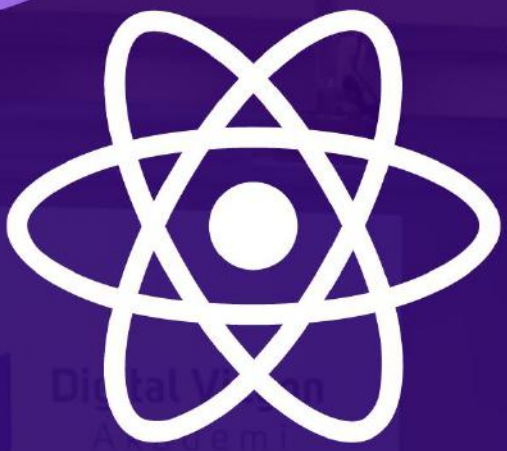

TEST-DRIVEN DEVELOPMENT (TDD) USING REACT.JS AND ES6 EĞİTİMİ

5 GÜN

Digital Vizyon Akademi

www.digitalvizyon.net

İçindekiler

Eğitim Hakkında

Test-Driven Development (TDD) Using React.js and ES6 eğitimi, web uygulamalarının kaliteli ve etkili bir şekilde nasıl geliştirilebileceğine odaklanan kapsamlı bir eğitim programıdır. TDD yönteminin gücünden faydalanarak, öğrenciler React.js ve ES6 kullanarak gerçek dünyaya uygun web uygulamaları geliştirmeyi öğrenecekler.

Eğitim başladığında, ilk olarak Test-Driven Development (TDD) kavramına derinlemesine bir giriş yapıyoruz. Bu bölümde, TDD'nin temel prensipleri, test yazma süreçleri ve TDD'nin uygulama geliştirme sürecine nasıl entegre edileceği üzerinde durulur. TDD'nin, kodun kalitesini artırma ve gelecekteki hataları azaltma gibi avantajları detaylı bir şekilde ele alınır.

Daha sonrasında, React.js kütüphanesinin temel kavramlarına geçiyoruz. React'ın component tabanlı yapısı, JSX, state ve props gibi özellikleri açıklanır. ES6 ile gelen yeniliklere, özellikle arrow fonksiyonları, sınıflar ve modüller gibi kavramlara vurgu yapıyoruz.

Bir sonraki adımda, TDD yöntemiyle React.js ve ES6 kullanılarak bir uygulama geliştirmeye başlıyoruz. Ayrıca, katılımcıları popüler test kütüphaneleri olan Jest ve Enzyme ile de tanıştırıyoruz.

Uygulama geliştirme süreci boyunca, performans optimizasyonu ve best practices konularına ağırlık veriyoruz. Öğrenciler, performansı artırmak ve uygulamalarını daha verimli hale getirmek için farklı teknikler ve araçlar öğrenirler.

Ayrıca, eğitimde sürekli entegrasyon (CI) ve sürekli dağıtım (CD) kavramları da ele alınır. Öğrenciler, kodlarını düzenli olarak test edip entegre etme ve üretim ortamına sürekli dağıtım yapma süreçlerini öğrenir.

Eğitim sona erdiğinde, katılımcılar React.js ve ES6 ile TDD yöntemini kullanarak etkili, güvenilir ve yüksek kaliteli web uygulamaları geliştirme konusunda yetenekli hale gelirler.

Test-Driven Development (TDD) Using React.js and ES6 eğitimi, kariyerlerinde ilerlemek, kaliteli kod yazma becerilerini geliştirmek, bununla beraber modern web geliştirme tekniklerini öğrenmek isteyen herkes için son derece değerlidir.

Neler Öğreneceksiniz?

Test-Driven Development (TDD) Using React.js and ES6 eğitimi sırasında, öğrenciler şunları öğrenecekler:

- Test-Driven Development (TDD) yaklaşımının ne olduğu ve nasıl kullanılabileceği.
- js ve ES6 kullanarak web uygulamalarının nasıl test edileceği.
- Jest ve benzeri test araçlarının nasıl kullanılabileceği.
- Uygulama bileşenlerinin nasıl test edileceği.
- Test-driven yaklaşımı kullanarak kodların güvenilirliğini ve kalitesini nasıl artırabileceği.

- Hızlı ve verimli bir şekilde uygulama geliştirme yapmanın yolları.
- Web uygulamalarını optimize etmenin ve geliştirmenin yolları.

Bu öğrenmeler, öğrencilerin gelecekteki web uygulama geliştirme projelerinde daha güvenli ve verimli bir şekilde çalışmasına yardımcı olacak ve onların kariyerlerine pozitif katkı sağlayacaktır.

