General Sir John Kotelawala Defence University

Faculty of Management, Social Sciences and Humanities

Department of Languages

BSc in Applied Data Science Communication

Advanced SQL and Cloud Databases / LB2224

Group Assignment

P Laksia – D/ADC/23/0013

PRM Perera – D/ADC/23/0031

MM Jayasinghe – D/ADC/23/0035

RYN Sanduprabha – D/ADC/23/0046

Advanced SQL and Cloud Databases / LB 2224

Year 2: Semester 1

Assignment number 1

15.12.2024

Content

Employee Leave Requests

1. Introduction and Overview

Employee leave management is a crucial aspect in the workplace administration which ensures the effective management of the employee absences while minimizing the disruptions to business operations. The traditional method where organizations use manual methods in handling and updating employee leave requests are often considered inefficient since it is a time-consuming method which is also suspectable to errors for Human Resource (HR) teams and other employees.

This project aims to streamline the employee leave management process by leveraging **Microsoft Power Apps** and **Microsoft Power Automate**. The solution includes a custom leave request application and an automated workflow for request approval, calendar updates and notifications. With the implementation of this solution, organizations will be able to improve the efficiency in the operations of the organization, enhance communication, reduce time consumption, reduce the administrative burden on HR teams and other employees.

2. Project Description

2.1. Scenario Description

At present, most companies and organizations utilize manual methods in recording and updating employee leave requests which have led to many inefficiencies such as:

i. **Data entry errors** - where HR teams and other employees are prone to errors on regards to entering data about the details of the employees' leave requests.

ii. **Time consumption** – due to manually reviewing the requests to approve or deny them, updating employee calendars and notifying respective employees about the approved leaves, it consumes a lot of time.

iii. **Decrease in productivity** – delays in the steps in employee leave management can lead to miscommunication, conflicts in scheduling leaves to employees which in turn increases the frustration of employees and managers likewise can cause to the decrease in the productivity of all the employees involved during the process and the organization indirectly.

The proposed solution addresses the concerns or the challenges by automating the employee leave request process where the employee will be able to apply for an employee leave request through a customized Power App, where an automated workflow handles the approval routing, updating the leave balances, and notifications through Power Automate. This solution will provide a streamlined effective employee leave request process, ensuring timely communication, accurate record keeping and increased productivity throughout the process.

2.2. Requirements

Functional requirements

i. **Power Apps: Leave Request Application**

User friendly interface for employees to:

- Submit leave requests with details (dates, leave type, and reason).
- View the status of submitted requests.

Manager view to:

- Review and approve or deny leave requests.

ii. **Power Automate: Automate Workflow**

- A workflow which is triggered by new leave request submissions.
- Sends notifications to managers to be reviewed.
- Tracks the status of the employee leave requests – whether they are Pending, Approved or Denied.
- Updates the employee's leave balance upon the approval of the leave requests.
- Sends notifications to the relevant teams and employees about the approved or declined leave.

Non - Functional requirements

i. **Scalability:** The solution should be able to handle multiple requests simultaneously.

ii. **Accessibility:** Both employees and managers should have the ability to use the app on mobile and desktop devices.

iii. **Security:** Protects the sensitive data of employees while limiting the access to relevant roles depending on their position in the workplace hierarchy. Example: employees, managers and HR team.

iv. **Reliability:** Ensure that the automated workflows execute without any errors or delays.

v. **Usability:** A simple, user-friendly and creative interface for all users.

Technical requirements

i. **Data Source:** Storing employee information, leave records and approval statuses using SharePoint and Office365 Users.

ii. **Integrations:** Microsoft Outlook and Microsoft Teams are used for email notifications and updates.

iii. **Power Platform Contents:**

- Power Apps: For front to end leave request application.
- Power Automate: For backend automated workflow.

3. Task 01 – Power Apps

3.1. Design Details

The Power App is designed to simplify the submission, approval and tracking of employee leave requests. The app consists of the following components:

User Roles:

i. **Employee View:**

- A request form to submit new leave requests.
- A dashboard to view the pending, approved and denied requests
- A dashboard to view the employee's leave balance list.
- A page view which provides information about the vacation types that the organization provides and another page view which provides details about the company holidays.

ii. **Manager View:**

- A dashboard to view the pending requests assigned to them along with the approved and rejected requests.
- Approve or deny requests of the employees.
- A page view which provides information about the vacation types that the organization provides and another page view which provides details about the company holidays.

Screens:

i. **Home Screen:**

- Navigation buttons for employees to submit requests and view request status.
- Navigation for managers to review requests.

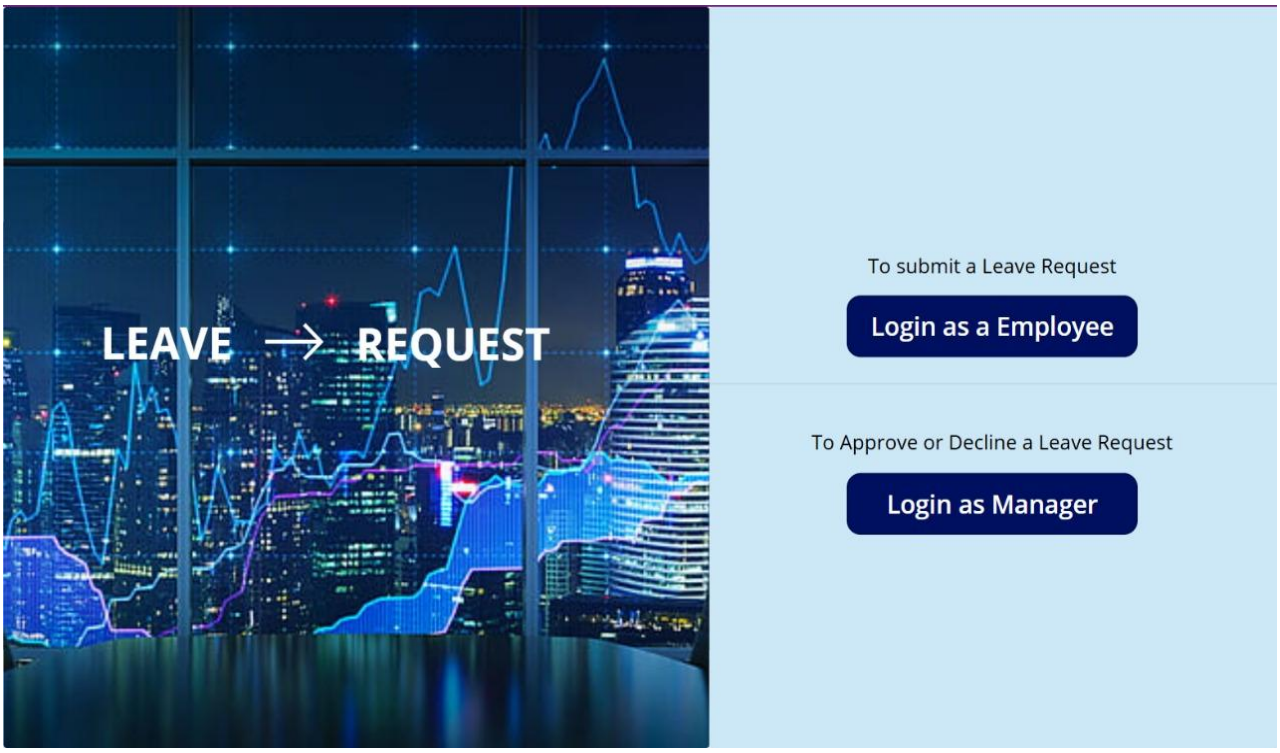

Figure 01

ii. **Create a Request Screen:**

- Form fields such as;
  a. **Employee Name** – Auto filled based on the login credentials.
  b. **Leave Type** – A gallery to select the type of leave (**Annual Leave, Casual Leave, Sick Leave, Maternity Leave, Paternity Leave, Public Leave, Special Leave**).
  c. **Start Date** and **End Date** – Date pickers.
  d. **Reason for Leave** – A single line text box and a multiline text box where the details of the leave request are mentioned.
  e. **Submit request to** – the person the employee is requesting the leave from.
- Submit button to send the request to the manager.
- Reset button to clear the request form.
- Back button navigates back to the employees **Leave Balance** page view.

