for CSS styling

      Practice Elements:

      - Document structure with `<header>`, `<main>`, `<footer>`
        - Content organization with `<article>`, `<section>`
          - Navigation with `<nav>`
            - Supplem content with `<aside>`

      ## Module 4: CSS Fundamenta

Moder CSS Layout

CSS Grid - Two-Dimensiona Layout:

```
.grid-
  disp
  grid
  gap:
}
```

Flexbox - One-Dimensiona Layout:

```
.flex-
  disp
  just
  alig
}
```

When to Use Each:

- Grid: Complex layouts, two-dimensio control
  - Flexbo Comp alignm one-dimen flow

Resp
Desig
Princ

Mobile-First Approac

```
/* I
.co
  w
}

/*
@me
  .

  }
}

/* I
@me
  .

  }
}
```

Key Breakpoi

- Mobile
  <
  768px
  - Tab
    768
    -
    102
    -
    -
    -
    -

La
0
-
C
Fl

Lea
Ob