Ön Koşullar

Test-Driven Development (TDD) Using React.js and ES6 eğitimine katılmak için şu ön koşullar gereklidir:

- JavaScript: Eğitim boyunca React.js ve ES6 kullanılacağı için temel JavaScript bilgisi gerekmektedir.
- js: Öğrencilerin React.js'i temel seviyede anlaması ve kullanması gerekmektedir.
- Web Geliştirme: HTML, CSS ve JavaScript gibi web geliştirme teknolojileri hakkında temel bilgiye sahip olmak gerekmektedir.
- Test Otomasyonu: Test otomasyonu konusunda temel bilgi sahibi olmak faydalı olabilir ancak gerekli değildir.

Bu ön koşulların yerine getirilmesi, öğrencilerin eğitim süresince daha verimli çalışmasını ve daha iyi anlamasını sağlayacaktır.

Kimler Katılmalı

Test-Driven Development (TDD) Using React.js and ES6 eğitimi aşağıdaki profesyonellere yöneliktir:

- Web Geliştiricileri: Eğitim, React.js ve ES6 kullanarak web uygulamalarının nasıl test edileceğini öğrenmek isteyen web geliştiricilere yöneliktir.
- Test Otomasyonu Uzmanları: Eğitim, React.js ve ES6 kullanarak test otomasyonu yapmayı öğrenmek isteyen test otomasyonu uzmanlarına yöneliktir.
- Başlangıç veya İleri Seviyede JavaScript Kullanıcıları: Eğitim, JavaScript ve React.js kullanarak test-driven yaklaşımı öğrenmek isteyen başlangıç veya ileri seviyede JavaScript kullanıcılarına yöneliktir.

Bu eğitim, katılımcıların profesyonel yetkinliklerini artırmasına ve gelecekteki web uygulama geliştirme projelerinde daha verimli ve güvenli çalışmasına yardımcı olacaktır.

Outline

React Quickstart with Create-React-App Development Ecosystem

- Code Editors and IDEs
- Lab 1: Installing and Configuring WebStorm IDE
- js

- EventEmitter
- Node Streams
- Node Modules
- Lab 2 – Getting Started with js
- Git
- What is Version Control
- History of Git
- What is Git?
- 3 States of Git
- Git Workflow
- Lab 3 – Version Control With Git
- Command Prompt
- Know Your Shell

Reproducible Builds

- Why Automate Your Build?
- Build Requirements
- npm
- Lab 4 – Initialize npm
- node_modules
- json
- npm install
- Lab 5 – Using npm as a Build Tool
- Lab 6 – Managing External Dependencies

Static Code Analysis

- Lint tools
- Configuring ESLint
- ESLint: What Can Be Configured?
- ESLint Rules
- Lab 7 – Automate Linting
- Lab 8 – Configure a Local Web Server
- Browser Development Tools

Test-Driven Development

- Goal of TDD
- The TDD Cycle
- Red, Green, Refactor
- Assertions

- JavaScript Testing Frameworks
- JS Exception Handling
- Jasmine Overview
- How Jasmine Works
- Test Suites
- Specs
- Expectations
- Matchers
- TDD BDD
- Lab 9 – Get Started with Jasmine
- Lab 10 – TDD Practice
- Automated Cross-browser Testing
- Karma
- Lab 11 – In-browser Testing with Karma

Modularity – Why is Modularity Important?

- CommonJS
- RequireJS
- ES6 Modules
- Front-end Modules
- Manage Modules Manually
- Front End Package Management with npm

Building and Refactoring

- Building the dist directory
- webpack
- Lab 12: Deploying with Webpack
- Lab 13 README Update and Refactoring

ES2015 (ES6)

- Variable Scoping with const and let
- let var
- Block-scoped Functions
- Arrow Functions
- Default Parameter Handling
- Rest Parameter
- Spread Operator

- Template Literals
- Enhanced Object Properties
- Array Matching
- Object Matching
- Symbol Primitive
- User-defined Iterators
- For-Of Operator
- Creating and Consuming Generator Functions
- Class Definition
- Class Declaration
- Class Expressions
- Class Inheritance
- Advanced JavaScript Topics
- “use strict”
- Understanding this
- 4 Rules of this
- What is this?
- Implicit Binding
- Explicit Binding
- new Binding
- window Binding
- map()
- Promises
- Promises vs. Event Listeners
- Why use Promises?
- Demo: Callback vs. Promise
- Using Promises
- Babel
- Lab 14: Transpiling with Babel
- Lab 15: Converting to ES6

