TASK-1

<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script type="text/javascript">

$('input[value=Save]').click(function ()

{

PreSaveAction();

});

function PreSaveAction() {

var joinedDesignation = $('select[title="Designation when joined"] option:selected').val();

var designation = $('select[title="Designation"] option:selected').val();

if(joinedDesignation == 'ELT-Entry Level Trainee'){

if(designation != 'PAT-Programmer Analyst Trainee'){

alert("Please select higher designation.");

return false;

}

else {

return true;

alert("Details saved.");

}

}

else if(joinedDesignation == 'PAT-Programmer Analyst Trainee'){

if(designation != 'A-Associate'){

alert("Please select higher designation.");

return false;

}

else{

return true;

alert("Details saved.");

}

}

else if(joinedDesignation == 'A-Associate'){

if(designation != 'SA-Senior Associate'){

alert("Please select higher designation.");

return false;

}

else{

return true;

alert("Details saved.");

}

}

else if(joinedDesignation == 'SA-Senior Associate'){

if(designation != 'M-Manager'){

alert("Please select higher designation.");

return false;

}

else{

return true;

alert("Details saved.");

}

}

}

</script>

**TASK-2**

<button onclick="GetListItems()">Update Salary List</button>

<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script type="text/javascript">

function GetListItems() {

$.ajax({

async: true,

url: "https://maneeshachebrolu.sharepoint.com/sites/Assignment1/HR/\_api/web/lists/GetByTitle('Designation')/items",

method: "GET",

headers: {

"accept": "application/json;odata=verbose",

"content-type": "application/json;odata=verbose"

},

success: function(data) {

data = data.d.results;

for( var i=0; i<data.length; i++){

InsertItems(i,data);

}

},

error: function(error) {

alert(JSON.stringify(error));

}

});

}

function InsertItems(i,designationData) {

var id = (designationData[i].ID).toString();

var name = designationData[i].Title;

var desig = designationData[i].Designation;

if(desig == 'ELT-Entry Level Trainee'){

$.ajax({

async: true,

url: \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/GetByTitle('Salary')/items",

method: "POST",

data: JSON.stringify({

'\_\_metadata': {

'type': 'SP.Data.SalaryListItem'

},

'EmployeeID': id,

'Title': name,

'Designation': desig,

'MinimumSalary': 10000,

'MaximumSalary' : 20000

}),

headers: {

"accept": "application/json;odata=verbose",

"content-type": "application/json;odata=verbose",

"X-RequestDigest": $("#\_\_REQUESTDIGEST").val()

},

success: function(data) {

alert("List updated successfully", "success");

},

error: function(error) {

alert(JSON.stringify(error));

}

});

}

else if(desig == 'PAT-Programmer Analyst Trainee'){

$.ajax({

async: true,

url: \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/GetByTitle('Salary')/items",

method: "POST",

data: JSON.stringify({

'\_\_metadata': {

'type': 'SP.Data.SalaryListItem'

},

'EmployeeID': id,

'Title': name,

'Designation': desig,

'MinimumSalary': 25000,

'MaximumSalary' : 40000

}),

headers: {

"accept": "application/json;odata=verbose",

"content-type": "application/json;odata=verbose",

"X-RequestDigest": $("#\_\_REQUESTDIGEST").val()

},

success: function(data) {

alert("List updated successfully", "success");

},

error: function(error) {

alert(JSON.stringify(error));

}

});

}

else if(desig == 'A-Associate'){

$.ajax({

async: true,

url: \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/GetByTitle('Salary')/items",

method: "POST",

data: JSON.stringify({

'\_\_metadata': {

'type': 'SP.Data.SalaryListItem'

},

'EmployeeID': id,

'Title': name,

'Designation': desig,

'MinimumSalary': 45000,

'MaximumSalary' : 60000

}),

headers: {

"accept": "application/json;odata=verbose",

"content-type": "application/json;odata=verbose",

"X-RequestDigest": $("#\_\_REQUESTDIGEST").val()

},

success: function(data) {

alert("List updated successfully", "success");

},

error: function(error) {

alert(JSON.stringify(error));

}

});

}

else if(desig == 'SA-Senior Associate'){

$.ajax({

async: true,

url: \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/GetByTitle('Salary')/items",

method: "POST",

data: JSON.stringify({

'\_\_metadata': {

'type': 'SP.Data.SalaryListItem'

},

'EmployeeID': id,

'Title': name,

'Designation': desig,

'MinimumSalary': 65000,

'MaximumSalary' : 80000

}),

headers: {

"accept": "application/json;odata=verbose",

"content-type": "application/json;odata=verbose",

"X-RequestDigest": $("#\_\_REQUESTDIGEST").val()

},

success: function(data) {

alert("List updated successfully", "success");

},

error: function(error) {

alert(JSON.stringify(error));

}

});

}

else if(desig == 'M-Manager'){

$.ajax({

async: true,

url: \_spPageContextInfo.webAbsoluteUrl + "/\_api/web/lists/GetByTitle('Salary')/items",

method: "POST",

data: JSON.stringify({

'\_\_metadata': {

'type': 'SP.Data.SalaryListItem'

},

'EmployeeID': id,

'Title': name,

'Designation': desig,

'MinimumSalary': 85000,

'MaximumSalary' : 100000

}),

headers: {

"accept": "application/json;odata=verbose",

"content-type": "application/json;odata=verbose",

"X-RequestDigest": $("#\_\_REQUESTDIGEST").val()

},

success: function(data) {

alert("List updated successfully", "success");

},

error: function(error) {

alert(JSON.stringify(error));

