Micro Frontend Application Development - 5 days

By Dr. Vishwanath Rao

You Will Learn How To:

- Create device-independent and technology independent applications
- Create applications using Micro Frontend patterns
- How to reuse old web applications instead of migration
- Develop Components using web frameworks, Templates, and Decorators
- Consume REST services using Observables and do intercommunications using patterns
- Modularize applications with the Component Router
- Capture and validate input with template-driven forms

Requirements

Experience developing web pages at the level of JavaScript Essentials for the Full Stack Developer Using ECMAScript

Lab setup

Windows 10, Linux or Mac OS with minimum 8 GB RAM
Open internet to download from repositories
Chrome and Firefox browser
Visual studio code editor

Day 1

Architectural overview

- Simplifying development of complex modern applications
- The organization of an Angular app

- Supporting multiple client devices
- Configuring an Angular development environment
- Bootstrapping your first Angular application

Getting started with TypeScript

- Transpiling TypeScript to JavaScript
- Building an app with TypeScript

Typescript Deep Dive-Type Script

Defining components

- Component meta data
- Templates&views
- Template syntax
- Data binding
- User input
- String interpolation
- Component styles
- Angular element
- Angular directives
- Structural Type Directives
- Attribute Type Directives
- HostsListeners
- HostBinding
- Attribute
- Property& Attribute binding
- Property Binding vs Interpolation

- Angular Pipes
- Angular Services
- Angular Forms - only introduction

- Observables and RxJS

Day 2

Introduction to React

Now we will start learning how React works. This lecture will include fundamentals of React.

- Introduction to ES6 modules
- Components and elements in React

- Component Lifecycle
- Introduction to JSX
- Functional components
- Component state management

React Fundamentals 2

In this lecture we will cover more React concepts and features.

- Error Boundaries
- Fragments
- Refs
- Handling events & building forms
- Pure class components & React.Memo (new)
- React Context API
- Typechecking With PropTypes

- Creating Reusable Components
- React.Component vs React.pure Component
- Composition vs Inheritance
- Code Reusability and Optimization
- Fragments
- Bundling
- Deploying

- Stateful Components
- Stateless Components
- Local Storage
- Routing
  - Basic Routing and Passing Params
  - Hyperlinks

Day 3

Overview of micro-frontends and the problems they solve Available options
Example implementations
Sync and async services
How to use iframes to split frontend apps
Incremental upgrades
Simple, decoupled codebases
ES6 Javascript for Microfrontends Universal Rendering

Client Side Integration
Micro services Architecture
Modern Web Application Development
Monolithic Frontends
Teams with Micro Frontends
Documents-to-Applications Continuum Progressive Web Apps

Day 4

Advanced DOM for MicroFrontends Custom Elements
Team Product and its development Page Composition
Page transition
Prototypes
Parent Child Communication DOM Modification
DOM Events
Client Parent Communication
Sibling Communication
Server Side Rendering or Universal Rendering Custom Elements with server side includes Data Fetching
Loading the states
Navigating between Pages

Day 5

Isolated CSS
Coherent User Interface
Style Guides & Pattern Libraries Performance on initial load Performance while using the site Loading CSS
Loading JS
Navigating between pages
soft vs. hard navigation
universal router
Server-side template composition
Autonomous teams
Build-time integration
Run-time integration via Web Components
Run-time integration via JavaScript Run-time integration via iframes
Integration Testing