

An ISO 9001 : 2015 Certified Company



SOFTWARE TRAINING & DEVELOPMENT

☎ 040-2374 6666 | 23734842

✉ info@nareshit.com

🌐 www.nareshit.com

📘 nareshit

📺 nareshitech

📺 nareshit

Find Latest IT Jobs
SeshaJobs.com

Opp. Satyam Theatre, Durga Bhavani Plaza, Ameerpet, Hyd-16

UI Full Stack Web Development with React

React using JavaScript & TypeScript

Introduction

- UI Developer roles and responsibilities
- UX designer roles
- Technologies needed
- Power of UI
- Current market requirements on UI
- Basic Technologies needed
- Difference between Front end and Backend
- Sample web pages
- Crawling and Meta tags

Basics – HTML

- Exploring existing pages
- Browsers & Editors
- DOM
- Structure of HTML Page
- Mandatory tags in html page (html, head, body)
- Heading tags (H1...H6), Tags and attributes (Class, Id, style...etc.)
- Inline and block level elements

CSS

- What is CSS
- Different ways of applying CSS for elements, and priority chain of CSS
- CSS Properties (color, font, size, border...etc.)
- Box model, Margin & Padding
- Positioning Elements, Floating Elements

More HTML Tags

- Including external page links in a page using

- anchor tags and its properties
- Working with row & column data using table tags
- Hiding and un-hiding elements using display property
- img tag, p tag, ul & ol tags, li, nobr, hr, breac
- Layouts, forms, buttons
- Input fields (textbox, radio button, checkbox, dropdown, text area etc.)
- Debugging HTML & CSS (Firebug, IE and Chrome developer tool)
- Creating Tabs and menu lists

More CSS Properties

- Adding borders, font, Pseudo classes & Pseudo Elements
- positioning elements (absolute, relative, fixed & static)
- Image spiriting
- Box model (margins, padding)
- Floating elements (float left, right etc.)
- Including external resources
- Absolute and Relative paths
- Including external resources like CSS, images etc.
- Display Property (inline, block, inline block etc)
- Display none and hidden elements.

Form Elements

- Get & Post Communication

- Validating input values in a form.
- Form action and type

JavaScript

- Data types and data structures in Js
- Control structures, if, if-else, while, for, switch case statements
- Hosting & type casting

JavaScript Supported Data Structures

- Arrays and Predefined methods.
- Working on logical programs using Arrays.
- Predefined methods in arrays
- Strings and predefined methods
- Objects
- JSON, SET, MAP
- Iterating through Dynamic JSON Object.
- Exploring Predefined Date object.

Adv. JavaScript

- Inheriting Static Object using proto.
- Inheritance using Object.create()
- Java Script Classes
- Prototyping
- Inheritance using prototyping.
- Global and local variables
- Securing Data using Closures
- Exception Handling
- Handling Predefined and user defined exceptions
- Exploring try, catch, finally and throws.

Sass

- Sass Intro
- Sass Installation
- Sass Variables
- Sass Nesting
- Sass @import
- Sass @mixin
- Sass @extend

jQuery

- History and version explore
- Difference between Minified and non-minified JS files.
- On load and on ready difference
- jQuery selectors
- Multiple ways of referring DOM elements using jQuery selectors
- jQuery methods
- Adding dynamic properties for DOM elements
- Toggling elements
- Creating dynamic elements using jQuery

jQuery Traversing methods

- Traversing Siblings, Children's&Parent Elements.
- Traversing Ancestors.
- Finding elements using jQuery techniques
- Filtering elements

Events using jQuery

- Binding events, Dynamic binding

- List of events been supported in jQuery (blur, change, click, dblclick....etc.)s

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
- Cross domain interactions using JSONP
- Jquery Promises with multiple conditions
- Handling Multiple AJAX Calls using Jquery Promises.

jQuery Animations

- Animation Effects using
- Hide, Show
- Sliding up and down, Fading, Animate Method.

jQuery Templating

- Loading DOM dynamically using Jquery templates
- loading templates using AJAX

HTML 5

- Difference between HTML5 and HTML 4
- List of Browsers support HTML5
- Doc type&Semantic Tags
- Multithreading Using Web Workers.
- Media Support (audio and video tags)
- Graphics using Canvas tag and SVG Tags.
- Drag and Drop features
- Offline Application using Application Cache
- Exploring Navigator Object
- Working on locations lat and lng using Geolocation
- Storing user preferences using Local storage & Session Storage

CSS 3

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

Responsive Design

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

Bootstrap – 4 along with components

- Basic-Typography
- Text-Alignments, Floats-Positions
- Colors-Backgrounds, Margins-Paddings
- Sizings, Breakpoints, Buttons, Navbars
- List-Groups,Forms, Input-Groups
- Alerts-Progress
- Tables, Cards, Carousel, Collapse, Modal
- Grid-System, Grid-Alignments, FlexBoxes

ECMA 7 (Latest JavaScript)

- Const and let keywords
- Arrow functions, advantages
- Extended parameter handling, default parameters
- Template literals
- Enhanced object parameters
- Creating direct classes, inheritance
- Implementing promises
- new data structures sets and maps
- New string built-in methods
- Using iterators

