

# Enhancing User Experience with a Custom Footer using SPFx

## Introduction:

Creating a custom footer in SharePoint Online has several advantages. For instance, one can share essential information, quick links, or branding components for all pages of a particular site. Using the SharePoint Framework (SPFx) Application Customizer, it is possible for developers to place a foot note in most if not all areas of share point which will ensure that the user has a better experience while using the brand inducing software. A footer may also contain information such as contacts, links for navigation, icons for social networks, or a tagline. A custom footer can be designed in a way that suits a client's requirement and more so it can be stylized through programming.

## Technology Used:

A framework where customizations are possible in SharePoint Online from the client.

**React:** To develop the UI component of the footer.

**TypeScript:** So that, static typing is ensured and the quality of the code is improved.

**Office UI Fabric/Fluent UI:** To build uniformity in designing the UI components.

**SharePoint REST API (optional):** To retrieve variations of data that can be used in the footer.

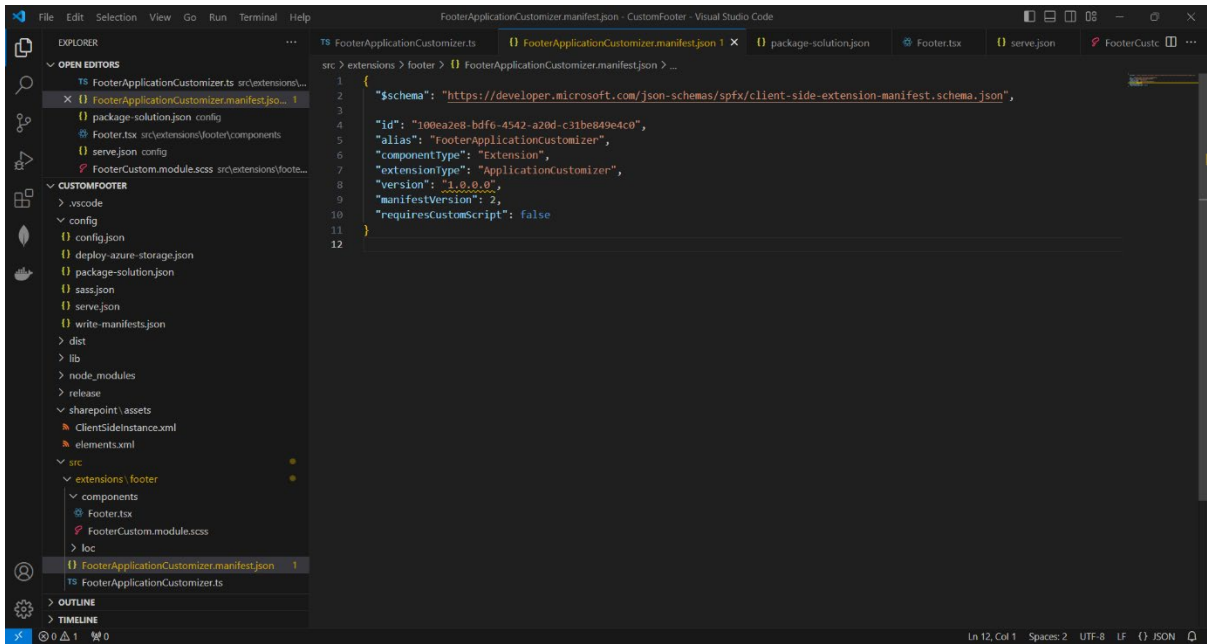
**CSS:** In order to have a corporate branded footer.

# .ts File(FooterApplicationCustomizer.ts)

```
src > extensions > footer > TS FooterApplicationCustomizer.ts > ...
1 import { Log } from '@microsoft/sp-core-library';
2 import { BaseApplicationCustomizer, PlaceholderContent, PlaceholderName } from '@microsoft/sp-application-base';
3 import * as React from 'react';
4 import * as ReactDOM from 'react-dom';
5 import Footer from './components/Footer';
6
7 export interface IFooterApplicationCustomizerProperties {
8     Bottom: string;
9 }
10
11 const LOG_SOURCE: string = 'FooterApplicationCustomizer';
12
13 export default class FooterApplicationCustomizer extends BaseApplicationCustomizer<IFooterApplicationCustomizerProperties> {
14     private _bottomPlaceholder: PlaceholderContent | undefined;
15
16     public onInit(): Promise<void> {
17         log.info(LOG_SOURCE, 'Initialized FooterApplicationCustomizer');
18
19         // Render the footer when placeholders are available
20         this.context.placeholderProvider.changedEvent.add(this, this._renderPlaceholders);
21         this._renderPlaceholders();
22
23         return Promise.resolve();
24     }
25
26     private _renderPlaceholders(): void {
27         // Handling the bottom placeholder
28         if (!this._bottomPlaceholder) {
29             this._bottomPlaceholder = this.context.placeholderProvider.tryCreateContent(
30                 PlaceholderName.Bottom,
31                 { onDispose: this._onDispose }
32             );
33
34             // If the bottom placeholder is available, render the footer component
35             if (this._bottomPlaceholder) {
36                 const element: React.ReactElement<> = React.createElement(footer, {});
37                 ReactDOM.render(element, this._bottomPlaceholder.domElement);
38             }
39         }
40     }
41
42     private _onDispose(): void {
43         log.info(LOG_SOURCE, 'Disposing of FooterApplicationCustomizer');
44         if (this._bottomPlaceholder && this._bottomPlaceholder.domElement) {
45             ReactDOM.unmountComponentAtNode(this._bottomPlaceholder.domElement);
46         }
47     }
48 }
49
```

```
src > extensions > footer > TS FooterApplicationCustomizer.ts > ...
26 private _renderPlaceholders(): void {
27     // Handling the bottom placeholder
28     if (!this._bottomPlaceholder) {
29         this._bottomPlaceholder = this.context.placeholderProvider.tryCreateContent(
30             PlaceholderName.Bottom,
31             { onDispose: this._onDispose }
32         );
33
34         // If the bottom placeholder is available, render the footer component
35         if (this._bottomPlaceholder) {
36             const element: React.ReactElement<> = React.createElement(footer, {});
37             ReactDOM.render(element, this._bottomPlaceholder.domElement);
38         }
39     }
40
41     private _onDispose(): void {
42         log.info(LOG_SOURCE, 'Disposing of FooterApplicationCustomizer');
43         if (this._bottomPlaceholder && this._bottomPlaceholder.domElement) {
44             ReactDOM.unmountComponentAtNode(this._bottomPlaceholder.domElement);
45         }
46     }
47 }
48 }
49
```