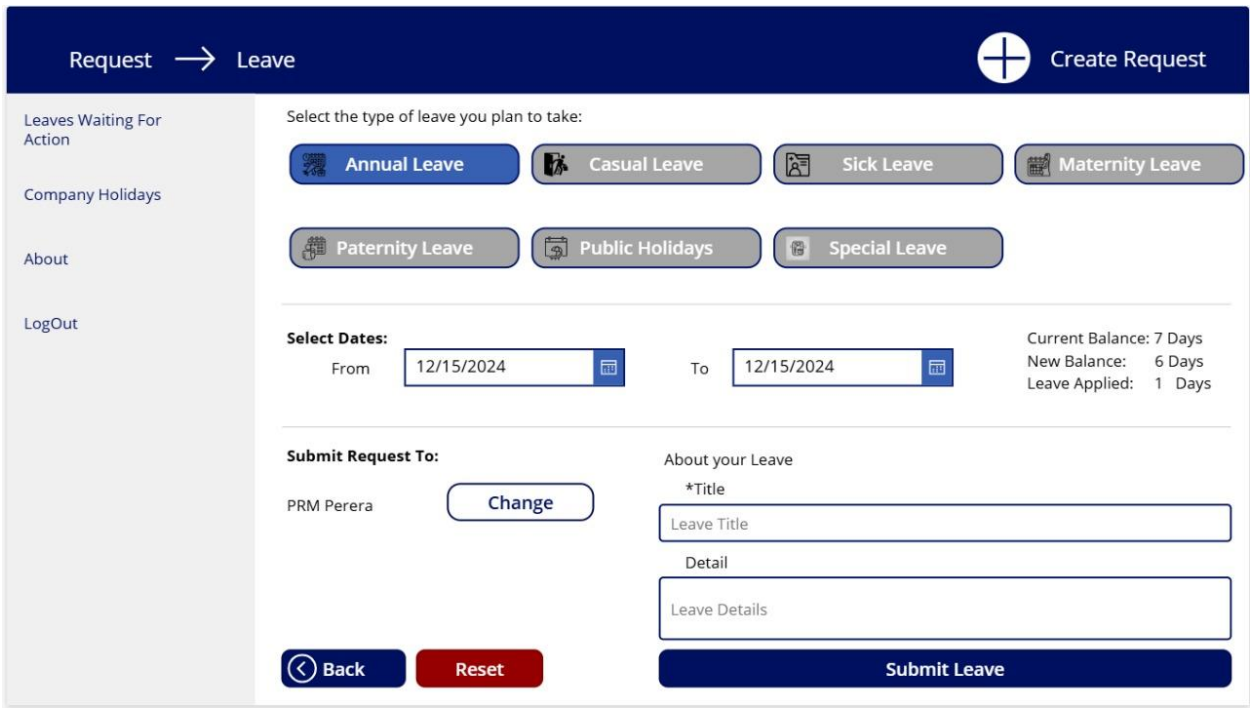

Figure 02

iii. **My Leave Requests Screen:**

- Gallery displaying the details of the employees' leave requests along with their statuses (**Pending, Approved, Denied**).
- An option to reset the request option.

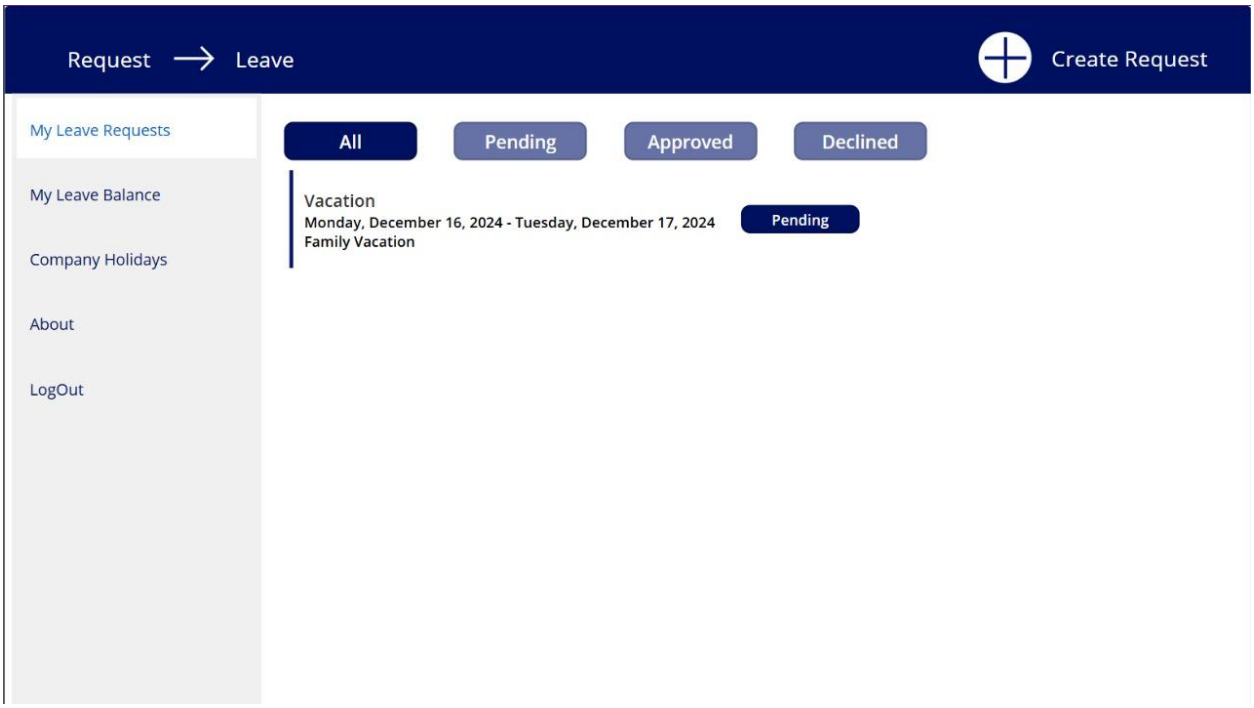

Figure 03

iv. **My Leave Balance Screen:**

- Provides all the leave types to the employee while providing the number of days available to that employee.
- A leave request form.
- An information button which provides information about the leave types.

