**Step 1 – Create and run new Angular Application**

 Navigate to any local folder, Open visual studio code and open terminal window

Check Angular CLI version

>ng version

If not latest version or not installed

>npm install -g @angular/cli@10.0.1

CLI command to create new application

>ng new angularLab

Run the application

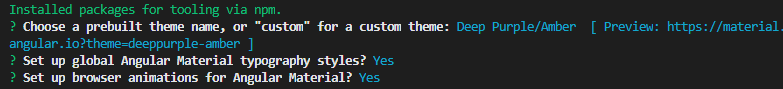
>ng serve -open

**Step 2 – Initial setup of Angular Material, Angular CDK and Animations**

Install angular material

>ng add @angular/material

It prompts for theme selection, select following options



 Adds BrowserAnimationsModule in app module

**Step 3 - Creating Material module**

Creating new module and adding the reference in App Module

>ng g module material -m App

 Though can be imported in app.module.ts for better architecture add new module and refer

 In the material.module.ts, add the highlighted code -

 Add export section, create materials array, import required controls

import { NgModule } from '@angular/core';

import { MatButtonModule } from '@angular/material/button';

const materials = [

    MatButtonModule

  ]

@NgModule({

  declarations: [],

  imports: [

    materials

  ],

  exports: [

    materials

  ]

})

export class MaterialModule { }

**Step 4 – Check out CLI help commands for awareness**

>ng help

>ng [command name] --help

>ng generate --help

>ng generate component --help

>ng g c --help

Step 5 – Install following packages required for session 2

**Add products component**

**>**ng g c products

**Add new component - order**

>ng g c orders

**Add SideNav component**

**>**ng g c SideNav

**Add register component**

>ng g c register

**Add login component**

>ng g c login

**npx** is also a CLI tool whose purpose is to make it easy to install and manage dependencies hosted in the npm registry.

From <[*https://www.freecodecamp.org/news/npm-vs-npx-whats-the-difference/*](https://www.freecodecamp.org/news/npm-vs-npx-whats-the-difference/)>

npm install -g npx

From <[*https://www.freecodecamp.org/news/npm-vs-npx-whats-the-difference/*](https://www.freecodecamp.org/news/npm-vs-npx-whats-the-difference/)>

npx your-package

npx will check whether <command> or <package> exists in $PATH, or in the local project binaries, and if so it will execute i

From <[*https://www.freecodecamp.org/news/npm-vs-npx-whats-the-difference/*](https://www.freecodecamp.org/news/npm-vs-npx-whats-the-difference/)>

**Add Header component (Create with Skipping tests)**

**>**ng g c Header -skip-tests

**Update - app.component.html**

Remove content in app.component.html

Add the content

<div>   <app-header></app-header> </div>

<div>   <app-side-nav></app-side-nav> </div>

**And content to header component**

Navigate to <https://material.angular.io/> (Components), show 4 possible themes

Mention it has flexibility to customizing the themes

Navigate to "Toolbar" component - "Examples" tab, take tool bar with right side icons and css also

<mat-toolbar color="primary">

  <button mat-icon-button class="example-icon" aria-label="Example icon-button with menu icon">

    <mat-icon>menu</mat-icon>

  </button>

  <span>Angular Lab</span>

  <span class="example-spacer"></span>

    <button mat-icon-button routerLink="/register">

      <mat-icon>account\_box</mat-icon>

    </button>

    <button mat-icon-button routerLink="/login">

      <mat-icon>login</mat-icon>

    </button>

</mat-toolbar>

In case of any angular material not rendering errors - restart >ng serve

Add this in package.json (for scripts section) -

"postinstall": "ngcc"

Navigate to <https://material.io/resources/icons/?style=baseline>

For register and login icons

**Update app-routing.module.ts with routes**

import { ProductsComponent } from "./products/products.component";

import { OrdersComponent } from "./orders/orders.component";

import { RegisterComponent } from "./register/register.component";

import { LoginComponent } from "./login/login.component";

const routes: Routes = [

  {path: "products", component: ProductsComponent},

  {path: "orders", component: OrdersComponent},

  {path: "register", component: RegisterComponent},

  {path: "login", component: LoginComponent}

];

**Add content to SideNav component**

Navigate to <https://material.angular.io/> (Components) - "SideNav" component - "Examples" tab

Pick Sidenav open & Close behaviour

And clean the checkbox & Events controls and bring it as -

<mat-sidenav-container class="example-container">

  <mat-sidenav #sidenav mode="side">

    Sidenav content

  </mat-sidenav>

  <mat-sidenav-content>

    <p><button mat-button (click)="sidenav.toggle()">Toggle</button></p>

      main content

  </mat-sidenav-content>

</mat-sidenav-container>

Also add the css from Material site

Navigate to <https://material.angular.io/> (Components) - "NavList" component - "Examples" tab

Copy paste the Nav list code and customize

 <mat-nav-list>

      <a mat-list-item routerLink="/products"> Products </a>

      <a mat-list-item routerLink="/orders" > Orders </a>

   </mat-nav-list>

**Add content to products page**

Navigate to <https://material.angular.io/> (Components) - "Table" component - "Examples" tab

Take table with Filer code (All - HTML, CSS, TS)

**Bind table from http**

Add content to productsDetails page

**Add content to Orders page**

**Add content to Register page**

**Add content to Login page**