RoadMap For Intern

Week : 1

Day 1: HTML Basics

- Study the basic structure of an HTML document: <!DOCTYPE html>, <html>, <head>, and <body>.
- Learn about headings (<h1> to <h6>), paragraphs (<p>), and line breaks (
).

Day 2: HTML Elements

- Explore lists (<ul>, <ol>, and <li>), links (<a>), and images (<img>).
- Learn how to create hyperlinks to other web pages and internal links within the same page.

Day 3: HTML Forms

- Study form elements like <form>, <input>, <textarea>, <select>, and <button>.
- Learn how to create basic forms to collect user input.

Day 4: HTML Semantic Elements

- Understand the importance of semantic elements like <header>, <nav>, <main>, <section>, <article>, <footer>, etc.
- Learn how semantic elements improve the structure and accessibility of web pages.
- Practice creating a functional web form.

NOTE: READ INTERVIEW QUESTION FOR HTML

Day 5: CSS Basics

- Learn the CSS syntax and how to apply styles to HTML elements.
- Understand the concept of selectors and how to target HTML elements with CSS rules.
- Practice changing colors, fonts, margins, padding, and backgrounds.
- Deep dive into Flexbox for more flexible layouts.
- Create a responsive navigation bar using Flexbox

Week : 2

Day 6: CSS Layout and flex

- Learn about the CSS box model (margin, padding, border) and how it affects the layout of elements.
- Explore different positioning techniques, such as static, relative, absolute, and fixed.
- Create simple layouts using floats and understand their limitations.
- Dive into CSS Flexbox for more advanced and flexible layouts.

Day 7: CSS Responsive Design and Project

- Study responsive web design principles.
- Implement media queries to make your website responsive.
- Create responsive designs using media queries to adapt your layout to different screen sizes.
- Work on a small project to apply your HTML and CSS skills, like building a simple landing page or a personal portfolio.. with responsive using media query.

NOTE: READ INTERVIEW QUESTION FOR CSS

Day 8: Introduction to Bootstrap

- Start with the official Bootstrap website's "Getting Started" section.
- Learn about the key features and benefits of Bootstrap.
- Familiarize yourself with the Bootstrap documentation layout and navigation.
- Explore the basic Bootstrap template structure.

Day 9: Bootstrap Grid System and Flexbox

- Study the Bootstrap grid system in detail on both W3Schools and the official website.
- Understand the grid classes, containers, and responsive layout principles.
- Practice creating responsive layouts using the Bootstrap grid.
- Delve deeper into Bootstrap's Flexbox-based layout system.

Day 10: Bootstrap Components , and Advanced Features

- Explore the various Bootstrap components on both platforms (W3Schools and official site).
- Learn how to use components like buttons, cards, forms, and more.
- Implement these components in sample projects to reinforce your understanding.
- Explore any advanced features or components you find interesting.
- Work on a mini-project combining various Bootstrap components and features with responsive.

Week : 3

Day 11: Bootstrap Utilities and Customization

- Study the utility classes offered by Bootstrap for quick styling.
- Learn how to customize Bootstrap to match your design preferences.
- Experiment with the theme customization options provided by Bootstrap

NOTE: READ INTERVIEW QUESTION FOR BOOTSTRAP

Day 12: Introduction to JavaScript

- Start with the JavaScript tutorial on W3Schools.
- Learn about the basics: variables, data types, operators, and basic syntax.
- Study Type Conversions , Comparisons in JavaScript.
- Learn alert, prompt and confirm
- Practice writing simple JavaScript code to perform calculations and display output.

Day 13: JavaScript Functions and Conditionals

- Continue with the JavaScript tutorial on W3Schools.
- Study functions and their importance in JavaScript.
- Understand conditional statements (if/else) and use them in your code.
- Implement simple functions and conditionals in practical examples.

Day 14: JavaScript Arrays and Object

- Learn about JavaScript arrays and their various methods.

- Explore array manipulation and iteration.
- Study object , '**This'** method and optional chaining in JavaScript.

