

Updating Unit Tests

In order to keep your unit tests running, you will need to import **MaterialModule** to any component's **spec** file that uses Angular material:

```
*.component.spec.ts
```

```
...
```

```
  beforeEach(
```

```
    async(() => {
```

```
      TestBed.configureTestingModule({
```

```
        ...
```

```
        imports: [..., MaterialModule, NoopAnimationsModule],
```

```
      }).compileComponents()
```

```
    })
```

```
  )
```



Updating unit tests

You will also need to update any test, including e2e tests, that search for a particular HTML element.

For example, since the app's title, LocalCast Weather, is not in an `h1` tag anymore, you must update the `spec` file to look for it in a `span` element:

src/app/app.component.spec.ts

```
expect(compiled.querySelector('span').textContent).toContain('LocalCast Weather')
```

Similarly, in e2e tests, you will need to update your page object function to retrieve the text from the correct location:

e2e/app.po.ts

```
getParagraphText() {  
    return element(by.css('app-root mat-toolbar span')).getText()  
}
```



Custom themes

1. Create a new file under `src` called `localcast-theme.scss`
2. Material theme guide, located at <https://material.angular.io/guide/theming>, includes an up-to-date starting file. I'll break down the contents of the file further
3. Start by including the base theming library:

`src/localcast-theme.scss`

```
@import '~@angular/material/theming';
```

4. Import the `mat-core()` mixin, which includes all common styles used by various Material components:

`src/localcast-theme.scss`

```
@include mat-core();
```



Custom Themes

<https://material.io/collections/color/#>

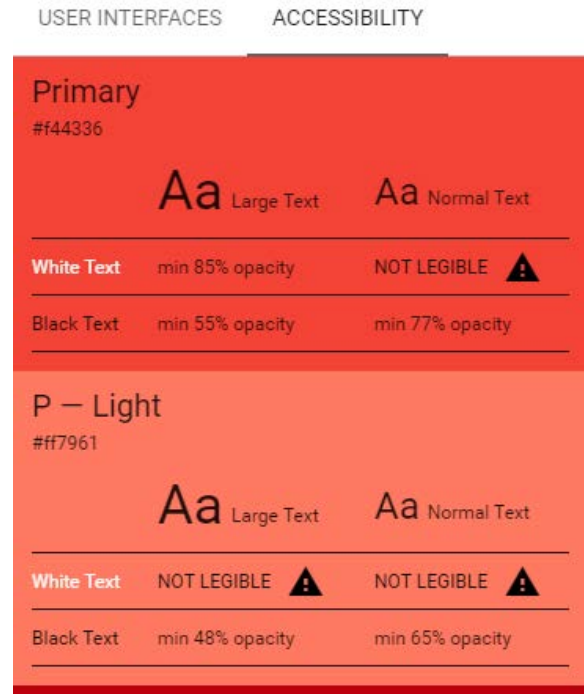
5. Using Material Palette, select a **Primary** and a **Secondary** color:

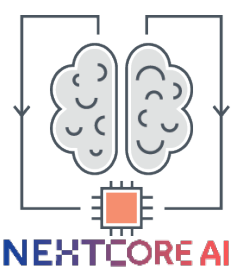
Primary selection is red with a hue value of 500

Secondary selection is indigo with a hue value of A400

6. Observe how your selections would apply to a material design app by going through the 6 prebuilt screen on the left

7. Evaluate the accessibility implications of your selections, as shown:





Mat-palette

```
mat-palette($base-palette, $default: 500, $lighter: 100, $darker: 700)
```

8. Define the **primary** and **secondary** **mat-palette** objects using the default hue from the tool:

```
src/localcast-theme.scss
```

```
$localcast-primary: mat-palette($mat-red, 500);
```

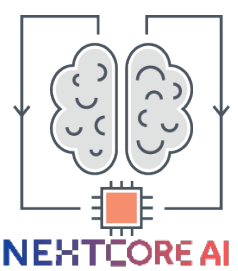
```
$localcast-accent: mat-palette($mat-indigo, A400);
```

9. Create a new theme and apply it:

```
src/localcast-theme.scss
```

```
$localcast-app-theme: mat-light-theme($localcast-primary, $localcast-accent);
```

```
@include angular-material-theme($localcast-app-theme);
```



Styles.input

10. In `angular.json`, locate the `apps.styles` attribute
11. Prepend the list `localcast-theme.scss` while removing the `styles.input` attribute