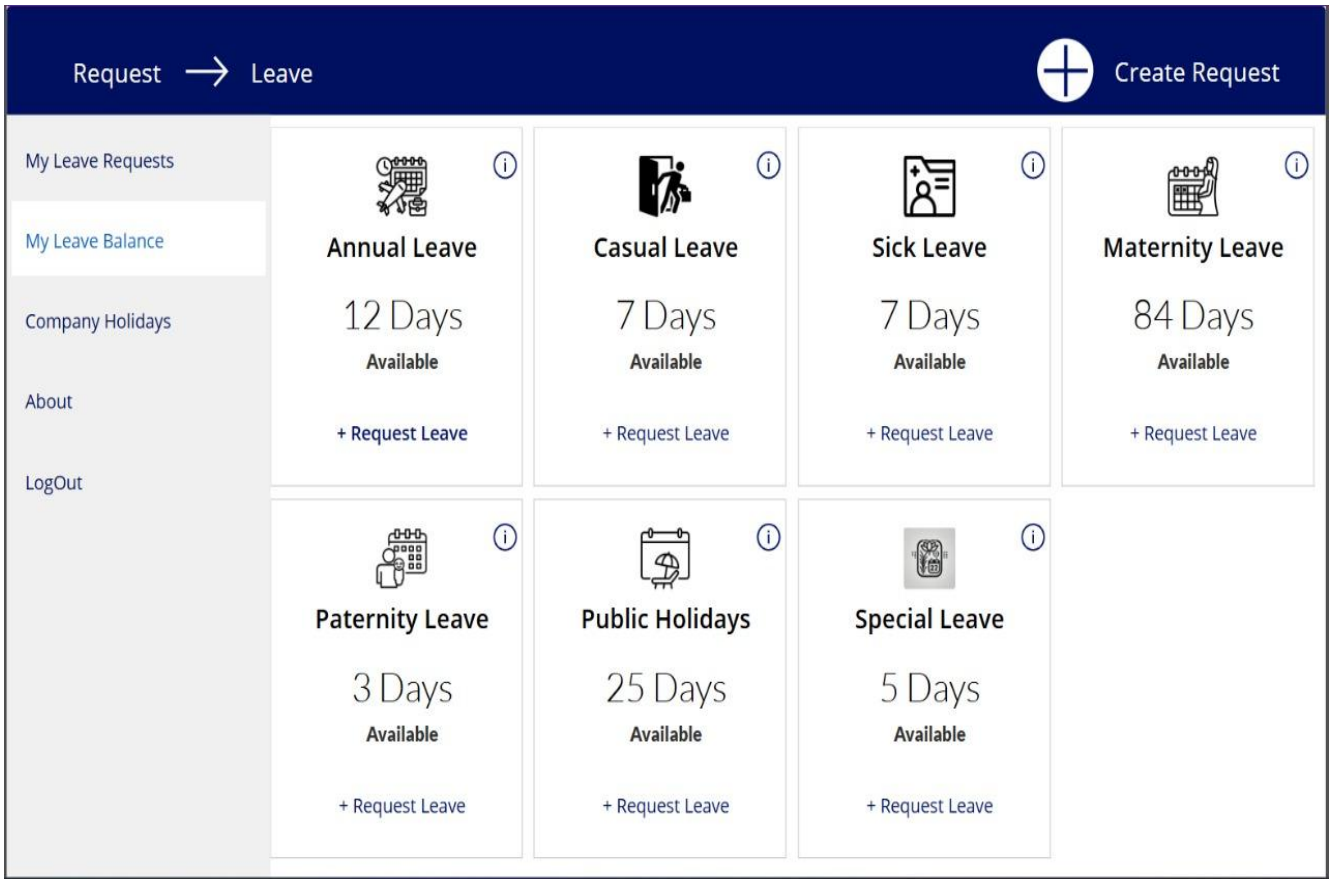

Figure 04

v. **Company Holidays:**

- Provides the details about the paid holidays provided by the organization which are observed during the closed days of the company.

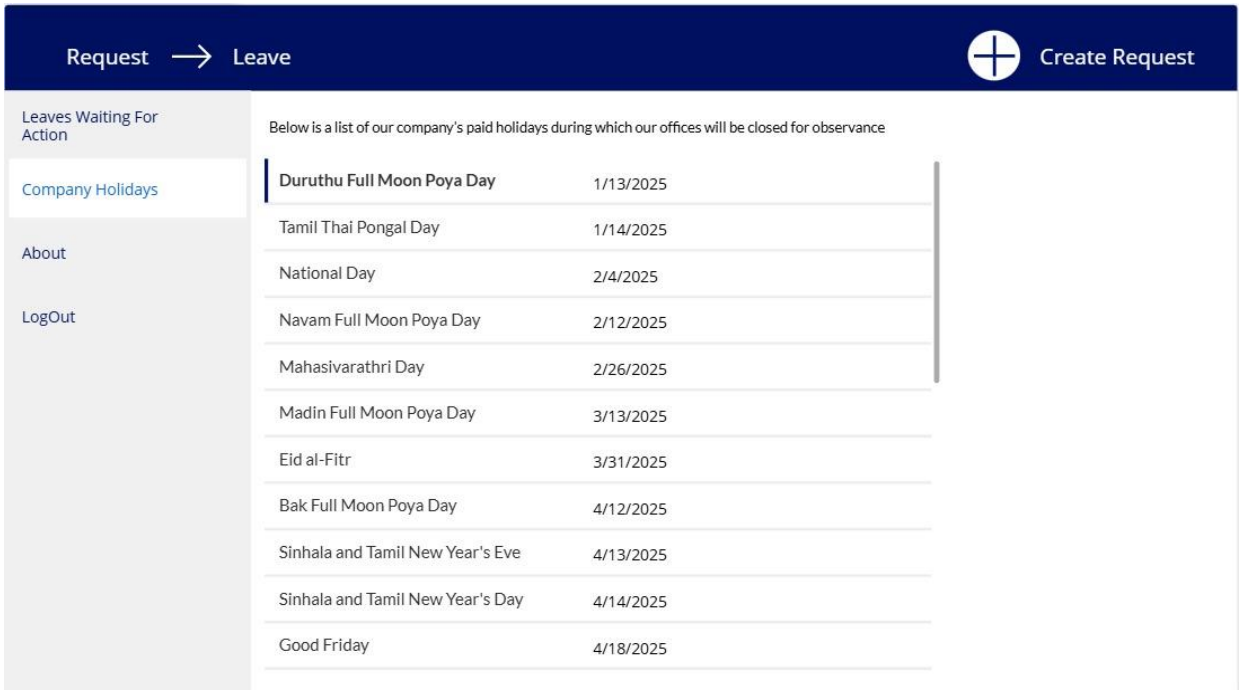

Figure 05

vi. **About:**

- A page view which provides information about the vacation types that the organization provides.