Day 15: JavaScript Events and Loops

- Study JavaScript events and how to attach event handlers to elements.
- Understand loops (for, while) and use them to perform repetitive tasks.
- Practice creating interactive applications with events and loops.
- Study Arrow functions in JavaScript.

NOTE: READ INTERVIEW QUESTION FOR JAVASCRIPT

Week : 4

Day 16: JavaScript Classes and Error Handling

- Study basic Syntax of classes.
- Study Private and public classes and class inheritance in JavaScript.
- Study Try…Catch and Custom error handling
- Perform it's practical

Day 17: Introduction to TypeScript

- Basic Introduction, Installing Compiler
- Learn About Typescript Simple And Special Types
- Perform it's practical

Day 18: Typescript Array,Tuples And Object Types

- Learn how to Define Array and Object Types
- Functions In Typescript
- Perform it's practical

Day 19: Typescript Classes,Type Alias,Modules

- Learn how to Define Class
- Use Of Type Alias, InterFaces, Interfaces with classes
- Perform it's practical

Day 20: Typescript Generics,Enums ,Keyof ,Ts null

- Learn how to use Generics,and Enums
- Learn about TS Script Mode, Strict Null checks,Narrowing in TS
- Perform it's practical

Day 21: Create a Mini Project

- Create simple form using HTML,CSS,BootStrap and Perform CRUD operation
- Use All the Possible Functionality of Typescript.
- **READ INTERVIEW QUESTION FOR TYPESCRIPT**

Angular RoadMap

READ AND TAKE REFERENCE FOR ANGULAR.IO

Day 1: Introduction to Angular and component

- Overview of Angular framework and its features
- Setting up the development environment (Node.js, Angular CLI)
- Creating your first Angular project and running it
- Understanding components and their role in Angular
- Creating a new component and using it in your project
- Working with templates and data binding

Day 2: Directives , Pipes, and Data binding

- Data binding: Interpolation, property binding, and event binding.
- Directives: ngIf, ngFor, and ngSwitch.
- Introduction to directives and their types (structural and attribute)
- Using built-in directives and creating custom ones
- Working with pipes to format data in templates

Day 3: Services and Dependency Injection

- Understanding services and their purpose in Angular
- Creating and using services to manage application logic
- Learning about dependency injection and its role in Angular

- Services and dependency injection: Create a basic service and inject it into a component.
- Introduction to RxJS and Observables.

Day 5: Routing and Navigation
- Implementing basic routing in an Angular application
- Creating multiple pages using the Angular Router
- Learn types of routing in Angular
- Advanced routing techniques: Route guards and lazy loading modules.
- Navigating between different components

Day 6: Forms in Angular using Angular Material
- Forms in Angular: Template-driven forms.
- Reactive forms and form validation.
- Handling form validation and user input
- Introduction to Angular Material design components
- Integrating Material components into your project
- Enhancing the UI/UX of your application with Material components
- Link of Angular material

Day 7: HTTP and Data Retrieval
- Making HTTP requests in Angular
- Handling responses and error handling
- Integrating external APIs into your application
- https://api.github.com/users/hadley/orgs get data into table format

Day 8: State Management with NgRx (Optional)
- Understanding the principles of state management
- Setting up NgRx in your Angular application
- Managing application state using NgRx store, actions, and reducers
- State management with NgRx: Introduction to reactive state management.

Day 9: Authentication and Security
- Implementing user authentication in an Angular app
- Securing routes and managing user sessions

- Exploring common security practices in Angular

Day 10: End-to-End Testing with Cypress (Optional)

- Introduction to end-to-end testing with Cypress
- Writing end-to-end tests for your Angular app
- Automating testing for better efficiency

