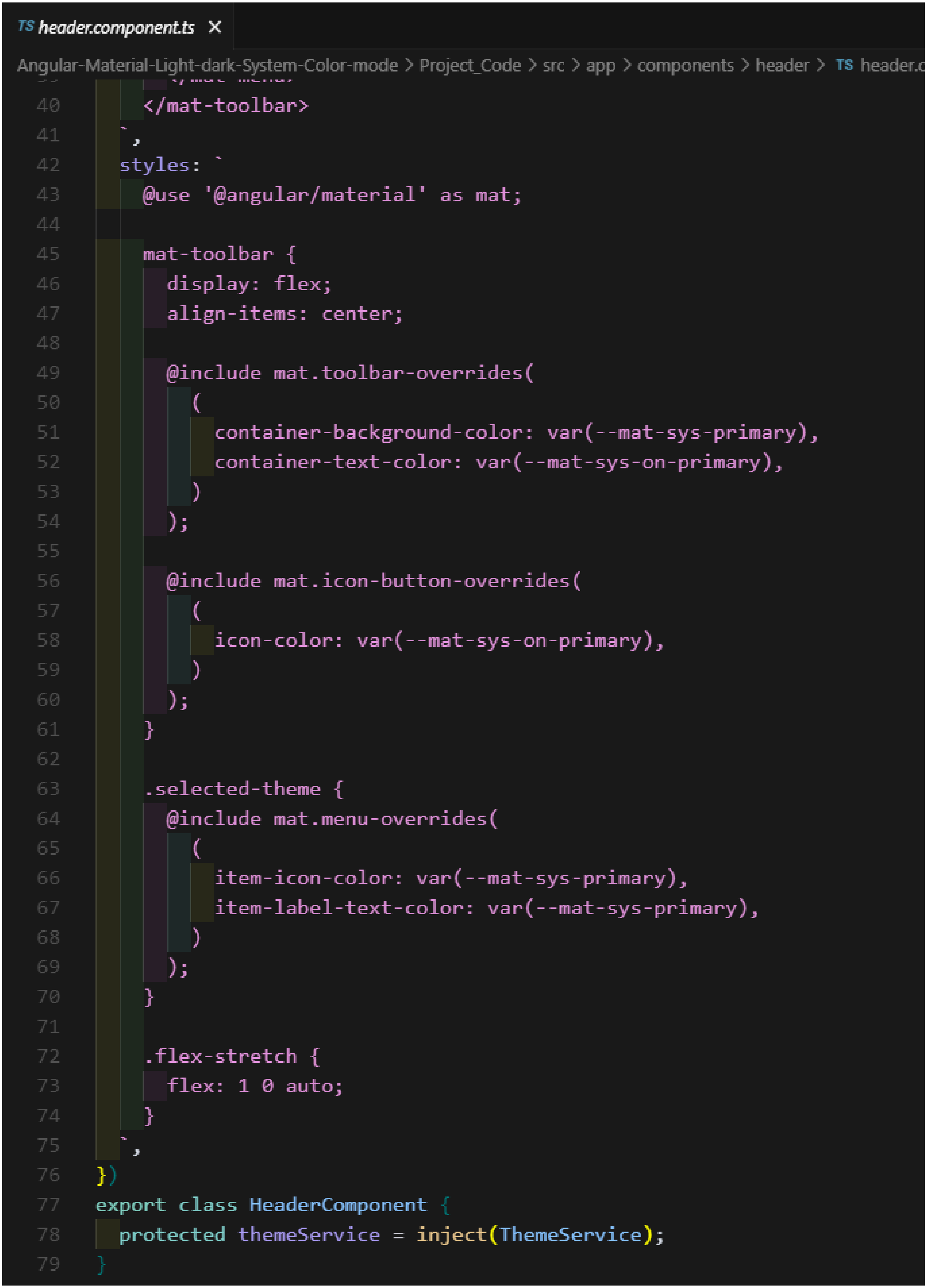
