Creating the Emoticons Webpart in SharePoint using React and TypeScript involves several steps. You'll need to set up your development environment, create the web part, and implement the features outlined in your requirement document. Here's a high-level overview of the steps to create this web part:

1. **Set Up Your Development Environment:**

a. Install Node.js: Make sure you have Node.js installed on your development machine.

b. Install Yeoman and the SharePoint generator: Use npm to install Yeoman and the SharePoint generator globally.

npm install -g yo @microsoft/generator-sharepoint

Create a new SharePoint solution: Run the Yeoman generator to create a new SharePoint solution project.

yo @microsoft/sharepoint

1. **Implement the Emoticons Web Part:**

In your SharePoint project, you'll create a custom web part with React and TypeScript that meets your requirements. Here's a basic outline of how to proceed:

a. Create a React component for the emoticon strip and reactions feature. This component will include the emoticons and logic for recording reactions.

b. Add the necessary SharePoint libraries and dependencies to your project, such as SharePoint Framework (SPFx) libraries and Office UI Fabric React components.

c. Design and implement the user interface for the emoticons, which could include a header icon or a floating menu.

d. Implement the logic to record reactions when a user clicks an emoticon.

e. Use SharePoint REST API or other SharePoint APIs to store the recorded reactions for analytics purposes.

f. Add the web part properties as needed to customize the behavior of the web part.

1. **Implement Emoticons on Individual Storyline and Home Page Sections:**

Implement the emoticons at the end of story pages and in sections on the home page as per your requirement document. This may involve creating separate components for the home page sections and integrating them into the web part.

1. **Testing and Debugging:**

Thoroughly test your web part on a SharePoint development site. Ensure that reactions are recorded correctly and that the user interface behaves as expected.

1. **Deployment:**

Package your SharePoint solution for deployment. You can use the SharePoint Framework's build and package commands for this purpose.

gulp bundle --ship

gulp package-solution –ship

1. Deploy the solution package to your SharePoint site using the App Catalog.
2. **Usage:**

Add the Emoticons Web Part to SharePoint pages as needed. Configure its properties and observe reactions from users.

1. **Documentation and Maintenance:**

Document your web part, including how to install, configure, and use it. Regularly update and maintain the web part as needed.

Please note that the above steps provide a high-level overview, and the actual implementation details can be complex depending on the specific requirements of your Emoticons Web Part. You may need to delve deeper into SharePoint Framework development, React, and TypeScript to complete this project successfully. Additionally, SharePoint Framework documentation and tutorials can be valuable resources as you work on this project.

please implement the above steps you mentioned

ChatGPT

Creating a complete SharePoint web part with React and TypeScript is a substantial project that can't be fully implemented in a single response. However, I can provide you with a simplified example to get you started on the right track. This example will cover the basic structure of a SharePoint web part with a simple emoticon strip for recording reactions. Please keep in mind that this is a simplified demonstration and not a production-ready solution. You'll need to expand upon it to meet your specific requirements.

**Prerequisites:** Before you begin, make sure you have Node.js, npm, Yeoman, and the SharePoint generator installed. You should also have access to a SharePoint site where you can deploy the web part.

**Step 1: Create a SharePoint Web Part Project**

In your command line, run the following commands to create a SharePoint web part project:

yo @microsoft/sharepoint

yo @microsoft/sharepoint

Follow the prompts to configure your project.

**Step 2: Create React Components**

In your project, you'll create React components for the emoticon strip and reactions.

