# Using Material 3 (M3) with Angular

# Dharmen Shah

I am a Developer, Blogger and Open Source Contributor

You can find me on twitter @shhdharmen

SOLVATIVE

*Solve Forward*

“Innovation made Human”

For more information

https://solvative.com/

# Never left it Alone!

- Blogs
- Open Source Libraries
- Helping other developers

1000+ GitHub Stars

# Agenda


User Experience


Material Design


Angular Material


M3 with Angular Material

# User Experience

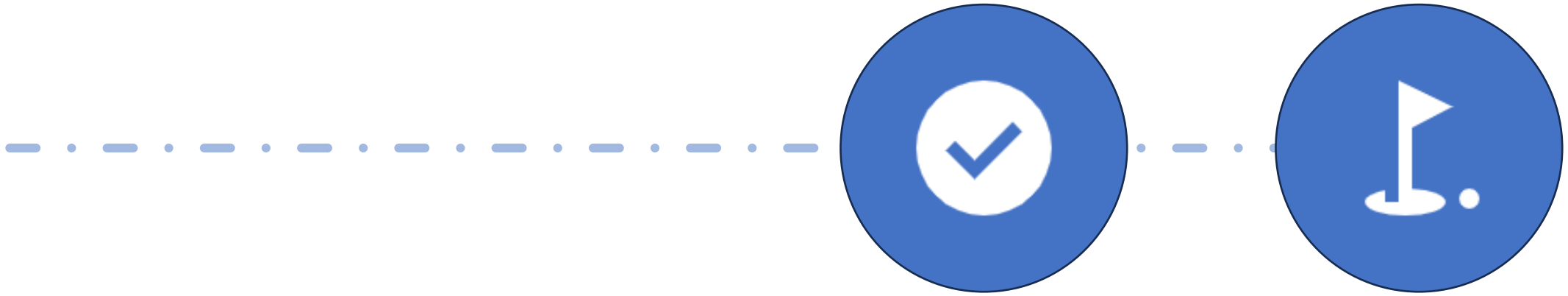
“User experience is the summary of everything”.


User experience encompasses all aspects of the end user's interaction

# Then what is User Interface?

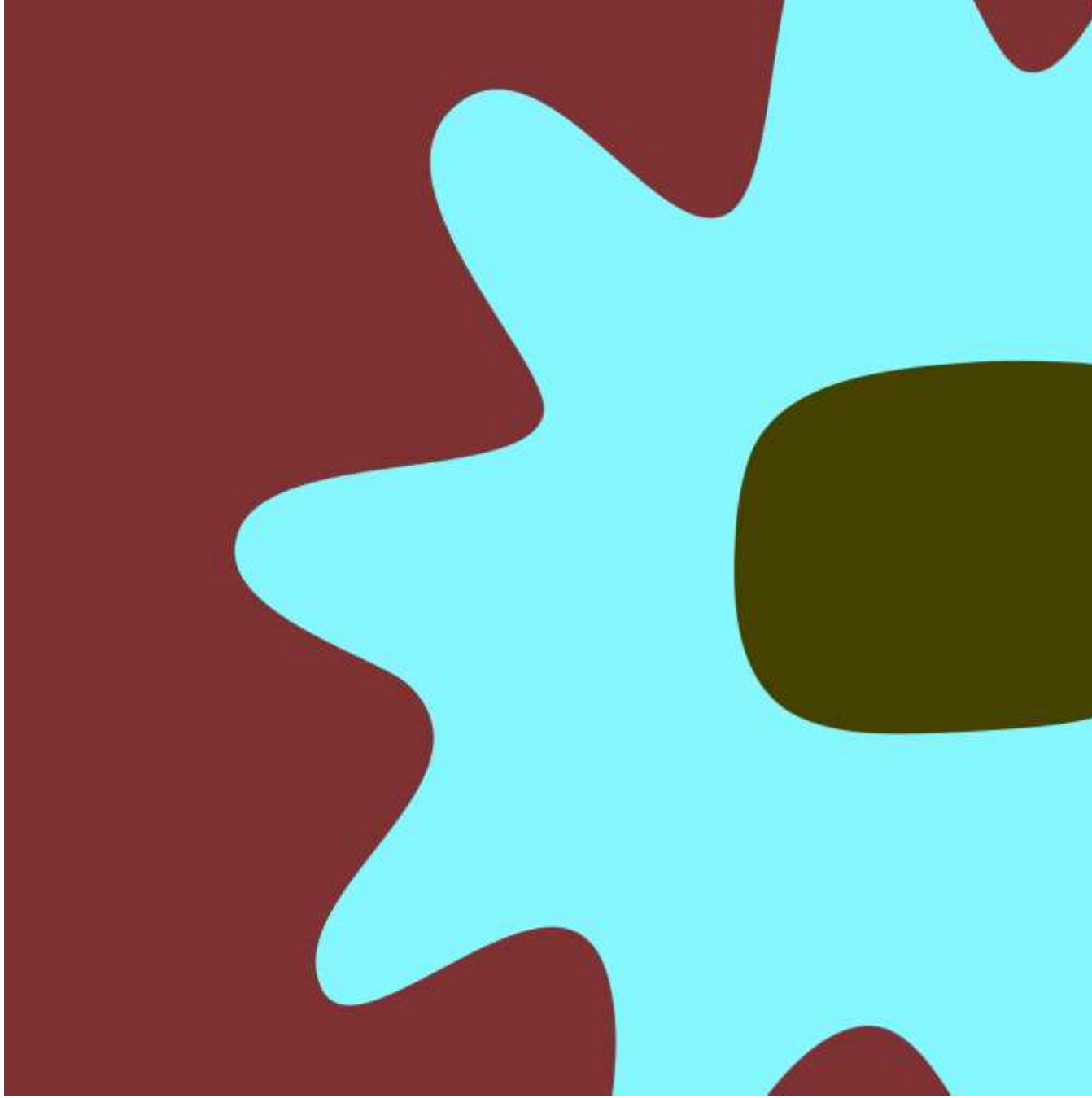
- Visual touchpoint or asset the user interacts with
- Color, spacing, grids, icons, and buttons
- Shape, decoration, and presentation