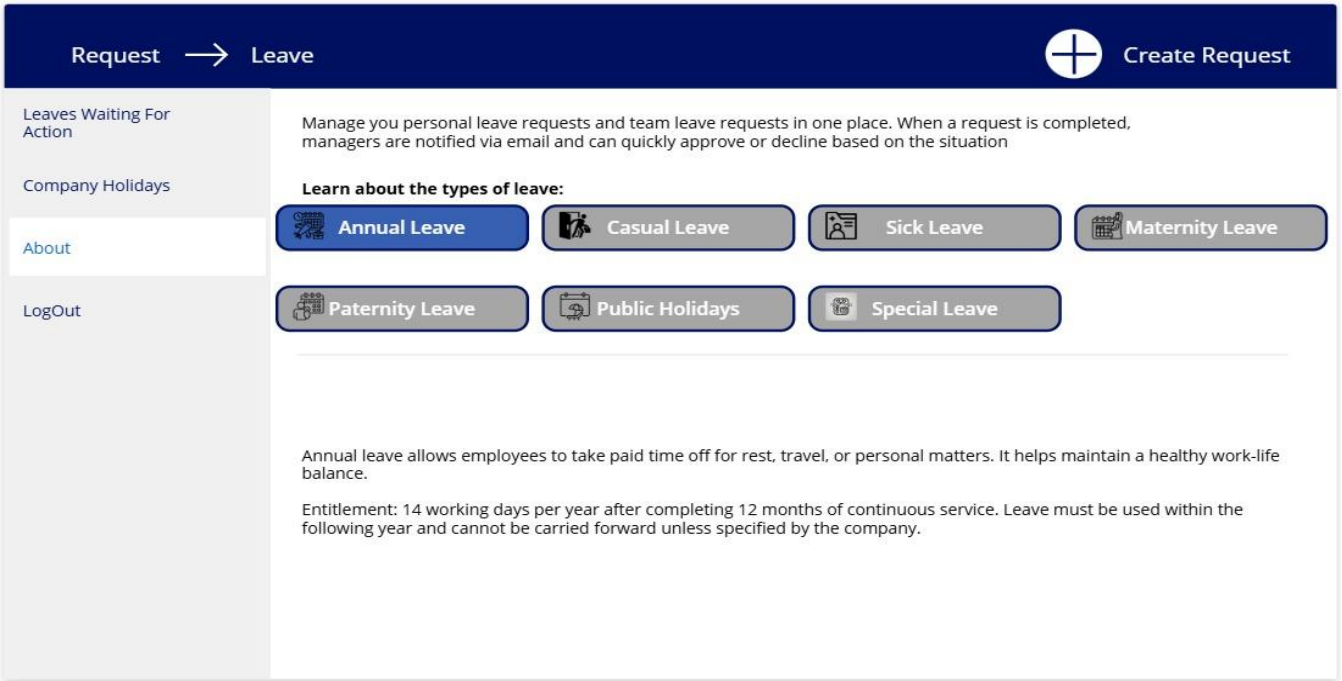

Figure 06

vii. **Manager Dashboard Screen:**

- A gallery displaying all pending requests for manager's review.
- Buttons to approve or deny employee requests.

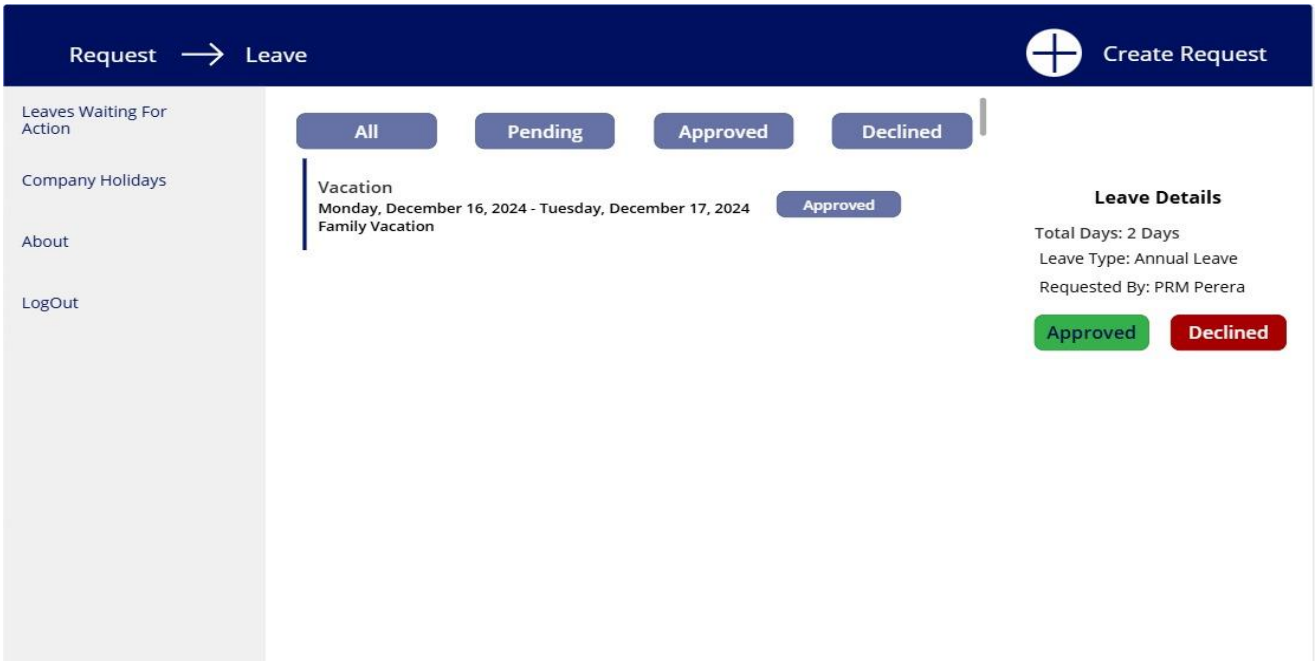

Figure 07

Data Source:

Three SharePoint lists were created as the data sources for the app. The lists created along with the column details are mentioned below:

i. **Balance List:**

- Leave Type – Type of leave taken (**Annual Leave, Casual Leave, Sick Leave, Maternity Leave, Paternity Leave, Public Leave, Special Leave**).
- Total Leave – Total leaves available for the employee.
- Applied Leave – The number of leaves applied.
- Balance Leave – The number of leaves remaining.
- Leave Description – The description of the leave requested.
- Leave ID – Unique Identification for each leave.

