**BASIC TODO APPLICATION WITH ANGULAR MATERIAL**

* Create a angular project by running the command in the command line as:

**ng new <project name>**

* Run the command : **npm install**
* Run the command: **npm install –g @angular/cli**
* Now install angular material by running the command:

npm install –save @angular/material @angular/cdk

* For animations: run the following command:

npm install –save @angular/animations

all browsers do not support angular animations: so to make it supported by all the browsers add

**import {BrowserAnimationsModule} from ‘@angular/platform-browser/animations’;**

in the **app.module.ts** file

* Now import the components module.

import {MatToolbarModule,MatButtonModule,MatInputModule, MatTableModule,MatCardModule,MatIconModule} from '@angular/material';

* For gesture support install hammerjs by running the following command

**npm install –save hammer js**

* after installing import it in the **main.ts** file.
* To have icons ; add the following link in the **index.html** file.

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

* Open the editor Visual Studio Code and open the project folder in it
* Configure the following files as:

app.component.ts:

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

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent{

title="My Todos";

todos=[

{

label:'Bootstrap',

done:false,

priority:3

},

{

label:'Angular 2',

done:true,

priority:1

},

{

label:'React',

done:false,

priority:3

},

{

label:'Express',

done:false,

priority:5

}

];

addTodo(newTodoLabel){

var newTodo={

label: newTodoLabel,

priority: 1,

done: false

};

this.todos.push(newTodo);

}

deleteTodo(todo){

this.todos=this.todos.filter(t => t.label !== todo.label );

}

}

app.module.ts

import { BrowserModule } from '@angular/platform-browser';

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

import {BrowserAnimationsModule} from '@angular/platform-browser/animations';

import { AppComponent } from './app.component';

import {MatToolbarModule,MatButtonModule,MatInputModule, MatTableModule,MatCardModule,MatIconModule} from '@angular/material';

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule,

BrowserAnimationsModule,

MatToolbarModule,MatButtonModule,MatInputModule,MatTableModule,MatCardModule,MatIconModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

app.component.html:

<mat-toolbar style="background-color:blue">

<h1> {{title}} </h1>

</mat-toolbar>

<br><br>

<mat-card>

<mat-form-field>

<input matInput placeholder="Enter the task to be completed" mat-maxlength="40" #newTodo />

</mat-form-field>

<button mat-button color="primary" (click)="addTodo(newTodo.value)"><i class="material-icons">add\_circle</i></button><br><br>

</mat-card>

<mat-card>

<br>

<table>

<tr>

<th>Label</th>

<th>Priority</th>

<th>Done</th>

<th>Action</th>

</tr>

<tr \*ngFor="let todo of todos">

<td>{{todo.label}}</td>

<td>{{todo.priority}}</td>

<td>{{todo.done ? '&#10003;' : '&#10005;'}}</td>

<td>

<button mat-button color="warn" (click)="deleteTodo(todo)"><i class="material-icons">delete</i></button>

</tr>

</table>

</mat-card>

Index.html:

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>ToDoApplication</title>

<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="icon" type="image/x-icon" href="favicon.ico">

<link href="node\_modules/@angular/material/prebuilt-themes/deeppurple-amber.css" rel="stylesheet">

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

</head>

<body>

<app-root></app-root>

</body>

</html>

Main.ts:

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

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';

import { environment } from './environments/environment';

import 'hammerjs';

if (environment.production) {

enableProdMode();

}

platformBrowserDynamic().bootstrapModule(AppModule)

.catch(err => console.log(err));

**FINAL OUTPUT:**

