HTML

HTML Introduction:

- Introduction to HTML
- Tags
- Paragraph
- Heading Tags
- Ancor Tag
- Image Tag
- List tag
- Table Tag

HTML Intermediate:

- Block vs Inline element
- Semantic element
- Forms
- Tables
- Button
- Form validation

CSS

Basic CSS

- Inline CSS
- Internal CSS
- External CSS
- How to Debug CSS Code
- The Anatomy of CSS Syntax
- CSS Selectors
- Classes vs. Ids
- Text and Font
- Images In CSS
- Padding , margin
- px vs em vs rem

Intermediate CSS

- Flexbox
- Responsive UI
- Box Model

Javascript

- Basics: Variables, Data Types, Operators & Functions
- Efficient Development & Debugging
- Working with Control Structures (if Statements, Loops, Error Handling)
- Past (ES3, ES5) & Present (ES6+) of JavaScript
- Functions Advance
- Working with the DOM (Browser HTML Code) in JavaScript
- Arrays & Iterables
- Objects
- Class and OOPS
- Events
- Number and String
- Promises and call back
- Http request
- Js Tooling and workflow
- Browser storage
- Typescript intro

Bootstrap

- Intro
- Typography
- CSS Component
- Grid System
- Forms
- Tables
- Javascript Widget
- Project :1

Typescript

Introduction to TypeScript

- Overview of TypeScript
- Why TypeScript?

Installation of tools

- Installing VS – an IDE
- Installing TypeScript using npm

Datatypes in TypeScript

- User-defined Datatypes
- Primitive Data types
- Tuples and Unions
- Variables and Operators

Decision Makers and Loops

- if, if-else, switch statements
- for, while loops
- forEach in TypeScript

Functions in TypeScript

- Defining and Calling a Function
- Returning functions
- Parameterized functions
- Arrow functions

Inbuilt methods in TypeScript

- String methods
- Array methods
- Other inbuilt methods

Oops in TypeScript

- Introduction to Classes and Objects
- Interfaces
- Inheritance

Winding up with TypeScript

- Revising the TypeScript Concepts
- Bridging TypeScript to Angular

ANGULAR

Installation

- Angular Inroduction
- Installing the latest version of nodeJS
- Installing Angular 9 using a node package manager (npm)
- Installing an IDE like Visual studio code
- List of commands for creating, running and building a project

Angular Fundamentals

- Create a new Angular project
- Building blocks of angular
- The folder structure of an Angular Project
- Linking bootstrap with Angular
- Linking icons, fonts with the Projects

Angular Components and Modules

- Creating new components
- Structure of a component
- Calling a component
- Nesting components
- Component lifecycle
- Introduction to Modules

Displaying Data and Event Handling

- Interpolation
- Property Binding
- Attribute Binding
- Style Binding
- Event Binding
- Two way binding

Pipes

- InBuild pipes
- Custom Pipes

Component Communication

- Using @Input @output
- View child
- Reference variable
- Services

Directive

- ngIF

- ngFor
- ngClass
- ngStyle
- Custom Directive

Forms

- Template-driven forms
- Reactive forms
- Adding validations to forms
- Displaying error messages

Navigation using Routers

- Introduction to Angular routings
- Nested routes
- Relative path vs Absolute paths
- Lazy loading modules
- Pre-loading modules
- Preventing unauthorized access using protected routes

Angular Observables and RxJS

- Introduction to Observables
- Using observables to pass data
- Type-ahead suggestions using Observables
- Introduction to RxJS library

Server Interactions using HTTP

- Creating a Service
- Introduction to Http Client Services
- HTTP get vs post
- Requesting and sending data to the server
- Sending get Request
- Sending put Request
- Adjusting Request Headers

Deployment of the Application

- Building the Application
- Steps to host the application on some server
- Hosting the Application

Building a Real-time Application

- Creating the Project Setup
- Creating modules and grouping Components
- Implementing Lazy Loading and adding Routings
- Adding necessary Material Components
- Completing the contents
- Hosting the Website

React Js

Introduction

Required Javascript

1. Modules
2. var/let/const
3. Arrow function
4. Export and Import
5. Understanding Classes
6. Spread operator
7. Array

React Fundamentals

1. Modules
2. Components
3. Project Creation
4. JSX
5. React Works
6. Custom Components
7. Adding Basic CSS Styling
8. Outputting Dynamic Data
9. Passing Data via "props"
10. Splitting Components
11. Splitting Components
12. Organizing Files

Events

- Key Events
- Listening to Events & Working with Event Handlers
- Working with "State"
- "useState" Hook
- Form Inputs
- Listing Form Inputs
- Working with Multiple States
- Using One State Instead
- Handling Form Submission
- Adding Two-Way Binding
- Child-to-Parent Component Communication (Bottom-up)
- Lifting The State Up

List Rendering

- Rendering List Data in UI
- Using Stateful Lists
- Using "Keys"
- Conditional Content
- Adding Conditional Return Statements
- Adding Dynamic Styles

Styling React Components

- Dynamic Inline Styles
- CSS Classes Dynamically
- Introducing Styled Components
- Styled Components & Dynamic Props
- Styled Components & Media Queries
- Using CSS Modules
- Dynamic Styles with CSS Modules

Debugging

- Understanding React Error Messages
- Analyzing Code Flow & Warnings
- Working with Breakpoints

Fragment, Portal & Ref

- Creating a Wrapper Component
- React Fragments
- Introducing React Portals
- Working with Portals
- Working with "ref"s
- Controlled vs Uncontrolled Components

Handing Side Effects, Using Reducer & Using the context api

- What are "Side Effects" & Introducing useEffect
- Using the useEffect() Hook
- useEffect & Dependencies
- Using the useEffect Cleanup Function
- Using the useReducer() Hook
- useReducer & useEffect
- Adding Nested Properties As Dependencies To useEffect
- useReducer vs useState for State Management
- Using the React Context API
- Making Context Dynamic
- Building & Using a Custom Context Provider Component
- Learning the "Rules of Hooks"

Optimization Technique

Class-based Component

- Class-based Component
- Working with State & Events
- The Component Lifecycle (Class-based Components Only!)
- Lifecycle Methods In Action
- Class-based Components & Context

HTTP Calls

- Connecting Backend
- Sending a GET Request
- Using async / await
- Handling Loading & Data States
- Handling Http Errors
- Using useEffect() For Requests
- Sending a POST Request
- Sending a PUT Request

Building Custom react hook

Working with forms and user input

- Form Submission
- Getting User Input Values
- Adding Basic Validation
- Providing Validation Feedback
- Managing The Overall Form Validity
- Applying Hook

Redux
Advanced Redux
React Router

- Installing React Router
- Defining & Using Routes
- Working with Links
- Using NavLinks
- Adding Dynamic Routes with Params
- Extracting Route Params
- Using "Switch" and "exact" For Configuring Routes
- Nested Routes
- Adding a "Not Found" Page
- Working with Query Parameters
- Getting Creative With Nested Routes
- Writing More Flexible Routing Code

Tools

- GitHub
- Jira
- Basic photoshop
- npm