Angular SPA

- Angular remains one of the world's most popular JavaScript frameworks. The redesigned Angular continues to be a robust framework for building Single Page Applications.
- **Angular** is an open source, TypeScript based frontend web application framework. It has been released by **Google's Angular** community.
- It is a TypeScript based full-stack web framework for building web and mobile applications.
- It supports for web application that can fit in any screen resolution.
- Angular application is fully compatible for mobiles, tablets, laptops or desktops.
- It has an excellent user interface library for web developers which contains reusable UI components.

CONTD…

- It helps to create Single Page Applications (SPA). SPA is reactive and fast application.
- For Example, if we have a button in single page and click on the button then the action performs dynamically in the current page without loading the new page from the server.
- Angular 8 is Typescript based object oriented programming and support features for server side programming as well.

Angular Versions

AngularJS

- AngularJs is very powerful JavaScript framework. It was released in October 2010. AngularJS based on Model View Controller (MVC) architecture and automatically handles JavaScript code suitable for each browser.
- AngularJS is a client side JavaScript MVC framework to develop a dynamic web application.
- AngularJS was originally started as a project in Google but now, it is open source framework. AngularJS is entirely based on HTML and JavaScript, so there is no need to learn another syntax or language.
- AngularJS changes static HTML to dynamic HTML. It extends the ability of HTML by adding built-in attributes and components and also provides an ability to create custom attributes using simple JavaScript.

Angular 8 New Features

- **Bazel support** − If your application uses several modules and libraries, Bazel concurrent builds helps to load faster in your application.
- **Lazy loading** − It splits **AppRoutingModule** into smaller bundles and loads the data in the DOM.
- **Differential loading** − When you create an application, Angular CLI generates modules and this will be loaded automatically then browser will render the data.
- **Web worker** − It is running in the background, without affecting the performance of a page.
- **Improvement of CLI workflow** − Angular 8 CLI commands ng-build, ng-test and ng-run are extended to third party libraries.
- **Router Backward Compatibility** − Angular router backward compatibility feature helps to create path for larger projects so user can easily add their coding with the help of lazy coding.
- **Opt-in usage sharing** − User can opt into share Angular CLI usage data.

Prerequisite

- Angular is written in **TypeScript**. We need **Node** and **npm** to compile the files into **JavaScript.**
- **Node.js** must be installed in our system.

  node –version
- **npm** is used to install **Angular 8** CLI. Once **Node.js** is installed, **npm** is also installed.

  npm –v
- Install Angular 8 CLI using npmas follows −

  npm install -g @angular/cli@^8.0.0

Check Angular version

ng version

CONTD…

- **ng** is the CLI application used to create, manage and run Angular Application.
- It written in JavaScript and runs in NodeJS environment.
- ng **serve** is the sub command used to compile and run the Angular application using a local development web server. **ng server** will start a development web server and serves the application under port, 4200.

Angular framework is based on four core concepts and they are as follows −

- Components.
- Templates with **Data binding** and **Directives**.
- Modules.
- Services and dependency injection.

CONTD…

- The core of the Angular framework architecture is **Angular Component**. It is the building block of every Angular application.
- Every angular application is made up of one more **Angular Component**.
- It is basically a plain JavaScript / Typescript class along with a HTML template and an associated name.
- The HTML template can access the data from its corresponding JavaScript / Typescript class.
- Component's HTML template may include other component using its selector's value (name).
- The Angular Component may have an optional CSS Styles associated it and the HTML template may access the CSS Styles as well.

Example

```
// src/app/app.component.ts
import { Component } from '@angular/core';
@Component({
   selector: 'app-root',
  templateUrl: './app.component.html',
   styleUrls: ['./app.component.css']
})
export class AppComponent {
   title = 'Expense Manager';
}
```

CONTD…

- Components are defined using the @Component decorator function, which takes in metadata about the component. It is used to convert a normal Typescript class to **Angular Component**.
- **app-root** is the selector / name of the component and it is specified using **selector** meta data of the component's decorator. **app-root** can be used by application root document, **src/index.html** as specified below

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>ExpenseManager</title>
    <base href="/">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" type="image/x-icon" href="favicon.ico">
  </head>
  <body>
    <app-root></app-root>
  </body>
</html>
```

CONTD…

- **app.component.html** is the HTML template document associated with the component. The component template is specified using **templateUrl** meta data of the **@Component** decorator.
- **app.component.css** is the CSS style document associated with the component. The component style is specified using **styleUrls** meta data of the **@Component** decorator.
- **AppComponent** property (title) can be used in the HTML template as mentioned below –**{{ title }}**

Template

- Template is basically a super set of HTML. Template includes all the features of HTML and provides additional functionality to bind the component data into the HTML and to dynamically generate HTML DOM elements

DATA BINDING

- Used to bind the data from the component to the template.

{{ title }}

Directives

Used to include logic as well as enable creation of complex HTML DOM elements.

ngIf and **showToolTip** are directives

<p *ngIf="canShow">

This sectiom will be shown only when the *canShow* propery's value in the corresponding component is *true* </p>

<p [showToolTip]='tips' />

MODULES

- **Angular Module** is basically a collection of related features / functionality. **Angular Module** groups multiple components and services under a single context.

Advantages Of Angularjs

- Open source JavaScript MVC framework.
- Supported by Google
- No need to learn another scripting language. It's just pure JavaScript and HTML.
- Supportsseparation of concerns by using MVC design pattern.
- Built-in attributes(directives) makes HTML dynamic.
- Easy to extend and customize.
- Supports Single Page Application.
- Uses Dependency Injection.
- Easy to Unit test.
- REST friendly

REACT

- Unlike Angular, which offers a full Model-View-Controller pattern implementation, React is only concerned with views.
- React, sometimes referred to as a frontend JavaScript framework, is a JavaScript library created by Facebook.
- React is a tool for building UI components.
- It's not a framework, just a library. There are a number of libraries that are designed to be used with React to produce rich single page applications.
- One of React's most important features is its use of a virtual DOM.
- React creates a VIRTUAL DOM in memory.

CONTD…

- The virtual DOM provides React with several advantages, including performance (the virtual DOM can optimize which parts of the actual DOM need to be updated) and testability (no need to have a browser to test React and its interactions with its virtual DOM).
- Rather than having a strict separation between code and markup (with references to JavaScript appearing in HTML attributes perhaps), React adds HTML directly within its JavaScript code as JSX. JSX is HTML-like syntax that can compile down to pure JavaScript.

CONTD…

- React is a JavaScript library for building user interfaces.
- React is used to build single-page applications.
- React allows us to create reusable UI components.

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>
    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js" crossorigin></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  </head>
  <body>
    <div id="mydiv"></div>
    <script type="text/babel">
      function Hello() {
        return <h1>Hello World!</h1>;
      }
      ReactDOM.render(<Hello />, document.getElementById('mydiv'))
    </script>
  </body></html>
```