The Document Object Model

- What is the DOM?
- Understanding Nodes
- EventTarget
- DOM Events
- Other Events
- Element
- Manipulating HTML with the DOM
- Manipulating HTML with JQuery
- Manipulating HTML with React

Introduction to React.js

- What is js
- Imperative API Declarative API
- Imperative Declarative Screen updates
- One-way Data Flow
- Virtual DOM
- Virtual DOM HTML DOM
- State Machines
- Lab 16, Part 1: Hello, React!
- Understanding Components
- render()
- ReactDOM
- React Development Process
- Props vs. State
- Setting Initial State
- super()
- Lab 16, Parts 2-3: Your first Component

JSX

- What is JSX?
- Using JSX
- JSX is not Exactly HTML
- Using React with JSX
- Using React without JSX
- Expressions in JSX
- Precompiled JSX
- Lab 17 – HTML to JSX

React Components

- Creating Components
- Pure Functions
- Benefits of Pure Functions
- I.R.S.T
- Single Responsibility
- Communication Between Components
- Props
- Ref Callback

- Lab 18: Passing Props
- Styles in React
- Style Components
- Lab 19: Style in React
- Forms
- Forms Have State
- Form Events
- Controlled Components
- Uncontrolled Components
- Lab 20: Controlling the Form
- Stateless Functional Components
- Lab 21: Refactoring the App
- Component Life-Cycle Events
- Life-Cycle Methods
- Mount/Unmount
- Mount/Unmount Life-Cycle Methods
- Data Life-Cycle Methods
- Component Life Cycle
- Events
- Lab 22: Life Cycle and Events
- Composition
- Reusable Components
- Presentational Components
- Container Components
- PropTypes
- Lab 23: PropTypes
- Testing React Components
- What to Test in a React Component
- Jest
- Mocking
- Snapshot Testing
- TestUtils
- Enzyme
- Shallow Rendering
- Lab 24: Testing React Components
- Lab 24.5: Testing with Jest and Enzyme
- Lab 25: Multiple Components
- React Router
- Lab 26: React Router
- Lab 27: React Router, Part 2

Flux and Redux

- Flux
- Flux Flow
- Flux Action
- Flux Dispatcher
- Flux Stores
- EventEmitter
- Redux
- Stores & Immutable State Tree
- Redux Actions
- Reducers
- Things You Should Never Do in a Reducer
- Reducer Composition
- Redux Store
- Redux Pros and Cons
- Lab 28: Redux Thermometer
- Lab 29: Implementing Redux
- React AJAX Best Practices
- Redux with Ajax
- What is Redux Middleware?
- What is Middleware Good For?
- Thunk
- Redux Saga
- Using Sagas
- Lab 30: SwimCalc App
- Lab 31: Redux Middleware
- create-react-app
- Lab 32: create-react-app and enzyme
- Relay and GraphQL

React Quickstart with Create-React-App Development Ecosystem

- Code Editors and IDEs
- Lab 1: Installing and Configuring WebStorm IDE
- js
- EventEmitter
- Node Streams
- Node Modules
- Lab 2 – Getting Started with js
- Git
- What is Version Control
- History of Git

- What is Git?
- 3 States of Git
- Git Workflow
- Lab 3 – Version Control With Git
- Command Prompt
- Know Your Shell

Reproducible Builds

- Why Automate Your Build?
- Build Requirements
- npm
- Lab 4 – Initialize npm
- node_modules
- json
- npm install
- Lab 5 – Using npm as a Build Tool
- Lab 6 – Managing External Dependencies

Static Code Analysis

- Lint tools
- Configuring ESLint
- ESLint: What Can Be Configured?
- ESLint Rules
- Lab 7 – Automate Linting
- Lab 8 – Configure a Local Web Server
- Browser Development Tools

Test-Driven Development

- Goal of TDD
- The TDD Cycle
- Red, Green, Refactor
- Assertions
- JavaScript Testing Frameworks
- JS Exception Handling
- Jasmine Overview
- How Jasmine Works
- Test Suites
- Specs
- Expectations

- Matchers
- TDD BDD
- Lab 9 – Get Started with Jasmine
- Lab 10 – TDD Practice
- Automated Cross-browser Testing
- Karma
- Lab 11 – In-browser Testing with Karma

Modularity – Why is Modularity Important?

