

# Full Stack ASP.NET 10 with React

13-Days (104-Hour) Training Course

**Duration:** 13 Days (108 Hours, 8 Hours/Day)

**Prerequisites:** Basic programming knowledge (JavaScript, C# is a plus)

## What's New in .NET 10 (Course Highlights)

.NET 10 is the latest Long-Term Support (LTS) release (supported for three years, through November 2028) and ships throughout this course. The following highlights are introduced where relevant in each module:

### Languages & SDK

- C# 14: field-backed properties (the *field* keyword), *nameof* with unbound generics, and less boilerplate
- File-based apps – run a single *.cs* file directly (no project/solution) using the *#:sdk* directive
- New *dnx* command to run .NET tools directly, plus the new *.slnx* solution format

### Runtime & Performance

- JIT improvements: better inlining, devirtualization, and stack allocation
- NativeAOT enhancements and new CPU instruction support (AVX10.2)

### ASP.NET Core 10

- Blazor enhancements (WebAssembly preloading), improved form validation and diagnostics
- OpenAPI improvements, minimal API updates, and passkey support for Identity

### Libraries & Data

- EF Core 10: vector/embedding support and native JSON type for SQL Server
- Post-quantum cryptography APIs (ML-KEM, ML-DSA, SLH-DSA), *WebSocketStream*, async ZIP, and stricter JSON serialization

## Day 1: JavaScript ES6 & Development Setup (8 Hours)

### Morning Session (4 Hours)

#### JavaScript ES6 Essentials

- Let, Const vs Var
- Arrow Functions & Template Literals
- Destructuring & Spread Operator
- Promises & Async/Await
- ES6 Modules

### Afternoon Session (4 Hours)

#### Development Environment Setup

- Installing Node.js, .NET 10 SDK, and VS Code
- Creating a React App with Vite
- Creating an ASP.NET 10 API with Minimal APIs

#### Lab

- ✓ Hands-on ES6 coding exercises
- ✓ Set up a simple “Hello World” app in React and .NET 10

## Day 2: React & .NET 10 API Fundamentals (8 Hours)

### Morning Session (4 Hours)

#### React Basics

- JSX & Functional Components
- Props & State (useState, useEffect)
- Handling Events & Forms

### Afternoon Session (4 Hours)

#### .NET 10 API Fundamentals

- Creating Routes & Endpoints
- Middleware & Dependency Injection
- Handling HTTP Requests & Responses

#### Lab

- ✓ Build a React component with state management
- ✓ Create a basic API with endpoints and test with Postman

## Day 3: Database, API Development & State Management (8 Hours)

### Morning Session (4 Hours)

#### Database Integration with Entity Framework Core

- Setting Up a Database
- Migrations & Seeding Data
- CRUD Operations

### Afternoon Session (4 Hours)

#### State Management & React Forms

- Controlled vs Uncontrolled Components
- Formik & React Hook Form
- React Context API

#### Lab

- ✓ Implement a database and CRUD operations in .NET 10
- ✓ Build a React form that submits data to the .NET API

## Day 4: Authentication & Authorization (8 Hours)

### Morning Session (4 Hours)

#### User Authentication in .NET 10

- JWT Authentication
- Secure API Endpoints & Role-Based Access Control

### Afternoon Session (4 Hours)

#### Managing Authentication in React

- Storing Tokens & Managing Sessions
- Protecting Routes

#### Lab

- ✓ Implement user authentication and protect React routes

## Day 5: Advanced Backend & API Integrations (8 Hours)

### Morning Session (4 Hours)

#### Advanced API Development

- Repository & Service Layers
- Error Handling & Logging
- Async Programming in .NET 10

### Afternoon Session (4 Hours)

#### Consuming APIs in React

- Fetch API vs Axios
- Handling API Responses & Errors

#### Lab

- ✓ Refactor API using a repository pattern
- ✓ Fetch and display API data in React

## Day 6: File Uploads, Image Handling & Real-Time Communication (8 Hours)

### Morning Session (4 Hours)

#### File Uploads & Image Handling

- Uploading & Storing Files in .NET 10
- Integrating Cloud Storage (Azure, AWS)

### Afternoon Session (4 Hours)

#### Real-Time Communication with SignalR

- Introduction to WebSockets & SignalR
- Implementing Live Notifications

#### Lab

- ✓ Implement a file upload feature
- ✓ Create a live chat system with SignalR

## Day 7: Performance Optimization & Testing (8 Hours)

### Morning Session (4 Hours)

#### Performance Optimization

- Lazy Loading & Memoization in React
- Caching with Redis in .NET 10
- Query Optimization in EF Core