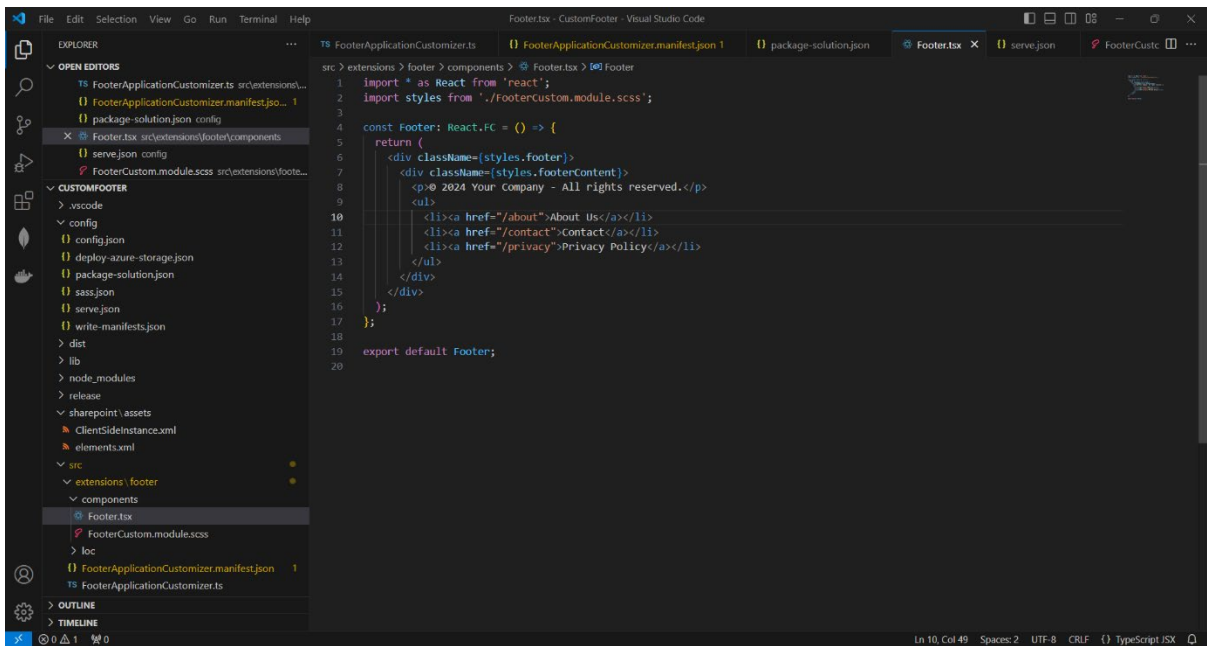
## .manifest.json File



The screenshot shows the Visual Studio Code editor with the `FooterApplicationCustomizer.manifest.json` file open. The file content is as follows:

```
1 {
2   "$schema": "https://developer.microsoft.com/json-schemas/spfx/client-side-extension-manifest.schema.json",
3
4   "id": "100ea2e8-bdfe-4542-a28d-c31be8a9e4c0",
5   "alias": "FooterApplicationCustomizer",
6   "componentType": "Extension",
7   "extensionType": "ApplicationCustomizer",
8   "version": "1.0.0.0",
9   "manifestVersion": 2,
10  "requiresCustomScript": false
11 }
12
```

## src\extensions\footer\components\Footer.tsx



The screenshot shows the Visual Studio Code editor with the `Footer.tsx` file open. The file content is as follows:

```
1 import * as React from 'react';
2 import styles from './FooterCustom.module.scss';
3
4 const Footer: React.FC = () => {
5   return (
6     <div className={styles.footer}>
7       <div className={styles.footerContent}>
8         <p>© 2024 Your Company - All rights reserved.</p>
9         <ul>
10          <li><a href="/about">About Us</a></li>
11          <li><a href="/contact">Contact</a></li>
12          <li><a href="/privacy">Privacy Policy</a></li>
13        </ul>
14      </div>
15    </div>
16  );
17 };
18
19 export default Footer;
```

## src\extensions\footer\components\FooterCustom.module.scss

```
1 .footer {
2   background-color: #0078d4;
3   padding: 10px;
4   text-align: center;
5   color: white;
6 }
7
8 .footerContent {
9   display: flex;
10  justify-content: space-between;
11  align-items: center;
12  font-size: 14px;
13 }
14
15 .footerContent ul {
16  list-style-type: none;
17  margin: 0;
18  padding: 0;
19 }
20
21 .footerContent li {
22  display: inline;
23  margin: 0 15px;
24 }
25
26 .footerContent a {
27  color: white;
28  text-decoration: none;
29 }
30
31 .footerContent a:hover {
32  text-decoration: underline;
33 }
34
```

## Output Screenshots:

