CLOUD CONTRAPTIONS
COMPUTER PROGRAMMING & TRAINING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Full-Stack Web Development with Svelte and ASP.NET Core

Duration

5 days

Description

The Full-Stack Web Development with Svelte and ASP.NET Core course offers a comprehensive overview of Svelte, a modern JavaScript framework for building user interfaces, and its integration with ASP.NET Core for server-side operations. The course covers everything from setting up a development environment, creating static and dynamic pages, to handling event and state management. It also explores how to build a REST API with ASP.NET Core, create a database, and handle authentication and authorization. The course is perfect for professionals who want to learn how to build robust, full-stack applications using Svelte and ASP.NET Core.

Objectives

- Understand the fundamentals of Svelte and how it compares to other frameworks.
- Set up a development environment for Svelte and learn how to use SvelteKit.
- Create static and dynamic pages using Svelte, including understanding their structure and how to handle images, CSS, and JavaScript content.
- Understand the principles of template reactivity and component basics in Svelte.
- Learn about event handling, data binding, and forms in Svelte.
- Understand how to handle lifecycle events and state management in Svelte.
- Learn about routing, error handling, and asynchronous data in Svelte.
- Understand how to use server data with ASP.NET Core, including building a REST API, creating a database, and handling authentication and authorization.

Prerequisites

All students must have C#, JavaScript, and HTML programming experience. Experience with CSS is helpful, but not required.

Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

CLOUD CONTRAPTIONS
COMPUTER PROGRAMMING & TRAINING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Student will need permission to install .NET SDK, Node.js, and Visual Studio Code on their computers. Also, students will need permission to install NuGet Packages, NPM Packages and Visual Studio Extensions. If students are unable to configure a local environment, a cloud-based environment can be provided.

Outline

- Introduction
  - What is Svelte?
  - What problem does Svelte solve?
  - Svelte vs. Other Frameworks
  - Svelte Compiler
- Development Environment
  - Requirements
  - SvelteKit
  - Svelte Files
  - Svelte Extension for Visual Studio Code
  - Run/Debug Svelte App in Visual Studio Code
  - Svelte Extension for WebStorm
  - Run/Debug Svelte App in WebStorm
- SvelteKit Overview
  - Vite Tooling
  - Development Server
  - Routing
  - Deployment
  - Server-side rendering
  - Unit Testing
- Getting Started
  - Exploring the REPL
  - Svelte Layout
  - Svelte Page
  - Svelte Component
  - Svelte Architecture
  - Svelte Element Directives

  - Compiling Svelte Files
- Static Pages
  - What is a Static Page?
  - What problem do Static Pages solve?
  - Static Page File Structure
  - Setting Head Content
  - HTML Content
  - CSS Content
  - Comments
  - Scoped CSS
  - Handling Images
  - Hot Module Reloading
  - Server Pre-rendering
  - Page Routing
- Dynamic Pages
  - What is a Dynamic Page?
  - What problem do Dynamic Pages solve?
  - Client-Side Rendering
  - Dynamic Page File Structure
  - JavaScript Content
  - Using Variables
  - Using Expressions
  - Data Binding
  - Class and Style Directive
  - Event Binding
  - Logic Blocks
  - Debug Tag
- Template Reactivity
  - Principles
  - Changing Data through Assignments
  - Reactive Statements
  - Updating Arrays and Objects
- Component Basics
  - What is a Component?
  - What problem does it solve?

  - Calling Components vs HTML Elements
  - Component File Structure
  - Component Props
  - Component Events
- Component Composition
  - What is Component Composition?
  - What problem does it solve?
  - Nested Components
  - Passing Data to Child Components
  - Handling Events and Receiving Data from Child Components
  - Component Tree Best Practices
- Event Handling
  - Event Handling Element Directives
  - DOM Events
  - Adding Event Handlers
  - In-line Handlers
  - Event Modifiers
  - Dispatching Component Events
  - Forwarding Events
- Data binding
  - Top-down data binding by default
  - Communication with props and events
  - Using two-way data binding
- Forms
  - HTML Form Element
  - Named Form Actions
  - Form Validation
  - Form Submission
  - Progressive Enhancement
- Lifecycle
  - Mount
  - Destroy
  - Before Update
  - After Update
  - Tick

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- State Management
  - Stores
  - Writable Stores
  - Auto-subscriptions
  - Readable Stores
  - Derived Stores
  - Custom Stores
  - Store Bindings
  - Page Store
  - Navigation Store
  - Updated Store
- Routing
  - What is Routing?
  - What problem does Routing solve?
  - Pages
  - Layout
  - Route Parameters
  - API Routes
- Errors and Redirects
  - Handling Errors and Redirects
  - Error Pages
  - Fallback Errors
  - Redirects
- Asynchronous Data
  - Promises & async/await
  - Fetching data from a REST API
  - Subscriptions
  - Stores
- Using Server Data
  - REST APIs with ASP.NET Core Web API
  - Web Sockets with SignalR
- ASP.NET Core
  - Configuration
  - Dependency Injection
  - Request Pipeline

  - Middleware
  - CORS
  - Swagger
  - Authentication
  - Authorization
  - Logging
- Web API
  - What is a Controller API?
  - Differences from Minimal API
  - REST Conventions
  - Build a REST API with ASP.NET Core Web API
  - Calling a REST API from a Svelte Component
  - Endpoint Metadata
  - Triple-Slash Comments and Swagger
  - Effective Error Handling
  - Action Filters
- Server Data
  - Create a Database with SQL Server
  - Unit of Work and Repository Patterns
  - Code to an Interface
  - Dependency Injection
  - Entity Framework Core / Dapper
  - Migrations (if EF Core selected)
- Authentication and Authorization
  - What is Authentication?
  - What problem does Authentication solve?
  - What is Authorization?
  - What problem does Authorization solve?
  - Securing Endpoint with JSON Web Tokens
  - Authorize Attribute
  - Svelte Login Form Component
  - Svelte Registration Form Component
  - Svelte Change Password Form Component
  - Svelte Login Status Component
- SignalR

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

  - Connect a Blazor App to SignalR
  - Hubs
  - Two-Way Data Transfer
  - Connect to SignalR from a Svelte Component
- Conclusion