NODE JS

- Introduction to Server-Side Scripting
- Node JS Features and Drawbacks
- Setup Environment for Node.js
- Node JS Program architecture
- Node JS Web Server
- Node JS Global Objects
- Node JS OS Objects
- Node JS Error Handling
- Node JS Event Loop
- Node JS File System
- Async and Sync
- Connecting with Database
- Handling CRUD Operations
- Express
- Routing in Express
- Response Methods in Express
- Serving Static Files
- Express JSON Parser
- Express CORS
- Creating Web API

Express JS

- Exploring Express module
- Setting up express environment
- Creating http server using express

Express Sessions

Data Communication

- Sending and receiving Post & GET Data
- Reading data from GET/POST requests

Templates

- Creating templates using PUG/JADE
- Generating Dynamic html pages from server
- Responding a template for a client request

File system

- Exploring File system, Reading and writing to files

REST API & WebServices

- Introduction to REST API
- REST Architecture

DB Connection

- Connecting to Mongo DB & Connecting to SQL

MONGO DB

- Introduction to MongoDB
- Configuring Server and Client
- MongoDB Compass
- Creating Database
- MongoDB Commands
- MongoDB CRUD Operations

React

REACT INTRODUCTION

- Overview of frameworks, libraries for client side Web applications
- React version history
- What's New in React 18
- Understanding "what" and "why" React
- Environment Setup for React Application
- Understanding NPM commands
- Using VS Code
- VS Code extensions for ES6, React

REACT ESSENTIAL FEATURES AND SYNTAX

- React App Project Directory Structure
- Overview of Webpack, Babel
- React Component Basic
- Create React Component
- Understanding JSX
- Limitations of JSX
- Working with Components and Reusing Components
- Helloworld app in React

REACT COMPONENTS , PROPS AND STATE

- Functional/Class/Pure Components
- Understanding and using Props and State
- Handling Events with methods
- Manipulating the State
- Two way data-binding
- Functional (Stateless) VS Class (Stateful) Components
- Between component child communication
- Dynamically rendering contents
- Showing Lists, List and keys
- Refs

STYLING COMPONENTS

- CSS Styling
- Scoping Styles using Inline Styles
- Limitations of inline styles
- Inline Styles with Radium
- Using Psuedo classes/media queries with inline styles

- CSS Modules, importing css classes
- Adding Bootstrap, Semantic UI to React apps
- Using react-bootstrap, reactstrap packages

DEBUGGING REACT APPS

- Understanding React Error Messages
- Handling Logical Errors,
- Debugging React apps using google developer tools and React DevTool
- Understanding Error Boundaries

REACT COMPONENT LIFE CYCLE

- Updating life cycle hooks
- PureComponents
- React's DOM Updating Strategy
- Returning adjacent elements
- Fragments

REACT HOOKS

- Introducing Hooks
- Hooks at a Glance
- Using the State Hook
- Using the Effect Hook
- Rules of Hooks
- Building Your Own Hooks
- Hooks API Reference
- Hooks FAQ

REACT COMPONENT IND ETAILS

- Higher Order Components
- Passing unknown Props
- Validating Props
- Using References
- React Context API
- Best practices for React Projects
- Demo apps

HTTP REQUESTS/AJAX C ALLS

- HTTP Requests in React
- Introduction of Axios package

- HTTP GET Request, fetching & transforming data
- HTTP POST, DELETE, UPDATE
- Handling Errors
- Creating/Using Axios instances

REACT ROUTING V5 and V6

- Routing and SPAs
- Setting Up the Router Package
- react-router vs react-router-dom
- Preparing the Project For Routing
- Switching Between Pages, Routing-Related Props
- The "withRouter" HOC & Route Props
- Passing & extracting route/query parameters
- Using Switch to Load a Single Route
- Navigating Programmatically

REACT FORMS AND FORM VALIDATION

- Creating a Custom Dynamic Input Component
- Setting Up a JS Config for the Form
- Dynamically Create Inputs based on JS Config
- Adding a Dropdown Component
- Handling User Input
- Handling Form Submission
- Adding Custom Form Validation
- Fixing a Common Validation
- Adding Validation Feedback
- Showing Error Messages
- Handling Overall Form Validity

DEPLOYING REACT APP TO THE WEB REACT REDUX

- Redux principles
- Install and setup redux

- Creating actions, reducer and store
- What is React Redux
- Why React Redux
- Install and setup
- Presentational vs Container components
- Understand high order component
- Understanding mapStateToProps and mapDispatchToProps usage

REACT MATERIAL UI ERROR HANDLING

UNIT TESTING IN REACT

- Understand the significance of unit testing
- Understand unit testing jargon and tools
- Unit testing react components with Jest
- Unit testing react components with enzyme

WEBPACK PRIMER

- What is webpack
- Why webpack
- Install and setup webpack
- Working with webpack configuration file
- Working with loaders
- Quick word on code splitting, lazy loading, tree shaking
- Setting up Hot Module Replacement

SERVER -SIDE RENDERING WITH REACT

- What is server-side rendering (SSR)?
- Why SSR
- Working with renderToString and renderToStaticMarkup methods

DEPLOYING ON CLOUD

- Firebase Deployment