**UI Technologies Training**

**Introduction**

* UI Developer roles and Responsibilities
* UX designer roles
* Technologies needed
* Power of UI
* Current market requirements on UI

**HTML**

* DOM
* Structure of HTML Page
* Including external page links in a page using anchor tags and its properties
* Working with row and column data using table tags
* Hiding and unhiding elements using display property
* img tag, p tag, ul and ol tags, li, nobr, hr, br etc
* Layouts, forms, buttons
* Input fields (textbox, radiobutton, checkbox, dropdown, textarea etc)

**HTML5**

* Difference between HTML5 and HTML 4
* List of Browsers support HTML5
* Doctype
* Media tags (audio and video tags)
* Graphics using Canvas tag
* Drag and Drop features
* Working on locations lat and lng using Geolocation
* Storing userpreferences using Localstorage.

**CSS 3**

* Difference between CSS2 and CSS3
* Adding borders and backgrounds
* Advanced text effects(shadow)
* 2D and 3D Transformations
* Adding Transitions to elements
* Adding animations to text and elements

**Responsive Designs**

* Difference between multiple devices, making a page to work on multiple devices
* Media queries
* Introduction to Bootstrap CSS API

**More CSS properties**

* Mandatory tags in html page (html, head, body)
* What is CSS
* Different ways of applying css for elements, and priority chain of css
* Heading tags(H1…H6), Tags and attributes(Class, Id, style..etc)
* Inline and blocklevel elements
* Adding borders, font, pseudo classes,
* Positioning elements (absolute, relative, fixed and static)
* Image spriting
* Boxmodel (margins, padding)
* Floating elements (float left, right etc.)
* Including external resources
* Absolute and Relative paths
* Including external resources like css, images etc

**Form Elements**

* Get & Post
* Validating input values in a form.
* Form action and type.

**JavaScript**

* Introduction
* Data types and data structures in Js
* Control structures, if, ifelse, while, for, switch case statements
* Dynamic creation and maniplation of dom elements using js
* Adding dynamic event listerners to dom elements
* Event capturing and event bubbling
* Validations using key charcodes
* Arrays
* Predefined methods in arrays
* Strings and predefined methods
* Objects
* JSON

**Advanced JavaScript**

* Prototyping in JavaScript
* Closures in JavaScript
* Inheritance in JavaScript
* Adding methods for an object

**jQuery Framework**

* Onload and onready diffrence
* jQuery selectors
* Multiple ways of refering dom elements using jQuery selectors
* jQuery methods
* Adding dynamic properties for dom elements
* Toggleing elements
* Creating dynamic elements using jQuery
* Finding elements using jQuery techniques
* Filtering elements
* Binding events
* Dynamic binding
* List of events been supported in jQuery(blur, change, click, dbclick….etc)

**AJAX**

* Advantages with Ajax and its limitations
* Samples working with Ajax
* Different data formats used in Ajax (string, xml, Json, etc)
* XML and JSON difference
* Crossdomain interactions using JSONP

**AngularJS Framework**

* Introduction
* Power of ngjs
* Controllers and modules in ngs with sample program
* Data binding with sample program
* What is filter and adding filters to dom elements with sample code.
* Loading multiple page content into a single page using ng-routing – Working example.
* Developer Tools Addons:
* Firebug, chrome and IE Developer tools.

**NodeJS**

* Node.js Environment setup
* First application,REPL terminal
* Package Manager
* Callbacks concept
* Event loop and Emitter
* Buffers
* Streams
* FileSystem
* Global objects
* Utility modules
* Web module
* Express framework
* RESTful API
* Packaging

**REACTJS**

* Virtual DOM
* React without JSX
* Npm Tools
* Gulp vs Grunt
* Gulp basic
* Gulp build react
* ES5 vs ES6
* Babel trans-compiler
* Babel options (.babelrc)
* React using ES6 syntax
* Key differences
* Stateless vs Stateful components
* Parent - Child components
* Smart Parent
* **Lifecycle of Components**
* Forcing component rendering
* Reusable Components  
  Controlled Components
* **Server Side Communication**
* Promises using Fetch
* CORS
* Get vs POST
* Other REST methods
* **Routing and SPA**
* Configuration based routing
* Passing parameters
* **React Architecture with Flux / Redux**
* State container
* Reducers
* Principles
* Store
* Actions
* Dispatcher
* Routing
* **Unit Testing**
* Cucumber
* Enzyme
* Jasmine TestFramework

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |