1. Installa **Angular Material**

Nel Terminal (git bush):

Ng add @angular/material

Chose: Indigo/Pink ; no; yes.

Se rilancio la direttiva dopo che è stata lanciata mi permette di cambiare il tema e le altre impostazioni.

In Angular14, tutti i passaggi che completano l’installazione vengono eseguiti in automatico

-Angular.json

In Angular14 lo style viene compilato in automatico.

"styles": [

              "./node\_modules/@angular/material/prebuilt-themes/deeppurple-amber.css",

Style: ./node\_modules/@angular/material/prebuilt-themes/indigo-pink.css

-Index.html

In automatico inserisce gli stili tra la riga 9 e 11. Cioè queste 3 righe.

<link rel="preconnect" href="https://fonts.gstatic.com">

  <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500&display=swap" rel="stylesheet">

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

-Styles.css:

In automatico ha inserito

html, body { height: 100%; }

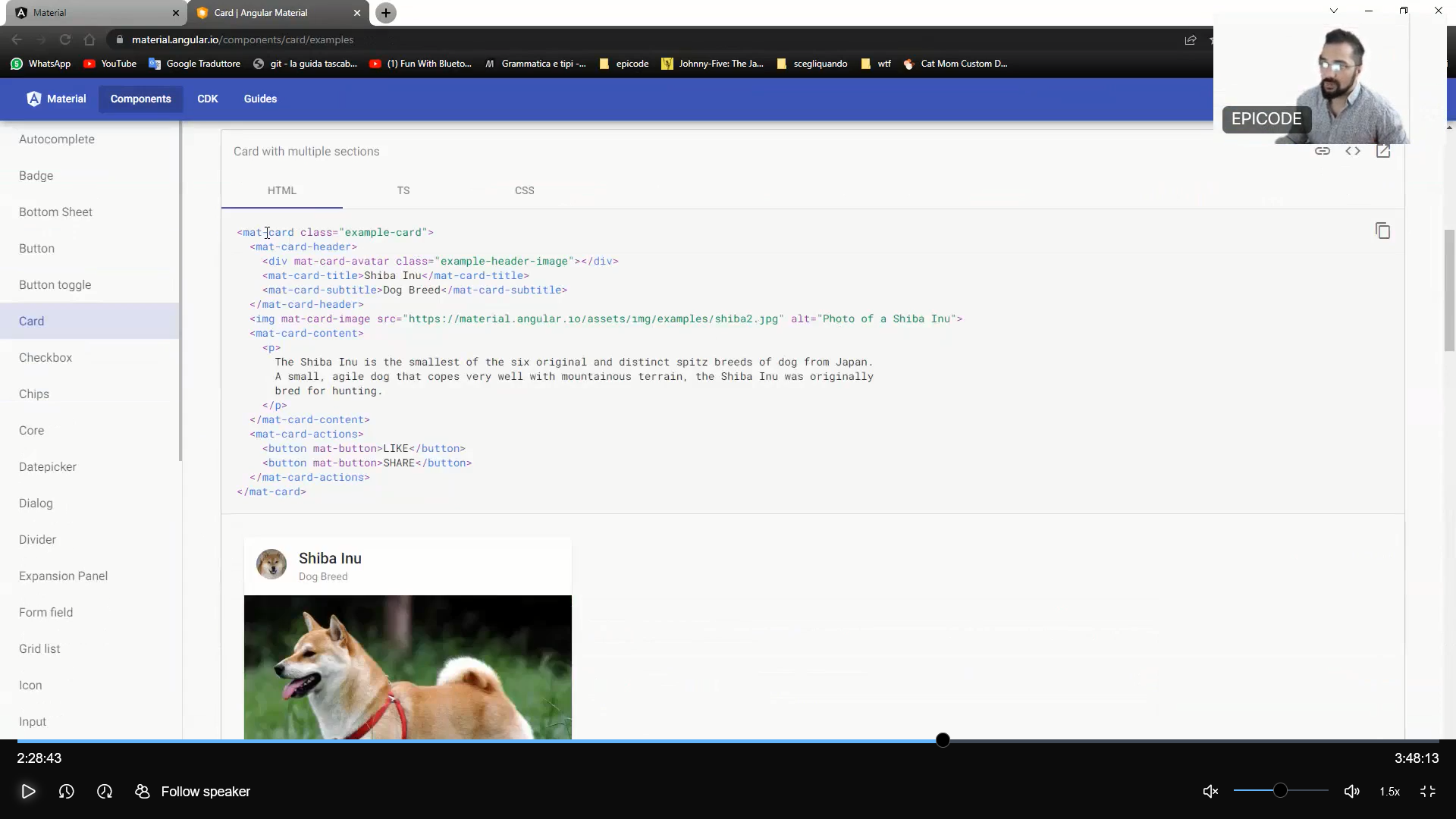
body { margin: 0; font-family: Roboto, "Helvetica Neue", sans-serif; }

-App.module.ts

In automatico ha aggiunto:

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

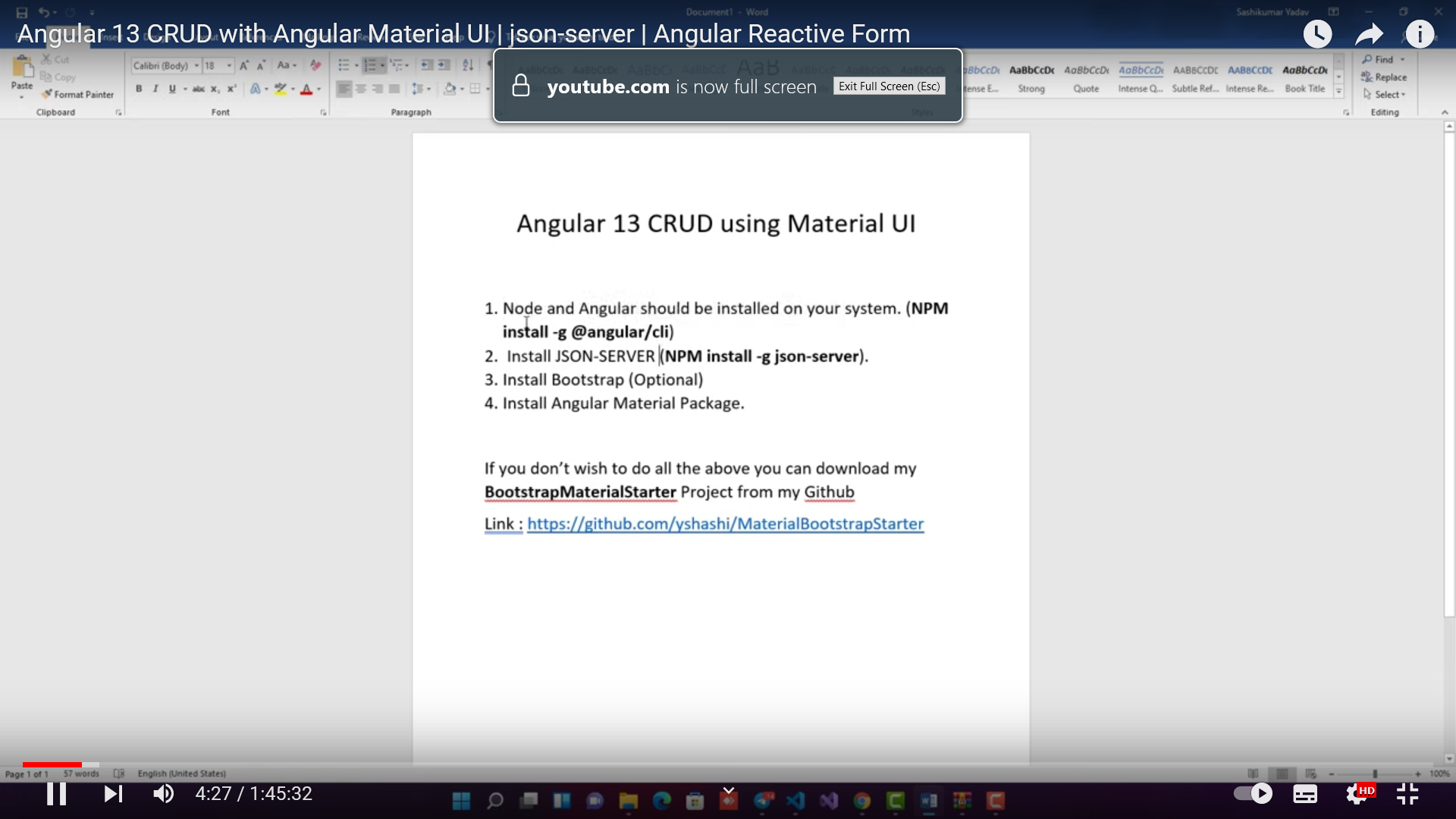
Esempio su come fare una card:



Per ciascun elemento c’è un html (con dei tag propri di angular material, un TS e un CSS).

Video youtube che sto seguendo per fare il progetto:

**Angular 13 CRUD with Angular Material UI | json-server | Angular Reactive Form**



**TOOLBAR:**

Su Angular Material scelgo una toolbar e vado a copiare l’html che trovo dentro al tasto che ha questo simbolo < >. Lo incollo su app.component.html. Dà subito 4 errori perché dobbiamo andare a fare degli import sul TS.

In app.module.ts importo: import {MatToolbarModule} from '@angular/material/toolbar';

Occorre anche importatre delle icone, quindi andiamo nella sezione icon; API; e copio import {MatIconModule} from '@angular/material/icon';sempre in app.module.ts. Nello stesso file ricordiamoci di copiare negli imports anche :

 MatToolbarModule,

    MatIconModule

I bottoni appaiono bruttini, perché dobbiamo andare ad inserire anche l’import sui bottoni.

Copio il CSS per sistemare la posizione dei bottoni.

Per inserire un’icona che desidero , la scelgo su “mat-icon list” e incollo nell’html.

<mat-icon>supervised\_user\_circle</mat-icon>

Cancello i bottoni che non mi servono e incollo quello che mi piace , scelto dalla sezione dei bottoni.

**Apertura Finestra di dialogo al click**

Per far si che al click del bottone “Add new Post” si apra una finestra di dialogo che mi permetta di inserire i dati del nuovo post, nel tag del bottone inserisco il metodo <button mat-button (click)="openDialog()”.

-Creiamo un componente “dialog” dando sul terminale ng g c dialog.

-App.component.ts qui incollo il metodo "openDialog()” che copio da Angular Material, aggiungo anche l’import e creo il constructor..

Ho modificato la app.component.html mettendo un menù con i collegamenti home e blog .

Ho creato i corrispettivi component ng g c home e ng g c blog.

La finestra di dialogo si apre dal blog cliccando su Add new Post.

Sulla finestra di dialogo vado a creare il form. Copio l’html, importo l’API nel app.module.ts.

-Nel dialog.component.ts ho aggiunto

import { FormGroup, FormBuilder, Validators } from '@angular/forms';

e inserito una formBuilder su ngOnInit, poi inserito i formControlName=”” sui campi di input.

Db.json

(Se non avessi già il json-server installato, su un nuovo terminale lancerei: json-server –watch db.json. Ma io ce l’ho già installato).

Creo un file db.json

Creiamo un API service.

$ ng g s services/api

1h10min