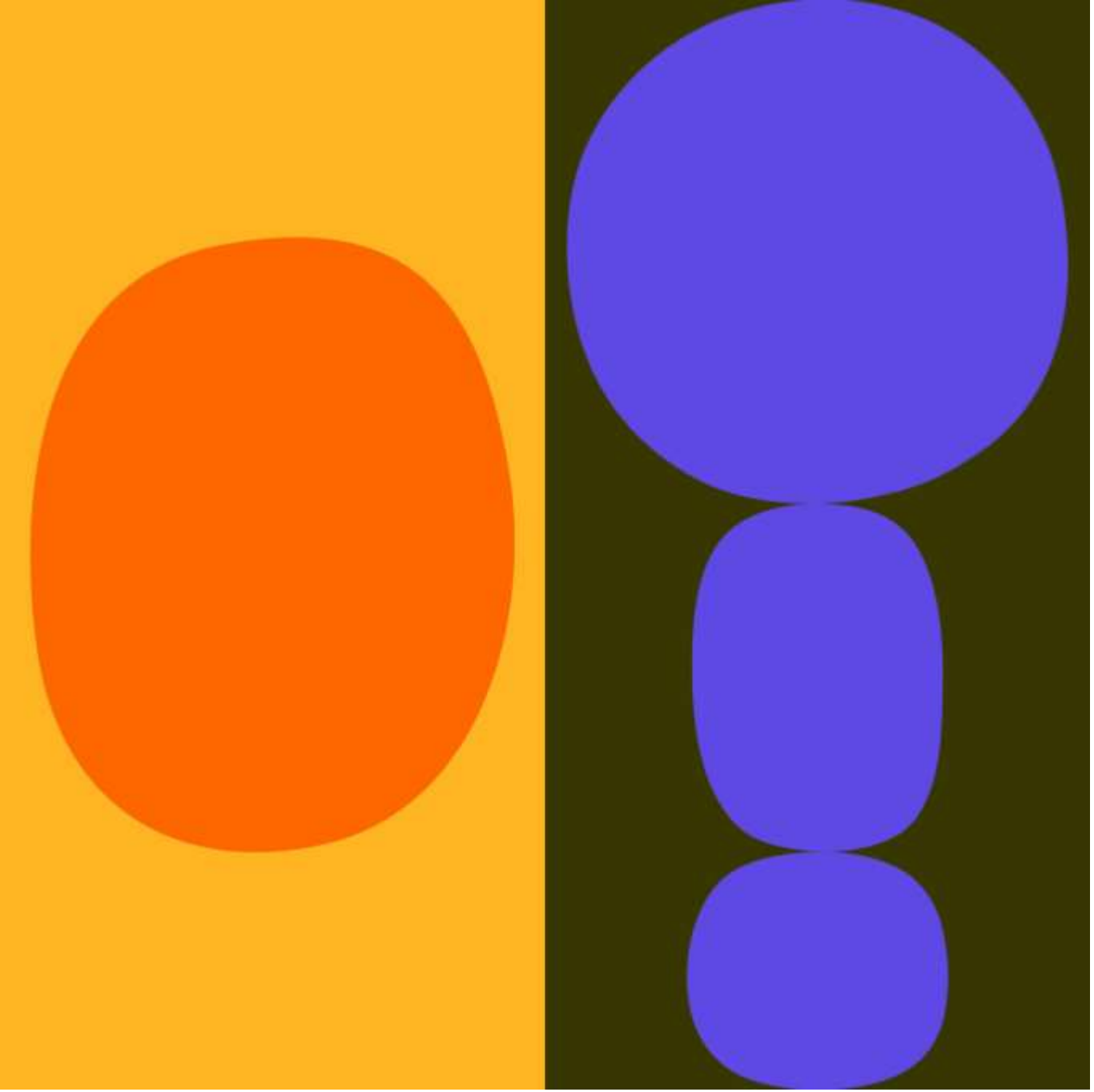

Through
User Interface
Enhance
User Experience

# Material Design

Material 3 is the latest version of Google’s open-source design system. Design and build beautiful, usable products with Material 3.



Video from Material Design



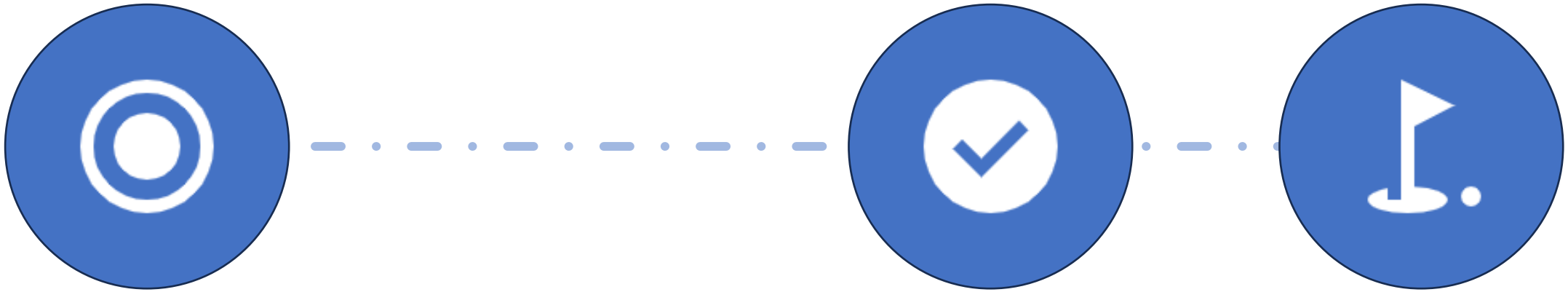
- Increased focus on personalization

- Updated theming

- New & improved components

- Emphasis on motion and responsiveness

Video from Foundations — Material Design 3



Using Material Design

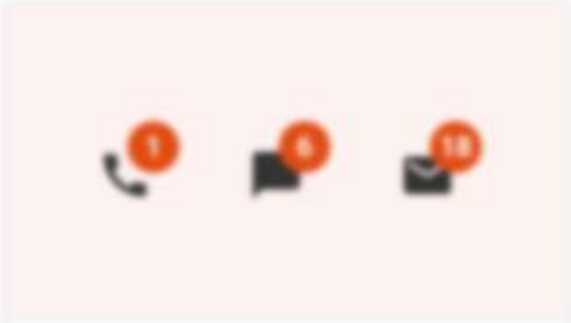
Through User Interface

Enhance User Experience

# Angular Material


## Autocomplete

Suggests relevant options as the user types.




## Badge

A small value indicator that can be overlaid on another object.


## Bottom Sheet

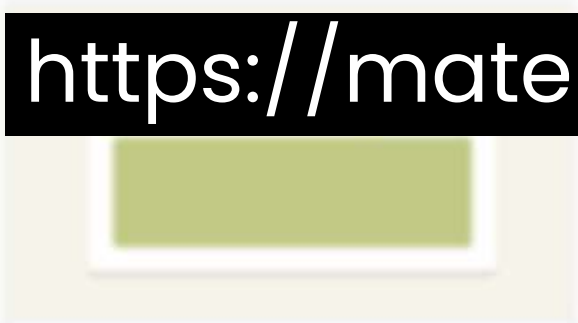
A large interactive panel primarily for mobile devices.


## Button

An interactive button with a range of presentation options.

## Button toggle

A groupable on/off toggle for enabling and disabling options.



## Card

A styled container for pieces of itemized content.


## Checkbox

Captures boolean input with an optional indeterminate mode.


## Chips

Presents a list of items as a set of small, tactile entities.



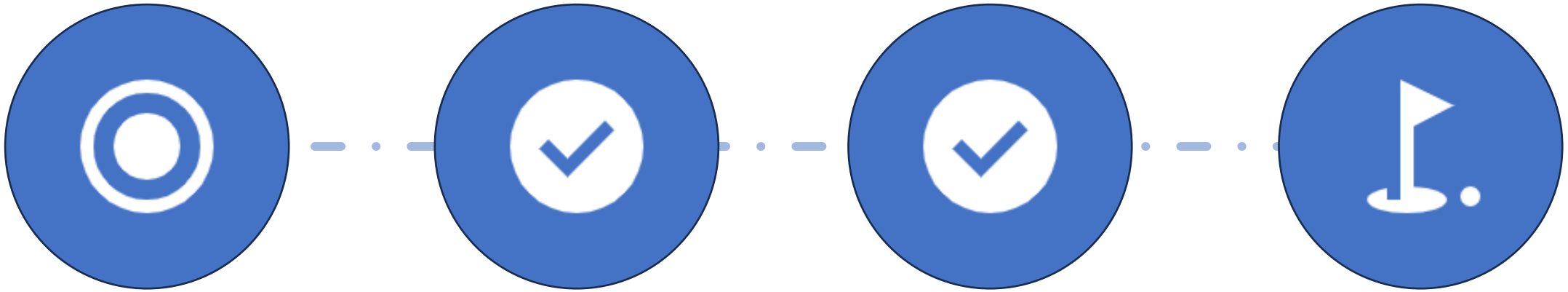
https://material.angular.io/

# Angular Material

- Material Design components for Angular
- High quality, versatile and frictionless
- Customizing base, color, typography and density
- APIs built with SASS (https://sass-lang.com/)
- Easy to setup thanks to schematics

# M3 in Angular Material

- ✋ As of v17.2.0, Angular Material includes **experimental support** for M3 styling in addition to M2
- M3 is implemented in Angular Material as an alternate Sass theme


Using
Material 3 Design
Through
Angular Material
Through
User Interface
Enhance
User Experience

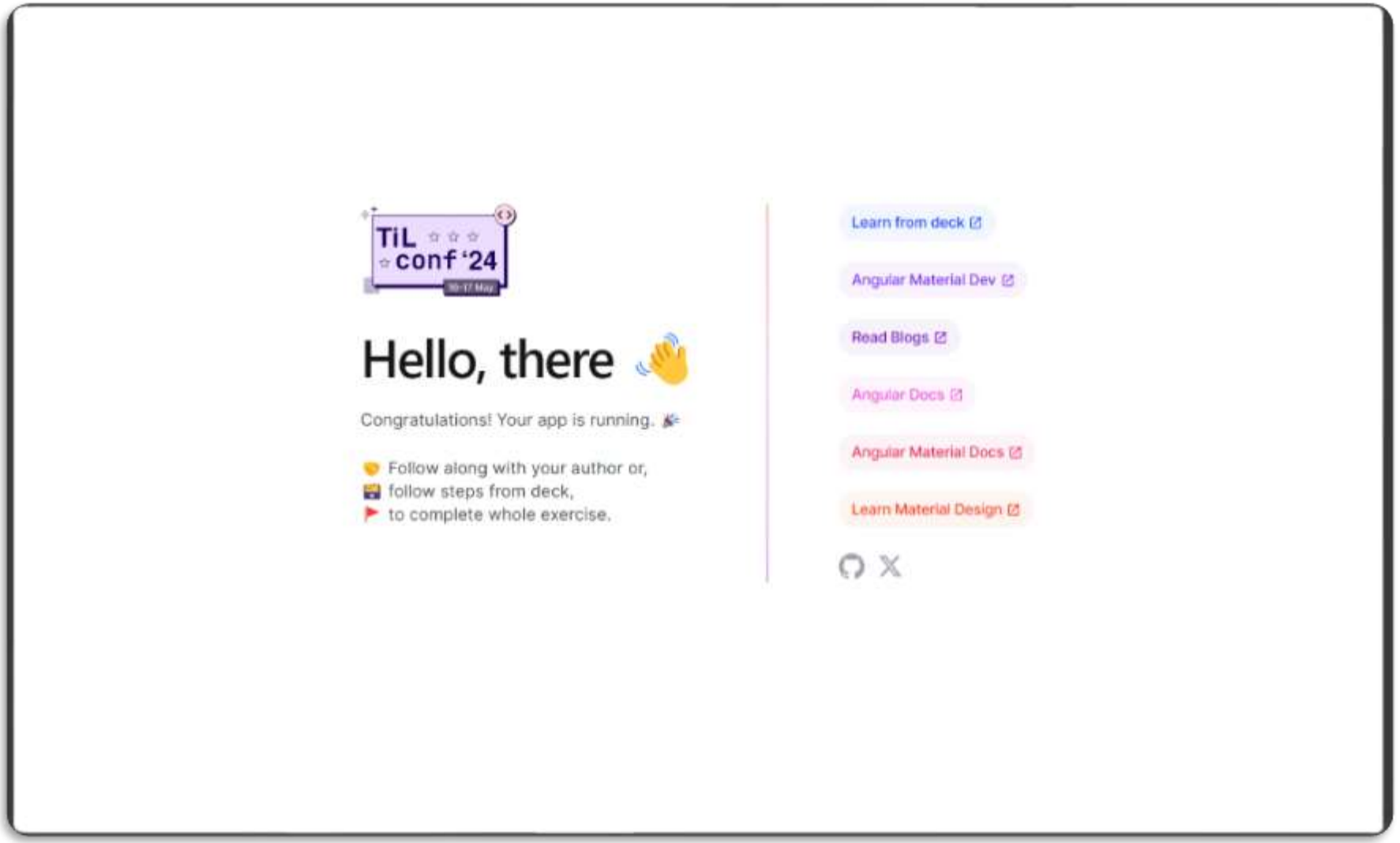
# Time to code

- Clone the starter
- Or Create new Angular project
  - With scss
  - Install dependencies
    - @angular/material
    - @angular/cdk
    - @angular/material-experimental
- Setup M3 theme
- Create application skeleton
  - Layout with sidenav
- Create pages
- Create theme manager

# Clone the starter

```
> git clone
https://github.com/shhdharm
en/tilconf-m3-angular.git -
-branch at-installed-deps
> cd tilconf-m3-angular
> npm i
> npm start
```

# Output after cloning and running



# Create new Angular project

```
> ng new m3-angular --defaults --styles scss
> cd m3-angular
```

# Install dependencies

```
> ng add @angular/material
? Choose a prebuilt theme name, or "custom"
for a custom theme: Custom
? Set up global Angular Material typography
styles? Yes
? Include the Angular animations module?
Include and enable animations
```

# Install dependencies

```
> npm i @angular-material/experimental
```

# Time to code

- ~~Clone the starter~~
- ~~Or Create new Angular project~~
  - ~~With scss~~
  - ~~Install dependencies~~
    - ~~@angular/material~~
    - ~~@angular/cdk~~
    - ~~@angular/material-experimental~~
- Setup M3 theme
- Create application skeleton
  - Layout with sidenav
- Create pages
- Create theme manager

# Setup M3 Theme

1 Use material-experimental

```
@use "@angular/material-experimental" as matx;
```

2 Create M3 theme

```
$light-theme: matx.define-theme(
  (
    color: (
      primary: matx.$m3-azure-palette,
      tertiary: matx.$m3-blue-palette,
    ),
  )
);
$dark-theme: matx.define-theme(
  (
    color: (
      theme-type: dark,
      primary: matx.$m3-magenta-palette,
      tertiary: matx.$m3-violet-palette,
    ),
  )
);
```

3 Use M3 theme

```
html {
  @include mat.all-component-themes($light-
theme);
  background: mat.get-theme-color($light-theme,
surface);

  &.dark {
    @include mat.all-component-colors($dark-
theme);
    background: mat.get-theme-color($dark-theme,
surface);
  }
}
```

