

.NET
Core

**INTRODUCTION TO ASP.NET
CORE DEVELOPMENT USING REACT
WITH REDUX EĞİTİMİ
5 GÜN**



Digital Vizyon
Akademi

www.digitalvizyon.net



İçindekiler

Eğitim Hakkında.....	3
Neler Öğreneceksiniz?	3
Ön Koşullar	4
Kimler Katılmalı.....	4
Outline	4
.NET Core SDK.....	4
ASP.NET Core Application Architecture	5
Application Configuration	5
Request Routing	5
Models.....	5
Controllers	5
Web APIs.....	5
Application State	6
Error Handling	6
Logging	6
Testing.....	6
Introduction to React and Redux.....	6
Development Tools.....	6
Functional Components	6
Class-Based Components	7
Hooks.....	7
Redux.....	7
Connect React to Redux	8
Unit Testing.....	8
React Router	8
Advanced Forms	9
Authentication.....	9

Eđitim Hakkında

“Introduction to ASP.NET Core Development using React with Redux Eđitimi”, ASP.NET Core ve React/Redux kullanan modern web uygulamalarının geliştirilmesine odaklanan geniş kapsamlı bir eğitim sunar. Bu eğitim, bu teknolojilerin temel konularını ve başlangıç seviye becerileri ele alır.

Eđitim, ASP.NET Core ve React/Redux’un temel özelliklerini, uygulama tasarımı, performans iyileştirme, güvenlik ve diđer kilit kavramları öğretir. Katılımcılar, gerçek hayattan örnekler ve projeler aracılığıyla bu konular hakkındaki bilgilerini artırır.

Ayrıca, bu eğitim, ASP.NET Core ve React/Redux kullanılarak web uygulamalarının tasarlanması ve geliştirilmesi sürecinde hangi araç ve teknolojilerin nasıl kullanılabileceđini öğretir. Katılımcılar, veri işleme, performans iyileştirme, uygulama tasarımı ve güvenlik ve diđer temel özelliklerin nasıl yönetileceđini öğrenirler. Ayrıca, ASP.NET Core ve React/Redux ile web uygulamalarının nasıl tasarlanıp yönetileceđini de anlarlar.

“Introduction to ASP.NET Core Development using React with Redux Eđitimi”, modern web uygulamalarını tasarlama ve yönetme sürecinin tüm temel yönlerini kapsar. Katılımcılar, etkili web uygulamaları tasarlamaya ve geliştirmeye başlamadan önce ihtiyaç duyacakları temel becerileri kazanırlar.

Eđitim programı, ASP.NET Core ve React/Redux konularının temelleriyle başlar. Katılımcılar, uygulama tasarımı, performans iyileştirme ve güvenlik gibi temel kavramları öğrenirler. Ayrıca, ASP.NET Core ve React/Redux tabanlı web uygulamalarının tasarlanması ve yönetilmesi konusunda nasıl bir rol oynadıđına dair bilgi sahibi olurlar. Bu bilgiler, katılımcıların web uygulamaları tasarlama sürecinde ihtiyaç duyacakları temel yapı taşlarını oluşturur.

Eđitimde, ASP.NET Core, React/Redux ve bunlarla ilişkili temel özellikler ve bileşenler üzerinde duruyoruz. Bu, katılımcılara veri işleme, uygulama tasarımı ve güvenlik gibi temel yetenekleri kazandırır. Konu akışları, uygulama tasarımı ve uygulama yönetimi gibi temel konuları işler.

Son olarak, bir web uygulamasının nasıl ASP.NET Core ve React/Redux tabanlı geliştirileceđi hakkında bilgi veriyoruz. Bu süreç, uygulamanın tasarımını, testlerini yapmayı, ve en sonunda web uygulamasının ASP.NET Core ve React/Redux tabanlı geliştirilmesini içerir. Bu bilgiler, katılımcıların web uygulamalarını başarılı bir şekilde ASP.NET Core ve React/Redux ile tasarlamalarına ve geliştirmelerine yardımcı olur.

Neler Öğreneceksiniz?

Introduction to ASP.NET Core Development using React with Redux eğitiminde şunları öğreneceksiniz:

- ASP.NET Core framework’ünün temel fonksiyonallikleri ve yapısı hakkında bilgi
- React with Redux kullanarak ASP.NET Core web uygulamalarının geliştirilmesi
- React component’lerinin oluşturulması, render edilmesi ve state yönetimi



- Veri erişimi için ASP.NET Core API'lerinin kullanımı
- React with Redux'un state yönetimi ve dispatch mekanizması hakkında bilgi
- React with Redux ile performanslı ve esnek web uygulamalarının geliştirilmesi.

Bu eğitim, katılımcıların ASP.NET Core ve React with Redux teknolojilerinin birleştirilmesiyle oluşan gelişmiş web uygulamaları geliştirebilmelerini hedefler ve katılımcıları bu teknolojileri etkili bir şekilde kullanmaya hazırlar.

Ön Koşullar

Introduction to ASP.NET Core Development using React with Redux eğitimine katılmak için aşağıdaki önkoşulların sağlanması gerekir:

- Temel web geliştirme becerileri ve HTML, CSS ve JavaScript deneyimi
- NET Core ve React teknolojilerine temel düzeyde hakimiyet
- Object Oriented Programming (OOP) kavramlarına hakimiyet.

