Creating a new App

1. Angular is backward compatible and built on node js. Rewritten on 2016.
2. Used to create single page responsive.
3. Cmd - ng new [app-name] --no-strict --standalone false [more parameters]

Angular apps are made by collection of components.  
  
Use components as much as possible.

Add Bootstrap CSS with cmd : npm install --save bootstrap

Creating a Component

Creating components by ng cli: ng generate component [component-name] or ng g c [component-name]

\*Use “emmet” plugin to write html faster in vscode

You can create standalone components

Assignment 1 success and warning alert components create

Data Binding

**Output Data :**

String Interpolation - {{ data }}

Property Binding - [property]=”data”

**React to user events :**

(event)=”expression”

**Two way data binding :**

[(ngModel)]=”data”

Directives

Data binding concept explained more here to illustrations:

ngIf and ngFor usages.

Check Assignment 3 for more details.

Debugging  
  
Using chrome developer console.   
Use intuition. Understand what error is shown on the console and also dev tools tab to debug

Using Source Maps to debug is good as well.

Components and Data Binding

Use @Input() to specify that property is bindable and initialised

Use @Output() to listen to events

You can specify aliases by providing it as a param. I.e., @Input(‘custom-name’)  
ng-content is a directive used instead of data binding. Helps against cross site scripting attacks. Use when complex HTML code needs to be viewed.

Lifecycle Hooks

ngOnChanges : called when bound input property changes.

ngOnInit: When components are created.

ngDoCheck: called when every change detection runs.

ngAfterContentInit: called after ng-content has been projected in view.

ngAfterContentChecked: called every time the projected content has been checked.

ngAfterViewInit: called after child’s view has been initialized.

ngAfterViewChecked: called after child’s view is checked.

ngOnDestroy: called right before the component is about to be destroyed.

Directives

Attribute directives

Structural directives

ng generate directive [directive-name]

Create attribute directives and add it in html tags to enable them. Eg: <input backgroundDirective />

Used as a way to change attributes of the ElementRef it is used for.

@HostListener, @HostBinding

Services

Created a Logging Service  
In @Component decorator add providers as the Service that you created. It will help.

We can use inject() to do the same but in in general approaches it is recommended to provide constructor injection for services.

Hierarchical Injector: If you inject a service then it will be available only to child components and not to parent components. It will all have only one instance of that service.  
If you do not provide the service in providers attribute of the @Component then it will create multiple instances of that very service available.  
@Injectable() decorator can be used if there is any service is injected into a service. *It is recommended to have to decorator added to service layer.*

This syntax can be used for lazy loading capability which will help with better performance overall.

Routing

Faster than reloading a website. State management.

<a [routerLink]=”/path” /> or <a routerLink=”/string-path” />

Taught relative path and absolute path

routerLinkActive=”active” is used to show if a route is active or not

[routerLinkActiveOption]={options…}

Use Router to route in typescript files. Make an instance of Router as router and use

ativatedRoute: ActivatedRoute

this.router.navigate[‘/path’, {relativeTo: this.activatedRoute}]

To add child route in the configurations add a parameter as chuldren :{ add the routes in array with path and component name}. Refer code

Use AuthGuard. Helps to open router links only if authenticated and authorised. CanActivate is used to guard routes.  
Two things canActive and canDeativate to provide and revoke access routes.