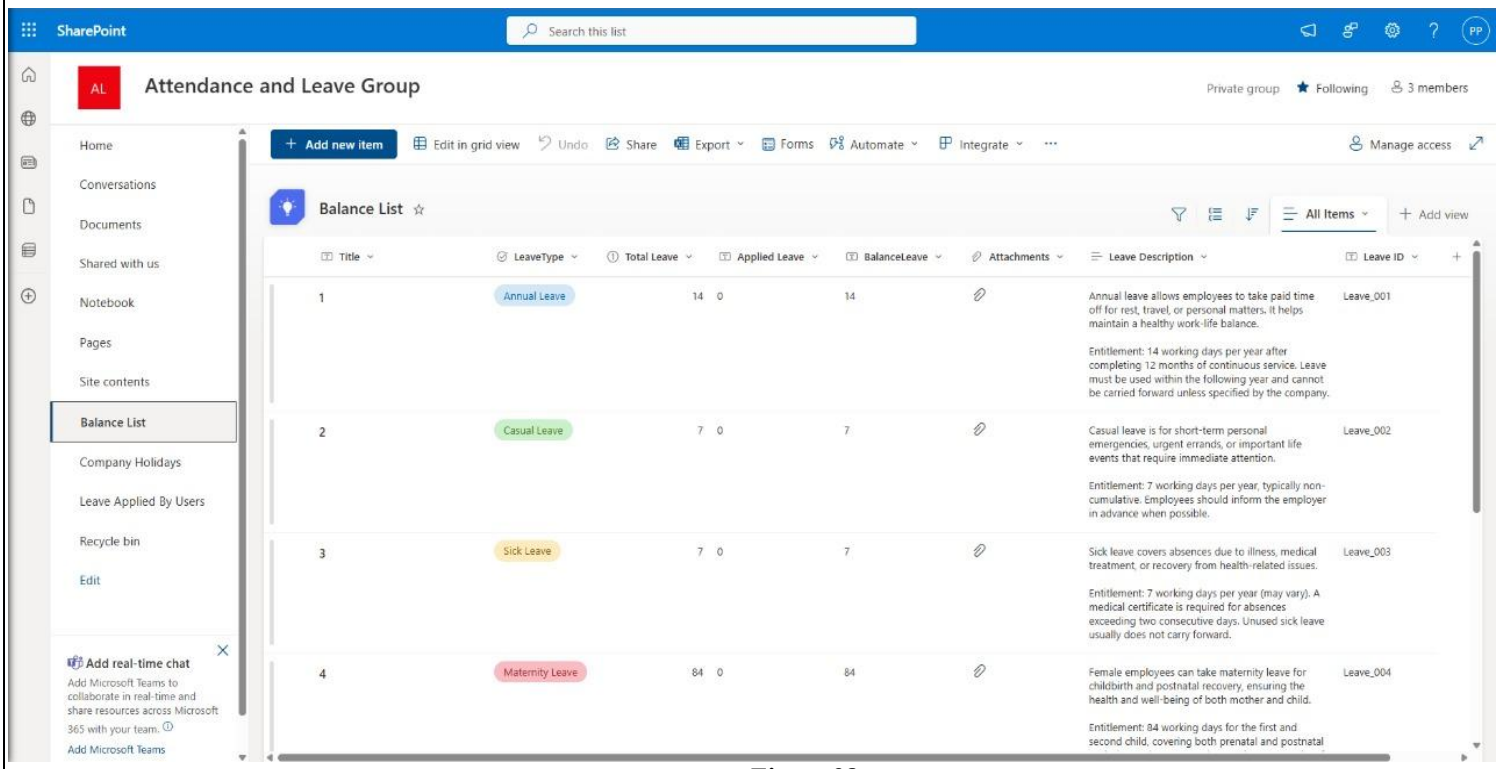

Figure 08

ii. **Company Holidays:**

- Holiday – Type of paid holidays provided by the organization which are observed during the closed days of the company.
- Date – The date in which the holiday falls in the calendar.

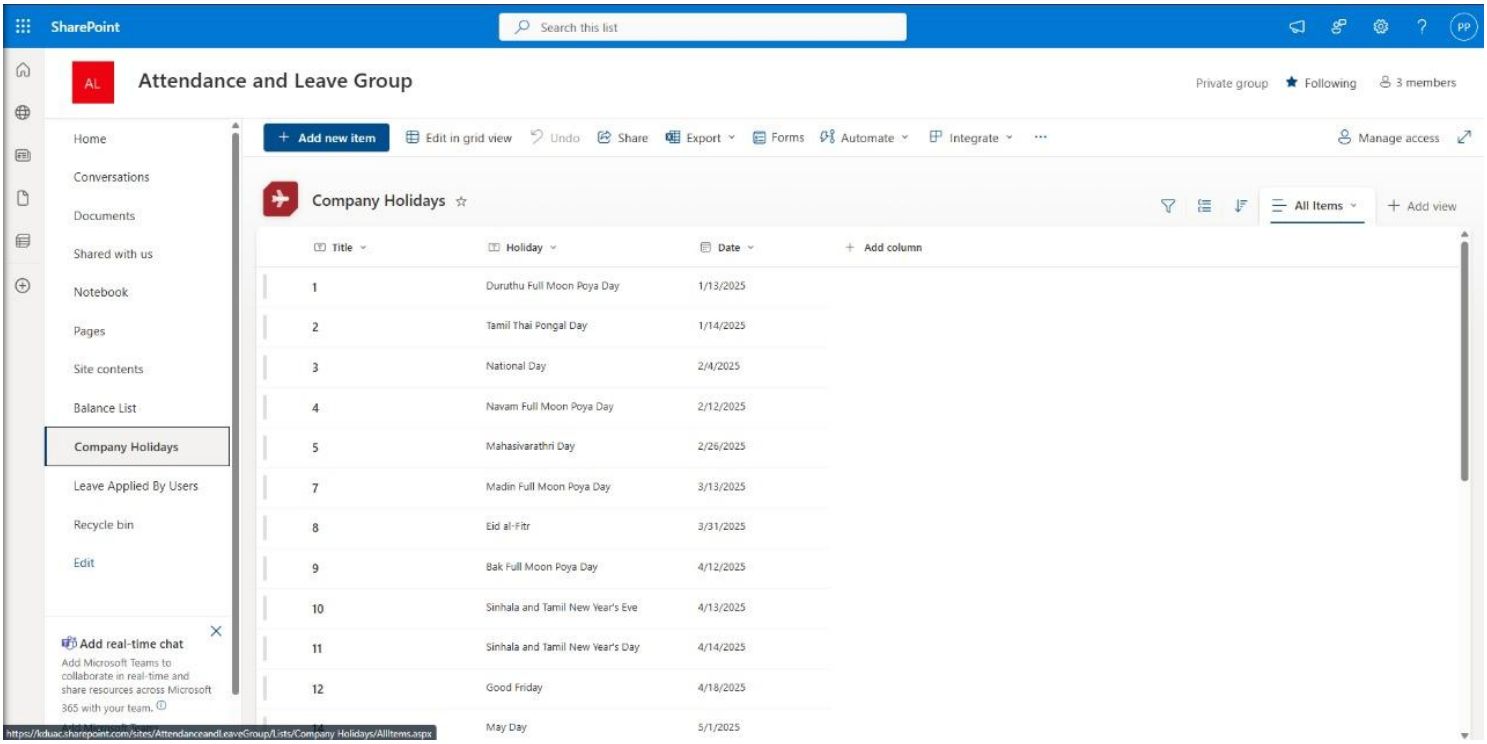

Figure 09

iii. **Leave Applied by Users:**

- Leave ID – Unique Identification for each leave.
- Leave Type – Type of leave taken (**Annual Leave, Casual Leave, Sick Leave, Maternity Leave, Paternity Leave, Public Leave, Special Leave**).
- Requested By – The name of the employee who is requesting the leave which is auto filled based on the login credentials.
- Requested To – The name of the person the employee is requesting the leave from.
- From Date – The starting date of the leave request.
- To Date – The ending date of the leave request.
- Total Days – The total number of days requested for the leave.
- Leave Title – The main subject title of the leave requested.
- Leave Description – The detailed description of the leave requested.
- Leave Status – The status of the leave requesting (**Approved, Pending, Denied**).

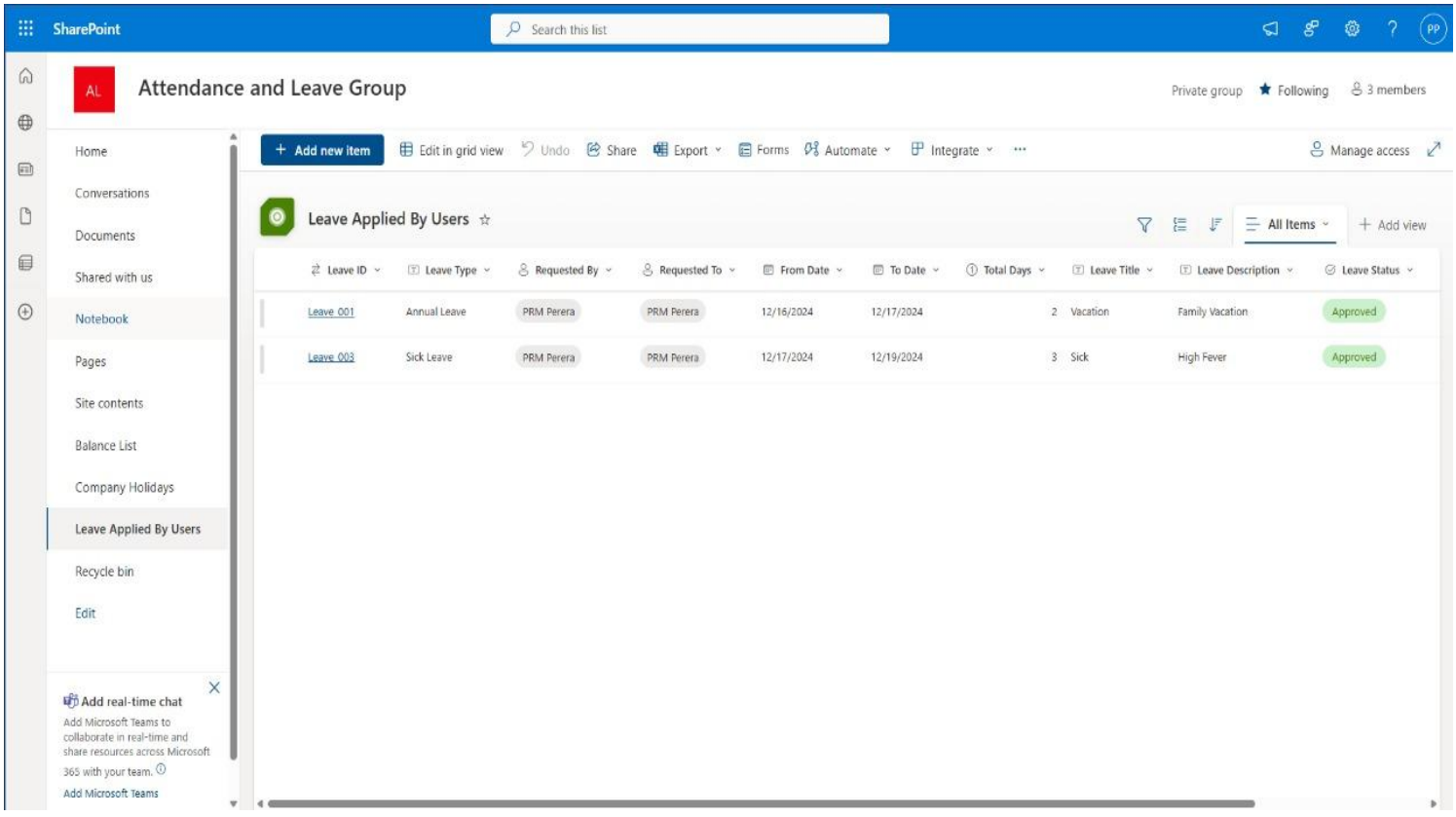

Figure 10

3.2. Implementation Process

Data Source Setup:

Create three SharePoint lists with columns for the necessary data points. The SharePoint Lists are mentioned in **Section 3.1. Design Details**. The SharePoint lists are mentioned as follows:

i. Balance List **(*Figure 08*)**.
ii. Company Holidays **(*Figure 09*)**.
iii. Leave Applied by Users **(*Figure 10*)**.

App Creation:

The **Canvas App** is used to create a customized Employee Leave Request App. After the Canvas App from Power Apps Studio is opened the three SharePoint Lists are created in order to setup the Data Source for the App.

Screen Design:

Seven separate screens were designed for the app which are mentioned in **Section 3.1. Design Details**. The Screens designed are mentioned as follows:

i. Home Screen **(*Figure 01*)**.
ii. Create a Request Screen **(*Figure 02*)**.
iii. My Leave Requests Screen **(*Figure 03*)**.
iv. My Leave Balance Screen **(*Figure 04*)**.
v. Company Holidays **(*Figure 05*)**.
vi. About **(*Figure 06*)**.
vii. Manager Display Screen **(*Figure 07*)**.

App Screen View Design:

The first screen which is the Home Screen where two login buttons namely, **Login as a Employee** and **Login as Manager** are designed. These buttons are the main components of this screen.

Figure 11

The app screen view design is consistent for the whole app. It consists of navigations, components, collections and galleries.

Figure 12

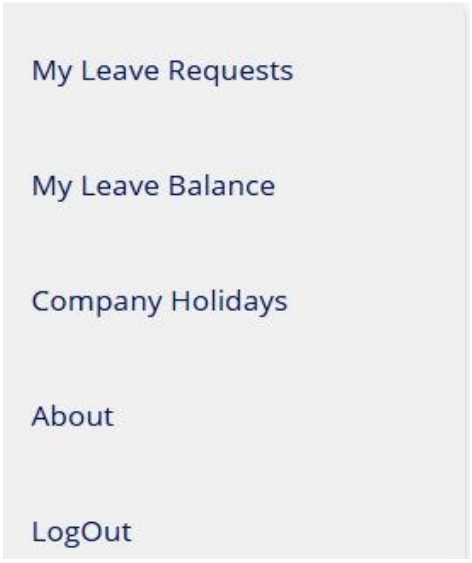

Figure 13

Figure 14

For the above navigation designed to work, a custom property was created as a *NavigationItem*. A closer look of the code for the NavigationItem is as follows:

```
Table({Title: "My Leave Requests", TargetScreen: App.ActiveScreen},{Title: "My Leave Balance", TargetScreen: App.ActiveScreen},{Title: "Company Holidays", TargetScreen: App.ActiveScreen},{Title: "About", TargetScreen: App.ActiveScreen},{Title: "LogOut", TargetScreen: App.ActiveScreen})
```

Figure 15

A gallery is added to the above navigation component where when selecting an item from the on select the code is as follows:

```
Navigation.NavigationItem
```

Figure 16

In order to navigate and transition to the selected item from on select the following code is utilized.

```
true; Navigate(ThisItem.TargetScreen,ScreenTransition.Fade)
```