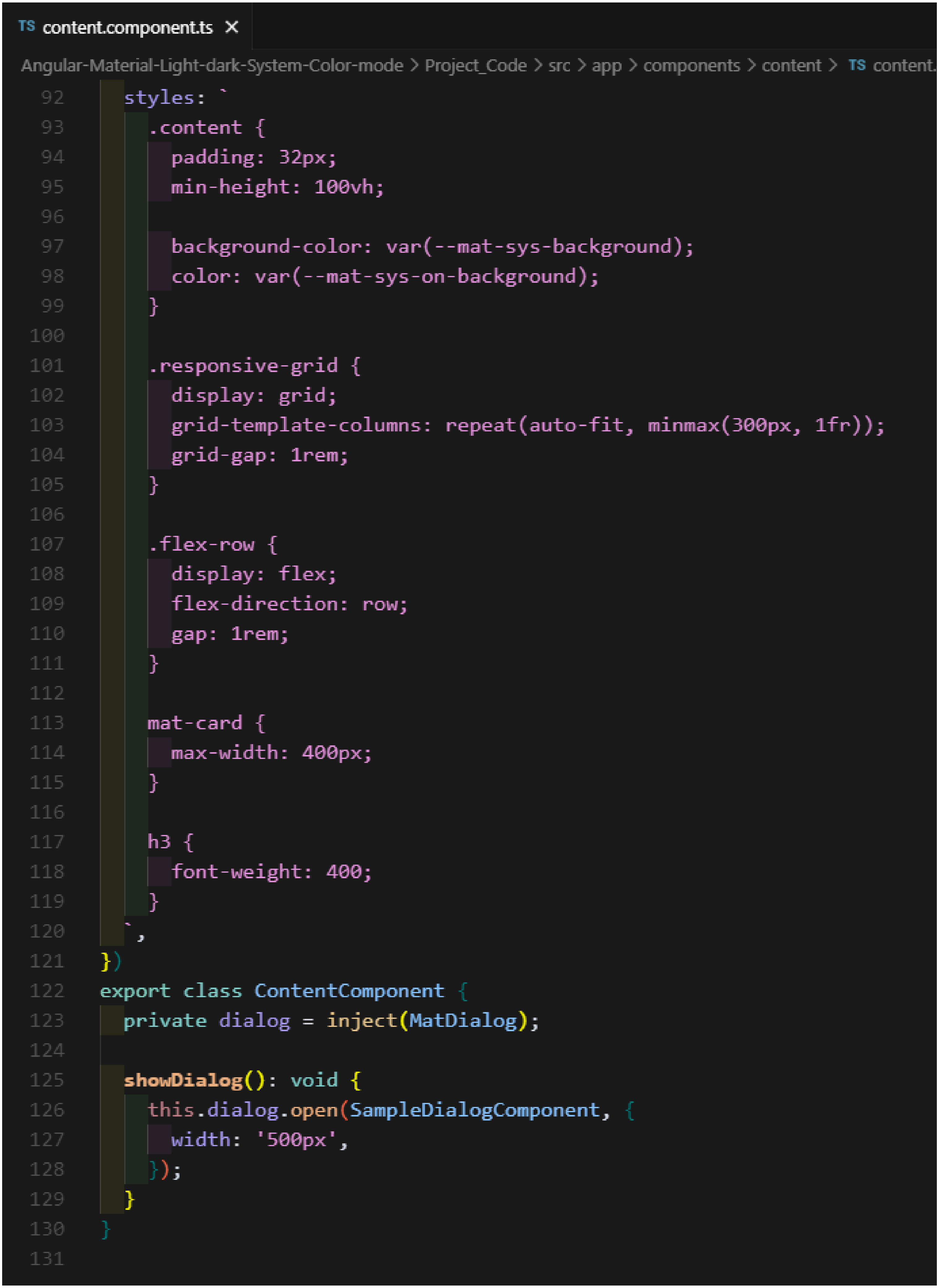
