Java Script Full Stack Experts

Course 4579 – 130 Hours

Overview:

These days, web based services tend to be more atomic, focused and intensively released. These characteristics forces a new generation of Web Developers that are familiar with the different wide aspects of a web software development. These aspects include 3 basic tiers (Front-end, Backend & Integration), the preferred technologies and best practices for each tier, the related configuration & basic administration and the understanding of each tier role & responsibility. This wide knowledge is what gives a developer the ability to rapidly update / change / develop feature for intensive releases – Full Stack Developer.
Java Script became a very popular solution for Full Stack Web Developers. There are several reason for it:

- Very effective server side framework and technologies lately appeared
- High performance of implemented web modules and web services in Java Script
- Availability of a server-side solution since it is based on a popular client-side language
- Mature of Client-MVC oriented frameworks
- Rich set of libraries for any kind of view and graphics
- Usage of Java Script in Single Page Applications
- JSON popularity as a self-descriptive text based protocol

On Completion, delegates will be able to:

Develop rich high strands, component based, full stack web applications.
Will have deep knowledge of latest JS technologies both for client and server side.

Target Audience:

Experienced developers who aim to learn the latest JS and web based technologies.

Pre-requisites:

- Rich Experience in Web Based Technologies such as HTML, CSS
- Rich experience with Java Script fundamentals
- Experience with JavaScript OOP
- Understanding of HTTP and AJAX

Course duration:

- 130 academic hours
- Lessons twice a week, between 17:30-21:45
- Studies will not take place on holidays. Recent vacations panel will be distributed to students

Certificate of Completion:

In order to be eligible for a course certificate participants have to meet the following requirements:

- Presence in 85% of the sessions at least

Course Contents:

Module Title	Hours
Advanced Java Script	**20**
Advanced CSS	**15**
REST Web Services	**5**
React.JS	**50**
Node.js & Express	**40**
Total	**130**

Module Title	Module Description
Advanced Java Script	
JavaScript Pitfalls	▪ Typeof operator ▪ Undeclared vs. Uninitialized variable ▪ Implicit Variable declaration ▪ No integral data type ▪ String is immutable ▪ Undefined value ▪ Strict vs. Abstract comparison ▪ Logical Operator ▪ Array is dynamic ▪ Don't mix object with array ▪ Where to declare variables inside function? ▪ Function overloading ▪ Function is an object ▪ Function.apply vs. Function.call ▪ Function creates scope ▪ Closure ▪ Self-executing function
Object Oriented JavaScript	▪ Module Pattern ▪ From Module to Class ▪ Function as Constructor ▪ Prototype ▪ Inheritance ▪ Namespace ▪ Objects and DOM ▪ me, self and that
ECMA Script 5.0 & 6.0	▪ Strict Mode ▪ Object.create ▪ Getters and Setters ▪ Reflection ▪ let ▪ Class

Module Title	Module Description
	▪ Module ▪ Iterator ▪ Generator ▪ Arrow function ▪ Binary data ▪ Collections ▪ Proxy ▪ Promise
jQuery	▪ jQuery Library Introduction ▪ jQuery basics – document ready, callback functions ▪ jQuery structure and components ▪ jQuery Selectors ▪ Traversing document elements ▪ Modifying CSS attributes ▪ Binding and unbinding events ▪ Using jQuery's AJAX Features ▪ jQuery Extensions (Plugins) ▪ jQuery UI plugin library ▪ Writing your own custom jQuery plugin
AJAX	▪ Why do we need it? ▪ XMLHttpRequest ▪ $.ajax ▪ AJAX Threading Model ▪ JSON ▪ JSONP
Advanced CSS	
Foundation	▪ Introduction ▪ Grid and layouts ▪ Fast prototyping ▪ Interactivity with Java Script Components
Bootstrap	▪ Obtaining and Using Bootstrap ▪ Bootstrap Grid System ▪ CSS Techniques with Bootstrap ▪ Typography - fonts and icons ▪ Tables ▪ Forms
Introduction to SaSS	▪ Nested Selectors ▪ Parent References ▪ Properties ▪ Processing SASS ▪ Examining Output ▪ SASS Variables ▪ Using SASS to strip the unavoidable repetition from your CSS ▪ Creating and Referencing SASS Variables ▪ Variable scope usable SASS programing with Mixin ▪ CSS Rules and Mixin ▪ Mixin parameters

