**CUSTOMIZED FRONTEND TRAINING COURSE OUTLINE**

**DURATION:** 08 Days (64 Hrs.)

**SOFTWARE/HARDWARE -**

* + Windows 10
  + 8GB RAM
  + Latest Chrome / Firefox
  + Any preferred IDE (VS Code)
  + Internet Connection
  + NodeJS Installer
  + Permission to download & install NPM libraries on participant machine

**COURSE OUTLINE -**

**Day 01:**

**HTML**

* Introduction to Web Programming
* Basic tags, block and inline level, paired and unpaired
* List and Table
* HTML 5 forms
* iframes, symbols and entities, video, audio

**Assignment**: Create a HTML profile page with links, images, lists, tables etc.

**CSS**

* Basic selectors and basic styles
* Inline, internal and external stylesheet, borders, box layout
* Media queries, animations, transitions, pseudocode
* Introduction to bootstrap

**Assignment**: Design the HTML page by applying the styles learned

**Day 02:**

**JavaScript**

* Data types, Variables(var, let, const), operators, dialog boxes, == and ===
* if-else, switch-case, loops, null/nan/undefined/reference error
* Arrays and basic array methods (map, filter and reduce)
* String and string methods
* Introduction to function (named functions, return and parameters)
* function expression and IIFE
* Closures
* introduction to DOM
* event handling and basic DOM methods
* JavaScript objects and Object.asssign()
* ES6 concepts - default , rest and spread parameters, destructuring, for-of
* fetch API and storage

**Assignment:** using JavaScript objects and fetch API implement an assignment

**Day 03**

**React**

* What is module, a basic demo on es6 class and constructor
* Import and export of modules
* Download and install node and create-react-app
* Using create-react-app and undesrtand the react folders
* Introduce JSX syntax, template literals
* Render HTML elements, build first class and functional component
* Pass data via props to functional and class component and difference between them
* State vs props in both functional and class component and children props
* Lift state up,

**Assignment**: creating components, pass props and using state

**Day 04:**

* Listen to events, event handlers, child to parent
* Conditional rendering
* Lifecycle method using class-based component
* Lifecycle method using functional based component
* Rendering lists
* Understand keys in list and add styles
* Form handling in react and change state of input fields
* 2-way binding
* Controlled vs uncontrolled component
* Other form element (radio, checkbox, dropdown)
* Form validation
* Handle form submission
* Higher order components
* Memorization

**Assignment:** form submission and CRUD operation

**Day 05:**

* Introduction to state management
* Different ways of state management.
* Context API - Problems with props drilling
* Mutable vs immutable state in react and how to modify using spread operator

**Redux**

* Introduction to redux
* introduction to redux terminologies and simple example for redux without react
* Steps to add redux in react
* Create reducer and actions in redux using redux-toolkit
* Provide store and dispatch actions from components
* Attach payloads to actions

**Assignment:** Using redux-toolkit for state management

* Introduction to redux thunk and why is it used
* Perform asynchronous operations using redux thunk
* Using axios library in react and
* CRUD operations using redux and axios (using json-server)

**Assignment:** CRUD operations using JSON Server

**Day 06:**

**Routing and SPA**

* What is SPA and why we need routing
* Introduction to react router, installation and basic configuration
* Understand how to create different routes and navigate
* Pass parameters to routes and nested routes
* Navigate programmatically
* Dynamic routes
* Index route and authentication in react app
* Login and logout functionality for react app authentication

**Assignment:** Authentication using JSON Server and loading the dynamic links

**Day 07:**

* NodeJS : Introduction
  + Foundation
    - The Node.js Platform
    - Installing Node.js
    - Using Node.js to execute scripts
* Node Projects
  + The Node Package Manager
  + Creating a project
  + The package.json configuration file
  + Global vs. local package installation
  + Configuring Node for JavaScript
* HTTP
  + The HTTP protocol
  + Building an HTTP server
  + Rendering a response
  + Processing query strings
* File System
  + Synchronous vs. asynchronous I/O
  + Path and directory operations
  + \_\_dirname and \_\_filename
  + Asynchronous file reads and writes
* Modules
  + Modularization
  + Defining modules with exports
  + Modules are singletons
  + Creating a package
  + Module scope and construction

**Task : CLI Application – Notes App**

**Day 08 :**

* Express
  + The model-view-controller (MVC) pattern
  + Building a front-end controller
  + Defining routes
  + Using REST – GET/POST/PATCH/DELETE
  + Adding middleware
  + Request / Response
* Data Sources
  + How Node.js connects to databases
  + RDBMS databases and NoSQL databases
  + Connecting to NoSQL databases (MongoDB)
  + Performing CRUD operations
  + Building client requests to web services

**Task : CRUD REST APIs for Todo App**

\*\*\*\*\*\*\*\*\*\*