# Time to code

- ~~Clone the starter~~
- ~~Or Create new Angular project~~
  - ~~With scss~~
  - ~~Install dependencies~~
    - ~~@angular/material~~
    - ~~@angular/cdk~~
    - ~~@angular/material-experimental~~
- ~~Setup M3 theme~~
- Create application skeleton
  - Layout with sidenav
- Create pages
- Create theme manager

# Create application skeleton

1. Use navigation schematics to generate layout

```
> ng generate @angular/material:navigation core/layout
```

2. Add content project in layout

```
<!-- Add Content Here -->
<ng-content></ng-content>
```

# Use layout component

1 Import layout in app component

```
import { LayoutComponent } from
'./core/layout/layout.component';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet, LayoutComponent],
  templateUrl: './app.component.html',
 })
export class AppComponent {}
```

2 Use layout in app template

```
<app-layout>
  <router-outlet></router-outlet>
</app-layout>
```

# Output after using layout

# Apply theme to layout

1 New file _layout.component.theme.scss

```scss
@use "@angular/material" as mat;

@mixin theme($theme) {
  .sidenav {
    background-color: mat.get-theme-color($theme, surface-bright);
  }
}
```

2 Use layout-theme in styles

```scss
@use "./app/core/layout/layout.component.theme" as layout-theme;

html {
  @include layout-theme.theme($light-theme);

  &.dark {
    @include layout-theme.theme($dark-theme);
  }
}
```

For all possible roles, visit Reading color roles

# Output after theming layout

# Time to code

- ~~Clone the starter~~
- ~~Or Create new Angular project~~
  - ~~With scss~~
  - ~~Install dependencies~~
    - ~~@angular/material~~
    - ~~@angular/cdk~~
    - ~~@angular/material-experimental~~
- ~~Setup M3 theme~~
- ~~Create application skeleton~~
  - ~~Layout with sidenav~~
- Create pages
- Create theme manager

# Create Pages

```
> ng generate @angular/material:dashboard dashboard
> ng generate @angular/material:address-form address-form
> ng generate @angular/material:table table
> ng generate @angular/material:tree tree
> ng generate @angular/cdk:drag-drop drag-drop
```

# Update routes

```
import { Routes } from '@angular/router';

export const routes: Routes = [
  {
    path: '',
    pathMatch: 'full',
    redirectTo: 'dashboard',
  },
  {
    path: 'dashboard',
    loadComponent: () =>
      import('./dashboard/dashboard.component').then(
        (c) => c.DashboardComponent
      ),
      title: 'Dashboard'
  },
];
```

# Updates in layout

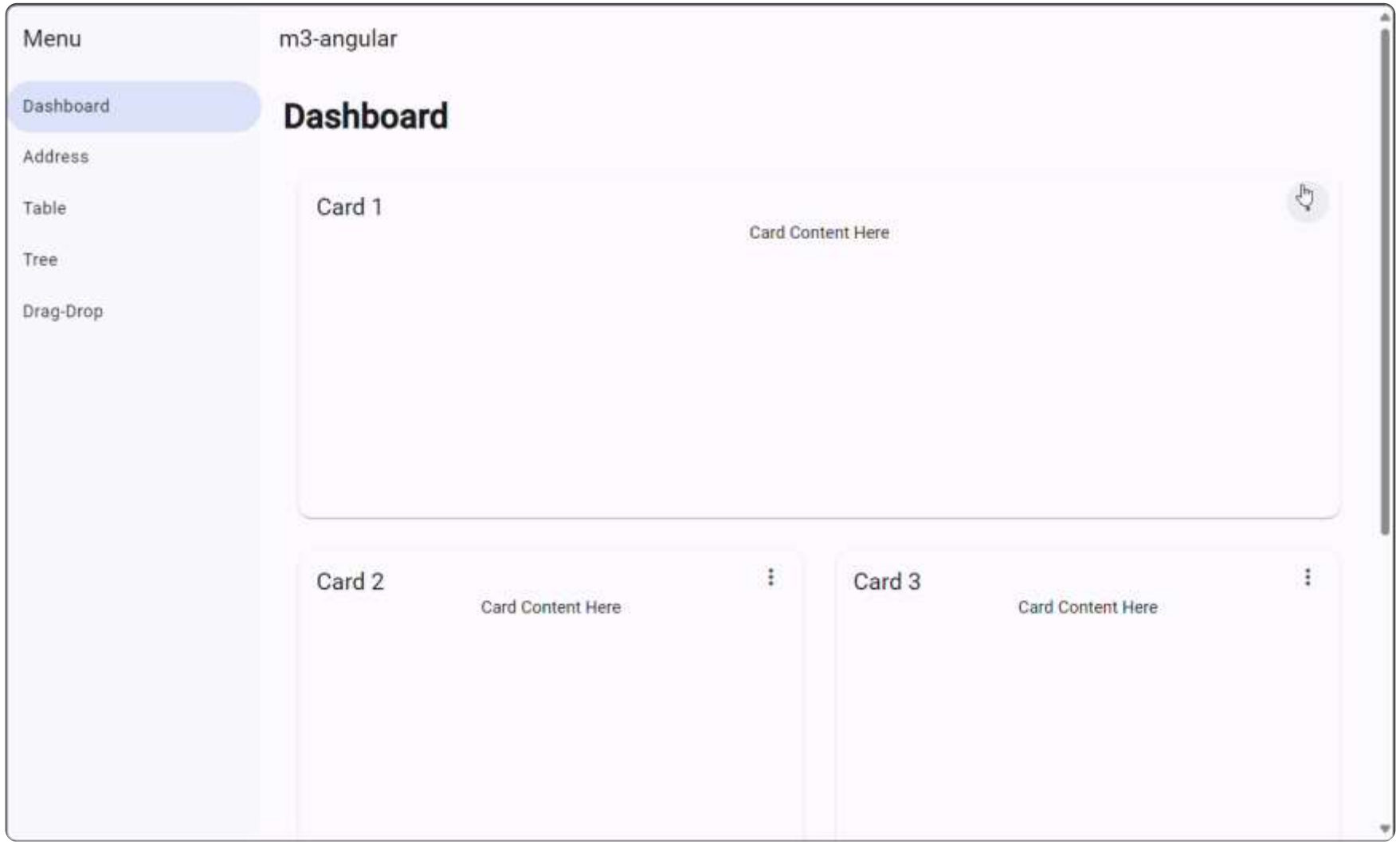
1 Use routes in layout component

```
@Component({
  selector: 'app-layout',
  imports: [
    RouterLink,
    RouterLinkActive
  ]
})
export class LayoutComponent {
  rootRoutes = routes.filter(r=>r.path);
}
```

2 Update layout template

```
<mat-nav-list>
  @for (item of rootRoutes; track $index) {
    <a
      mat-list-item
      [routerLink]="item.path"
      #link="routerLinkActive"
      routerLinkActive
      [activated]="link.isActive"
    >
      {{ item.title }}
    </a>
  }
</mat-nav-list>
```

# Output after pages and routes



# Time to code

- ~~Clone the starter~~
- ~~Or Create new Angular project~~
  - ~~With scss~~
  - ~~Install dependencies~~
    - ~~@angular/material~~
    - ~~@angular/cdk~~
    - ~~@angular/material-experimental~~
- ~~Setup M3 theme~~
- ~~Create application skeleton~~
  - ~~Layout with sidenav~~
- ~~Create pages~~
- Create theme manager

# Create Theme manager

1 Create theme-manager.service.ts

```
export type Theme = 'light' | 'dark';

@Injectable({providedIn: 'root'})
export class ThemeManagerService {
  theme = signal<Theme>('light');

  toggleTheme() {
    this.theme.update((value) => {
      return value === 'light' ? 'dark' : 'light';
    });
  }
}
```

2 Update document’s class

```
private _document = inject(DOCUMENT);

  constructor() {
    effect(() => {
      if (this.theme() === 'dark') {
        this._document.documentElement.classList.add('
dark');
      } else {
        this._document.documentElement.classList.remov
e('dark');
      }
    });
  }
```

