**Week-1 : JavaScript & Typescript**

**day-1:**

* big picture of javascript language
* data types
  + primitive & reference types
* execution-context / scope
  + variable hoisting
  + let & const keywords
* literal style objects
* error handling
* functional programming
* how to create/define functions
* function with default & rest parameters
* functional programming principles
* higher-order-functions ( HOF )
* closures
* imperative vs functional style coding

**day-2:**

* function binding
* mystery of 'this' keyword
* static vs dynamic function-binding
* dynamic function binding with call(),apply and bind() methods
* object oriented programming
* ES5 class / constructor-function
* prototype based inheritance
* ES6 class syntax
* class based inheritance
* class with static variables & methods
* data structures
* array/list
* set
* map

**day-3:**

* ES6 And Above
* Arrow Function
* DE structuring
* spread operator
* iterables
* for-of-loop
* obj-literal enhancements
* modules, packages & tools
  + how to organize javascript code? traditional & modern approach
  + Module standards
    - commonJS module standard
    - ES module standard
  + package managers
    - NPM & Yarn
  + transpiler
    - babel
  + module bundler & loader
    - webpack

**day-4:**

* javascript runtime internal
  + single-thread-model with event-loop
  + event-queue with event-loop
  + callbacks
  + non-blocking/asynchronous io
* async programming apis
  + promise api
    - async-await keywords
  + reactive programming with RxJS library ( overview )
* unit-testing
* why unit-testing important?
* TDD & BDD testing style
* unit-testing framework - JEST
* JEST - sync & async code testing
* JEST - matchers
* JEST - mock functions

**day-5:**

* Typescript
  + javascript vs typescript
  + basic & advanced types
  + interfaces
  + classes, enum and generics
  + decorators
* utility libraries ( self learning )
* lodash
* moment.js

**Week-2 : UI foundation**

**Day-6:**

* what makes web-UI?
  + Elements, Style and Behavior
* HTML5 new elements
* CSS3
  + common style properties
  + box model
  + flexbox
  + grid
  + Media Quires
  + Animations & Transitions
  + Web Font
* Responsive Web Design (RWD) Principles

**Day-7:**

* Responsive Web Design (RWD) Framework: bootstrap
  + Layout
  + Content
  + Forms
  + Components
  + Helper and Utilities
* Case-Study-1: web-ui using bootstrap

**Day-8:**

* Writing Maintainable styles : CSS pre-processor
  + Introduction to SASS
  + Variables
  + Nested rules
  + Mixins
  + Functions. And More
* Case-Study-2: web-ui using bootstrap

**day-9 & 10 :**

* Browser Apis core & HTML5 Apis
* Core apis
  + DOM, Timer and XHR/Fetch api
* connectivity
  + web sockets & server sent events (sse)
* offline & storage apis
  + session & local storage
  + service workers
* multi-media
  + audio & video
* graphics
  + canvas and svg
* performance & integration api
  + web workers
  + history api
  + drag & drop
  + geo-location
  + full screen api
  + detecting device orientation

**Week-3 : Javascript Lib and Frameworks**

**day-11:**

* challenges while developing UI applications
* compatibility issues
* performance & memory issues
* data-binding issues
* UI architectures
  + MVC vs component-based architectures
* component-based UI architecture
  + what is component?
  + characteristics of UI components
* how to create UI-components?
* Introduction to Progressive Web Apps (PWA)

**Day-12:**

* Challenges while working browser core-apis ( DOM & XHR )
  + Using jQuery Core
  + Events
  + Effects
  + Ajax
  + Plugins
  + Performance
  + Code Organization
  + jQuery UI
  + jQuery Mobile

**Day-13:**

* JSON and Backend API Integration
  + JSON – what & why we need
  + Understanding REST Api principles
  + Consuming API : XHR/Fetch and axios
  + Best practices

**Day-14:**

* UI and UX basics
* Case-study

**Day-15:**

* Final case-study implementation
* Q & A