- CommonJS
- RequireJS
- ES6 Modules
- Front-end Modules
- Manage Modules Manually
- Front End Package Management with npm

Building and Refactoring

- Building the dist directory
- webpack
- Lab 12: Deploying with Webpack
- Lab 13 README Update and Refactoring

ES2015 (ES6)

- Variable Scoping with const and let
- let var
- Block-scoped Functions
- Arrow Functions
- Default Parameter Handling
- Rest Parameter
- Spread Operator
- Template Literals
- Enhanced Object Properties
- Array Matching
- Object Matching
- Symbol Primitive
- User-defined Iterators
- For-Of Operator

- Creating and Consuming Generator Functions
- Class Definition
- Class Declaration
- Class Expressions
- Class Inheritance
- Advanced JavaScript Topics
- “use strict”
- Understanding this
- 4 Rules of this
- What is this?
- Implicit Binding
- Explicit Binding
- new Binding
- window Binding
- map()
- Promises
- Promises vs. Event Listeners
- Why use Promises?
- Demo: Callback vs. Promise
- Using Promises
- Babel
- Lab 14: Transpiling with Babel
- Lab 15: Converting to ES6

The Document Object Model

- What is the DOM?
- Understanding Nodes
- EventTarget
- DOM Events
- Other Events
- Element
- Manipulating HTML with the DOM
- Manipulating HTML with JQuery
- Manipulating HTML with React

Introduction to React.js

- What is js
- Imperative API Declarative API
- Imperative Declarative Screen updates

- One-way Data Flow
- Virtual DOM
- Virtual DOM HTML DOM
- State Machines
- Lab 16, Part 1: Hello, React!
- Understanding Components
- render()
- ReactDOM
- React Development Process
- Props vs. State
- Setting Initial State
- super()
- Lab 16, Parts 2-3: Your first Component

JSX

- What is JSX?
- Using JSX
- JSX is not Exactly HTML
- Using React with JSX
- Using React without JSX
- Expressions in JSX
- Precompiled JSX
- Lab 17 – HTML to JSX

React Components

- Creating Components
- Pure Functions
- Benefits of Pure Functions
- I.R.S.T
- Single Responsibility
- Communication Between Components
- Props
- Ref Callback
- Lab 18: Passing Props
- Styles in React
- Style Components
- Lab 19: Style in React
- Forms
- Forms Have State

- Form Events
- Controlled Components
- Uncontrolled Components
- Lab 20: Controlling the Form
- Stateless Functional Components
- Lab 21: Refactoring the App
- Component Life-Cycle Events
- Life-Cycle Methods
- Mount/Unmount
- Mount/Unmount Life-Cycle Methods
- Data Life-Cycle Methods
- Component Life Cycle
- Events
- Lab 22: Life Cycle and Events
- Composition
- Reusable Components
- Presentational Components
- Container Components
- PropTypes
- Lab 23: PropTypes
- Testing React Components
- What to Test in a React Component
- Jest
- Mocking
- Snapshot Testing
- TestUtils
- Enzyme
- Shallow Rendering
- Lab 24: Testing React Components
- Lab 24.5: Testing with Jest and Enzyme
- Lab 25: Multiple Components
- React Router
- Lab 26: React Router
- Lab 27: React Router, Part 2

Flux and Redux

- Flux
- Flux Flow
- Flux Action
- Flux Dispatcher
- Flux Stores

- EventEmitter
- Redux
- Stores & Immutable State Tree
- Redux Actions
- Reducers
- Things You Should Never Do in a Reducer
- Reducer Composition
- Redux Store
- Redux Pros and Cons
- Lab 28: Redux Thermometer
- Lab 29: Implementing Redux
- React AJAX Best Practices
- Redux with Ajax
- What is Redux Middleware?
- What is Middleware Good For?
- Thunk
- Redux Saga
- Using Sagas
- Lab 30: SwimCalc App
- Lab 31: Redux Middleware
- create-react-app
- Lab 32: create-react-app and enzyme
- Relay and GraphQL

React Quickstart with Create-React-App Development Ecosystem

- Code Editors and IDEs
- Lab 1: Installing and Configuring WebStorm IDE
- js
- EventEmitter
- Node Streams
- Node Modules
- Lab 2 – Getting Started with js
- Git
- What is Version Control
- History of Git
- What is Git?
- 3 States of Git
- Git Workflow
- Lab 3 – Version Control With Git
- Command Prompt
- Know Your Shell

Reproducible Builds

- Why Automate Your Build?
- Build Requirements
- npm
- Lab 4 – Initialize npm
- node_modules
- json
- npm install
- Lab 5 – Using npm as a Build Tool
- Lab 6 – Managing External Dependencies

Static Code Analysis

- Lint tools
- Configuring ESLint
- ESLint: What Can Be Configured?
- ESLint Rules
- Lab 7 – Automate Linting
- Lab 8 – Configure a Local Web Server
- Browser Development Tools

Test-Driven Development

- Goal of TDD
- The TDD Cycle
- Red, Green, Refactor
- Assertions
- JavaScript Testing Frameworks
- JS Exception Handling
- Jasmine Overview
- How Jasmine Works
- Test Suites
- Specs
- Expectations
- Matchers
- TDD BDD
- Lab 9 – Get Started with Jasmine
- Lab 10 – TDD Practice
- Automated Cross-browser Testing
- Karma
- Lab 11 – In-browser Testing with Karma

Modularity – Why is Modularity Important?

