**Javascript concepts**

### Javascript

* Not a Strictly typed
* Declaring scalar variables
* Knowing the type of variables
* Arrays
* Javascript objects
* Concepts related to arrays [ indexes and properties]
* Process array using for [ in vs of in for loop ]
* Object processing keys ( Object.keys() )
* Pass by reference
* Spread operator
* Spread Operator - Array and objects
* Destructuring Operator
* Functions
* Fuction Parameters [ Behaviour of JS in parameter passing)
* Function return statement [ Meaning of undefined]
* Window scope and Global Scope (node js) - this keyword
* Anonymous Function
* Function returning another function
* Closures [ Outer function scope and inner function scope ]
* Keywords Related to Variables[ const, var , let and global variables]
* global vs function vs local scopes
* Callback [ Passing function as a parameter ]
* Prototype Functions
* Classes
* Arrow Functions
* Binding this keyword
* resolving bind using arrow functions

**API Based website**

HTTP URL

url format:

http://host:port/resource

HTTP REQUEST

REQUEST = header + body

HTTP VERB

GET POST PUT DELETE PATCH

Synchronous Communication [ response is html+Css+JS] Multi page application

Single page appication

asynchronous commuication

asychrochrounous communication

is initiated using JS Program

Synchronous communication is initiated because of

* url change
* anchor tag click
* form submission

in http , if server responds with html/css and js the server execution unit is called web application

if the server responds with REST [ represented state / media type] execution unit is called WEB API/REST API/REST SERVICE/ REST WEB SERVICE

the technology used to call restapi using javascript is called AJAX

we used to or still can make ajax calls using XMLHttpRequest

But in the mordern JS we got a library called fetch api to do the same

### Angular Directives

The custom html elements and custom attributes that we create in angular

are called angular Directives

there are three kinds of Directives in Angular

* Component Directives
* Structural Directives
* Attribute Directives

#### Component Directives

Every component that we create is a custom html element

EVery component that you create is a component directive

#### Structural Directives

Structural Directives generate html template, hides/unhides the template

Basically Structural directives manipulate the layout[html]

There are few inbuilt stuructural directies

Example: ngFor, ngIf, ngSwitch

We can also create our own custom structural directives

Note: while using structural directives we must prefix them with asterisk(\*)

### Attribute Directives

Attribute Directives changes the behaviour of target elements

Behaviour change ? -- manipulate css attributes like size,color

manipulate data

listening and handling events

Attribute directives are further categorized into three kinds

* input directive
* output directive
* input and output directive

#### input Directive

input directive is meant to receive a value from the right side of the

expression and change the behaviour of target element

and usually the right side value would be a variable defined in the class

some inbuilt input directives incude

```

style, ngStyle,ngClass, RouterLink, formControlName.....

```

We have to enclose the input directives within []

<h1 [sample]="data"></h1>

in the above case, h1 is the target element

sample is the input directive

data is the value, data is a variable that would be present in the component class

if the same above thing defined in the following way

<h1 sample="data"></h1> OR

<h1 [sample]="'data'"></h1>

in this case, data would be considered a string , that is is directly

taken as value

#### Output Directives

output directives listens to the appropriate Event associated with it on the target element

click, keyup,keydown, change, mouseover,mouseaway.....

in angular , events on the target elements must be handled by using output directives

output directives must be enclosed within ()

in the right side usually we will give a funciton call and if necassary with a parameter

example

<button (click)="doThis()"></button>

in the above case whenever you click button doThis will be called

click is the output directive

#### input output directives

This directive is bidirectional

it listens to the changes and at the same time receives value from the right side

usually this directives are used in form elements

and the only known popular input output directive is ngModel

it must be enclosed with [()]

<input type="text" [(ngModel)]="fname">

in the above case if fname ="John", the text box will be filled with John

and suppose if you type "peter" in the text box the value fname will be changed

to peter

### Parent child communication

#### Parent to child

if a parent needs to send data to the child

the child must have variables ready to receive the value from parent and those variables must be decorated with @Input

and from that point onwards those variables will behave like input directives but the target element can be only child component

### Child to parent

## Services

Service is a class in angular

It is usually used to hold some reusable logic that can be called and utilized by components , other angular units like pipes, directives and other services

Service is a class that is decorated with decorator called @Injectable

The reason we decorate a service with Injectable is we should not create object for it , instead we must get it injected by angular