Bu eğitim, katılımcıların ASP.NET Core ve React with Redux teknolojilerinin birleştirilmesiyle oluşan gelişmiş web uygulamaları geliştirebilmelerini hedefler. Eğitim sürecinde teorik bilgi verilen konuların yanı sıra, pratik örnekler ve uygulamalarla pekiştirilmesi amaçlanır.

Kimler Katılnmalı

Introduction to ASP.NET Core Development using React with Redux eğitimine aşağıdaki profesyoneller katılabilir:

- Web Geliştiricileri
- ASP.NET Core ve React kullanmayı planlayan ya da henüz başlamış olan geliştiriciler
- İleri seviyede web uygulamaları geliştirme becerilerine sahip olan geliştiriciler
- Front-end ve back-end geliştiricileri
- Büyük ölçekli web uygulamalarının geliştirilmesine katılmak isteyen profesyoneller.

Bu eğitim, geliştiricilerin ASP.NET Core ve React with Redux teknolojilerini birlikte kullanarak daha verimli ve kaliteli web uygulamaları geliştirebilmelerini hedefler. Eğitim, profesyonellerin mevcut becerilerini geliştirmelerine ve daha da ileri seviyede web uygulamaları geliştirebilmelerine yardımcı olacaktır.

Outline

.NET Core SDK

- Installation
- Version Management
- Command-Line Interface (CLI)



- Hello World Application
- Using Visual Studio Code for C# Coding
- Optional: Using Visual Studio 2019 for C# Coding
- Optional: Using Visual Studio for Mac for C# Coding

ASP.NET Core Application Architecture

- NuGet Packages and Metapackages
- Application Startup
 - Hosting Environments
 - Middleware and the Request Processing Pipeline
 - Services and Dependency Injection

Application Configuration

- Configure and ConfigureServices
- Configuration Providers and Sources
- Configuration API
- Options Pattern
- HTTPS, GDPR, and HTTP/2

Request Routing

- RESTful Services
- Endpoint Routing
- Attribute-Based Routing
- Route Templates
- Route Constraints

Models

- Persistence Ignorance
- Object-Relational Mapping
- Entity Framework (EF) Core
- AutoMapper

Controllers

- Responsibilities
- Requirements and Conventions
- Dependencies
- Action Results

Web APIs

- Introduction
- CRUD Operations



- Bad Requests
- Cross-Origin Resource Sharing (CORS)

Application State

- Client-Side Server-Side
- Items
- Session State

Error Handling

- Best Practices
- HTTP Error Status Codes
- Status Code Pages
- Developer Exception Page
- Exception Filters

Logging

- Configuration
- ILogger
- Serilog and Seq

Testing

- Unit Testing
- xUnit
 - Testing Controllers
 - Integration Testing

Introduction to React and Redux

- What is React?
- What problem does React solve?
- Development Ecosystem
- React versus other frameworks

Development Tools

- Create React App project generator
- React Developer Tools
- Running and Debugging a React Application
- Role of js
- Purpose of React and ReactDOM

Functional Components

- What are Components?



- Create Element and JSX
- Benefits of JSX
 - Fragments
 - JavaScript Arrow Functions
 - ES2015 Modules
 - JSX and Expressions
 - Displaying Collections of Data
 - JavaScript Array Maps and React Keys
 - Passing Data with Props
 - Validating Props with PropTypes
 - Default Props
 - Using Memo

Class-Based Components

- JavaScript Classes and Extends
- Configuring State
- Lifecycle Methods
- Google Performance Tool
- Context of Event Handlers
- Class Properties and Class Arrow Functions
- PropTypes and Default Props on Classes

Hooks

- Overview of Hooks
- State Hook
- Effect Hook
- Ref Hook
- Callback Hook
- Advanced Components
- Composition Inheritance
- Patterns: Specialization, Containment, and Higher Order Components
- Lifting State Up
- Forwarding Refs
- Context

Redux

- Managing Application State
- Three Principles of Redux
- Pure Functions
- Reducer Functions
- Composing Reducer Functions
- Dispatching Actions



- Action Creators

Connect React to Redux

- Connect React to Redux with React-Redux
- React-Redux Higher Order Components
- React-Redux Hooks
- Using State Selectors
- Optimizing State Selectors
- Connect React to ASP.NET Core REST API
- JavaScript Review: Callbacks, Promises & Async/Await
- Using the Fetch API with NET Core REST API
- Asynchronous Operations and React/Redux using Saga
- JavaScript Generators
- Sagas Helpers
- Declarative Effects
- Error Handling
- Connecting to Redux

Unit Testing

- Using Jest
- Organizing Tests and Test Suites
- Setup and Teardown of Tests
- Performing Assertions with Expect
- Using Spies
- Snapshot Testing
- DOM Testing
- Shallow Testing
- Generating Code Coverage Reports
- React Unit Testing
- Redux Unit Testing

React Router

- What is routing?
- URL as State
- React Router Hooks
- Configuring Routes
- Page Pattern
- Error Page
 - Redirects
 - Animated Transitions
 - Nested Routes



Advanced Forms

- What is Formik?
- Challenges with React and Forms
- Formik Higher Order Components
- Formik Hooks
- Form-Level Validation
- Field-Level Validation
- Synchronous and Asynchronous Validation
- Form Submission
- Form Submission Phases

Authentication

- Introduction
- NET Core Identity
- Cookie Middleware
- Authorization
 - Claims-Based Authorization
 - React Login Form
 - Login Error Handling
 - Integrating Authorization with Routing