UI FULL STACK COURSE CONTENT

TOPICS	SUB TOPICS
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
	● HTML3,4,5
	● Crawling and Meta tags
GIT Basics	● Introduction to version control
	● Understanding Git basics (init, add, commit, push, pull)
	● Branching and merging
	● Resolving merge conflicts
	● Git workflows (e.g., feature branches)
	● Collaboration on GitHub
	● Apply version control (Git/GitHub) to manage project history
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
	● **Project :** Building a Personal Blog Page
	Apply HTML concepts to create a basic blog page
	Implement proper document structure and semantics
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
	● Nested Tables
	● Fieldset, Option Groups
	● Hiding and un-hiding elements using display property
	● img tag, p tag, ul&ol tags, li, nobr, hr, bretc
	● 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,Selectors
	● 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.)
Form Elements	● Get & Post Communication
	● Validating input values in a form
	● Form action and method attributes
JavaScript	● Data types and data structures in Js
	● Control structures (Conditional, Looping, Case)
	● Variable Hoisting & 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
	● Solving some logical problems through Control structure
	● DOM Traversal, DOM Events using JS?
	● Functions and types (Inner, Recursive, Anonymous, Pure, impure, etc.)
Adv. JavaScript	● JavaScript 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, throw, and throws
	● Implementing AJAX through fetch method
	● OOPS concepts supported in JS
	● Why Polymorphism is not supported in JS
Sass (optional)	● Sass Intro
	● Sass Installation
	● Sass Variables
	● Sass Nesting
	● Sass @import
	● Sass @mixin
	● Sass @extend
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
	● Promises with multiple conditions
	● Handling Multiple AJAX Calls using Promises

UI FULL STACK COURSE CONTENT

Templating	● Loading DOM dynamically using templates
	● Loading templates using AJAX
	● Exploring “Handlebars” and “Load template” External API
Axios	● Handling AJAX through Axios library
HTML 5	● Difference between HTML5 and HTML 4
	● List of Browsers support HTML5
	● Doc type and 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 Objects
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
	● Implementing Gradients (linear, Conic, radial)
	● Keyframes
	● The @ import rule
	● Multi-Column Layouts
	● How Var() works in CSS3
Responsive Design	● Difference between multiple devices, making a page to work on multiple devices
	● Media queries
	● Introduction to Bootstrap CSS API
Bootstrap	● Basic-Typography
	● Text-Alignments, Floats-Positions

	● Colors-Backgrounds, Margins-Paddings
	● Sizing’s, Breakpoints, Buttons, Navbars
	● List-Groups, Forms, Input-Groups
	● Alerts-Progress
	● Tables, Cards, Carousel, Collapse, Modal
	● Grid-System, Grid-Alignments, FlexBoxes
ECMA Standards (Latest JavaScript)	● Const and let keywords
	● Arrow functions, advantages
	● Function parameters with default values
	● Extended parameter handling, default parameters
	● Template literals
	● Enhanced object parameters
	● Creating direct classes, inheritance
	● Implementing promises
	● New data structures Set and Map
	● New string built-in methods
	● Using iterators
	● Spread and Rest operators
	● New Array methods (map, find, filter, includes, etc.)
	● Async and await keywords
	Project: Design and implement a basic E-commerce Website with necessary functionalities using HTML, CSS and JS
NODE JS	● Introduction to Server-Side Scripting and NPM
	● Node JS Features and Drawbacks
	● Setup Environment for Node.js
	● Node JS Program architecture
	● Nodemailer
	● MVC Design Pattern
	● 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 JS	● Introduction
	● Routing in Express
	● Response Methods in Express
	● Serving Static Files
	● Express JSON Parser
	● Middleware, middleware chaining?
	● Express CORS
	● Postman
	● Creating Web API
	● Exploring Express module
	● Setting up express environment
	● Creating http server using express
Express Sessions	● Session management in Express
	● Setting up session middleware
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 the 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
Clustering	● Creating and handling multiple clusters in Node js
DB Connection	● Connecting to Mongo DB&Connecting to SQL
Webservices with DB interactions	● Creating web services which communicate with Database
MONGO DB	● Introduction to MongoDB

	● Configuring Server and Client
	● MongoDB Compass
	● Creating Database
	● MongoDB Commands
	● MongoDB CRUD Operations
	Project: Design and Implement Back-End and DB for Previous E commerce site.
Authentication:	● Introduction to Authentication
	● Session-based Authentication
	● Token-based Authentication
Security:	● Basic Security Concepts
	● Handling SQL Injection and Cross-Site Scripting
Handling CORS:	● Understanding CORS
	● Solving CORS issues
Error Handling and Debugging:	● Best practices for Error Handling
	● Debugging Node.js Applications
REST API Testing:	● Tools for Testing REST APIs
	● Writing Test Cases
Deploying Node.js Application:	Deploying Node.js App on Heroku
ADVANCED NODE.js:	● WebSockets
	● GraphQL
ADVANCED EXPRESS:	● Middleware Functions
	● Creating Custom Middleware
GraphQL with Express:	● Introduction to GraphQL
	● Setting up GraphQL with Express
	● Creating GraphQL Schemas
WEB SOCKETS:	● Understanding WebSockets
	● Implementing WebSocket Communication
CACHING STRATEGIES:	● Introduction to Caching
	● Implementing Caching in Node.js
STREAMING DATA:	● Working with Streams in Node.js
	● Streaming Data to and from the Client

MICROSERVICES ARCHITECTURE:	● Introduction to Microservices
	● Pros and Cons of Microservices
DOCKER:	● Introduction to Docker
	● Creating Docker Images
	● Docker Compose
KUBERNETES:	● Overview of Kubernetes
	● Setting Up Kubernetes Cluster
	● Deploying Apps on Kubernetes
	PROJECT : Building a Full-Stack Application
DEPLOYEMENT	Deploy full Stack Application in NETLIFY, vercel or AWS or Azure

FOR REACT JS:

REACT INTRODUCTION	● 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 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 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
	● Other Hooks like useCallBack,useRef,useLayoutEffect,useMemo etc..,
	● 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
	● Handing 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
	● 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 Message
	● Handling Overall Form Validity
	● Redux principles
	● Install and setup redux
	● Creating actions, reducer and store

DEPLOYING REACT APP TO THE WEB REACT REDUX	● 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	● Material UI Overview
	● Installation and Setup
	● Basic Usage and Components
	● Theming and Styling
	● Handling Forms and Inputs
	● Using Icons and SVGs
ERROR HANDLING:	● Common Errors in React
	● Debugging Techniques
	● Best Practices for Error Handling
NEW FEATURES OF REACT 18	● Automatic Batching
	● Transitions
	● Suspense Features
	● New Strict Mode Behaviors
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
	● Deploy full Stack Application in Netlify, Vercel, AWS, or Azure
	Project: Design and Implement E commerce website with suitable UI using React. Implement state management using Redux and handle situations like Prop Drilling By Context API.