By default an angular service is singleton

singleton? - no matter how many places you request the object in the code , you will get the same object

note : if we want a fresh object not a common single object, we can still get it

### **Server side routing**

Generally when we perform any of the following operations in a webpage, the url in the address bar will change and a request will be sent to server and the server will resolve it

* Using anchor tag and Href
* Submitting a form
* typing the url directly

since server resolves it and gives a new html page we call it server side routing

and since new page is loaded we cannot categorize the app is SPA and basically the app is multipage application

### **client Side routing**

When doing SPA, we tend to avoid making a server request while we change the url in the address bar and hence we need to avoid submit button click, anchor tag + href combination , direct typing of url [even done will be resolved using client library ]

If url change is resolved by client side library(angular) itself it is called as client side routing

typically in angular, we do it in two ways

1. Anchor tag without href
2. Programmatically changing the url

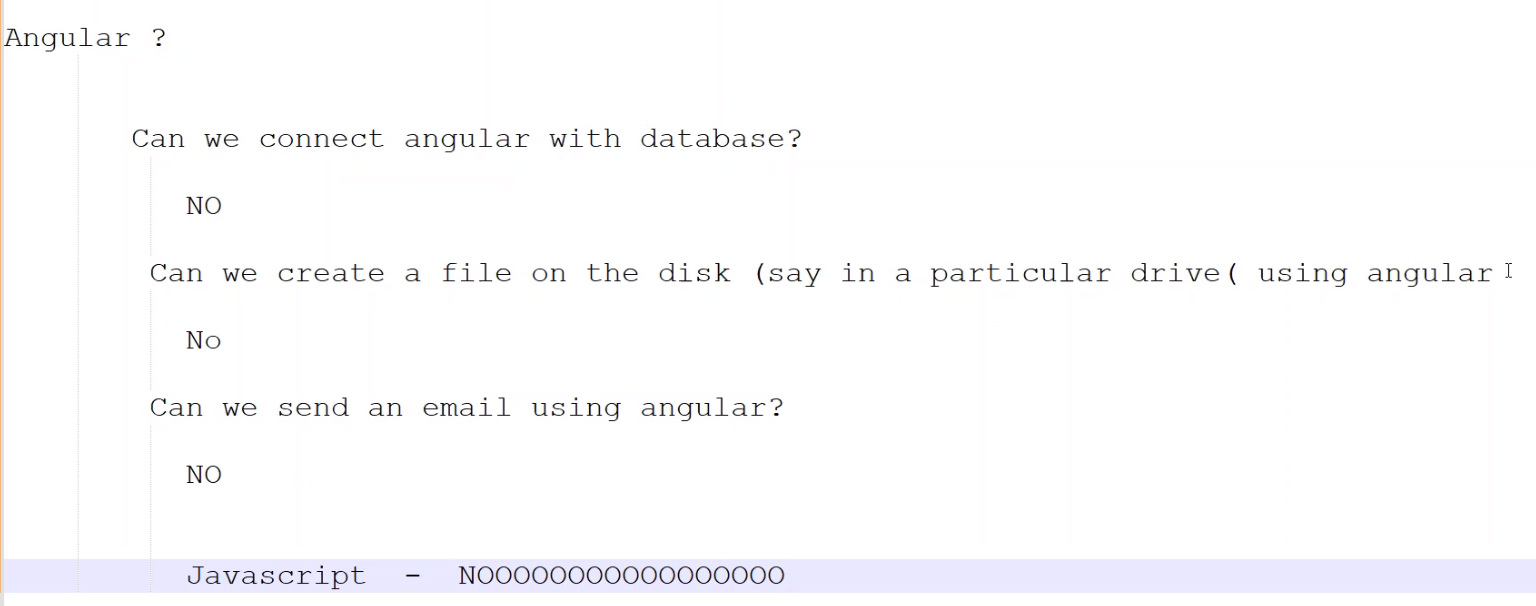
When we are configuring angular Routing we should focus on Three things

1. Target Template location where the associate route component to be loaded (for this we call a component called router-outlet which is part of routing module)
2. We should configure our anchor tags (if at all you are handling routes using anchor tag) with input directive called routerLink instead of href
3. add the routing configuration where we map paths with components

**Angular running on Browser**

Can be converted to Mobile app from Angular.