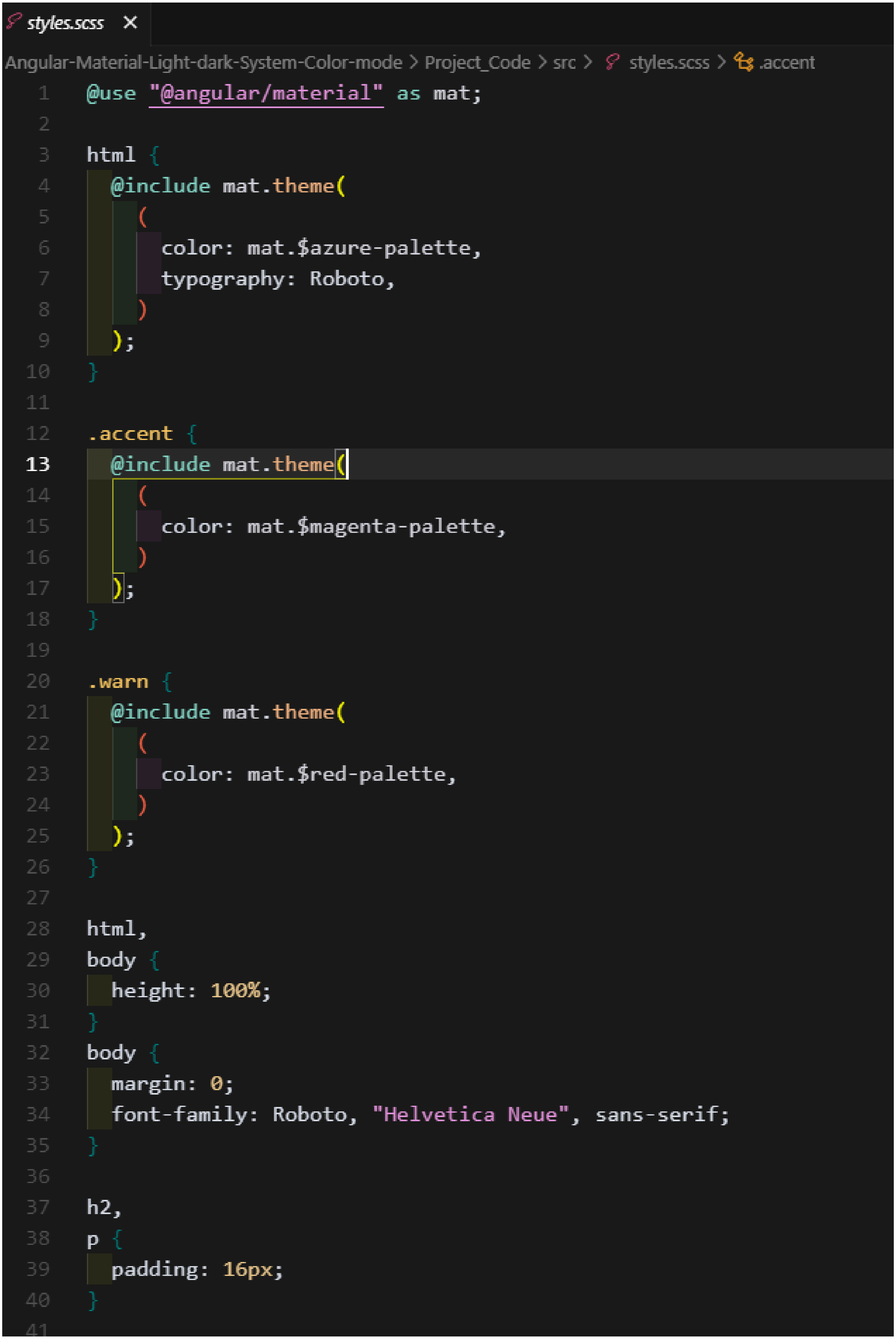