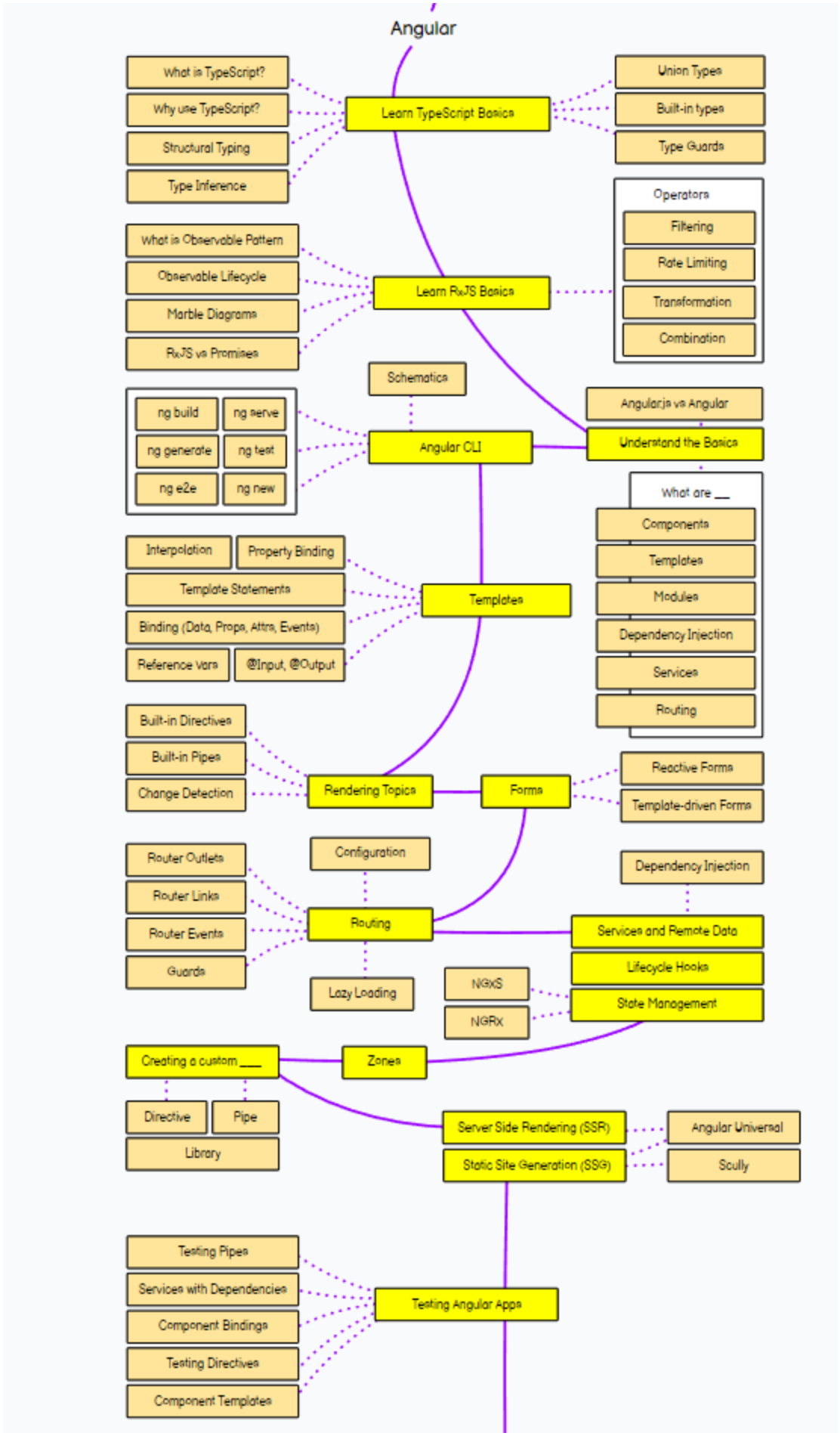
Day 11: Deployment and Performance Optimization

- Preparing your Angular app for deployment
- Strategies for optimizing performance
- Deploying your Angular app to a hosting platform
- Progressive Web Apps (PWA) with Angular

Day 12:Implement Project

- Spend the day working on a personal project using Angular
- Review and consolidate your learning
- Plan your next steps for further Angular development
- Deployment: Build your Angular app for production.
- Host your app on a web server or deploy it to platforms like Firebase or Netlify.