### Afternoon Session (4 Hours)

#### Testing & Debugging

- Unit Testing with xUnit
- Integration Testing
- Debugging & Logging in .NET 10

### Lab

- ✓ Optimize API queries and implement caching
- ✓ Write unit tests for API endpoints

## Day 8: Advanced React Features & Custom Hooks with Telerik Kendo UI (8 Hours)

### Morning Session (4 Hours)

#### Advanced React Concepts & Kendo UI

- React Portals & Higher-Order Components (HOC)
- Error Boundaries & Handling UI Errors
- Code Splitting & Lazy Loading
- Introduction to Telerik Kendo UI (KendoReact): Grid, Charts, Inputs, and Scheduler
- Theming and integrating Kendo UI components with the React frontend Integration

### Afternoon Session (4 Hours)

#### Custom Hooks & Performance Optimization

- Writing Reusable Custom Hooks
- Optimizing React Applications
- Webpack & Performance Tuning
- Custom UI with Kendo UI

### Lab

- ✓ Implement a custom hook for API calls
- ✓ Integration of Kendo UI
- ✓ Optimize a React application for better performance

## Days 9 & 10: End-to-End Full Stack Project (16 Hours)

### Project Overview: Building a Full-Stack Web Application

Students will apply all the learned concepts to build a real-world full-stack application. Examples include:

#### Project Options

- E-commerce App (User authentication, product management, cart, orders)
- Task Management App (Users, tasks, real-time updates with SignalR)
- Job Board (Job postings, user profiles, application submissions)

### Day 9 (8 Hours)

#### Morning: Backend Development

- Set up .NET 10 API with authentication
- Define data models & database integration
- Implement API endpoints

#### Afternoon: Frontend Development

- Set up React UI with routing
- Implement state management & API calls

### Lab

- ✓ Start developing a full-stack project, integrating backend & frontend

## Day 10 (8 Hours)

### Morning: Finalizing Features & Debugging

- Implement authentication & authorization
- Add file uploads, real-time notifications (if applicable)
- Optimize API performance

### Afternoon: Project Review & Presentation

- Final code review and debugging
- Deploy the app (optional)

## Day 11: Windows Services Migration to ASP.NET Core Web Applications

### Session 1: Introduction to Windows Services & ASP.NET Core

#### Topics

- Understanding Windows Services
- Limitations of Windows Services in Modern Applications

### Session 2: Converting Windows Services to Background Services in ASP.NET Core

#### Topics

- Introduction to IHostedService & BackgroundService
- Implementing a simple background service in ASP.NET Core

### Session 3: Using the Repository Pattern (IRepository)

#### Topics

- Why use the Repository Pattern?
- Implementing IRepository for database operations
- Dependency Injection in ASP.NET Core with IRepository
- Hands-on: Creating a generic IRepository for an application

### Session 4: Hands-on Migration

#### Topics

- Converting a sample Windows Service to an ASP.NET Core Background Service
- Integrating IRepository for data persistence
- Debugging & logging in background services

## Day 12: Advanced Windows Service Migration & Performance Optimization

### Session 1: Handling Background Tasks & Scheduling

#### Topics

- Using Hosted Services and Worker Services
- Scheduled tasks with IHostedService

### Session 2: Error Handling & Monitoring

#### Topics

- Exception handling in background services
- Implementing logging using Serilog & Application Insights
- Performance considerations in background services

### Session 3: Deployment Considerations

#### Topics

- Deploying ASP.NET Core Background Services

## Day 13: Reporting & UI with Telerik Reporting in ASP.NET 10

### Session 1: Introduction to Telerik Reporting

#### Topics

- Overview of Telerik Reporting
- Installing & configuring Telerik Reporting in ASP.NET 10
- Understanding .trdp / .trdx report definition files

### Session 2: Integrating Telerik Reporting in an ASP.NET Core Application

#### Topics

- Generating reports dynamically with the Telerik Reporting REST Service
- Connecting Telerik Reporting with Entity Framework Core
- Passing parameters to Telerik Reports

### Session 3: Exporting, Displaying Reports

#### Topics

- Embedding the Telerik Report Viewer in a React application
- Exporting reports to PDF, Excel, and Word