# Use Theme manager

1 Inject theme manager in layout

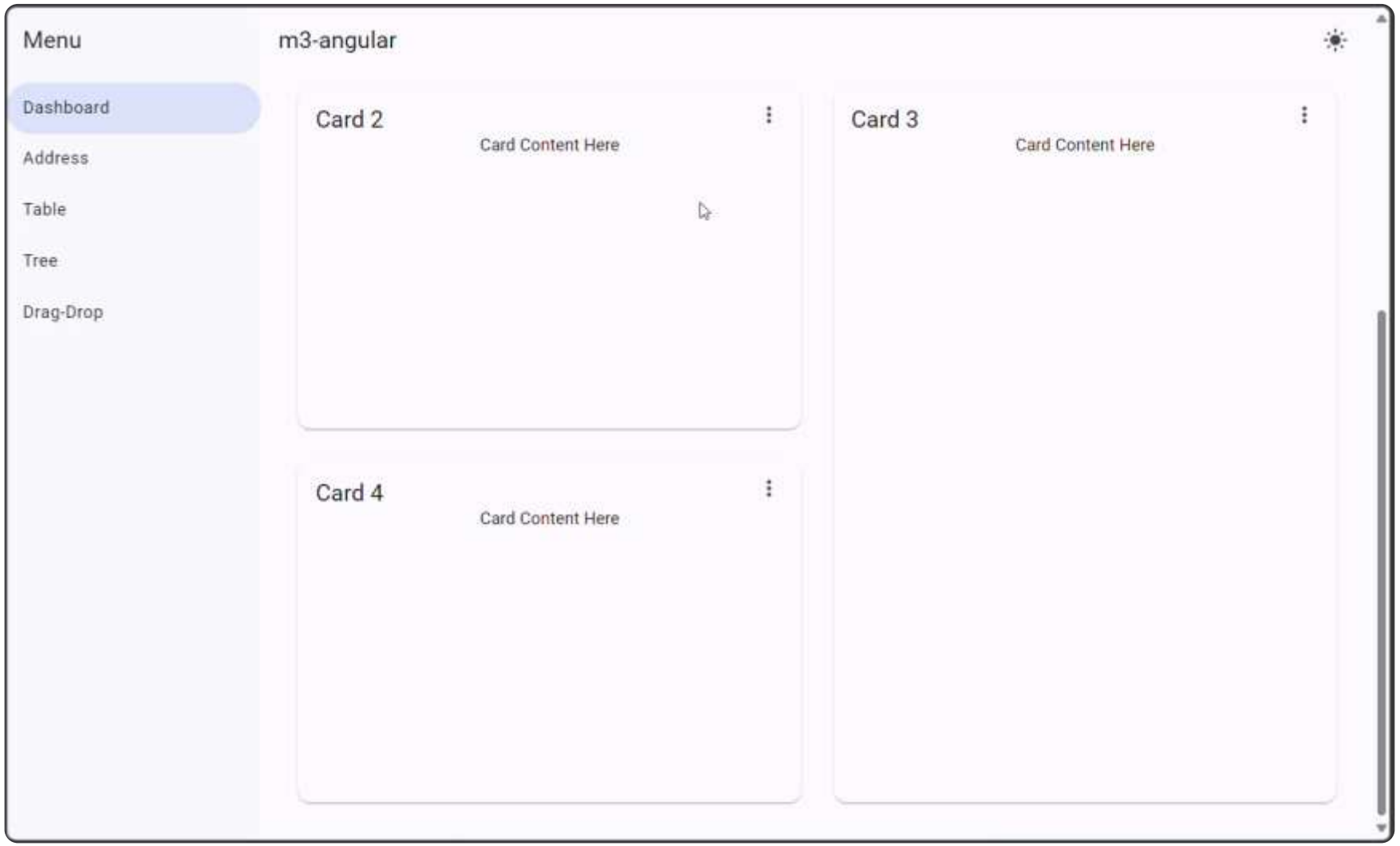
```
private themeManager = inject(ThemeManagerService);
    theme = this.themeManager.theme;

    toggleTheme() {
      this.themeManager.toggleTheme();
    }
```

2 Add theme switch in layout template

```
<button
        class="theme-switch"
        type="button"
        aria-label="Toggle theme"
        mat-icon-button
        (click)="toggleTheme()"
        matTooltip="Toggle theme"
      >
        <mat-icon>
          {{ theme() === "light" ? "light_mode" :
"dark_mode" }}
        </mat-icon>
      </button>
```

# Output after theme manager



# Quick recap

1. UX vs UI
2. Material Design
3. Angular Material

4. Using Material 3
    - Cloning the starter with dependencies
    - Creating & using M3 theme
    - Creating layout skeleton
    - Creating pages
    - Create theme manager

# https://angular-material.dev

One place for everything related to Material Design in Angular

# Q&A

# Thank you!

I am available online

https://twitter.com/**shhdharmen**

https://www.linkedin.com/in/**shhdharmen**

https://github.com/**shhdharmen**

Code and deck is available at

**https://github.com/shhdharmen/tilconf-m3-angular**