Figure 17

A *ClearCollect* function is utilized to navigate menus for managers and employees with titles and target screens.

```
ClearCollect(
    NavManagerView,
    {
        Title: "Leaves Waiting For Action",
        TargetScreen: 'Leaves Waiting For Action'
    },
    {
        Title: "Company Holidays",
        TargetScreen: 'Company Holidays'
    },
    {
        Title: "About",
        TargetScreen: About
    },
    {
        Title: "LogOut",
        TargetScreen: Home
    }
);ClearCollect(
    NavEmployeeView,
    {
        Title: "My Leave Requests",
        TargetScreen: 'My Leave Requests'
    },
    {
        Title: "My Leave Balance",
        TargetScreen: 'My Leave Balance'
    },
    {
        Title: "Company Holidays",
        TargetScreen: 'Company Holidays'
    },
    {
        Title: "About",
        TargetScreen: About
    },
    {
        Title: "LogOut",
        TargetScreen: Home
    }
```

Figure 18

To all the Screens that has the Navigation Menu where the NavigationItem is present this code is given as follows:

```
If(ManagerView=false,NavEmployeeView,NavManagerView)
```

Figure 19

Another gallery is introduced where a code is used to filter the request status to the given buttons. The code is as follows:

```
Table({Filter:"All"},{Filter:"Pending"},{Filter:"Approved"},{Filter:"Declined"})
```

Figure 20

The following code is used to filter and collect leave requests for the logged-in user based on the selected filter from the gallery button where, if the Filter is set to All, it collects all leave requests made by the logged-in user (*User().Email*). If a specific filter like Pending, Approved, or Declined is selected, it collects only the leave requests that match the selected status (*Leave Status.Value*) and are made by the logged-in user. The results are stored in the *MyLeaveColl* collection for display.

```
If(ThisItem.Filter="All",ClearCollect(MyLeaveColl,Filter('Leave Applied By Users','Requested By'.Email=User().Email)),
ClearCollect(MyLeaveColl,Filter('Leave Applied By Users','Leave Status'.Value=ThisItem.Filter,'Requested By'.Email=User().Email)));
```

Figure 21

When there is no available data, a code is introduced to show the output **No Record Found**.

```
If(IsEmpty(MyLeaveColl),UpdateContext({NoData:"No Record Found"}),UpdateContext({NoData:""}));
```

Figure 22

A gallery is created in the Company Holidays Screen where the Company Holidays SharePoint List is connected to the galary using the following code:

```
'Company Holidays_1'
```

Figure 23

To get the labels of the gallery the following codes are used:

```
ThisItem.Holiday

ThisItem.Date
```

Figure 24

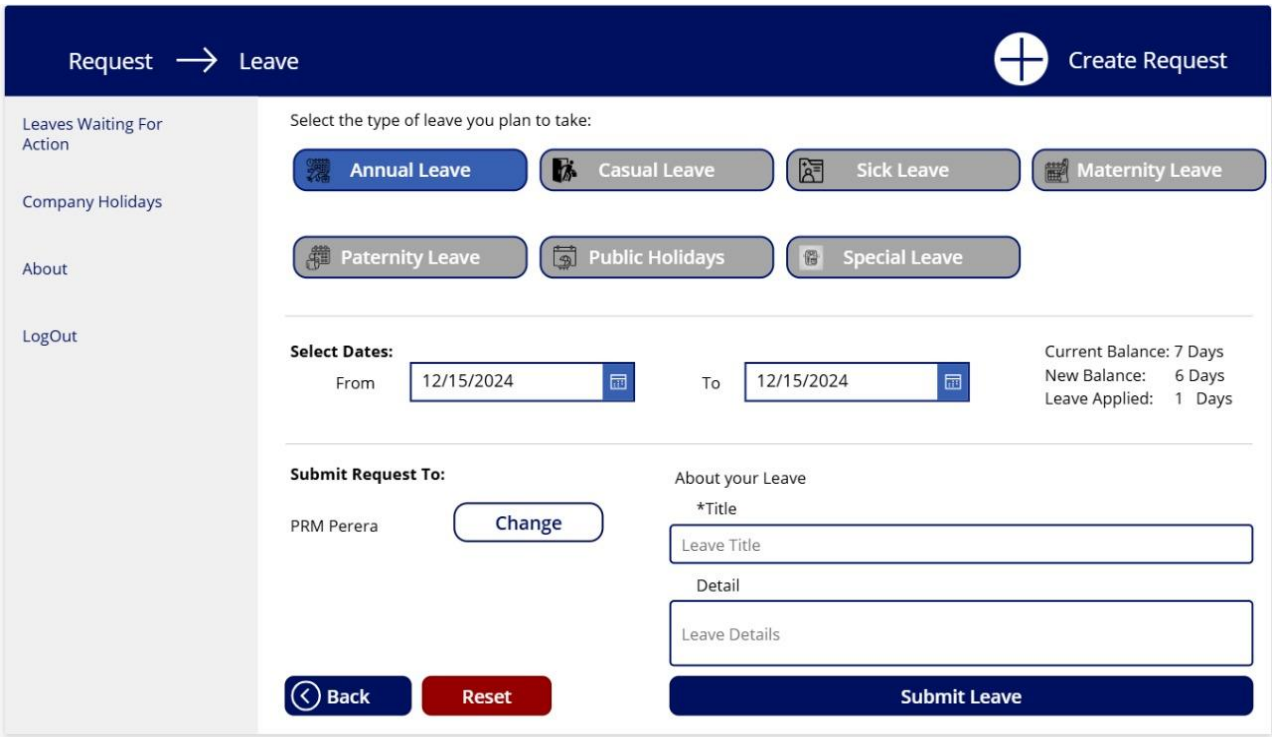

Figure 2

A gallery is added by connecting to the Balance List Share Point where the user will be able to select the type of leave he/ she wants. The following code is used to allow the user to select the type of leave.

```
ClearCollect(LeaveSum,Sum(Filter('Leave Applied By Users','Leave ID'.Value=Gallery5.Selected.'Leave ID' && 'Requested By'.
Email=User().Email),'Total Days'));

ClearCollect(TotalLeave,Sum(Filter('Balance List', 'Leave ID'=Gallery5.Selected.'Leave ID'),ThisItem.'Total Leave'));
```

Figure 25

Two date pickers are included for the user to select the starting and the ending dates of the leave which is shown in the code given below.

```
UpdateContext({daysdiff: RoundDown(DateDiff(StartDate.SelectedDate, EndDate.SelectedDate, TimeUnit.Days)/7,0)*5+
Mod(5+Weekday(EndDate.SelectedDate)-Weekday(StartDate.SelectedDate),5)})
```

Figure 26

The default for the two date pickers re shown as **Today().**

```
Today()
```

Figure 27

The code for the OnChange is as follows.

```
UpdateContext({daysdiff: RoundDown(DateDiff(StartDate.SelectedDate, EndDate.SelectedDate, TimeUnit.Days)/7,0)*5+
Mod(5+Weekday(EndDate.SelectedDate)-Weekday(StartDate.SelectedDate),5)})
```

Figure 28

A button called **Change** is introduced for the employee to select the person he/ she needs to get the approval from.

```
UpdateContext({_showSelectApprover:true});
```

Figure 29

To display the name of the person the employee selected for the approval the following code is used to the text property in the text label.

```
If(!IsBlank(_SelectApprover.DisplayName),_SelectApprover.DisplayName,"Select an approver by clicking""change")
```