Greater than 12 node.js



Web SQL – database connected from browser

Node Js – is a platform to run javascript from 2009

Browser – Renders HTML, CSS and execute Javascript

Type script – is not run on browser, but convert to javascript and run on browser

Angular run on browser

A picture containing text, screenshot, font, line

Description automatically generated

Javascript runs on both browser and back-end

HTML 5, CSS 3 defined on 2015.

HTML 5 new features – Notifications , Caching, Storage, Background scripts, drag and drop, drawing

CSS 3 – Responsiveness, Response Design [Responsive **web design** is about creating web pages that look good on all devices! A responsive **web design** will automatically adjust for different screen ...]

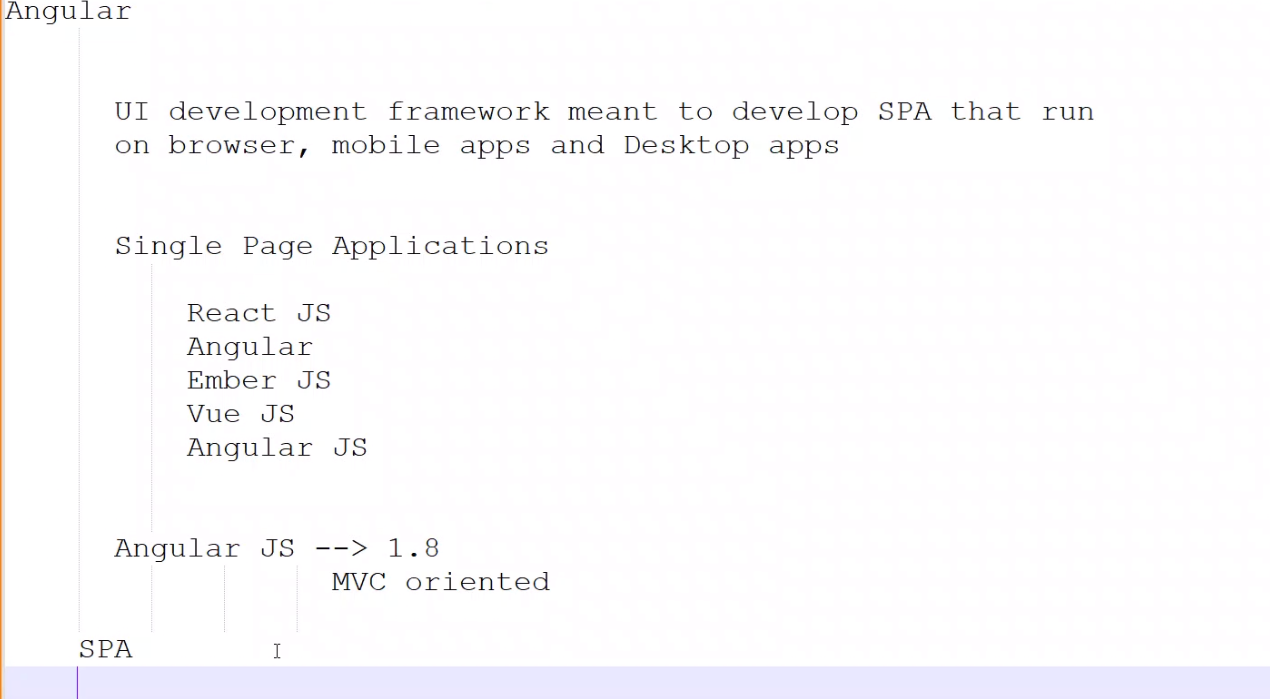
Javascript defined by ECMA – European computer manufacturer association.

ECMAScript

Now currently following script is es5. Every year releasing the es scripts

Es6 – released on 2015 [it contains the major changes]…can be called as es2015, es2016….es2022…

PWA – Progressing web Apps are web apps that use service workers, manifests, and other web-platform features in combination with progressive enhancement to give users an experience on par with native apps.



Single Page Application [SPA] - One HTML (Many JS files)

React JS

Angular

Ember JS

Vue JS

Angular JS

In reality,

Development Phase – many html files, type script files, many css files

Production phase – one html file, few js files (max 5 to 20 files), few css files

Add extension – Live server from Preferences in Visual studio code

Enable auto save in VSS from File menu

Javascript run on browser – HTML / CSS running on page - DOM renders [document object model]

Root object is window.