Create a new folder in your project, e.g., "components," and add the following files:

**EmoticonStrip.tsx**:

import \* as React from 'react';

interface EmoticonStripProps {

onReactionSelected: (reaction: string) => void;

}

const EmoticonStrip: React.FC<EmoticonStripProps> = ({ onReactionSelected }) => {

const reactions = ['like', 'happy', 'sad'];

return (

<div>

{reactions.map((reaction, index) => (

<button key={index} onClick={() => onReactionSelected(reaction)}>

{reaction}

</button>

))}

</div>

);

};

export default EmoticonStrip;

**EmoticonsWebPart.ts**:

import \* as React from 'react';

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

import { Version, Text, Button } from '@microsoft/sp-core-library';

import {

BaseClientSideWebPart,

IPropertyPaneConfiguration,

PropertyPaneTextField,

} from '@microsoft/sp-webpart-base';

import EmoticonStrip from './components/EmoticonStrip';

export interface IEmoticonsWebPartProps {

description: string;

}

export default class EmoticonsWebPart extends BaseClientSideWebPart<IEmoticonsWebPartProps> {

public render(): void {

const element: React.ReactElement = (

<div>

<EmoticonStrip onReactionSelected={this.recordReaction} />

<div>

<Text>Reactions: {this.properties.description}</Text>

</div>

</div>

);

ReactDom.render(element, this.domElement);

}

private recordReaction = (reaction: string): void => {

// Implement logic to record the user's reaction (e.g., send to a SharePoint list).

// Update the description property with the recorded reactions.

this.properties.description = `${this.properties.description} ${reaction}`;

this.render();

}

protected get dataVersion(): Version {

return Version.parse('1.0');

}

protected getPropertyPaneConfiguration(): IPropertyPaneConfiguration {

return {

pages: [

{

header: {

description: 'Emoticon Settings',

},

groups: [

{

groupName: 'Configuration',

groupFields: [

PropertyPaneTextField('description', {

label: 'Reactions',

}),

],

},

],

},

],

};

}

}

**Step 3: Build and Deploy the Web Part**

Build your project using:

gulp bundle --ship

gulp package-solution –ship

This will create a .sppkg file in the "sharepoint" folder of your project.

**Step 4: Deploy the Web Part**

Upload the .sppkg file to your SharePoint App Catalog.

**Step 5: Add the Web Part to a Page**

Navigate to a SharePoint page, edit the page, and add the "Emoticons Web Part" to the page.

**Step 6: Test**

Test the web part by clicking the emoticons, and the reactions should be recorded and displayed in the web part.

This is a basic example to get you started. You will need to expand upon it, add styling, implement server-side logic to store reactions, and handle user permissions as needed. SharePoint development often involves deeper integration with SharePoint features and services, which will depend on your specific requirements.

what will be the sequence diagram for above steps

ChatGPT

A sequence diagram visually represents the interactions between objects or components in a system over time. In the case of the SharePoint web part development described earlier, the primary interactions involve the user, the SharePoint web part, and the SharePoint services. Here's a simplified sequence diagram illustrating the interactions for creating, deploying, and using the Emoticons Web Part:

User Emoticons Web Part SharePoint Services

| | |

| 1. Access SharePoint| |

| | |

|---------------------| |

| |

| 2. Click "Edit Page"| |

| | |

|---------------------| |

| |

| 3. Add "Emoticons | |

| Web Part" | |

| | |

|---------------------| |

| |

| 4. Configure Web | |

| Part Properties | |

| | |

|---------------------| |

| |

| 5. Interact with | |

| Emoticon Strip | |

| (click emoticons)| |

| | |

|---------------------| |

| |

| | 6. Record Reaction (e.g., |

| | update description) |

| | |

|---------------------| |

| |

| 7. Save Page | |

| (Publish) | |

| | |

|---------------------| |

| |

| 8. View Page with | |

| Emoticons | |

| | |

|  |  |
| --- | --- |
|  |  |

Here's a breakdown of the steps illustrated in the sequence diagram:

1. The user accesses the SharePoint site.
2. The user clicks "Edit Page" to start editing the SharePoint page.
3. The user adds the "Emoticons Web Part" to the page.
4. The user configures the web part properties.
5. The user interacts with the Emoticon Strip by clicking emoticons.
6. The Emoticons Web Part records the user's reaction, possibly updating a description property.
7. The user saves the page (publishes it) with the web part and reactions.
8. The user views the published page with the Emoticons Web Part displaying recorded reactions.

