Angular

* JavaScript **framework** which allows us to create **single page applications**.
* **Framework:** Collection of predefined classes and methods which provides api’s for performing different operations when used in an application.
* **Single Page Applications**: It has only one HTML page. When we navigate around, only the content of that HTML page changes. The page itself never changes.

**Advantages of SPA**

1. Since we are using JavaScript to change the content of the page, it is much faster. Here we are not reaching out to the server to request a new piece of HTML data, every time we navigate to a different URL.
2. This allows us to create an application which is fast and reactive.
3. **Project Structure:**
4. node module -> 3rd party libraries are stored. Not deployed on production server
5. package.json -> every project which depends on npm has it. Project related configuration
   1. dependencies -> project directly dependencies
   2. devDependencies -> packages for development
6. package-lock.json -> used to ensure version of project dependencies
7. .editorconfig -> coding standard for the project
8. Angular.json -> angular related configuration
9. Tsconfig.json -> typescript compiler configuration
10. src/app -> app code goes here
11. every angular app has one module one component at least
12. src/assets -> public accessible resourse
13. src/index.html -. Main html page which is rendered we don’t have refrence to js and css they will be injected during the build process which will be injected using dependency injection
14. src/main.ts -> starting point of the application bootstrap the appmodule
15. src/style.css -> global style
16. **Bootstrapping:**
    1. **Phase 1: Build-Time (Compilation & Bundling): Happens when you run ng build or ng serve**
    2. Step 1: Angular CLI Starts the Build Process
       * The CLI reads angular.json to determine:
         + Entry file (main.ts).
         + Asset paths (HTML, CSS, etc.).
       * Build optimizations (AOT/JIT, production/dev).
    3. Step 2: TypeScript Compilation
       * .ts files (like main.ts, app.module.ts) are compiled to .js.
       * AOT (Production): Templates compiled to JS *during build* (faster runtime).
       * JIT (Development): Templates compiled *in the browser* (slower, but easier debugging).
    4. Step 3: Webpack Bundling
       * Angular CLI uses Webpack to:
         + Create Dependency Graph: main.ts → AppModule → AppComponent → app.component.html.
         + Generate Bundles:
           - runtime.js (Webpack runtime logic).
           - polyfills.js (browser compatibility).
           - main.js (your app code).
           - vendor.js (Angular + third-party libraries).
           - styles.css (global styles).
    5. Step 4: Inject Bundles into index.html

<script src="runtime.js"></script>

<script src="polyfills.js"></script>

<script src="vendor.js"></script>

<script src="main.js"></script>

* 1. **Phase 2: Runtime (Execution in the Browser)**

1. Step 1: Browser Loads index.html
   * + The HTML file is parsed: The <app-root> element is detected (but empty at first).
2. Step 2: Script Execution
   * + runtime.js → Initializes Webpack module loading.
     + polyfills.js → Ensures Angular works in older browsers.
     + vendor.js → Loads Angular framework dependencies.
     + main.js → Contains your app’s compiled code.
3. Step 3: Angular Platform Initialization
   * + **main.js runs the code from main.ts:**
4. Step 4: AppModule Bootstrapping: Angular loads AppModule (app.module.ts):
5. **Step 5: AppComponent Renders:** Angular finds AppComponent (app.component.ts):

Angular CLI

1. Create Angular Project: ng new project-name
2. Start Project: ng serve/ ng s