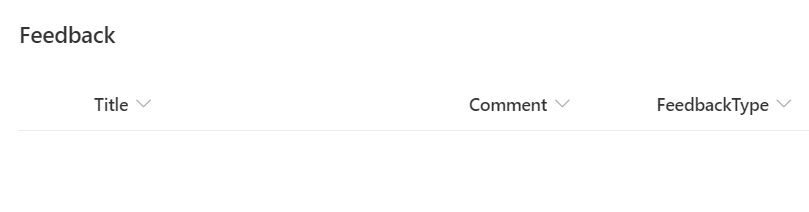
SPFx React Webpart using jQuery Plugin & Bootstrap

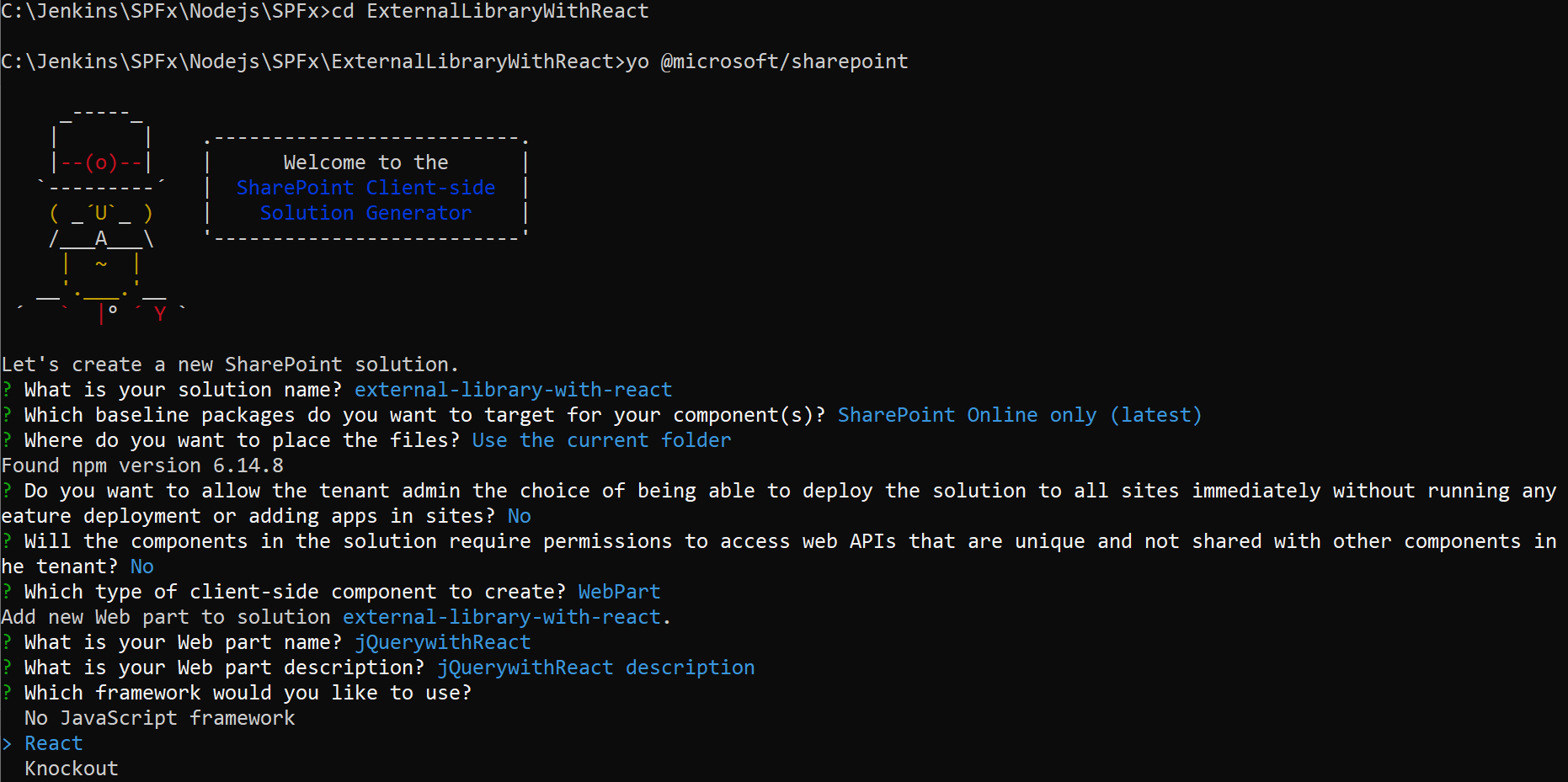
**Create a SharePoint List**





**Create a new SPFx web part project**

* Navigate to your favourite folder
* create new folder
* move to newly created folder
* Open Command prompt and run following comment to create a new web part by running the Yeoman SharePoint Generator
  + - yo @microsoft/sharepoint
* When prompted:
* Enter the webpart name as your solution name, and then select Enter.