TS header.component.ts ×
Angular-Material-Light-dark-System-Color-mode > Project_Code > src > app > components > header > TS header.c

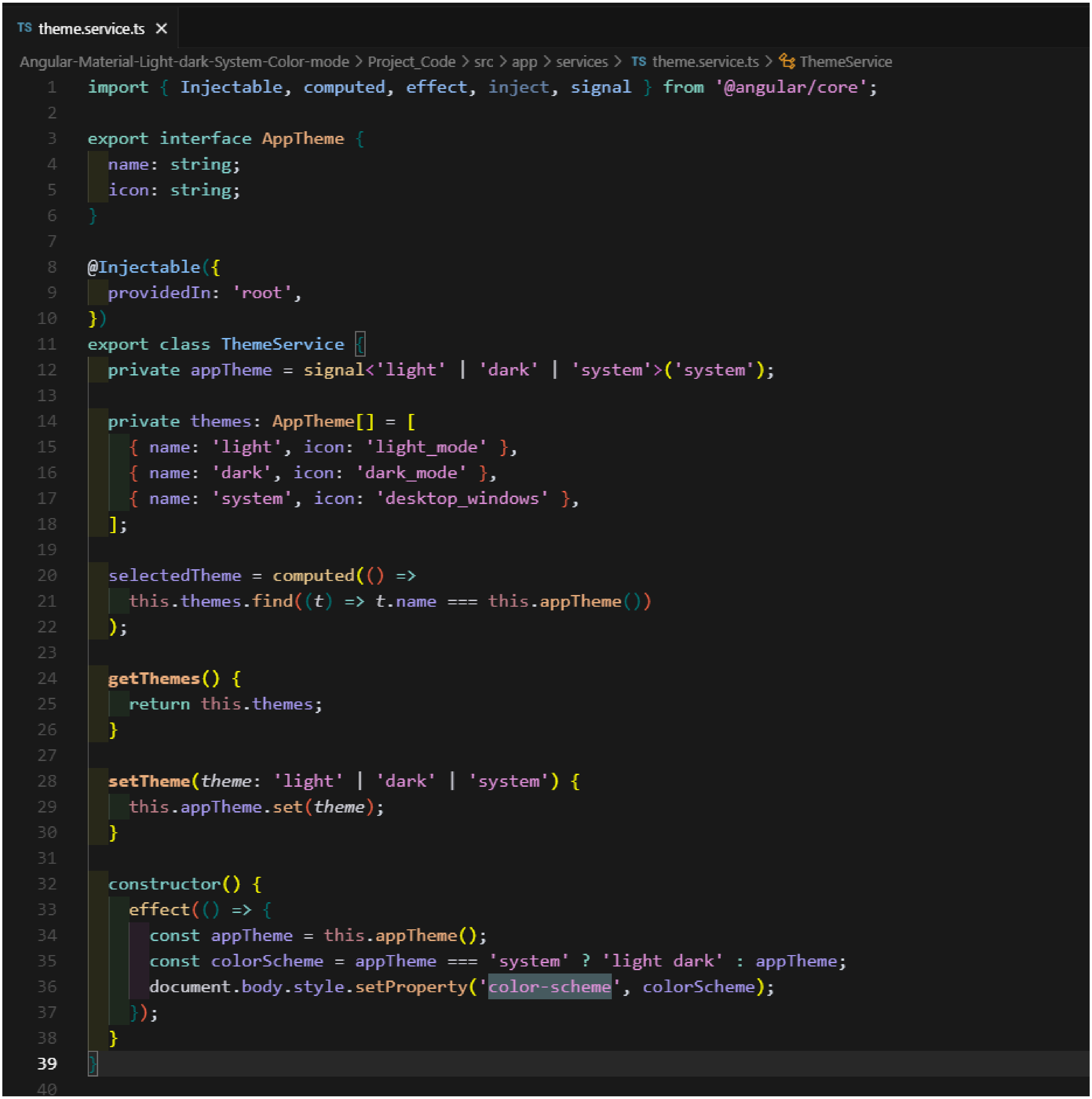
```
    </mat-toolbar>
  `,
  styles: `
    @use '@angular/material' as mat;

    mat-toolbar {
      display: flex;
      align-items: center;

      @include mat.toolbar-overrides(
        (
          container-background-color: var(--mat-sys-primary),
          container-text-color: var(--mat-sys-on-primary),
        )
      );

      @include mat.icon-button-overrides(
        (
          icon-color: var(--mat-sys-on-primary),
        )
      );
    }

    .selected-theme {
      @include mat.menu-overrides(
        (
          item-icon-color: var(--mat-sys-primary),
          item-label-text-color: var(--mat-sys-primary),
        )
      );
    }

    .flex-stretch {
      flex: 1 0 auto;
    }
  `,
})
export class HeaderComponent {
  protected themeService = inject(ThemeService);
}
```


TS content.component.ts ×
Angular-Material-Light-dark-System-Color-mode > Project_Code > src > app > components > content > TS content.

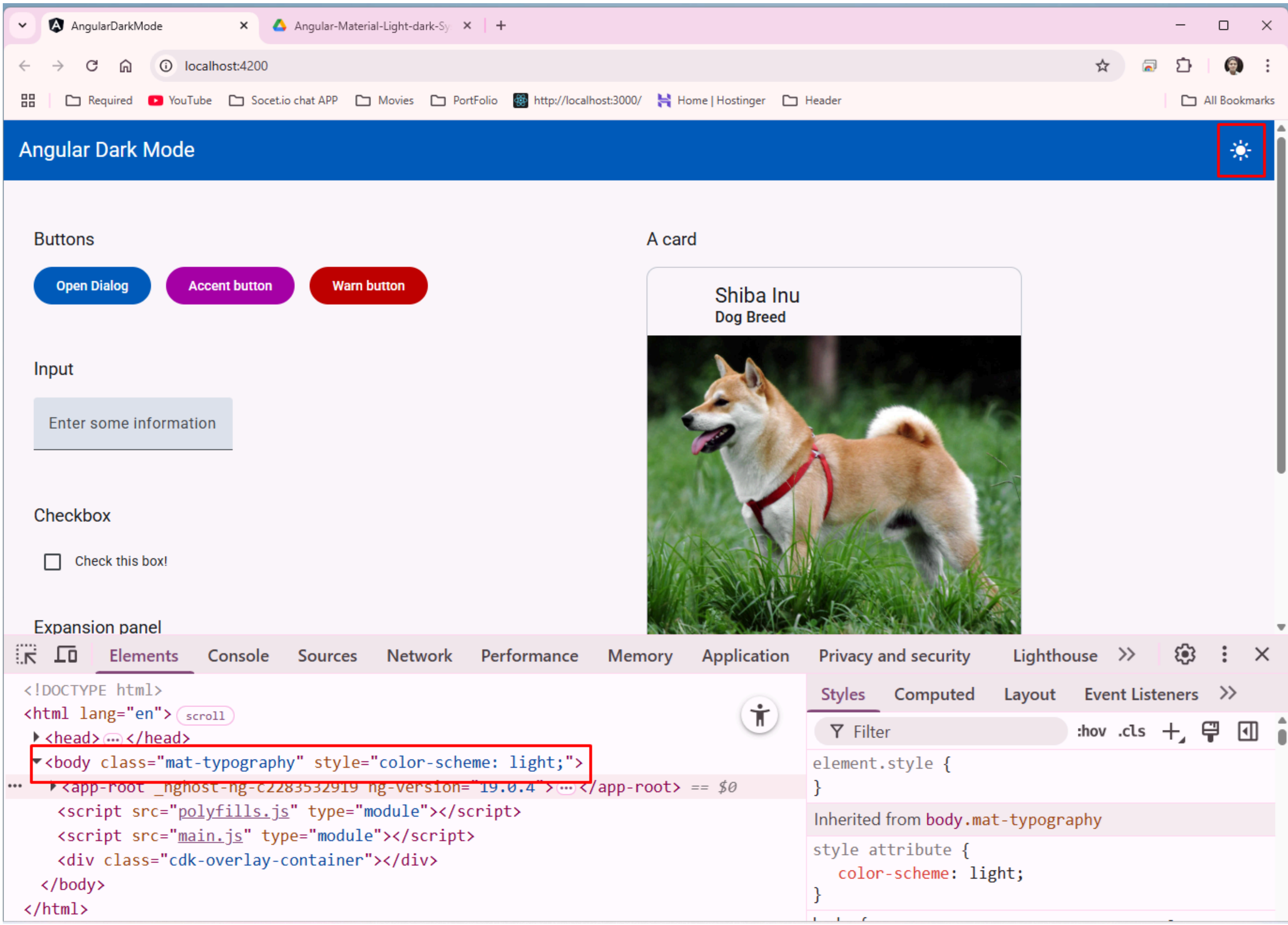
```
  styles: `
    .content {
      padding: 32px;
      min-height: 100vh;

      background-color: var(--mat-sys-background);
      color: var(--mat-sys-on-background);
    }

    .responsive-grid {
      display: grid;
      grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
      grid-gap: 1rem;
    }

    .flex-row {
      display: flex;
      flex-direction: row;
      gap: 1rem;
    }

    mat-card {
      max-width: 400px;
    }

    h3 {
      font-weight: 400;
    }
  `,
})
export class ContentComponent {
  private dialog = inject(MatDialog);

  showDialog(): void {
    this.dialog.open(SampleDialogComponent, {
      width: '500px',
    });
  }
}

