

Inside a panel Browse Document From file explorer(Webpart)

Introduction:

Document Upload web Part enables system users to search for files within the local system and upload them to a defined document library in SharePoint. The design adopts a panel based interface whose components include a button to browse for a file, a button to upload the file onto the document library and a button to close the panel.

Technology Used:

SPFx (SharePoint Framework): A framework for modern SharePoint development that enables developers to implement solutions using client-side coding.

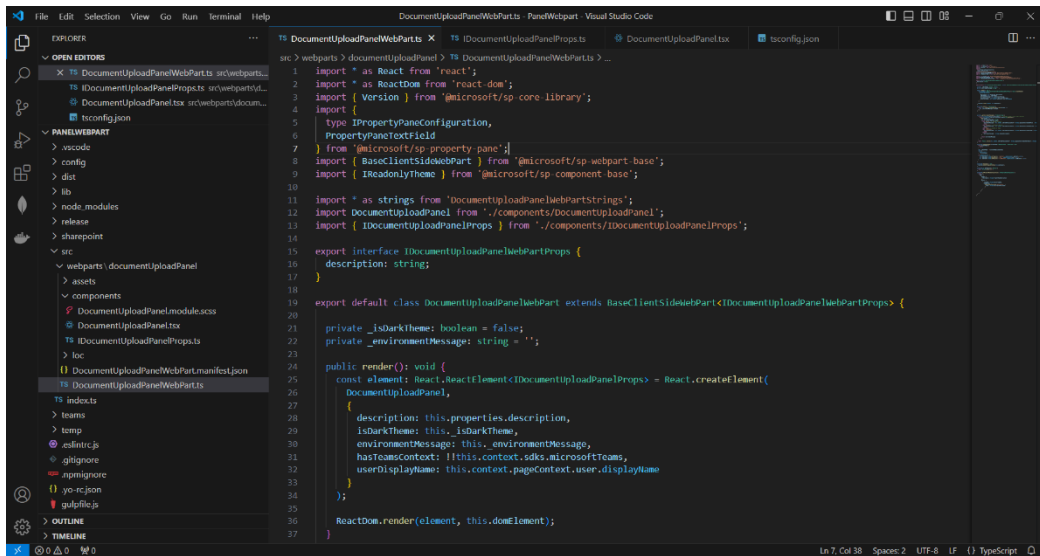
React: A JavaScript tool for building interface components. Used for constructing the panel and managing user input.

PnPJS: A javascript library intended to ease the development of applications that use SharePoint and Office 365 REST services.

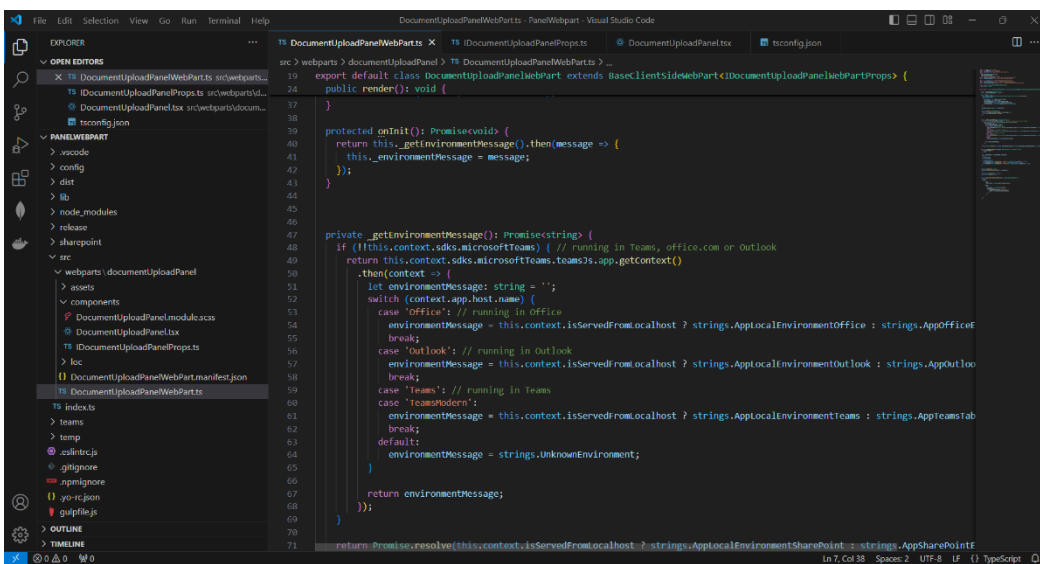
Fluent UI: A set of UX architecture for ensuring uniformity in the implementation of Microsoft's design.