Window.console.log

Window.document.write etc

Javascript – Intepreted language, strictly typed language [No data types definition before hand], it supports string, number, Boolean scaler types,

Literals – for Boolean we true for true, false for false

Literals – for string , define with single quote or double quote

Variables are not declared within function, then it considers as global

Supports Objects, array type [nothing but collection]

[Git Push Local Branch to Remote – How to Publish a New Branch in Git (freecodecamp.org)](https://www.freecodecamp.org/news/git-push-local-branch-to-remote-how-to-publish-a-new-branch-in-git/)

**How to Push the Main Branch to Remote**

If you want to push the main branch to remote, it’s possible you’re pushing for the first time. Before you attempt to push to remote, make sure you’ve executed these commands:

* git init for initializing a local repository
* git add . to add all your files that the local repository
* git commit -m ‘commit message’ to save the changes you made to those files
* To push the main repo, you first have to add the remote server to Git by running git remote add <url>.
* To confirm the remote has been added, run git remote -v:

To finally push the repo, run **git push -u origin <branch-name> [could be master folder]**  
(“main” is the name of that branch for me). It could be master or Main for you. Initially, it was “master”, so I ran git branch -M main to change it.

Download the nignx server  
[nginx: download](https://nginx.org/en/download.html) – link

Unzip the nginx. Run the command prompt. Run the start ngnix  
C:\nginx-server\nginx-1.25.0\nginx-1.25.0

A picture containing text, screenshot, font, document

Description automatically generated

A picture containing text, font, document, screenshot

Description automatically generated

A picture containing text, screenshot, font, algebra

Description automatically generated

A screenshot of a computer program

Description automatically generated with low confidence

In command prompt type it ng…..if it is not installed then put this command in the command prompt

Ng new <appName> – will install all **core packages** of Angular under the folder <appName>

npm install -g @angular/cli

A screenshot of a computer

Description automatically generated

Install typescript anywhere in the system

npm install -g typescript – check whether installed ..the command is tsc

-g means global path. Check the path C:\Users\gganesan5\AppData\Roaming\npm

A black background with white text

Description automatically generated with low confidence

A picture containing text, screenshot, font, algebra

Description automatically generated

A screenshot of a computer program

Description automatically generated with low confidence

After downloading the project.. run the command in the VSS npm install.  
This will refer the package.json – find the dependencies of the package and install automatically.

