

HTML and CSS

- HTML
 - DOCTYPE html
 - DOM
 - Semantic HTML
 - Difference between HTML and HTML5
 - When to use section, div or article?
 - section
 - article
 - div
 - Headings
 - Paragraph
 - Links / Anchor elements
 - Difference between <div> and <frame>
 - Images
 - Image Sizes
 - Image as links
- Styling Webpage
 - Types of styling
 - Internal
 - External
 - Inline
 - Selectors
 - .class
 - child .class
 - #id
 - element tag
 - Mix n match
 - Id and Class
 - Element and Class
 - Advanced Selectors
 - adjacent selector
 - attributes selector
 - Backgrounds
 - Colors
 - Borders
 - Fun with Border Radius
 - Shapes
 - Shorthand
 - Circle and leaf
 - Circle
 - Leaf
 - Display and position elements
 - Box model
 - Total width of an element
 - Total height of an element

- Margins
 - Margin On Individual Sides
 - Margin Shorthands
 - Auto Margin
- Paddings
 - Padding On Individual Sides
 - Padding Shorthands
- Display
 - Block
 - Inline
 - Inline-block
 - None
 - Visibility Hidden
 - Flex
- Positions
 - Static
 - Relative
 - Absolute
 - Fixed
- Centering
 - Centering Vertically
- CSS Float
 - Clearing Floats
 - Methods to clear float
- Flexbox and media query
 - Flexbox
 - Flex box container properties
 - Flex box item properties
 - Flexbox Examples
 - Media queries
 - Always Design for Mobile First
 - Orientation: Portrait / Landscape
 - Sizes - px vs em vs rem
 - How to calculate PX from REM
 - More on rem vs em
 - CSS Grids
 - Flexbox
 - Grids
 - Nested grids
 - Start-End points of Grid
 - FOUC
 - BEM naming convention
 - OOCSS - Object Oriented CSS
 - CSS User Agent Styles
 - Normalizing CSS
 - Reset CSS
 - Validate your CSS

- Testing Strategies
- Conditional CSS

JavaScript

- JavaScript Basics
 - Run just JavaScript
 - Variables
 - Data Types in JavaScript
 - Basic Operators
 - Special Operators
 - JavaScript as Object-Oriented Programming language
- Conditionals and Collections
 - Conditionals
 - If Else Condition
 - Ternary Operator
 - Advanced Ternary
 - Switch Statements
 - truthy and falsy values in JavaScript
 - For Loop
 - For-In loop
 - For-Of loop
 - While loop
 - Do-While loop
 - Map Reduce Filter
 - Map
 - Reduce
 - Filter
- JavaScript Objects and Functions
 - JavaScript Object Basics
 - Access Object Value
 - JavaScript Functions
 - Invoke Function
 - Local variables
 - Function Expressions
 - Scoping in JavaScript
 - Two Types
 - Constructor Functions
 - The this keyword
 - this with example
 - The new Operator
 - Creating an object with and without new operator
 - WITHOUT new operator
 - WITH new operator
 - The difference between the new operator and Object.create Operator
 - new Operator in JavaScript
 - Object.create in JavaScript

- Prototypes and Prototypal Inheritance
 - Prototype
 - Prototypal Inheritance
 - Prototypal Inheritance by an analogy
 - Why is Prototypal Inheritance better?
 - Linking the prototypes
 - Prototype Chain
- Advanced JavaScript (Closures, Method Chaining, etc.)
 - Hoisting in JavaScript
 - JavaScript Closures
 - IIFE
 - Closure And IIFE
 - JavaScript call() & apply() vs bind()
 - bind
 - Example using bind()
 - call()
 - apply
 - Asynchronous JavaScript
 - Callback Function
 - Callback in asynchronous programming
 - Promises
 - Promise.all
 - Async-await
 - Explanation via Example
 - Handle errors using async-await
- Next Generation JS - ES6 and Beyond
 - JavaScript Classes
 - Class methods
 - Class vs Constructor function
 - Using Function - ES5 style
 - Using Classes - ES6+ Style
 - let and const and Block scope
 - let
 - const
 - Tricky const
 - Arrow Functions
 - Lexical this