```


styles.scss ×
Angular-Material-Light-dark-System-Color-mode > Project_Code > src > styles.scss > .accent

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

html {
  @include mat.theme(
    (
      color: mat.$azure-palette,
      typography: Roboto,
    )
  );
}

.accent {
  @include mat.theme(
    (
      color: mat.$magenta-palette,
    )
  );
}

.warn {
  @include mat.theme(
    (
      color: mat.$red-palette,
    )
  );
}

html,
body {
  height: 100%;
}
body {
  margin: 0;
  font-family: Roboto, "Helvetica Neue", sans-serif;
}

h2,
p {
  padding: 16px;
}
```

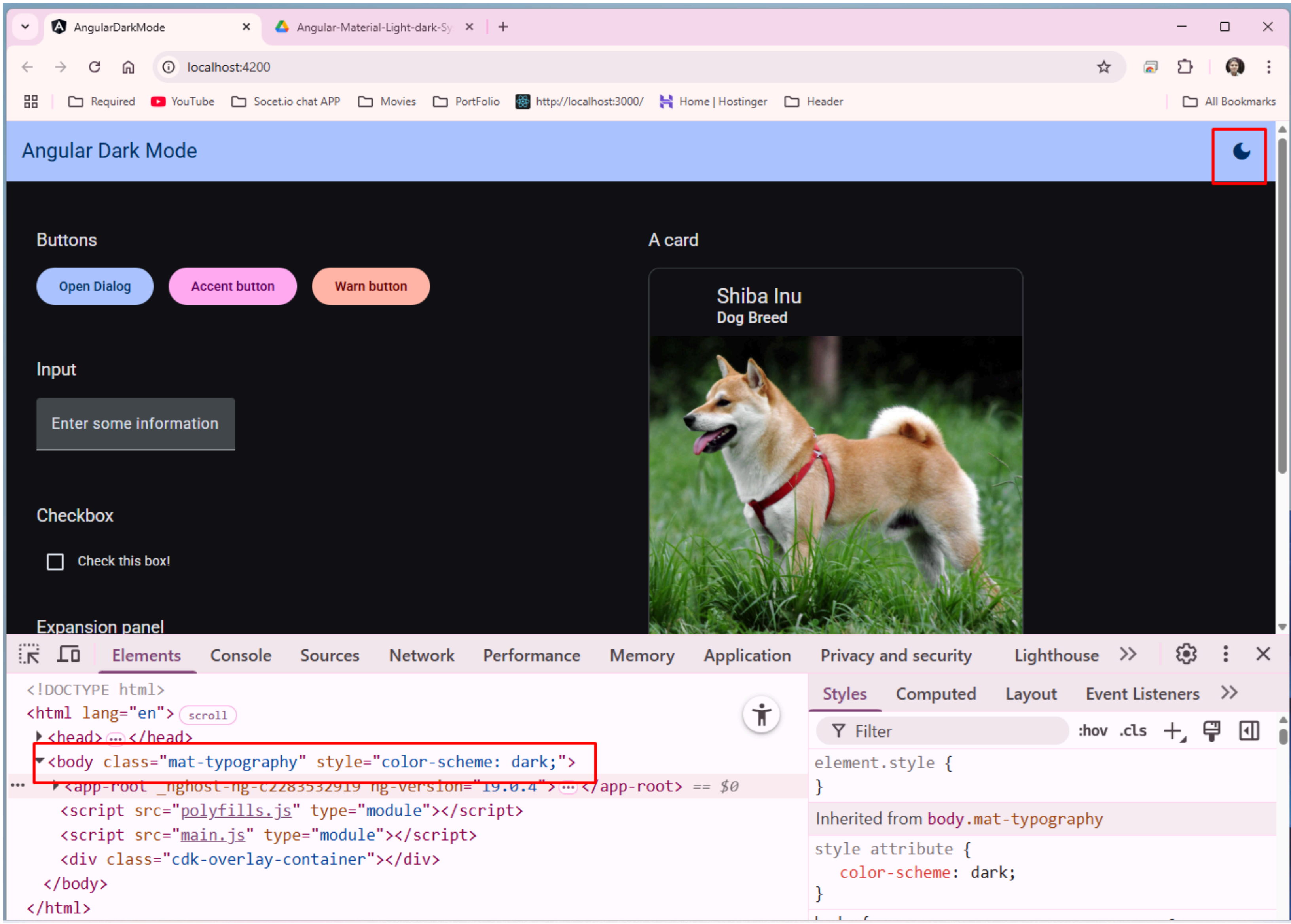

TS theme.service.ts ✕
Angular-Material-Light-dark-System-Color-mode > Project_Code > src > app > services > TS theme.service.ts > ThemeService