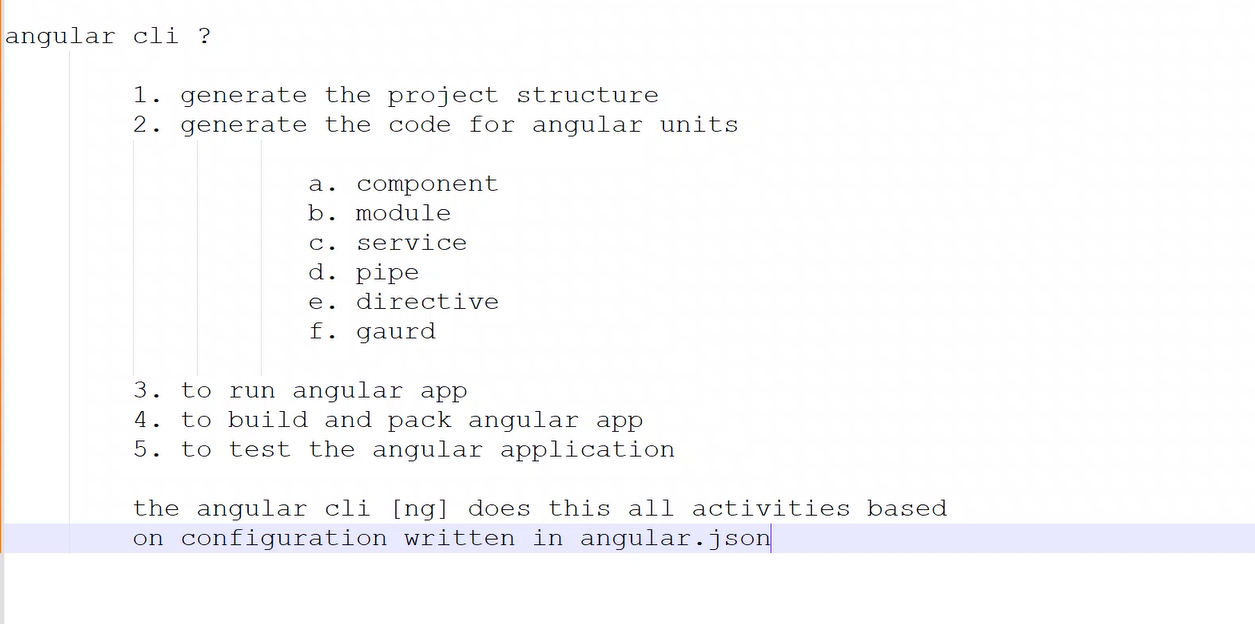
A screenshot of a computer

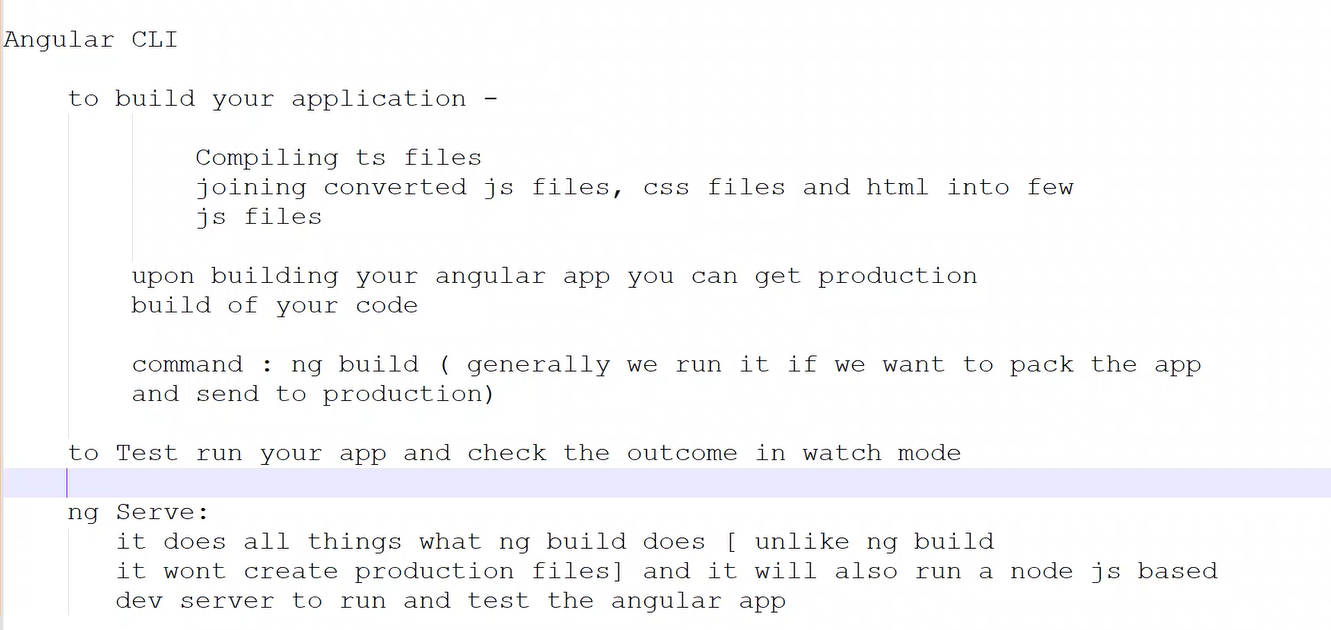
Description automatically generated with medium confidence

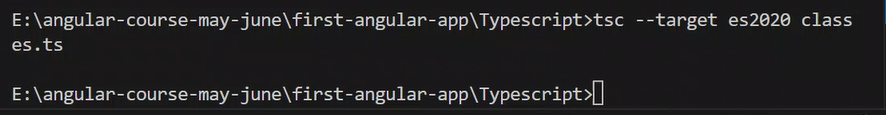
**What is the difference between NPM and Ng?**

NPM is basically a package manager which acts as a dependency provider. Similarly, YARN is another such example. NPM contains and manages many packages and modules, and NG is one such module which is a core module of Angular. functionality to run angular project.

.gitignore – folder --- those folders are not required to upload to git.







Until mention the ts version in the tsconfig.json, it will compile with es5 version



Angular.json -- configuration



<https://github.com/GGeetha123/AngularCourses>

Angular projects tips

Add additional components use the below command

\src\app>ng g c box – g for general , c for component, box is a component name

### Angular Directives

The custom html elements and custom attributes that we create in angular

are called angular Directives

there are three kinds of Directives in Angular

* Component Directives
* Structural Directives
* Attribute Directives

#### Component Directives

Every component that we create is a custom html element.

