1. What is a Web Part in SharePoint?

Answer:

A Web Part in SharePoint is a reusable component or building block that can be added to pages to display content, data, or functionality. It allows developers and users to customize SharePoint pages without writing extensive code.

2. What are the types of Web Parts in SharePoint?

Answer:

Classic Web Parts: Used in classic SharePoint pages; built using ASP.NET.

Modern Web Parts: Used in modern SharePoint pages; built using SPFx (SharePoint Framework).

Custom Web Parts: Developed by developers using SPFx to add custom functionality.

3. Which framework is commonly used to create custom modern Web Parts in SharePoint?

Answer:

Custom modern Web Parts are typically created using SharePoint Framework (SPFx), which supports TypeScript, React, and Node.js.

4. How can we retrieve data from a SharePoint list in a custom Web Part?

Answer:

We can use:

PnP JS Library for simplified SharePoint REST API calls.

SharePoint REST API directly to fetch list items.

Example:

typescript

Copy code

sp.web.lists.getByTitle("CompanyAnnouncements").items

.orderBy("Created", false)

.top(5)

.get()

.then((items) => {

console.log(items);

});

5. What steps are involved in deploying a custom SPFx Web Part?

Answer:

Develop the Web Part using SPFx.

Test locally using the Workbench.

Bundle and package using gulp bundle --ship and gulp package-solution --ship.

Upload the package to the SharePoint App Catalog.

Deploy the Web Part to the desired SharePoint site.

6. How can filtering functionality be implemented in the Web Part?

Answer:

Add a dropdown or filter UI in the Web Part.

Fetch filtered items using REST API queries or PnP JS:

typescript

Copy code

sp.web.lists.getByTitle("CompanyAnnouncements").items

.filter(Category eq '${selectedCategory}')

.get()

7. What are the best practices for creating custom Web Parts?

Answer:

Use PnP JS for efficient SharePoint data operations.

Keep UI lightweight and responsive.

Minimize API calls to avoid performance issues.

Use modern frameworks like React for dynamic components.

Follow security and governance guidelines for deployment.

**Problem Statement:**

Your organization is migrating to modern SharePoint Online and Microsoft Teams for internal collaboration. The HR and IT departments have requested the following features:

HR Tab in Teams:

HR needs a dedicated tab in Microsoft Teams to display HR Policies stored in SharePoint.

Employees should access policies quickly without navigating to the SharePoint site.

IT Requests Command Extension:

IT team manages requests in a SharePoint list “ITRequests”.

They want Approve and Reject buttons in the SharePoint list view for quick actions.

When a button is clicked, the item’s status should update in the list.

Your task is to analyze, design, and implement these requirements using SharePoint Tabs and Extensions (SPFx).

Use Case:

Title: Employee Self-Service and IT Quick Actions

Actors:

HR Team

IT Team

SharePoint Developer

Preconditions:

HR Policies page exists in SharePoint.

ITRequests list exists with columns: RequestID, Title, Status.

Modern SharePoint environment and Microsoft Teams integration enabled.

Steps:

HR team requests an HR Policies Tab in Teams.

Developer adds a SharePoint Tab in Teams channel linking to the HR Policies page.

IT team requests an extension to approve/reject requests directly in SharePoint list view.

Developer creates a ListView Command Set Extension in SPFx with Approve and Reject commands.

IT team uses the new commands to quickly update request statuses without opening the list item.

Expected Outcome:

Employees can access HR policies from Teams directly.

IT team can approve/reject requests faster, improving efficiency.

SharePoint modern experience remains responsive and lightweight.

**Problem Statement: Review Management System**

Organizations often collect feedback from employees, customers, or stakeholders to evaluate performance, product quality, and service satisfaction. Currently, feedback is collected through emails or manual forms, which leads to delayed responses, lack of tracking, and difficulty in analyzing data.

The management wants a centralized Review Management System that allows users to submit, track, and analyze reviews efficiently.

Requirements:

Review Submission:

Users can submit reviews with Reviewer Name, Date, Department/Product, Rating, and Comments.

Approval Workflow:

Reviews are routed to a manager or admin for approval before publishing.

Categorization & Search:

Reviews are categorized (HR Feedback, Product Review, Service Feedback).

Ability to search and filter by date, category, or rating.

Dashboard & Reports:

Summary of ratings, most-reviewed items, and trends over time.

Monthly reports for management insights.

Security & Permissions:

Only authorized users can approve/reject reviews.

General users can submit and view only approved reviews.

**Questions and Answers**

1. What is the purpose of a Review Management System?

Answer:

The purpose of a Review Management System is to collect, approve, and analyze reviews or feedback in a centralized platform, making it easier for management to track performance, improve services, and take data-driven decisions.

2. What are the key modules required in the system?

Answer:

Review Submission Form (for collecting reviews)

Approval Workflow (for manager/admin verification)

Categorization & Search Module (for filtering reviews)

Dashboard & Reports (for analytics and insights)

Security & Permissions (to control access and approval rights)

3. How can the approval process be implemented in SharePoint or SPFx?

Answer:

Use Power Automate Flow or SharePoint Designer Workflow to route reviews to the manager for approval.

Once approved, the review status can be updated to “Published” and visible in dashboards.

4. How can the system generate insights and reports?

Answer:

Use Power BI dashboards or SharePoint List Views with Grouping and Charts to visualize:

Average rating per department/product

Number of reviews over time

Percentage of positive/negative reviews

5. What are the benefits of implementing this system?

Answer:

Centralized and transparent review management.

Faster approval and publishing of reviews.

Data-driven insights for improving products, services, or employee performance.

Reduced dependency on manual tracking and emails.