Module Title	Module Description
REST Web Services	
REST	▪ Introduction ▪ RPC ▪ HTTP Based Integration ▪ REST API ▪ RESTful ▪ Relevance for SPA and future internet clients
React.js	
Introduction to React.js	▪ Github links for React libraries ▪ Environment setup ▪ Project setup ▪ Basic concepts and terms ▪ Most basic components ▪ The virtual DOM ▪ Introduction to React.js event model
React components	▪ Basic components ▪ Components structure ▪ Passing properties ▪ Managing component state ▪ Using references ▪ Pure components
Redux & React	▪ FLUX architecture ▪ Introduction to Redux o Redux actions o Redux reducers o Redux store o Redux middleware ▪ Combined with React o Use store with React o Using reducers and actions to manage component state
React routing	▪ React router ▪ Single routes ▪ Multiple routes
Advanced Redux - Middleware	▪ Async Actions ▪ Logging ▪ Crash reporting
Advanced React	▪ Project structure ▪ Some selected open-source supportive libraries ▪ Best practices
Node.js & Express	
Introduction to Node.js	▪ What is Node and what is it not ▪ Node.js Features ▪ Our first Node.js script: Hello World ▪ Hello Server: Building a web server in Node.js ▪ Debugging node applications
Building your Stack	▪ Pulling in other libraries ▪ Building custom libraries ▪ Asynchronicity and callbacks ▪ Blocking vs. non-blocking I/O ▪ Working within the event loop

Module Title	Module Description
Modular JavaScript with Node.js	▪ Writing Modular JavaScript with Node.js ▪ Core Modules ▪ Installing Packages ▪ Publishing packages
Avoiding common pitfalls with Async.js	▪ Introducing the Async problem ▪ Async.js Library to the rescue ▪ Collections ▪ Flow Controllers
Working with the file system	▪ Sync and Async operations ▪ Files manipulations ▪ Folder manipulations ▪ Putting the file-system module together Async.js
Data Access 01 – MySQL	▪ Installing MySQL node package ▪ Simple db connection ▪ CRUD example ▪ Putting together Async with MySQL operations
Command-line interfaces	▪ The built-in REPL ▪ Custom REPL ▪ Using external libraries ▪ Receive command-line arguments ▪ Build command-line tools
Unit-Testing Node Applications	▪ Introduction to unit testing ▪ Testing with Mocha and Should ▪ Suits, specs & Reporters ▪ Testing Synchronous code ▪ Testing Asynchronous code
Building Web applications with the Express Framework	▪ Introduction to Express, installation and basic setup ▪ Application configuration ▪ Routing ▪ Views and Templating options ▪ Persistence with Cookies, In-Memory Sessions and session-stores. ▪ Authenticating users with passport local ▪ Social Auth with Passport.js
Data Access 02 – mongo DB	▪ Tooling up – installing mongo, clients and drivers. ▪ Mongoose Schemas ▪ CRUD operations ▪ Single Page Applications with Express, Mongoose and Angular.js
Real-time communication	▪ Introduction to real-time applications ▪ Listen & emit ▪ Readable streams – streaming chunked data ▪ Piping Readable streams to Writable streams ▪ Sockets on the Server and the Client ▪ Build a chat application
Services, Observers and the RxJS library	▪ Understanding Reactive Programing ▪ Working with the RxJS library ▪ Working with data Observables ▪ Promises vs. Observables ▪ Implementing custom services