Code Structure:

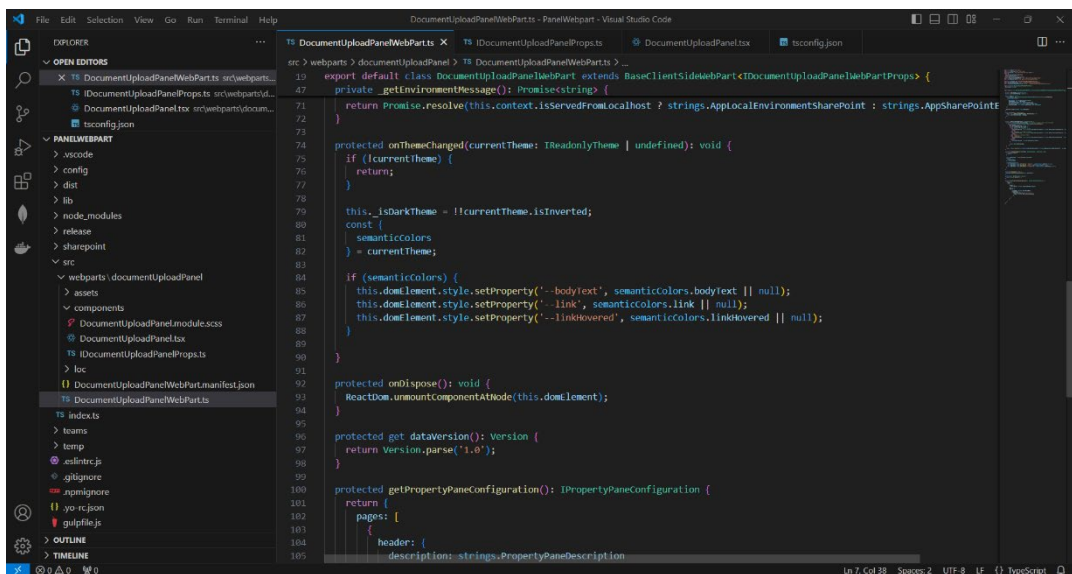
DocumentUploadWebPart.ts:-



```
src > webparts > documentUploadPanel > DocumentUploadPanelWebParts > ...
1 import * as React from 'react';
2 import * as ReactDOM from 'react-dom';
3 import { Version } from '@microsoft/sp-core-library';
4 import {
5   type IPropertyPaneConfiguration,
6   PropertyPaneTextField
7 } from '@microsoft/sp-property-pane';
8 import { BaseClientSideWebPart } from '@microsoft/sp-webpart-base';
9 import { IReadyOnlyTheme } from '@microsoft/sp-component-base';
10
11 import * as strings from 'documentUploadPanelWebPartStrings';
12 import DocumentUploadPanel from './components/documentUploadPanel';
13 import { IDocumentUploadPanelProps } from './components/IDocumentUploadPanelProps';
14
15 export interface IDocumentUploadPanelWebPartProps {
16   description: string;
17 }
18
19 export default class DocumentUploadPanelWebPart extends BaseClientSideWebPart<IDocumentUploadPanelWebPartProps> {
20   private _isDarkTheme: boolean = false;
21   private _environmentMessage: string = '';
22
23   public render(): void {
24     const element: React.ReactElement<IDocumentUploadPanelProps> = React.createElement(
25       DocumentUploadPanel,
26       {
27         description: this.properties.description,
28         isDarkTheme: this._isDarkTheme,
29         environmentMessage: this._environmentMessage,
30         hasTeamsContext: !!this.context.sdk.ms.teams,
31         userDisplayName: this.context.pageContext.user.displayName
32       });
33     ReactDOM.render(element, this.domElement);
34   }
35 }
```



```
36
37
38 protected onInit(): Promise<void> {
39   return this._getEnvironmentMessage().then(message => {
40     this._environmentMessage = message;
41   });
42 }
43
44 private _getEnvironmentMessage(): Promise<string> {
45   if (!!this.context.sdk.ms.teams) { // running in Teams, Office.com or Outlook
46     return this.context.sdk.ms.teams.teamsJS.getAppContext()
47       .then(context => {
48         let environmentMessage: string = '';
49         switch (context.app.hostName) {
50           case 'Office': // running in Office
51             environmentMessage = this.context.isServedFromLocalhost ? strings.AppLocalEnvironmentOffice : strings.AppOffice;
52             break;
53           case 'Outlook': // running in Outlook
54             environmentMessage = this.context.isServedFromLocalhost ? strings.AppLocalEnvironmentOutlook : strings.AppOutlook;
55             break;
56           case 'TeamsModern': // running in Teams
57             environmentMessage = this.context.isServedFromLocalhost ? strings.AppLocalEnvironmentTeams : strings.AppTeamsModern;
58             break;
59           default:
60             environmentMessage = strings.UnknownEnvironment;
61         }
62         return environmentMessage;
63       });
64   }
65   return Promise.resolve(this.context.isServedFromLocalhost ? strings.AppLocalEnvironmentSharePoint : strings.AppSharePoint);
66 }
```



```
67
68
69 protected onThemeChanged(currentTheme: IReadyOnlyTheme | undefined): void {
70   if (!currentTheme) {
71     return;
72   }
73   this._isDarkTheme = !currentTheme.isInverted;
74   const {
75     semanticColors
76   } = currentTheme;
77
78   if (semanticColors) {
79     this.domElement.style.setProperty('--body-text', semanticColors.bodyText || null);
80     this.domElement.style.setProperty('--link', semanticColors.link || null);
81     this.domElement.style.setProperty('--link-hovered', semanticColors.linkHovered || null);
82   }
83 }
84
85 protected onDispose(): void {
86   ReactDOM.unmountComponentAtNode(this.domElement);
87 }
88
89 protected get dataVersion(): Version {
90   return Version.parse('1.0');
91 }
92
93 protected getPropertyPaneConfiguration(): IPropertyPaneConfiguration {
94   return {
95     pages: [
96       {
97         header: {
98           description: strings.PropertyPaneDescription
99         }
100       }
101     ]
102   };
103 }
```

```

export default class DocumentUploadPanelWebPart extends BaseClientSideWebPart<IDocumentUploadPanelWebPartProps> {
  protected get dataVersion(): Version {
    return Version.one();
  }

  protected getPropertyValuePaneConfiguration(): IPropertyPaneConfiguration {
    return {
      pages: [
        {
          header: {
            description: strings.PropertyPaneDescription
          },
          groups: [
            {
              groupName: strings.BasicGroupName,
              groupFields: [
                PropertyPaneTextField('description', {
                  label: strings.DescriptionFieldLabel
                })
              ]
            }
          ]
        }
      ]
    };
  }
}