Every component that you create is a component directive

#### Structural Directives

Structural Directives generate html template, hides/unhides the template

Basically Structural directives manipulate the layout[html]

There are few inbuilt stuructural directies

Example: ngFor, ngIf, ngSwitch

We can also create our own custom structural directives

Note: while using structural directives we must prefix them with asterisk(\*)

### Attribute Directives

Attribute Directives changes the behaviour of target elements

Behaviour change ? -- manipulate css attributes like size,color

manipulate data

listening and handling events

Attribute directives are further categorized into three kinds

* input directive
* output directive
* input and output directive

#### input Directive

input directive is meant to receive a value from the right side of the

expression and change the behaviour of target element

and usually the right side value would be a variable defined in the class

some inbuilt input directives incude

```

style, ngStyle,ngClass, RouterLink, formControlName.....

```

We have to enclose the input directives within []

<h1 [sample]="data"></h1>

in the above case, h1 is the target element

sample is the input directive

data is the value, data is a variable that would be present in the component class

if the same above thing defined in the following way

<h1 sample="data"></h1> OR

<h1 [sample]="'data'"></h1>

in this case, data would be considered a string , that is is directly

taken as value

#### Output Directives

output directives listens to the appropriate Event associated with it on the target element

click, keyup,keydown, change, mouseover,mouseaway.....

in angular , events on the target elements must be handled by using output directives

output directives must be enclosed within ()

in the right side usually we will give a funciton call and if necassary with a parameter

example

<button (click)="doThis()"></button>

in the above case whenever you click button doThis will be called

click is the output directive

#### input output directives

This directive is bidirectional

it listens to the changes and at the same time receives value from the right side

usually this directives are used in form elements

and the only known popular input output directive is ngModel

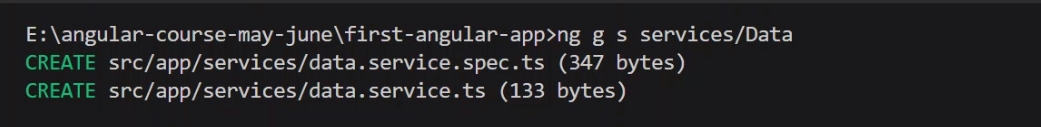
it must be enclosed with [()]

<input type="text" [(ngModel)]="fname">

in the above case if fname ="John", the text box will be filled with John

and suppose if you type "peter" in the text box the value fname will be changed

to peter



Create a new branch in github

In the command prompt run this…



A screen shot of a computer code

Description automatically generated with low confidence

Create a file “file.json”

Run the command json-server –port 4500 file.json

Run the post master

Run the localhost:4500/contact in the GET – it will give the result in the “body”

Link : [Getting started | Angular Material](https://material.angular.io/guide/getting-started)

Go to VSC – in the command run ng add @angular/material

Run the command in VSC Ng g m material --- our own model create

Copy from the link <https://raw.githubusercontent.com/vinodh1988/angular-more-features/master/src/app/material/material.module.ts> and paste it in to material.module.ts