Please note that this sequence diagram is a simplified representation, and in practice, there might be additional interactions and services involved, especially when storing and retrieving reaction data, handling user permissions, and ensuring proper rendering of the web part in the SharePoint environment.



Requirement Document for Emoticons Webpart What Needs to be Developed? • An emoticon strip will be provided with icons similar and limited to Viva Engage. The placement of these emoticons can be given through an icon embedded on header or a floating menu. This can be finalized at requirement detailing stage while considering technical feasibility (e.g., merging this into default SharePoint comment bar under a page does not seem feasible at this point). • User can record their reaction for a particular page using these emoticons. We are considering showing counts for reactions instead of disclosing names at this point. This data can be harvested in detail for Analytics purpose as covered in related section. Emoticons on Individual Storyline on Home Page We have considered provision of emoticons at the end of story pages. The emoticons will include Like sign and Happy to Sad faces with total 5 icons in all. We are proposing same emoticons strip for the sections on home page as well, to keep it consistent and also have consistent data for analysis in backend. For shortlisted sections on home page, a brief icon will be displayed on one top corner of the section. When user takes mouse over this icon, the emoticon strip will roll out as feasible based on the content area. User can select icon which will record the reaction. On home page, since it will add clutter, we are NOT considering display of earlier captured reactions (like 50 likes, 20 happy, 10 sad etc.). We are not considering the Emoticons for RSS Feeds. Elaboration has been given in section of RSS Feeds / Newsfeeds. Draw the sequence diagram for above mentioned to implement it in sharepoint using react and typescript.

ChatGPT

Creating a sequence diagram for the requirements you've described in the SharePoint Emoticons Web Part involves visualizing interactions between users, the web part, and the SharePoint services. Below is a sequence diagram that outlines the interactions for implementing the features as described:

User Emoticons Web Part SharePoint Services

| | |

| 1. Access SharePoint site | |

| | |

|---------------------------| |

| |

| 2. Navigate to a page | |

| | |

|---------------------------| |

| |

| 3. Click "Edit Page" | |

| | |

|---------------------------| |

| |

| 4. Add "Emoticons Web Part"| |

| | |

|---------------------------| |

| |

| 5. Configure Web Part | |

| Properties | |

| | |

|---------------------------| |

| |

| 6. Interact with Emoticon | |

| Strip (click emoticons)| |

| | |

|---------------------------| |

| |

| 7. Record Reaction | |

| (e.g., update counts | |

| for reactions) | |

| | |

|---------------------------| |

| |

| 8. Save Page (Publish) | |

| | |

|---------------------------| |

| |

| 9. View Published Page | |

| with Emoticons | |

| | |

|---------------------------| |

| |

| | 10. Store Reaction Data (e.g., |

| | update counts in SharePoint |

| | list for analytics) |

| | |

|---------------------------| |

| |

| 11. Hover over Section | |

| Emoticon Icon | |

| | |

|---------------------------| |

| |

| 12. Emoticon Strip Rolls | |

| Out | |

| | |

|---------------------------| |

| |

| 13. User Selects Emoticon | |

| for Section Reaction | |

| | |

|  |  |
| --- | --- |
|  |  |

This sequence diagram outlines the interactions from the user accessing a SharePoint site, adding the Emoticons Web Part, configuring it, and then interacting with the emoticon strip to record reactions. It also includes interactions for storing reaction data for analytics purposes, as well as the behavior of the Emoticon Strip when it rolls out for section reactions.

Please note that this is a simplified representation, and in a real-world scenario, you would need to consider more details related to the implementation, including handling user authentication, saving data in SharePoint lists, and implementing the UI behaviors. The sequence diagram can serve as a high-level overview of the interactions, and the actual implementation would require more detailed design and coding.