NARESH i technologies
An ISO 9001:2015 Certified Company

UI FULL STACK

Web Development with

REACT JS

HTML_CSS_Bootstrap

HTML Basics

- Introduction to HTML, Document Structure, Headings (h1 to h6)
- Paragraphs,Tags vs Elements, Text Formatting, Lists (ul, ol, li)
- comments, Links (anchor tags), Images (img tags), Block vs Inline
- Semantic Elements, Tables,symbols, miscellenous
- Forms(Ui Elements and their attributes)
- forms Validation
- media inserting tags(video,audio,picture, iframe,embed etc..,)

More HTML and CSS

- CSS Basics - Selectors, Colors, Fonts
- Box Model (margin, padding, border), Display Property
- pseudo selectors,pseudo classes
- Positioning,floats,clearfix
- FlexBox
- Grid
- Translate and Transitions and units(vw,vh,em,rem,%,px)
- animations and Responsive Basics
- media Queries
- miscellaneous(Filters, Blendering mode, designs, custom variables)
- Project 1(using HTML & CSS)
- Project 2(Using HTML &CSS)
- SASS Introduction(vaiables,Nesting,mixin,partials)
- SASS (control directives,extend,sample usage)
- Project 3(using HTML& CSs)

Bootstrap

- Introduction to Bootstrap, Libraries, Grid System
- Bootstrap Typography, Colors, Images
- Bootstrap Components (Navbar, Cards, Forms)

JavaScript

JavaScript

- Introduction to JavaScript, Variables
- Data types and its Methods-1
- Data types and its Methods-2
- Operators in JS
- Control Statements (if and switch)
- loops(while,do-while,for)
- variations of for(for,for..in,for..of,forEach)
- forEach,map,filter,reduce
- Function(declaration,invokation,parameters and arguments)
- Scope,closure,callbacks
- DOM Intro, Selection, modifying content
- Handling events(event handlers,listeners)
- Events(event propagation and delegation)
- creating and deleting elements
- Forms in JS
- Asynchronous JS,Callback Hell
- Promises
- HTTP Requests and Fetch API
- Async/Await
- Practice projects for DOM and API
- ES6 Features(Arrow Functions, Template Literals)
- Destructuring
- Classes -1
- Classes-2(include this,bind,apply,call)
- Modules-1
- Modules-2 and Prototypes and its chaining
- BOM(introduction and basics)
- Local Storage and Session Storage
- cookies
- HTML5(Canvas,svg,GeoLocation)
- AJAX(Asynchronous requests,XHR,JSON,CORS)
- JQUERY-1
- JQUERY-2
- PROJECT(Complete)
- PROJECT(Complete)
- PROJECT(Complete)

React JS

Getting Started with React

- 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 and VS Code extensions for ES6, React
- React App Project Directory Structure

Components and JSX

- React Component Basics
- Creating React Components
- Understanding JSX
- Limitations of JSX
- Working with Components and Reusing Components

State, Props, and Event Handling

- Helloworld app in React
- Functional/Class/Pure Components
- Understanding and using Props and State
- Handling Events with methods
- Manipulating the State
- Two-way data-binding

Advanced React Components

- Functional (Stateless) VS Class (Stateful) Components
- Between component child communication
- Dynamically rendering contents
- Showing Lists, List and keys
- Refs

Styling in React

- CSS Styling in React
- Scoping Styles using Inline Styles
- Limitations of inline styles
- Inline Styles with Radium
- Using Pseudo classes/media queries with inline styles
- CSS Modules, importing CSS classes
- Adding Bootstrap, Semantic UI to React apps
- Using react-bootstrap, reactstrap packages

Debugging and Error Handling

- Understanding React Error Messages
- Handling Logical Errors
- Debugging React apps using Google Developer Tools and React DevTool

- Understanding Error Boundaries
- Updating life cycle hooks
- PureComponent
- React’s DOM Updating Strategy
- Returning adjacent elements, Fragments

Introduction to Hooks

- Introducing Hooks
- Hooks at a Glance
- Using the State Hook
- Using the Effect Hook
- Other Hooks like useCallBack, useRef, useLayoutEffect, useMemo etc.
- Rules of Hooks
- Building Your Own Hooks
- Hooks API Reference
- Hooks FAQ

Higher Order Components and Context API

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

HTTP Requests and Forms in React

- Demo apps
- HTTP Requests in React
- Introduction of Axios package
- HTTP GET Request, fetching & transforming data
- HTTP POST, DELETE, UPDATE
- Handling Errors
- Creating/Using Axios instances
- Routing and SPAs

React Router and Forms

- 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
- Navigating Programmatically

Form Handling and 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 Message
- Handling Overall Form Validity

