Web Development Full-Stack

(Frontend, Server and Backend) - Syllabus

Sl. no	Topic	Hrs
1	**Module 1: Web Development**	
2	**Module 2: Html, Css, Bootstrap**	
3	**Module 3: Javascript, Typescript, Ecmascript**	
4	**Module 4: Angular**	
5	**Module 5: ReactJs**	
6	**Module 6: NodeJs, ExpressJs**	
7	**Module 7: MongoDb**	
8	**Module 8: Project Implementation with Angular as Client**	
9	**Module 9: Project Implementation with ReactJs as Client**	
1	**Module 1: Web Development**	
	What is Web Development, What is Mean Stack	
	Github	
2	**Module 2: Html, Css and Bootstrap**	
	Html	
	Introduction to Html	
	Html Elements, Attributes	
	Page Creation with Formatting Elements	
	Creating Links and Navigating, Redirecting Pages	
	Displaying the Images, Adjustment of Images along with Content	

	Organizing data with Tables	
	Displaying data in Listings	
	Creating Forms and Layouts with Fields and Input	
	Implementing the Semantic Elements (latest Html elements)	
	Working with Audio and Video (Media Objects)	
	Css	
	Introduction to Css	
	Implementing the Three Different ways of Styles: Inline, Embedded and External to Html Content	
	Applying the Web Colors to Html Content in Three Different ways: Direct Name, Numerical, and HexaDecimal	
	Applying different properties like: padding, margin, float, font, border, background, etc.,	
	Applying the Display and Position properties, etc.,	
	Creating the Web Page with Content	
	Creating the Grid Layout with Sample Content with Responsive	
	Working with Media Queries	
	Bootstrap	
	Introduction to Bootstrap, Usage, Advantages	
	Discussing about different types of Resolutions	
	Implementing the Bootstrap Required Links into Web Pages	
	Create the Fixed Width, Fluid Width Web Pages	
	Implementing the Components of Bootstrap:	
	Tables, Lists, Forms, Images,	
	Modals, Tabs, Cards, Navigations	

3	**Module 3: Javascript and Typescript**	
	Javascript	
	Introduction about Javascript, DOM etc?	
	Writing the Programs and Displaying the Data	
	Understanding about Keywords, Identifiers	
	Declarations: var, let, const	
	Implementing Statements: Decision Making Statements and Loop Statements	
	Working with Arrays and Implementing with Examples	
	Understanding the Concept of Functions: Call and Define, Functions with Parameters, Return Statement, Arrow Functions, Call Back Functions	
	Implementing the Object Oriented Programming in Javascript: What is Object, Class, Constructor, Method, Property etc., and its terminology	
	Creating the concept with Class and Object includes the Methods, Properties, and Accessing the Data	
	Proptotype, Closures concepts implementation	
	Promise, and Async and Await concepts implementation	
	Typescript	
	Typescript Introduction, Advantages of Typescript	
	Declaring Variables or Identifiers using let & const and accessing them	
	Working with Decision Making Statements	
	Working with Loop Statements	

	Working with Functions	
	Understanding and Implementing Tuple Operators	
	Creating Interfaces, accessing with Objects	
	Creating Classes, Interfaces and Declaring Objects	
	Implementing Inheritance	
	Creating Custom Modules	
	Ecmascript	
	Ecmascript Introduction	
	Declarations of Identifiers with let & const	
	Working with Loops	
	Functions, Passing Parameters, Return Statement, Arrow Functions, Call Back Functions	
	Spread Operators	
	Map Object with Set and Get	
	Promise, Async and Await with Promise	
4	**Angular**	
	Introduction to Angular, Features, Advantages, What is Single Page Application?	
	Setting up the Env., and Creating the Angular First Project	
	Understanding the Angular Project & Its Flow, Adjusting the Project Folder Structure in Precisely	
	What is Component? Creating the Components through Command	
	Adding additional Dependencies through NPM, Managing the Packages in Angular Project	

	What is Routing? and Implementing the Routing in Angular Project	
	Working with Property Binding, Event Binding	
	Property Binding with One Way & Two Way	
	Implementing the Two Way Property Binding with Forms	
	Creating the Forms with Template Driven and Implementing Form Validations with Regular Expressions using Pattern	
	Creating the Forms with Reactive Forms and Implementing Form Validation with Regular Expressions using Pattern	
	Creating the Form with File Upload with Validation	
	Creating the Form with Adding Dynamic Elements on Fly Based on the User Actions	
	Understanding the concept of Services, Creating the Services, and accessing the Services into Components	
	Implementing the concept of Observables	
	Implementing the concept of Dependency Injection	
	Understanding the concepts of @Input() and @Output() Decorators and Implementation	
	Understanding and Implementing the concept of Directives	
	Working with Built-in Directives, Creating the Custom Directives	
	Understanding and Implementing the concept of Pipes	
	Working with Built-in Pipes and Creating the Custom Pipes	
	Implementing the CRUD Operations	
	Implementing the concept of Lazy Loading	
	Implementing the concept of Life Cycle Hooks	
	Creating the Custom Modules and Implementing into the	