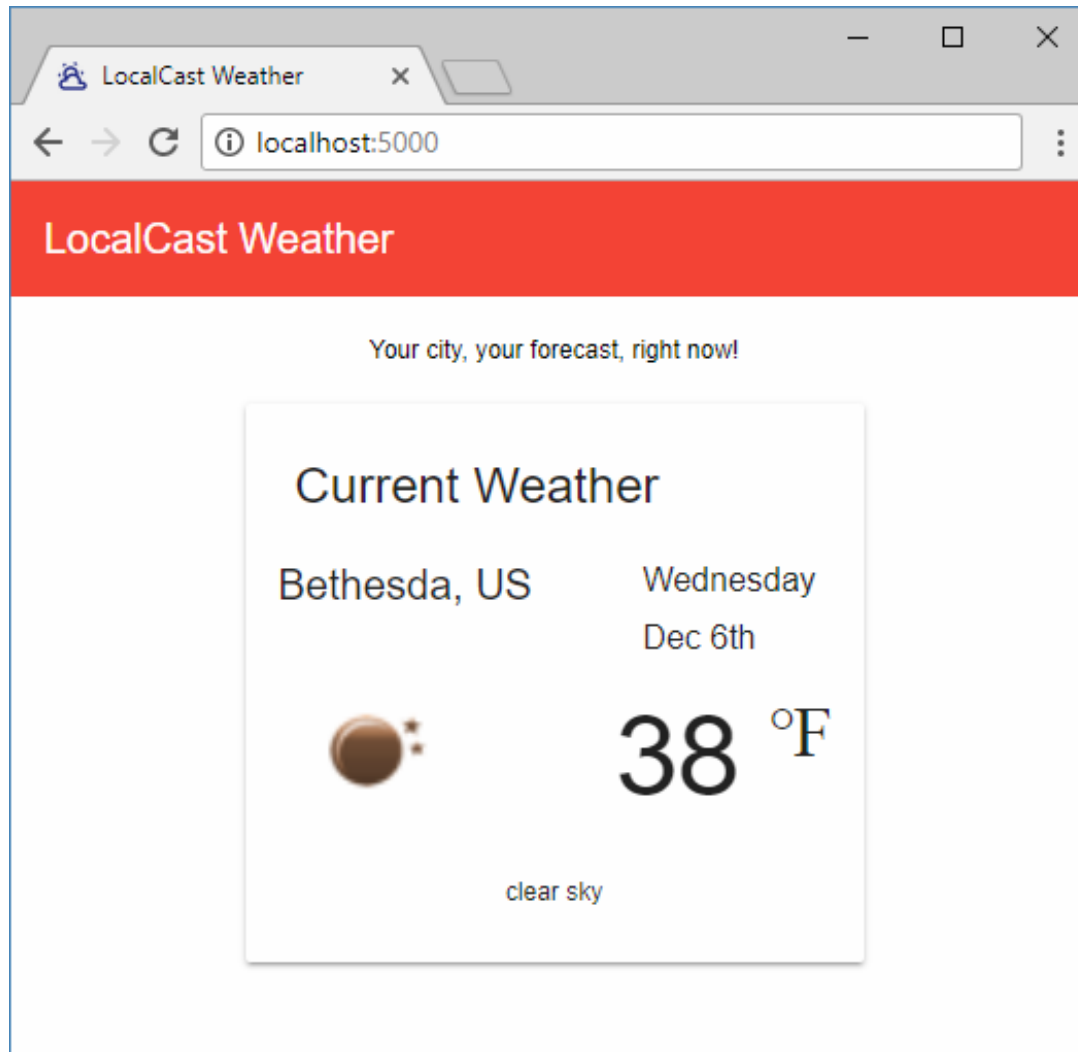
`angular.json`

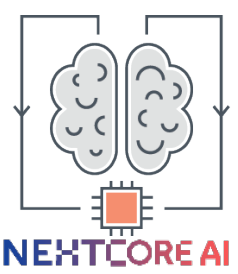
```
...  
"styles": [  
  "src/localcast-theme.scss",  
  "src/styles.css"  
],  
...
```



Final look

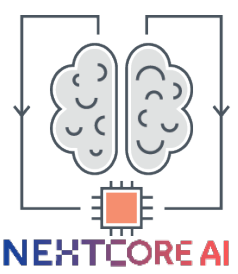
- Application should look like this:





More material

<http://mcg.mbitson.com>



we
BUILD