Go to VSC – in the command run ng g c common/menubar – this will create



Go to VSC – in the command run ng g c pages/home – this will create

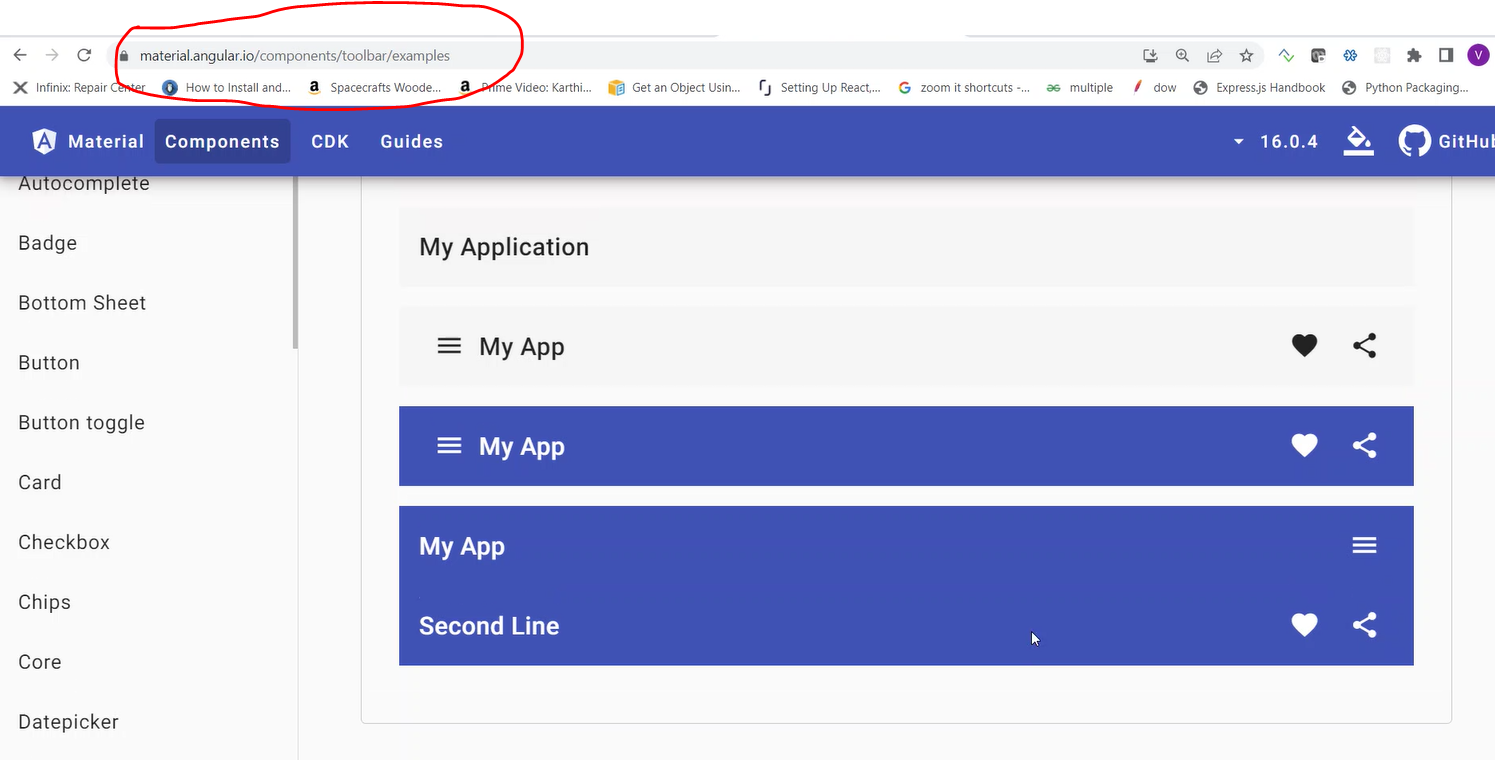
A close-up of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, font, document

Description automatically generated

Refer the url for patter of the pages and controls



To start ng server use the command ng serve in VSC. Default port of ng serve is 4200. So browse

http://Localhost:4200

Review menu --- code flow

1.modles.ts - creating the interface for reviews

2.reviews.service.ts - calling the URL using the kewword HttpClient and Observable.

3.reviews.component.ts - inside ngOnInit function subscribe is used with next and error using arrow functions, calling the function created in reviews.service.ts

4.review-box.component.html - current values to be shown using DIV tag - this only shows the values in the screen as expected

5.review-box.component.css - the styling done for the classes and ids created in point 3

6.review-box.component.ts - values are initialized using variable- here named as current

7.reviews.component.html - <app-review-box \*ngFor = "let x of reviews" [review] = "x"></app-review-box>

Training courses online video- share point portal  
[22 May - Batch 2 - Angular - OneDrive (sharepoint.com)](https://dedalusspa-my.sharepoint.com/personal/nsoundararaj_dedalus_eu/_layouts/15/onedrive.aspx?ga=1&id=%2Fpersonal%2Fnsoundararaj%5Fdedalus%5Feu%2FDocuments%2FShared%20with%20Everyone%2FDigiterati%20Training%2F22%20May%20%2D%20Batch%202%20%2D%20Angular&view=0)

Try this -- Exercise

[45e24d998fc020602344361a457d82bf.jpg (900×1214) (pinimg.com)](https://i.pinimg.com/originals/45/e2/4d/45e24d998fc020602344361a457d82bf.jpg)

A screenshot of a website

Description automatically generated with medium confidence