React JS

- React
 - Features
 - Benefits
 - Limitations
- SPAs
- Installing React
 - Prerequisites

- Install Node
 - Install Yarn
 - Install ReactJS using create-react-app
 - Online Playgrounds
- Deploying React App To Internet
 - Deploy to Netlify
 - Easy setup deploy
- Basics of React
 - React JSX
 - Virtual DOM
 - React Components
 - Component Render
 - Function Components
 - Class Components
 - Pure components
 - Reusing Components
 - States And Props
 - States
 - Props
 - Event Handling
 - Bind this
 - Passing Arguments
 - Two Way Binding
 - One way data binding
 - Two Way - 2 way binding
- Styling your components
 - Inline Styles
 - CSS Stylesheets
 - Dynamic Styles
- Advanced React
 - Conditional Rendering
 - Outputting Lists
 - Keys
 - Higher Order Components
 - Render Props
 - Component Lifecycle
 - initialization
 - mounting
 - componentWillMount()
 - componentDidMount()
 - static getDerivedStateFromProps()
 - updating
 - componentWillReceiveProps()
 - shouldComponentUpdate()
 - getSnapshotBeforeUpdate()
 - componentDidUpdate()

- componentDidUpdate()
 - unmount
 - Error handling
 - componentDidCatch()
- React Hooks
 - React Hooks Basics
 - useState React Hook
 - useEect React Hook
 - Cleanup
 - useRef React Hook
 - Two Use Cases
 - 1. Accessing DOM nodes or React elements
 - 2. Keeping a mutable variable
 - Context
 - React.createContext
 - Context provider
 - Consuming context
 - useContext
- App performance optimization
 - Improve React app performance
 - Memoization
 - useMemo
 - Lazy Loading
 - Suspense
 - Suspense - Data Fetching
 - Render-as-You-Fetch

Web Modules

- Web Development Tooling
 - Git and GitHub Basics
 - What is Git
 - Install Git
 - Create Git project
 - Adding a le to your repo
 - Committing your le
 - Git clone
 - How to clone Git repo?
 - Git Branching
 - Checkout the branch
 - Merging branches
 - Git status
 - Webpack
 - Installing Webpack
 - Webpack in-depth
 - Webpack config

- HTTP and API
 - What is HTTP
 - HTTP is a stateless protocol
 - HTTP Headers
 - SSL and TLS
 - SSL handshake process
 - Web Services
 - RESTful web services - REST API
 - GraphQL
 - Benefits of GraphQL / Why GraphQL?
 - Setup GraphQL
 - Create First GraphQL Application
 - Data Access Layer
 - Data store
 - Create a schema in schema.graphql
 - Creating your first resolver
 - Query
 - Nested Query
 - Dynamic Queries
 - Mutations
 - Setting up GraphQL client
 - Online Playgrounds
 - GraphQL vs RESTful APIs
 - Apollo
 - Apollo client
 - Apollo Server
- Web Application Performance
 - List of ways to reduce page load time
 - How to lazy load images
 - Web Worker
 - How Web Workers work
 - Terminating the worker thread
 - Web workers use cases
 - Important points about Web workers
 - Web workers cons
 - Server-side rendering SSR vs Client-side rendering CSR
 - Client-side rendering
 - Server-side rendering
- Web security
 - Authentication vs Authorization
 - Authentication
 - Authorization
 - OAuth
 - JSON Web Token - JWT
 - Structure of JWT
 - Popular analogy to explain JWT
 - Local storage vs Session storage
 - localStorage

- sessionStorage
- Web attacks
- CSRF
 - How does CSRF work?
 - Defense against CSRF
- XSS
 - How XSS work?
 - Defense against XSS
- CORS
- Principles of secure coding
- Secure your Web application
 - Disable cache for web pages with sensitive data
 - Use HTTPS instead of HTTP
 - Prevent Content-Type Sniffing
- Security in JavaScript