```typescript
import { Injectable, computed, effect, inject, signal } from '@angular/core';

export interface AppTheme {
  name: string;
  icon: string;
}

@Injectable({
  providedIn: 'root',
})
export class ThemeService {
  private appTheme = signal<'light' | 'dark' | 'system'>('system');

  private themes: AppTheme[] = [
    { name: 'light', icon: 'light_mode' },
    { name: 'dark', icon: 'dark_mode' },
    { name: 'system', icon: 'desktop_windows' },
  ];

  selectedTheme = computed(() =>
    this.themes.find((t) => t.name === this.appTheme())
  );

  getThemes() {
    return this.themes;
  }

  setTheme(theme: 'light' | 'dark' | 'system') {
    this.appTheme.set(theme);
  }

  constructor() {
    effect(() => {
      const appTheme = this.appTheme();
      const colorScheme = appTheme === 'system' ? 'light dark' : appTheme;
      document.body.style.setProperty('color-scheme', colorScheme);
    });
  }
}

```



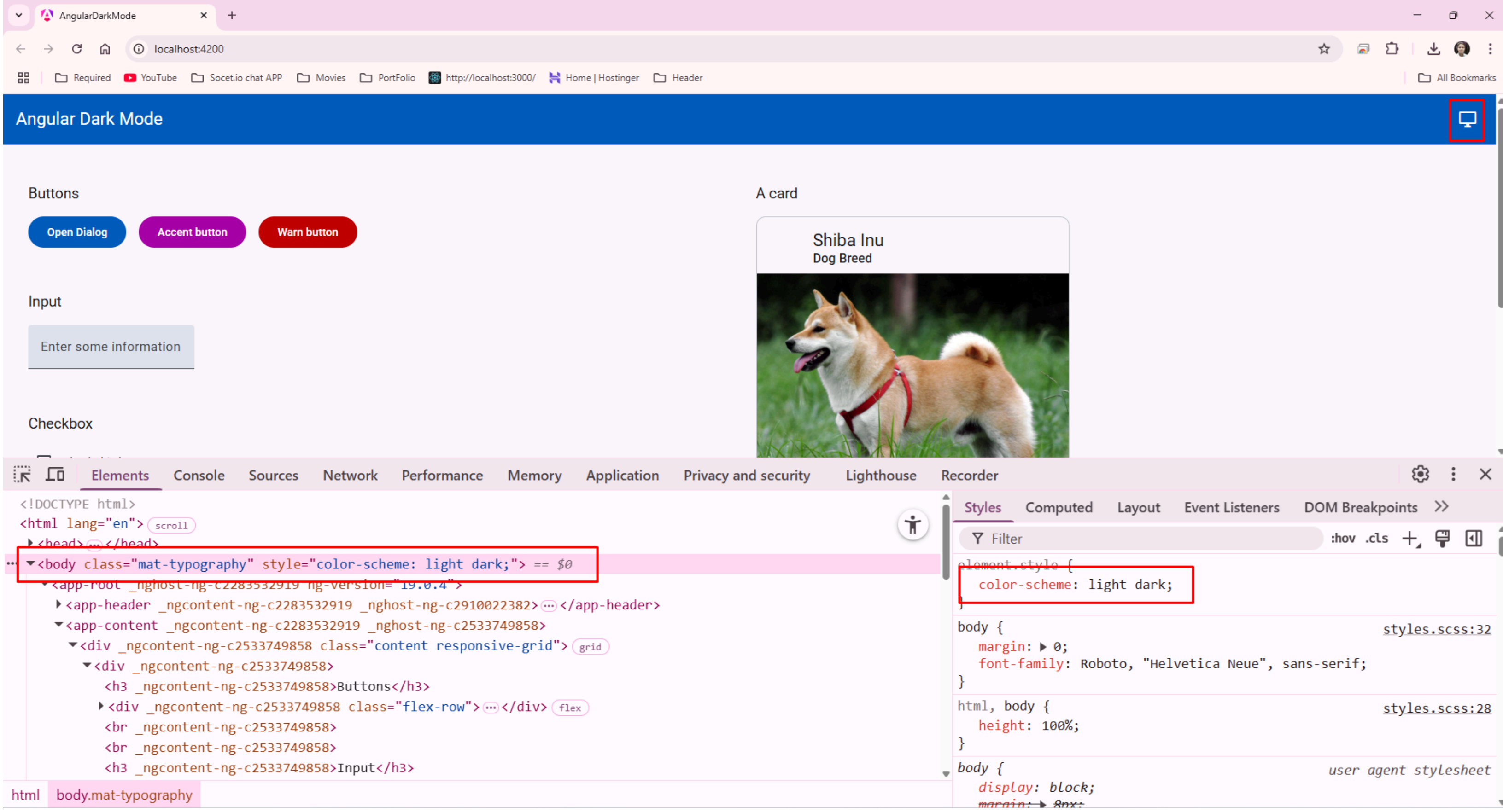
```
<!DOCTYPE html>
<html lang="en"> scroll
  <head> … </head>
  <body class="mat-typography" style="color-scheme: light;">
    <app-root _nghost-ng-c2283532919 ng-version="19.0.4"> … </app-root> == $0
    <script src="polyfills.js" type="module"></script>
    <script src="main.js" type="module"></script>
    <div class="cdk-overlay-container"></div>
  </body>
</html>
```