State Management with 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

Material UI and Common Errors

- Material UI Overview
- Installation and Setup
- Basic Usage and Components
- Theming and Styling
- Handling Forms and Inputs
- Using Icons and SVGs
- Common Errors in React
- Debugging Techniques
- Best Practices for Error Handling

Advanced React Features

- Automatic Batching
- Transitions
- Suspense Features
- New Strict Mode Behaviors

Unit Testing and Webpack

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

- What is server-side rendering (SSR)?
- Why SSR
- Working with renderToString and renderToStaticMarkup methods
- Firebase Deployment
- Deploy full Stack Application in Netlify, Vercel, AWS, or Azure

Hands-On

- Project: Design and Implement E-commerce Website
- Design and implement an E-commerce website with a suitable UI using React
- Implement state management using Redux
- Handle situations like Prop Drilling By Context API

Back_End Node

Introduction to Server-Side Scripting and NPM

- Introduction to Server-Side Scripting
- Overview of NPM (Node Package Manager)

Node.js Features, Drawbacks, and Environment Setup

- Features and Drawbacks of Node.js
- Setting up the Environment for Node.js

Node.js Program Architecture and Nodemailer

- Node.js Program Architecture
- Introduction to Nodemailer for email functionality

Node.js Core Concepts

- MVC Design Pattern
- Node.js Web Server
- Node.js Global and OS Objects
- Error Handling in Node.js
- Understanding the Event Loop

Node.js File System and Asynchronous Operations

- File System in Node.js
- Asynchronous and Synchronous Operations
- Connecting with Databases
- Handling CRUD Operations

Express.js

Express.js

- Express.js Basics
- Introduction to Express.js
- Routing in Express
- Response Methods in Express
- Serving Static Files
- Express JSON Parser
- Middleware and API Creation
- Middleware and Middleware Chaining
- Express CORS
- Postman for API Testing
- Creating Web API
- Exploring the Express Module

Advanced Express.js

- Express.js Sessions and Data Handling
- Session Management in Express
- Setting up Session Middleware
- Sending and Receiving Data (POST & GET)
- Reading Data from Requests
- Templates, File System, and REST API
- Creating Templates using PUG/JADE
- Generating Dynamic HTML Pages
- Exploring File System (Reading and Writing)
- Introduction to REST API and REST Architecture

Hands-On

- Project: Design and Implement Back-End and DB
- Applying knowledge to design and implement a back-end for an E-commerce site

Authentication and Security

- Introduction to Authentication
- Session-based Authentication
- Token-based Authentication
- Basic Security Concepts

Security Practices and Error Handling

- Handling SQL Injection and Cross-Site Scripting
- Understanding CORS and Solving CORS Issues
- Best Practices for Error Handling
- Debugging Node.js Applications

Testing and Deployment

- Tools for Testing REST APIs
- Writing Test Cases
- Deploying Node.js App on Heroku

WebSockets, GraphQL, and Middleware

- WebSockets and Implementation
- Introduction to GraphQL
- Setting up GraphQL with Express
- Creating Custom Middleware

Caching

- Introduction to Caching
- Implementing Caching in Node.js

Streams

- Working with Streams in Node.js

Microservices

- Introduction to Microservices
- Pros and Cons of Microservices

Docker

- Introduction to Docker
- Creating Docker Images and Docker Compose

Kubernetes

- Overview of Kubernetes and Setting Up a Cluster
- Deploying Apps on Kubernetes

Hands-On

- Full-Stack Application and Deployment
- Building a Full-Stack Application
- Deploying the Full Stack Application on NETLIFY, Vercel, AWS, or Azure

MongoDB

MongoDB and Database Connectivity

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

MySQL

Introduction

- DBMS vs RDBMS
- Intro to SQL
- SQL vs NoSQL
- MySQL Installation
- Data Types in SQL

Keys

- Primary Key
- Foreign Key

Constraints

- Unique
- Not NULL
- Check
- Default
- Auto Increment

CRUD Operations

- Create
- Retrieve
- Update
- Delete

SQL Languages

- Data Definition Language (DDL)
- Data Query Language
- Data Manipulation Language (DML)
- Data Control Language
- Transaction Control Language

SQL Commands

- Create
- Insert
- Alter, Modify, Rename, Update
- Delete, Truncate, Drop
- Grant, Revoke
- Commit, Rollback

- Select

SQL Clause

- Where
- Distinct
- OrderBy
- GroupBy
- Having
- Limit

Operators

- Comparison Operators
- Logical Operators
- Membership Operators
- Identity Operators

Wild Cards

Aggregate Functions

SQL Joins

- Inner Join & Outer Join
- Left Join & Right Join
- Self & Cross Join
- Natural Join