	Components	
	Creating the Single Page Application	
5	**ReactJS**	
	ReactJS Introduction	
	Scope of ReactJS	
	What is ReactJS?	
	Understanding the Terminology of ReactJS	
	What is Single Page Application?	
	Understanding the ReactJS Architecture	
	Setting up the ReactJS Environment	
	Creating the First Project in ReactJS	
	Understanding the Project Folder & Files Structure	
	Working with assets - Html, Css, Images, Javascripts in ReactJS	
	Understanding the Term: Component.	
	What is Component?	
	Creating the new Component(s)	
	Accessing the Created Component(s)	
	Working with Re-Usable Component(s)	
	Understanding the Terms: ‘props’ and ‘state’	
	Working with ‘props’	
	Implementing the ‘props’ into Component(s)	
	Creating the ‘state’ object, setting the ‘properties’ into it	
	Accessing the ‘properties’ from ‘state’ object into Components	
	Working with Event Handlers ex., onClick, onHover, onBlur, onFocus, onChange etc.,	
	Creating the methods into Component(s)	

	Writing the definition(s) of the methods	
	Passing ‘parameters’ into methods	
	Understanding the concept of Property Binding	
	Two types of Property Binding., 1. One Way Property Binding and 2. Two Way Property Binding	
	Asking the User to input the value, accessing those value(s)	
	Creating the group of Component(s) and accessing those through ‘state’ object	
	Creating the CRUD operations	
	Understanding the concept of Routing	
	Creating the group of Component(s) and aligning them into Routing	
	Fetching the Data from RESTAPI into Component(s) and accessing the Data	
	Understanding and Working with Material UI (ReactJS UI Framework)	
	Creating the Button, Radio, Form, Table etc.,	
	Working with Form, Two – Way Property Binding	
	Creating the Validation with Error Hints,	
	Applying the Regular Expressions to Input Fields	
	Working with the concept of Observables	
	Working with the concept of Services	
	Implementing the concept of Observables & Subscribe with Services	
	Implementing the concept of Observables & Subscribe with Services using RESTAPI	
	Understanding the Terms: babel, webpack	

	Understanding the concept of Redux	
	Understanding the Terms: Component, Action, Reducer, Data Store	
	Understanding and Implementing the concept of Redux	
	Understanding the Term: Sage	
	Implementing the functionality of Redux with Saga	
	Understanding the Terms: Life Cycle Hooks	
	What is the usage of Life Cycle Hooks	
	How many types of Life Cycle Hooks: 1. Class Based Life Cycle Hooks and 2. Functional Based Life Cycle Hooks	
	Understanding the Class Based Life Cycle Hooks and its methods: componentDidMount(), componentDidUpdate(), componentWillUnmount() etc.,	
	Understanding the Functional Based Life Cycly Hooks and its methods: useState(), useEffect(), useContext(), etc.,	
	Understanding the concept of Styled Components	
	Working with ReactJS Developer Tools	
	Implementing the concept of ContextAPI	
	Understand and Implementing the Unit Tests in ReactJS	
	Creating the Single Page Application in ReactJS	
6	**NodeJs and ExpressJS**	
	NodeJs	
	Introduction to NodeJs	
	What is NodeJs? Advantages of NodeJs	
	Traditional Web Server Model	
	NodeJs Process Model	

	Setup Dev., Environment	
	Install NodeJs on Windows, Mac and Linux	
	Working with REPL	
	NodeJs Console	
	NodeJs Modules	
	Functions	
	Buffer	
	Modules	
	Module Types	
	Core Modules	
	Local Modules	
	Module.exports	
	Node Package Manager	
	What is NPM?	
	Installing Packages Locally	
	Adding Dependency in package.json	
	Installing packages globally	
	Updating packages	
	Creating Web Server	
	Creating Web Server	
	Handling Http Requests	
	Sending Requests	

	File System	
	Reading a File Asynchronously / Synchronously	
	Writing a File Asynchronously / Synchronously	
	Opening a File	
	Deleting a File	
	IO Operations	
	Debugging NodeJs Application	
	Core NodeJs Debugger	
	Debugging with Visual Studio Code	
	Events	
	EventEmitter Class	
	Returning EventEmitter	
	Inhering Events	
	ExpressJs	
	Introduction to ExpressJs	
	Env., Setup of ExpressJs	
	Creating a Sample Program in ExpressJs	
	Creating a Static Routing & Dynamic Roting	
	Working with HTTP Methods	
	Creating the URL Building	
	Using Middleware Functions	
	Creating a Templating	
	Working with Static Files	
	Parsing Form Data	

	Parsing & Fetching & Connecting with Database	
	Cookies & Sessions	
	Authentication	
	RestFul Api’s	
	Scaffolding	
	Error Handling	
	Debugging	
7	**MongoDb**	
	Introduction to MongoDb	
	Differences between Sql vs NoSql	
	Features & Advantages of MongoDb	
	Installing MongoDb along with Compass	
	Understanding the Terminology: Database, Collection, Document, etc.,	
	Commands in MongoDb	
	Creating Database, Drop Database	
	Creating Collection, Drop Collection	
	Insert, Update, Delete the Documents	
	Writing Conditions while Retrieving, Updating the Documents	
	Data Aggregation	
	$match, $project, $group, $unwind, $sort, $limit, $set, $lookup	
	$count, $distinct,	
	Working with Indexes, Types of Indexes	
8	**Creating the Project with Angular as Client, NodeJs & ExpressJs as Server and MongoDb as Database**	

9	**Creating the Project with ReactJs as Client, NodeJs & ExpressJs as Server and Mongodb as Database**	