```
element.style {
}
Inherited from body.mat-typography
style attribute {
  color-scheme: light;
}
```



```
<!DOCTYPE html>
<html lang="en"> scroll
  <head> … </head>
  <body class="mat-typography" style="color-scheme: dark;">
    <app-root _nghost-ng-c2283532919 ng-version="19.0.4"> … </app-root> == $0
    <script src="polyfills.js" type="module"></script>
    <script src="main.js" type="module"></script>
    <div class="cdk-overlay-container"></div>
  </body>
</html>
```

```
element.style {
}
Inherited from body.mat-typography
style attribute {
   color-scheme: dark;
}
```



```
<!DOCTYPE html>
<html lang="en"> scroll
  <head> … </head>
  <body class="mat-typography" style="color-scheme: light dark;"> == $0
    <app-root _nghost-ng-c2283532919 ng-version="19.0.4">
      <app-header _ngcontent-ng-c2283532919 _nghost-ng-c2910022382> … </app-header>
      <app-content _ngcontent-ng-c2283532919 _nghost-ng-c2533749858>
        <div _ngcontent-ng-c2533749858 class="content responsive-grid"> grid
          <div _ngcontent-ng-c2533749858>
            <h3 _ngcontent-ng-c2533749858>Buttons</h3>
            <div _ngcontent-ng-c2533749858 class="flex-row"> … </div> flex
            <br _ngcontent-ng-c2533749858>
            <br _ngcontent-ng-c2533749858>
            <h3 _ngcontent-ng-c2533749858>Input</h3>
```

```
element.style {
   color-scheme: light dark;
}
body {                                                        styles.scss:32
   margin: ▶ 0;
   font-family: Roboto, "Helvetica Neue", sans-serif;
}
html, body {                                                  styles.scss:28
   height: 100%;
}
body {                                                user agent stylesheet
   display: block;
   ~~margin: ▶ 8px;~~
```