## **Libraries Used**

1. Bootstrap v4.5.2
2. jQuery v3.5.1
3. Tooltipster v4.2.8
4. Pnpjs v2.0.10

Use below commands to install other dependencies

npm install jquery --save

npm install tooltipster --save

npm install @pnp/logging @pnp/common @pnp/odata @pnp/sp @pnp/graph –save

npm i bootstrap

Open your solution in Visual Studio Code

Code .

Once the libraries are installed add the below json entry to the config\config.json file under the “externals”:

"externals": {

        "jquery": {

            "path": "node\_modules/jquery/dist/jquery.min.js",

            "globalName": "jquery"

        },

        "bootstrap": {

            "path": "node\_modules/bootstrap/dist/js/bootstrap.min.js",

            "globalName": "bootstrap"

        },

        "tooltipster": {

            "path": "node\_modules/tooltipster/dist/js/tooltipster.bundle.min.js",

            "globalName": "tooltipster"

        }

    },

Create a Feedback.tsx file under components folder and paste below code

import \* as React from 'react';

import \* as ReactDOM from 'react-dom';

import styles from './JQuerywithReact.module.scss';

import \* as $ from 'jquery';

require('tooltipster');

import { sp } from "@pnp/sp";

import "@pnp/sp/webs";

import "@pnp/sp/lists";

import "@pnp/sp/items";

import "@pnp/sp/site-users/web";

import { IItemAddResult } from "@pnp/sp/items";

require('../../../../node\_modules/bootstrap/dist/css/bootstrap.min.css');

require('../../../../node\_modules/tooltipster/dist/css/tooltipster.bundle.min.css');

require('../../../../node\_modules/tooltipster/dist/css/plugins/tooltipster/sideTip/themes/tooltipster-sideTip-punk.min.css');

interface MyProps {

}

interface MyState {

  ExcellentComment: string;

  GoodComment:string;

  MediumComment: string;

  PoorComment: string;

  MyEmail: string;

}

export default class Feedback extends React.Component<MyProps, MyState> {

  constructor(props) {

    super(props);

    this.state = {

      ExcellentComment: "",

      GoodComment:"",

      MediumComment: "",

      PoorComment: "",

      MyEmail: ''

    };

  }

  private $node;

  private $ttInstance;

  private hideStyle = {

    display: 'none'

  };

  private btnSubmitStyle = {

    cursor: 'pointer'

  };

  private icoPoorStyle = {

    color: 'red'

  };

  private icoMediumStyle = {

    color: 'orange'

  };

  private icoExcellentStyle = {

    color: 'green'

  };

  protected handleExcellentChanged(event) {

    this.setState({ ExcellentComment: event.target.value });

  }

  protected handleGoodChanged(event) {

    this.setState({ GoodComment: event.target.value });

  }

  protected handleMediumChanged(event) {

    this.setState({ MediumComment: event.target.value });

  }

  protected handlePoorChanged(event) {

    this.setState({ PoorComment: event.target.value });

  }

  public componentDidMount() {

    this.$node = $(this.refs.ttContainer);

    this.$ttInstance = this.$node.children().tooltipster({

      animation: 'grow',

      delay: 200,

      theme: 'tooltipster-punk',

      trigger: 'click',

      maxWidth: 350,

      minWidth: 300,

      interactive: true

    });

    sp.web.currentUser.get().then((user) => {

      this.setState({ MyEmail: user.Email });

      console.log(user);

    });

  }

  protected addExcellentComment = (event) => {

    console.log(this.state.ExcellentComment);

    sp.web.lists.getByTitle("Feedback").items.add({

      Comment: this.state.ExcellentComment,

      FeedbackType: "Excellent",

      Title: this.state.MyEmail,

    }).then((iar: IItemAddResult) => {

      console.log(iar);

    });

    $(this.refs.ttFeedback1).trigger("click"); // to close

    this.setState({ ExcellentComment: "" });

    event.preventDefault();

  }

  protected addGoodComment = () => {

    console.log(this.state.GoodComment);

    sp.web.lists.getByTitle("Feedback").items.add({

      Comment: this.state.GoodComment,

      FeedbackType: "Good",

      Title: this.state.MyEmail,

    }).then((iar: IItemAddResult) => {

      console.log(iar);

    });

    $(this.refs.ttFeedback2).trigger("click"); // to close

    this.setState({ MediumComment: "" });

    event.preventDefault();

  }

  protected addMediumComment = () => {

    console.log(this.state.MediumComment);

    sp.web.lists.getByTitle("Feedback").items.add({

      Comment: this.state.MediumComment,

      FeedbackType: "Average",

      Title: this.state.MyEmail,

    }).then((iar: IItemAddResult) => {

      console.log(iar);

    });