```

DocumentUploadPanel.tsx:-

```

import React from 'react';
import { Panel, PrimaryButton, DefaultButton } from '@fluentui/react';
import pnp, {} from 'sp-pnp-js';
import { IDocumentUploadPanelProps } from './IDocumentUploadPanelProps';
import { useState } from 'react';

const DocumentUploadPanel: React.FC<IDocumentUploadPanelProps> = (props) => {
  const [file, setFile] = useState<File | null>(null);

  const handleFileChange = (event: React.ChangeEvent<HTMLInputElement>) => {
    setFile(event.target.files[0]);
  };

  const handleUpload = async () => {
    if (file) {
      const filename = file.name;
      const arrayBuffer = await file.arrayBuffer();

      pnp.sp.web.getFolderByServerRelativeUrl('/Shared Documents')
        .files.add(filename, arrayBuffer, true)
        .then(() => {
          alert(`${filename} uploaded successfully!`);
        })
        .catch((error: Error) => {
          alert('Upload failed. Error: ${error.message}');
        });
    }
  };

  return (
    <Panel
      isOpen={true}
      onDismiss={() => { alert('Back button clicked!'); }}
      headerText="Upload Document"
    >
      <input type="file" onChange={handleFileChange} />
      <PrimaryButton text="Upload" onClick={handleUpload} />
      <DefaultButton text="Back" onClick={() => { alert('Back button clicked!'); }} />
    </Panel>
  );
};

export default DocumentUploadPanel;