READ INTERVIEW QUESTION FOR ANGULAR


Angular
What is TypeScript?
Why use TypeScript?
Structural Typing
Type Inference
Learn TypeScript Basics
Union Types
Built-in types
Type Guards
Operators
Filtering
Rate Limiting
Transformation
Combination
What is Observable Pattern
Observable Lifecycle
Marble Diagrams
RxJS vs Promises
Learn RxJS Basics
Schematics
ng build
ng serve
ng generate
ng test
ng e2e
ng new
Angular CLI
Angularjs vs Angular
Understand the Basics
What are __
Components
Templates
Modules
Dependency Injection
Services
Routing
Interpolation
Property Binding
Template Statements
Binding (Data, Props, Attrs, Events)
Reference vars
@Input, @Output
Templates
Built-in Directives
Built-in Pipes
Change Detection
Rendering Topics
Forms
Reactive Forms
Template-driven Forms
Router Outlets
Router Links
Router Events
Guards
Configuration
Routing
Lazy Loading
Dependency Injection
Services and Remote Data
Lifecycle Hooks
NGXS
NGRX
State Management
Creating a custom ___
Directive
Pipe
Library
Zones
Server Side Rendering (SSR)
Static Site Generation (SSG)
Angular Universal
Scully
Testing Pipes
Services with Dependencies
Component Bindings
Testing Directives
Component Templates
Testing Angular Apps

React RoadMap

READ AND TAKE REFERENCE FOR REACT.DEV

Day 1: Introduction to React
- Overview of React and its core concepts
- Setting up the development environment (Node.js, npm)
- Creating your first React project and running it

Day 2: JSX and Components
- Understanding JSX syntax and how it works
- Creating functional components and rendering them
- Composing components to build more complex UIsc

Day 3: Props and State
- Learning about props and how to pass data between components
- Introducing state and managing component state
- Understanding the difference between props and state

Day 4: Component Lifecycle
- Understanding the lifecycle methods of a React component
- Using lifecycle methods to perform actions at different stages
- Handling component updates and unmounting

Day 5: Handling Events
- Handling user events (e.g., click, input) in React components
- Using event handlers and updating state based on events
- Implementing interactive features in your application

Day 6: Lists and Keys
- Working with lists in React to render dynamic content
- Understanding the importance of keys in list elements
- Updating lists and handling list-related challenges

Day 7: Forms and Form Handling

- Building controlled components for form inputs
- Validating and handling form submissions in React
- Managing form state and synchronization with the UI
- Use material ui for design form
- Link for Material ui

Day 8: Styling in React

- Exploring different approaches to styling React components
- Using CSS modules, inline styles, or CSS-in-JS libraries
- Organizing and managing styles in larger applications

Day 9: React Router

- Implementing client-side routing with React Router
- Creating multiple pages and navigation in a React app
- Handling route parameters and nested routes

Day 10: React Hooks (Part 1)

- Understanding the fundamentals of React Hooks
- Exploring useState and useEffect hooks
- Refactoring class components to functional components with hooks

Day 11: React Hooks (Part 2)

- Working with other React hooks like useContext and useReducer
- Creating custom hooks for reusable logic
- Maximizing the benefits of React hooks in your code

Day 12: State Management with Redux (Optional)

- Understanding the principles of Redux and its core components
- Setting up Redux in a React application
- Managing state and actions using Redux and React Redux

Day 13: Redux Middleware and Asynchronous Actions (Optional)

- Implementing Redux middleware for tasks like logging and API calls
- Handling asynchronous actions with middleware like Redux Thunk

- Enhancing the Redux store for better performance

Day 14: React Performance Optimization
- Identifying and optimizing performance bottlenecks
- Using React's memoization techniques and pure components
- Employing tools like React DevTools for performance analysis

Day 15: Project Work and Wrap-Up
- Spend the day working on a personal project using React
- Review and consolidate your learning
- Plan your next steps for further React development

READ INTERVIEW QUESTION FOR REACT