    $(this.refs.ttFeedback2).trigger("click"); // to close

    this.setState({ MediumComment: "" });

    event.preventDefault();

  }

  protected addPoorComment = () => {

    console.log(this.state.PoorComment);

    sp.web.lists.getByTitle("Feedback").items.add({

      Comment: this.state.PoorComment,

      FeedbackType: "Poor",

      Title: this.state.MyEmail,

    }).then((iar: IItemAddResult) => {

      console.log(iar);

    });

    $(this.refs.ttFeedback3).trigger("click"); // to close

    this.setState({ PoorComment: "" });

    event.preventDefault();

  }

  public render() {

    return (<div>

      <ul className="nav justify-content-center" ref="ttContainer">

        <li ref="ttFeedback1" data-tooltip-content="#tooltip\_content1" className="nav-item"><a className="nav-link tt-tip" href="#">Excellent</a></li>

        <li ref="ttFeedback2" data-tooltip-content="#tooltip\_content2" className="nav-item"><a className="nav-link tt-tip" href="#">Good</a></li>

        <li ref="ttFeedback3" data-tooltip-content="#tooltip\_content3" className="nav-item"><a className="nav-link tt-tip" href="#">Medium</a></li>

        <li ref="ttFeedback4" data-tooltip-content="#tooltip\_content4" className="nav-item"><a className="nav-link tt-tip" href="#">Low</a></li>

      </ul>

      <div style={this.hideStyle} className={styles.tooltipTemplates}>

        <div id="tooltip\_content1">

          <div className="row">

            <div className="col-md-12">

              <div className="form-group">

                <label htmlFor="comment1">Comment:</label>

                <textarea value={this.state.ExcellentComment} onChange={(event) => this.handleExcellentChanged(event)} ref="comment1Ref" className="form-control" rows={5} id="comment1" />

              </div>

            </div>

          </div>

          <div className="row">

            <div className="col-md-6"></div>

            <div className="col-md-6 text-right">

              <div style={this.btnSubmitStyle} onClick={this.addExcellentComment} className="btnSubmit btnHappy btn btn-success">Submit</div>

            </div>

          </div>

        </div>

        <div id="tooltip\_content2">

          <div className="row">

            <div className="col-md-12">

              <div className="form-group">

                <label htmlFor="comment2">Comment:</label>

                <textarea value={this.state.GoodComment} onChange={(event) => this.handleGoodChanged(event)} className="form-control" rows={5} id="comment2" />

              </div>

            </div>

          </div>

          <div className="row">

            <div className="col-md-6"></div>

            <div className="col-md-6 text-right">

              <div style={this.btnSubmitStyle} onClick={this.addGoodComment} className="btnSubmit btnMedium btn btn-warning">Submit</div>

            </div>

          </div>

        </div>

        <div id="tooltip\_content3">

          <div className="row">

            <div className="col-md-12">

              <div className="form-group">

                <label htmlFor="comment3">Comment:</label>

                <textarea value={this.state.MediumComment} onChange={(event) => this.handleMediumChanged(event)} className="form-control" rows={5} id="comment3" />

              </div>

            </div>

          </div>

          <div className="row">

            <div className="col-md-6"></div>

            <div className="col-md-6 text-right">

              <div style={this.btnSubmitStyle} onClick={this.addMediumComment} className="btnSubmit btnMedium btn btn-warning">Submit</div>

            </div>

          </div>

        </div>

        <div id="tooltip\_content4">

          <div className="row">

            <div className="col-md-12">

              <div className="form-group">

                <label htmlFor="comment4">Comment:</label>

                <textarea value={this.state.PoorComment} onChange={(event) => this.handlePoorChanged(event)} className="form-control" rows={5} id="comment4" />

              </div>

            </div>

          </div>

          <div className="row">

            <div className="col-md-6"></div>

            <div className="col-md-6 text-right">

              <div style={this.btnSubmitStyle} onClick={this.addPoorComment} className="btnSubmit btn btn-danger">Submit</div>

            </div>

          </div>

        </div>

      </div>

    </div>);

  }

}

Go back to main webpart file in this case JQuerywithReact.tsx, open the file and reference to the feedback component

import Feedback from "./Feedback"

In the render() method add the <Feedback></Feedback>

i.e

<div className={ styles.jQuerywithReact }>

       <div className="container-fluid">

         <Feedback></Feedback>

       </div>

      </div>

Then go to JQuerywithReactWebPart.ts file

Import below headers

import { sp } from "@pnp/sp";

import "@pnp/sp/webs";

import "@pnp/sp/lists";

import "@pnp/sp/items";

import "@pnp/sp/site-users/web";

Add onInit method

public onInit(): Promise<void> {

    return super.onInit().then(\_ => {

      sp.setup({

        spfxContext: this.context

      });

    });

  }

Open JQuerywithReact.module.scss file under components

Add below css class

.tooltipTemplates {

            display: none;

        }