- CommonJS
- RequireJS
- ES6 Modules
- Front-end Modules
- Manage Modules Manually
- Front End Package Management with npm

Building and Refactoring

- Building the dist directory
- webpack
- Lab 12: Deploying with Webpack
- Lab 13 README Update and Refactoring

ES2015 (ES6)

- Variable Scoping with const and let
- let var
- Block-scoped Functions
- Arrow Functions
- Default Parameter Handling
- Rest Parameter
- Spread Operator
- Template Literals
- Enhanced Object Properties
- Array Matching
- Object Matching
- Symbol Primitive
- User-defined Iterators
- For-Of Operator
- Creating and Consuming Generator Functions
- Class Definition
- Class Declaration
- Class Expressions
- Class Inheritance
- Advanced JavaScript Topics
- “use strict”
- Understanding this

- 4 Rules of this
- What is this?
- Implicit Binding
- Explicit Binding
- new Binding
- window Binding
- map()
- Promises
- Promises vs. Event Listeners
- Why use Promises?
- Demo: Callback vs. Promise
- Using Promises
- Babel
- Lab 14: Transpiling with Babel
- Lab 15: Converting to ES6

The Document Object Model

- What is the DOM?
- Understanding Nodes
- EventTarget
- DOM Events
- Other Events
- Element
- Manipulating HTML with the DOM
- Manipulating HTML with JQuery
- Manipulating HTML with React

Introduction to React.js

- What is js
- Imperative API Declarative API
- Imperative Declarative Screen updates
- One-way Data Flow
- Virtual DOM
- Virtual DOM HTML DOM
- State Machines
- Lab 16, Part 1: Hello, React!
- Understanding Components
- render()
- ReactDOM

- React Development Process
- Props vs. State
- Setting Initial State
- super()
- Lab 16, Parts 2-3: Your first Component

JSX

- What is JSX?
- Using JSX
- JSX is not Exactly HTML
- Using React with JSX
- Using React without JSX
- Expressions in JSX
- Precompiled JSX
- Lab 17 – HTML to JSX

React Components

- Creating Components
- Pure Functions
- Benefits of Pure Functions
- I.R.S.T
- Single Responsibility
- Communication Between Components
- Props
- Ref Callback
- Lab 18: Passing Props
- Styles in React
- Style Components
- Lab 19: Style in React
- Forms
- Forms Have State
- Form Events
- Controlled Components
- Uncontrolled Components
- Lab 20: Controlling the Form
- Stateless Functional Components
- Lab 21: Refactoring the App
- Component Life-Cycle Events
- Life-Cycle Methods

- Mount/Unmount
- Mount/Unmount Life-Cycle Methods
- Data Life-Cycle Methods
- Component Life Cycle
- Events
- Lab 22: Life Cycle and Events
- Composition
- Reusable Components
- Presentational Components
- Container Components
- PropTypes
- Lab 23: PropTypes
- Testing React Components
- What to Test in a React Component
- Jest
- Mocking
- Snapshot Testing
- TestUtils
- Enzyme
- Shallow Rendering
- Lab 24: Testing React Components
- Lab 24.5: Testing with Jest and Enzyme
- Lab 25: Multiple Components
- React Router
- Lab 26: React Router
- Lab 27: React Router, Part 2

Flux and Redux

- Flux
- Flux Flow
- Flux Action
- Flux Dispatcher
- Flux Stores
- EventEmitter
- Redux
- Stores & Immutable State Tree
- Redux Actions
- Reducers
- Things You Should Never Do in a Reducer
- Reducer Composition
- Redux Store

- Redux Pros and Cons
- Lab 28: Redux Thermometer
- Lab 29: Implementing Redux
- React AJAX Best Practices
- Redux with Ajax
- What is Redux Middleware?
- What is Middleware Good For?
- Thunk
- Redux Saga
- Using Sagas
- Lab 30: SwimCalc App
- Lab 31: Redux Middleware
- create-react-app
- Lab 32: create-react-app and enzyme
- Relay and GraphQL