```

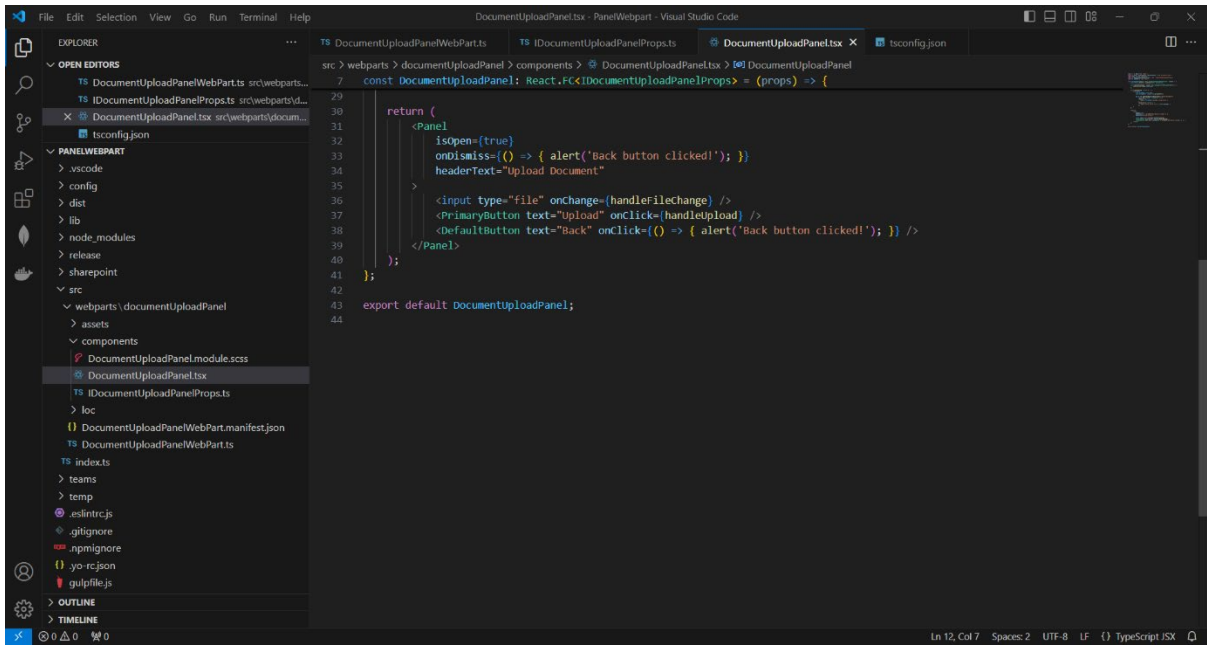
```

return (
  <Panel
    isOpen={true}
    onDismiss={() => { alert('Back button clicked!'); }}
    headerText="Upload Document"
  >
    <input type="file" onChange={handleFileChange} />
    <PrimaryButton text="Upload" onClick={handleUpload} />
    <DefaultButton text="Back" onClick={() => { alert('Back button clicked!'); }} />
  </Panel>
);
};

export default DocumentUploadPanel;

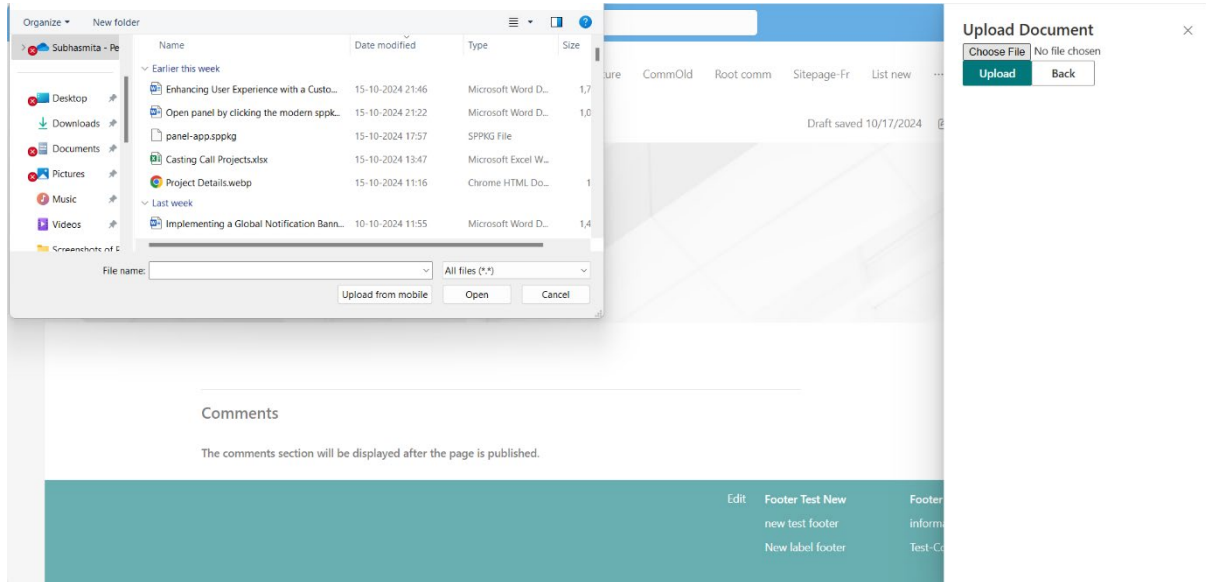
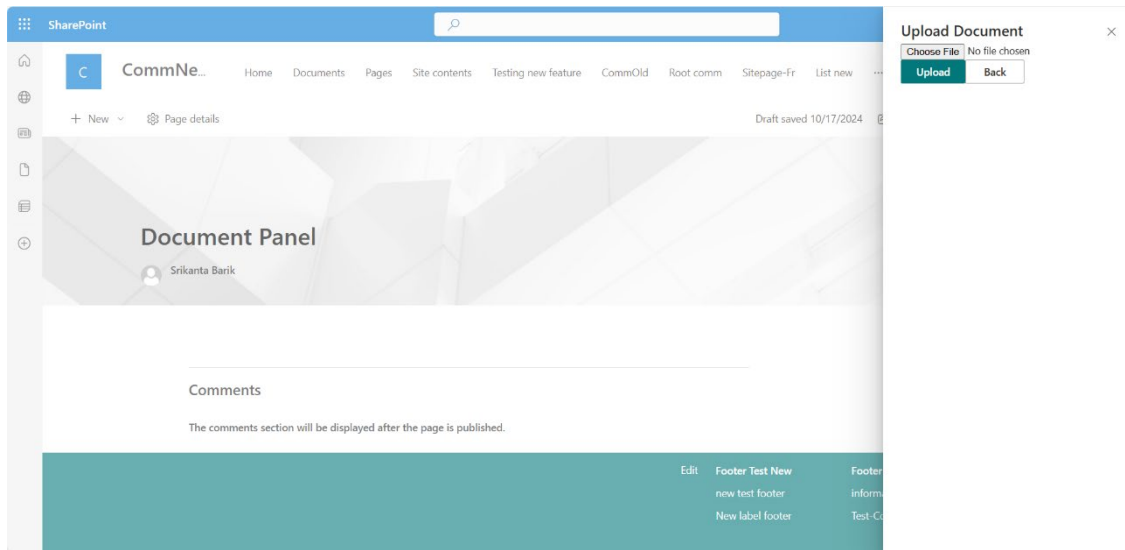
```

IDocumentUploadPanelProps.ts :



```
7  const DocumentUploadPanel: React.FC<IDocumentUploadPanelProps> = (props) => {
29
30
31     return (
32       <Panel
33         isOpen={true}
34         onDismiss={() => { alert("Back button clicked!"); }}
35         headerText="Upload Document"
36       >
37         <input type="file" onChange={handlefilechange} />
38         <PrimaryButton text="Upload" onClick={handleupload} />
39         <DefaultButton text="Back" onClick={() => { alert("Back button clicked!"); }} />
40       </Panel>
41     );
42   };
43   export default DocumentUploadPanel;
44 }
```

Output Webpart Images:-



The image shows a SharePoint interface with several key elements:

- SharePoint Header:** Located at the top left, it includes the SharePoint logo and the text "SharePoint".
- Search Bar:** A search bar is visible in the top right corner of the header area.
- Notification:** A white notification box in the top center reads "3gv3y3.sharepoint.com says" and "Back button clicked!". It features a blue "OK" button.
- Page Title and Navigation:** The page title is "CommNe...". Navigation links for "Home", "Documents", and "Pages" are present. A "New" button with a dropdown arrow and "Page details" are also visible.
- Document Panel:** A large section titled "Document Panel" is shown, with a user profile for "Srikanta Barik" and a "Draft saved 10/17/2024" status.
- Comments Section:** A section titled "Comments" contains the text: "The comments section will be displayed after the page is published."
- Footer:** A teal footer bar at the bottom contains links for "Edit", "Footer Test New", and "Footer", along with sub-links like "new test footer" and "New label footer".
- Upload Dialog:** An "Upload Document" dialog box is open on the right side. It shows "Choose File" (No file chosen) and buttons for "Upload" and "Back".