Full Stack Web Development Curriculum

MindVerse Academy

This comprehensive curriculum outlines the modules and topics covered in MindVerse Academy's Full Stack Web Development program. Designed to equip students with the skills necessary to build, deploy, and maintain modern web applications, this course progresses from foundational front-end technologies to robust back-end development and deployment strategies.

Module 1: HTML & CSS Basics

Duration: [Suggested Duration, e.g., 20 hours]
This module introduces the fundamental building blocks of the web, focusing on structuring content with HTML and styling it with CSS.

- **HTML Fundamentals:**
  - Basic document structure (<!DOCTYPE html>, <html>, <head>, <body>)
  - Semantic HTML5 elements (e.g., <header>, <nav>, <main>, <article>, <section>, <footer>)
  - Text formatting, links, images, lists, tables
  - Forms and input types
- **CSS Fundamentals:**
  - Selectors (Type, Class, ID, Attribute, Pseudo-classes, Pseudo-elements)
  - Box Model (margin, border, padding, content)
  - Typography and color properties
  - Backgrounds and borders
  - CSS positioning (static, relative, absolute, fixed, sticky)
  - Transitions and basic animations
- **Responsive Layouts:**
  - Introduction to responsive design principles
  - **Flexbox:**
    - Container and item properties
    - Creating one-dimensional layouts
  - **CSS Grid:**
    - Container and item properties
    - Creating two-dimensional layouts
  - Media Queries for adapting layouts to different screen sizes

Module 2: JavaScript & DOM

Duration: [Suggested Duration, e.g., 25 hours]
This module dives into JavaScript, the language of the web, covering core concepts and how

to interact with and manipulate the Document Object Model (DOM).

- **Core JavaScript Concepts:**
  - Variables (var, let, const) and data types
  - Operators (arithmetic, comparison, logical, assignment)
  - Control flow (if/else, switch, loops: for, while, do-while)
  - Functions (declarations, expressions, arrow functions)
  - Arrays and Objects (creation, manipulation, iteration)
  - ES6+ Features: Template literals, destructuring, spread/rest operators, modules (import/export)
  - Asynchronous JavaScript: Callbacks, Promises, Async/Await
  - Error Handling (try-catch)
- **DOM Manipulation & Events:**
  - Understanding the Document Object Model (DOM Tree)
  - Selecting elements (getElementById, querySelector, querySelectorAll)
  - Modifying element content (textContent, innerHTML) and attributes
  - Styling elements dynamically
  - Creating and appending new elements
  - Event Handling:
    - Event listeners (addEventListener)
    - Common events (click, submit, keydown, mouseover, etc.)
    - Event bubbling and capturing
    - event.preventDefault() and event.stopPropagation()

Module 3: Front-End Frameworks

Duration: [Suggested Duration, e.g., 40 hours]
This module introduces popular front-end frameworks and libraries that accelerate UI development and provide structured ways to build complex user interfaces.

- **UI Frameworks/Libraries:**
  - **Bootstrap:**
    - Grid system, components (Navbar, Cards, Modals, Carousels)
    - Utilities and customization
  - **Tailwind CSS:**
    - Utility-first approach
    - Responsive design with Tailwind
    - Customization and configuration
- **React (or Angular) - Basics to Intermediate:**
  - **Introduction to React:**
    - Why React? Component-based architecture
    - Setting up a React project (Create React App/Vite)

        - JSX syntax
    - **Components:**
        - Functional components vs. Class components
        - Props for data passing
        - State management with useState hook
        - Lifecycle methods/useEffect hook
    - **Conditional Rendering and List Rendering:**
    - **Event Handling in React:**
    - **Forms in React:** Controlled components
    - **React Router (Basic):** Client-side routing
    - **Context API (Basic):** State management across components
    - **Introduction to API Calls:** Fetching data from external APIs

Module 4: Backend with Node.js & Express

Duration: [Suggested Duration, e.g., 30 hours]
This module focuses on server-side development using Node.js and the Express.js framework to build robust RESTful APIs.

- **Introduction to Node.js:**
    - Node.js runtime environment
    - NPM (Node Package Manager)
    - Module system (CommonJS, ES Modules)
    - Asynchronous nature of Node.js
- **Express.js Framework:**
    - Setting up an Express application
    - Routing (defining API endpoints)
    - Middleware (parsing request bodies, logging, error handling)
    - Request and Response objects
    - Handling HTTP methods (GET, POST, PUT, DELETE)
- **Building REST APIs:**
    - Principles of RESTful design
    - Designing API endpoints for resources
    - Implementing **CRUD** operations (Create, Read, Update, Delete)
    - Status codes and error handling in APIs
    - Authentication and Authorization concepts (e.g., JWT introduction)

Module 5: Databases

Duration: [Suggested Duration, e.g., 25 hours]
This module covers database integration, focusing on either NoSQL (MongoDB) or Relational (MySQL) databases and their respective ORMs.

- **Introduction to Databases:**

    - Relational vs. NoSQL databases
    - Database concepts (tables/collections, schemas, relationships)
- **MongoDB (NoSQL) or MySQL (Relational):**
    - **MongoDB:**
        - Document-oriented database
        - Basic CRUD operations with MongoDB Shell
        - Data modeling for NoSQL
    - **MySQL:**
        - Relational database concepts (tables, columns, rows, primary/foreign keys)
        - SQL queries (SELECT, INSERT, UPDATE, DELETE, JOINs)
        - Database normalization (introduction)
- **Mongoose (for MongoDB) or Sequelize (for MySQL) ORM:**
    - **Introduction to ORMs:** Why use an ORM?
    - **Mongoose (for MongoDB):**
        - Connecting to MongoDB
        - Defining schemas and models
        - Performing CRUD operations through Mongoose
        - Validation and middleware
    - **Sequelize (for MySQL):**
        - Connecting to MySQL
        - Defining models and associations
        - Performing CRUD operations through Sequelize
        - Migrations and seeding (introduction)
- **Integrating Database with Node.js/Express:**
    - Connecting the backend application to the database
    - Performing database operations via API endpoints

Module 6: Deployment & Hosting

Duration: [Suggested Duration, e.g., 15 hours]
This module covers essential tools and platforms for version control and deploying full-stack applications to the cloud.

- **Git & GitHub:**
    - Version control concepts
    - Basic Git commands (init, add, commit, status, log)
    - Branching and merging
    - Working with remote repositories (clone, push, pull, fetch)
    - Collaborative workflows on GitHub
- **Deployment Strategies:**

    - Understanding hosting environments (PaaS, IaaS)
    - Environment variables
    - Build processes for front-end and back-end
- **Deploy on Netlify/Render/Vercel:**
    - **Netlify/Vercel (for Front-End):**
        - Connecting to GitHub repository
        - Automated deployments
        - Custom domains
    - **Render (for Back-End/Full-Stack):**
        - Deploying Node.js/Express applications
        - Connecting to databases (e.g., MongoDB Atlas, Render PostgreSQL)
        - Environment configuration
        - Monitoring and logging

Capstone Project: Build and Deploy a Full-Stack Application

Duration: [Suggested Duration, e.g., 30 hours]
The capstone project is the culmination of the course, where students apply all learned concepts to build a complete, real-world full-stack application from scratch and deploy it.

- **Project Planning:**
    - Requirement gathering and feature definition
    - Database design
    - API design
    - UI/UX planning
- **Development:**
    - Implementing front-end (React/Angular)
    - Developing back-end (Node.js/Express) with REST APIs
    - Integrating database (MongoDB/MySQL)
    - Implementing authentication/authorization
- **Testing & Debugging:**
    - Basic unit/integration testing concepts
    - Debugging techniques for front-end and back-end
- **Deployment:**
    - Deploying the complete application to cloud platforms (e.g., Netlify/Vercel for front-end, Render for back-end/database)
    - Post-deployment checks and troubleshooting

Upon completion of this curriculum, students will be able to:

- Design and develop responsive user interfaces using HTML, CSS, and modern front-end frameworks.
- Build robust and scalable RESTful APIs with Node.js and Express.js.

- Manage and interact with databases (both SQL and NoSQL).
- Utilize version control with Git and GitHub for collaborative development.
- Deploy full-stack applications to cloud hosting platforms.
- Independently plan, develop, and deploy a complete web application.