}

});

}

}

</script>

 #######EditForm.aspx#######  
<script type="text/javascript" src="<https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js>"></script>  
<script type="text/javascript">  
$('input[value=Save]').click(function ()    
    {    
        PreSaveAction();  
    });    
function PreSaveAction() {  
var minSalary = $('input[title="MinimumSalary"]').val();  
var maxSalary = $('input[title="MaximumSalary"]').val();  
var salary = $('input[title="Salary"]').val();  
if(salary < minSalary || salary > maxSalary){  
alert("Please enter salary within range.");  
return false;  
}  
else{  
return true;  
}  
}  
</script>

TASK-3

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

import {

  IPropertyPaneConfiguration,

  PropertyPaneTextField

} from '@microsoft/sp-property-pane';

import { BaseClientSideWebPart } from '@microsoft/sp-webpart-base';

import { SPHttpClient, ISPHttpClientOptions, SPHttpClientResponse } from '@microsoft/sp-http';

import { Web, List, ItemAddResult } from "sp-pnp-js";

import { escape } from '@microsoft/sp-lodash-subset';

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

import \* as strings from 'TravelRequestsWebPartStrings';

export interface ITravelRequestsWebPartProps {

  description: string;

}

export interface IListItem

{

  value: IListItem[];

}

export interface IListItem

{

  EmployeeId: string;

  Title: string;

  DateOfTravel: string;

  Place :string;

  Reason :string;

}

export default class TravelRequestsWebPart extends BaseClientSideWebPart<ITravelRequestsWebPartProps> {

  public render(): void {

    this.domElement.innerHTML = `

      <div class="${ styles.travelRequests }">

        <div class="${ styles.container }">

          <div class="${ styles.row }">

            <div class="${ styles.column }">

            <table>

            <tr>

            <td>Employee ID:</td>

            <td><input type="text" id="employeId"></td>

           </tr>

           <tr>

            <td>Employee Name:</td>

            <td><input type="text" id="employeName"></td>

           </tr>

           <tr>

            <td>DateOfTravel</td>

            <td><input type="date" id="dateOfTravel"></td>

           </tr>

           <tr>

            <td>Place</td>

            <td><input type="text" id="place"></td>

           </tr>

           <tr>

            <td>Reason</td>

            <td><input type="text" id="reason"></td>

           </tr>

           <tr><td><button id="saveItem">SaveItem</button></td></tr>

           </table>

            </div>

          </div>

        </div>

      </div>`;

      this.setButtonEventListeners();

  }

  private setButtonEventListeners ():void {

    const webPart : TravelRequestsWebPart = this;

    this.domElement.querySelector('#saveItem').addEventListener('click',()=>{webPart.FormValidation();});

  }

  private FormValidation(){

    var alertMessage="";

    if(document.getElementsByTagName("input")[1].value==""){

        alertMessage+=" Employee Id cannot be Empty !\n";

    }

    if(document.getElementsByTagName("input")[2].value==""){

        alertMessage+=" Name cannot be Empty !\n";

    }

    if(document.getElementsByTagName("input")[3].value==""){

        alertMessage+=" Date of Journey cannot be Empty !\n";

    }

    if(document.getElementsByTagName("input")[4].value==""){

        alertMessage+="> Enter a Valid place Name !\n";

    }

    if(document.getElementsByTagName("input")[5].value==""){

        alertMessage+="> Give the reason for the journey!";

    }

    if(alertMessage==""){

      this.Control();

    }

    else{

      alert(alertMessage);

    }

  }

  private Control(): void{

    this.getListData();

    this.CheckListAsync();

  }

  private CheckListAsync(): void

    {

      this.getListData().then((response) => { this.CheckList(response.value); });

    }

  private CheckList(items: IListItem[]): void {

    var count=0;

    items.forEach((item: IListItem) => {

      if( item.EmployeeId == document.getElementsByTagName("input")[1].value && item.DateOfTravel == document.getElementsByTagName("input")[3].value)

      {

          alert("You have another request on that day!");

          count=1;

      }

    });

    if(count==0){

      this.AddItem();

    }

    }

  private getListData(): Promise<IListItem>

  {

   return this.context.spHttpClient.get( "<https://maneeshachebrolu.sharepoint.com/sites/Assignment1/Travel/_api/web/lists/GetByTitle('TravelRequests')/Items>",SPHttpClient.configurations.v1)

       .then((response: SPHttpClientResponse) =>

       {

       return response.json();

       });

   }

   protected AddItem(): void {

    const web: Web = new Web(this.context.pageContext.web.absoluteUrl);

    web.lists.getByTitle("TravelRequests").items.add({

      'EmployeeId': document.getElementsByTagName("input")[1].value,

      'Title': document.getElementsByTagName("input")[2].value,

      'DateOfTravel': document.getElementsByTagName("input")[3].value,

      'Place': document.getElementsByTagName("input")[4].value,

      'Reason': document.getElementsByTagName("input")[5].value,

    }).then((result: ItemAddResult): void => {

      alert("Item Created Successfully");

      const item: IListItem = result.data as IListItem;

    });

  }

  protected get dataVersion(): Version {

    return Version.parse('1.0');

  }

  protected getPropertyPaneConfiguration(): IPropertyPaneConfiguration {

    return {

      pages: [

        {

          header: {

            description: strings.PropertyPaneDescription

          },

          groups: [

            {

              groupName: strings.BasicGroupName,

              groupFields: [

                PropertyPaneTextField('description', {

                  label: strings.DescriptionFieldLabel

                })

              ]

            }

          ]

        }

      ]

    };

  }

}