Figure 30

To submit the form a submit button is added. The code for the submit button which is on ***OnSelect*** property is as follows.

```
Patch(
    'Leave Applied By Users',
    Defaults('Leave Applied By Users'),
    {'Leave Type': Gallery5.Selected.LeaveType.Value},
    {'From Date': StartDate.SelectedDate},
    {'To Date': EndDate.SelectedDate},
    {'Leave Title': txtLeaveTitle.Text},
    {'Leave Description': txtLeaveSesc.Text},
    {'Total Days': daysdiff + 1},
    {
        'Requested By': {
            Claims: Concatenate(User().Email),
            Department: "",
            DisplayName: User().FullName,
            Email: User().Email,
            JobTitle: ".",
            Picture: "."
        }
    },
```

Figure 31 (a)

```
        'Requested To': {
            Claims: Concatenate(_SelectApprover.Mail),
            Department: "",
            DisplayName: _SelectApprover.DisplayName,
            Email: _SelectApprover.Mail,
            JobTitle: ".",
            Picture: "."
        }
    },{'Leave ID':{'@odata.type': "#Microsoft.Azure.Connector.SharePoint.SPListExpanded.Reference",
                   Id:Gallery5.Selected.ID,
                   Value:Gallery5.Selected.Title}}

);Navigate('My Leave Balance');
Notify("Leave Applied Successfully",NotificationType.Success);
```

Figure 31 (b)

In the Manager Dashboard Screen, the Leaves Waiting for Action screen consists of two buttons called **Approved** and **Declined** in the **Leave Details** gallery. The code for the OnSelect proprety for the **Approved** button is as follows.

```
Set(
    LookUp(
        'Leave Applied By Users',
        ID = Gallery10_3.Selected.ID
    )
);
Patch(
    'Leave Applied By Users',
    myItem,
    {
        'Leave Status': {
            '@odata.type': "#Microsoft.Azure.Connector.SharePoint.SPListExpanded.Reference",
            ID: 1,
            Value: "Approved"
        }
    }
);
Navigate('Leaves Waiting For Action',ScreenTransition.Fade);
```

Figure 32


```
Set(
    LookUp(
        'Leave Applied By Users',
        ID = Gallery10_3.Selected.ID
    )
);
Patch(
    'Leave Applied By Users',
    myItem,
    {
        'Leave Status': {
            '@odata.type': "#Microsoft.Azure.Connector.SharePoint.SPListExpanded.Reference",
            ID: 1,
            Value: "Approved"
        }
    }
);
Navigate('Leaves Waiting For Action',ScreenTransition.Fade);
```


Figure 33

3.3. Testing and Troubleshooting

i. Leave Request Not Submitting

Issue: Users cannot submit a leave request via the app.

Possible Causes:

- Required fields not filled out.
- Network connection failure.
- App errors or glitches.

Solutions:

- **Check Required Fields:** Ensure that all required fields (leave type, start and end dates, and reason for leave) are filled out before submission. Missing fields may prevent successful submission.
- **Verify Network Connection:** Confirm that the user's device has an active and stable internet connection. An unstable connection can result in the request not being submitted.
- **Restart the App:** If the issue persists, advise the user to restart the Power App to resolve any temporary glitches.

ii. Data Not Updating in Manager Dashboard

Issue: Data in the Manager Dashboard is not updating with new leave requests or statuses.

Possible Causes:

- Synchronization issues between the app and the data source (e.g., SharePoint).
- System cache or app lag.
- Permissions issues.

Solutions:

- **Refresh the App:** Ask the manager to refresh the app or restart it to load any newly submitted requests or updated statuses.
- **Check Data Source Synchronization:** Verify that the SharePoint list or data source that stores leave request data is syncing correctly with the app. Ensure that the data source is not experiencing issues.
- **Verify Permissions:** Ensure that the manager has the necessary permissions to view and process leave requests. Review the permissions settings to confirm that the manager has access to the relevant data.

iii. Notifications Not Received

Issue: Users (employees or managers) are not receiving email notifications for leave requests.

Possible Causes:

- Power Automate workflow is not triggered or configured properly.

- Email server issues.
- Emails being flagged as spam.

Solutions:

- **Check Power Automate Workflows:** Ensure that the Power Automate flows responsible for sending notifications are active and properly configured. Verify the workflow history to check for any errors.
- **Check Email Settings:** Verify that the email address associated with the app is correct and active. If the workflow is sending emails to an incorrect address, users will not receive notifications.
- **Check Spam/Junk Folders:** Ask the user to check their spam or junk folders. Sometimes, automated emails can be misclassified as spam by email clients.
- **Contact IT Support:** If the problem persists, contact IT support to check for any issues with the email server or configuration.

iv. Leave Request Status Not Updating

Issue: The status of a leave request (Pending, Approved, Denied) is not updating after a decision is made by the manager.

Possible Causes:

- Power Automate workflow not triggering correctly.
- Manual errors in updating statuses.
- Data source synchronization issues.

Solutions:

- **Check Workflow Execution:** Verify that the Power Automate flow responsible for updating the status is working correctly. Review the workflow history to identify if any errors are occurring during status updates.
- **Manually Update Status:** If the workflow fails, try manually updating the status in the data source (e.g., SharePoint list) to reflect the decision.
- **Check Data Source Syncing:** Confirm that the app is properly synced with the data source and that any changes made are reflected in the app.

3.4. Results and Findings

1. Application Performance:

Efficiency: The Leave Request App successfully automates the process of submitting, approving, and managing leave requests, eliminating manual intervention in most tasks.

Speed: The app processes requests in real-time, with no noticeable delays in submission or approval workflows. Power Automate flows ensure quick communication between employees and managers, and the app is responsive across devices (desktop and mobile).

Error Handling: The app performs well in error scenarios such as incomplete forms or submission failures, providing users with clear feedback for resolution.

2. User Adoption:

Employee Engagement: Employees are able to easily submit and track their leave requests through the user-friendly interface. Features such as status tracking and email notifications have enhanced the user experience.

Manager Utilization: Managers have successfully adopted the app for reviewing, approving, or denying leave requests. The Manager Dashboard provides a centralized location for managing requests, improving efficiency.

Access Across Devices: The app's responsiveness ensures that both employees and managers can access and use it seamlessly across mobile and desktop platforms.

3. Automation Efficiency:

Power Automate Integration: Power Automate flows significantly reduce manual work by automating notification emails, calendar updates, and request status updates. The automation of these tasks ensures that leave requests are processed in a timely manner.

Notifications and Reminders: Automatic reminders sent to managers for pending requests have reduced delays in the approval process. Employees also receive prompt updates on their leave status through email notifications.

4. Data Management:

Centralized Data Storage: Data is securely stored and managed using a SharePoint list or SQL Server, ensuring consistency and ease of access. Managers can easily view past requests, while employees can track their request history.

Real-time Updates: The app ensures that both employees and managers have access to the most up-to-date information regarding leave requests and approvals.

5. Usability and User Interface:

Employee Interface: The app provides an intuitive interface for employees, allowing them to submit and track leave requests with minimal training. Features like editing or canceling pending requests are easy to navigate.

Manager Interface: The Manager Dashboard is well-designed, enabling managers to efficiently process leave requests and view detailed request history. The dashboard allows managers to approve or deny requests with ease.

4. Task 02 – Power Automate

4.1. Design Details

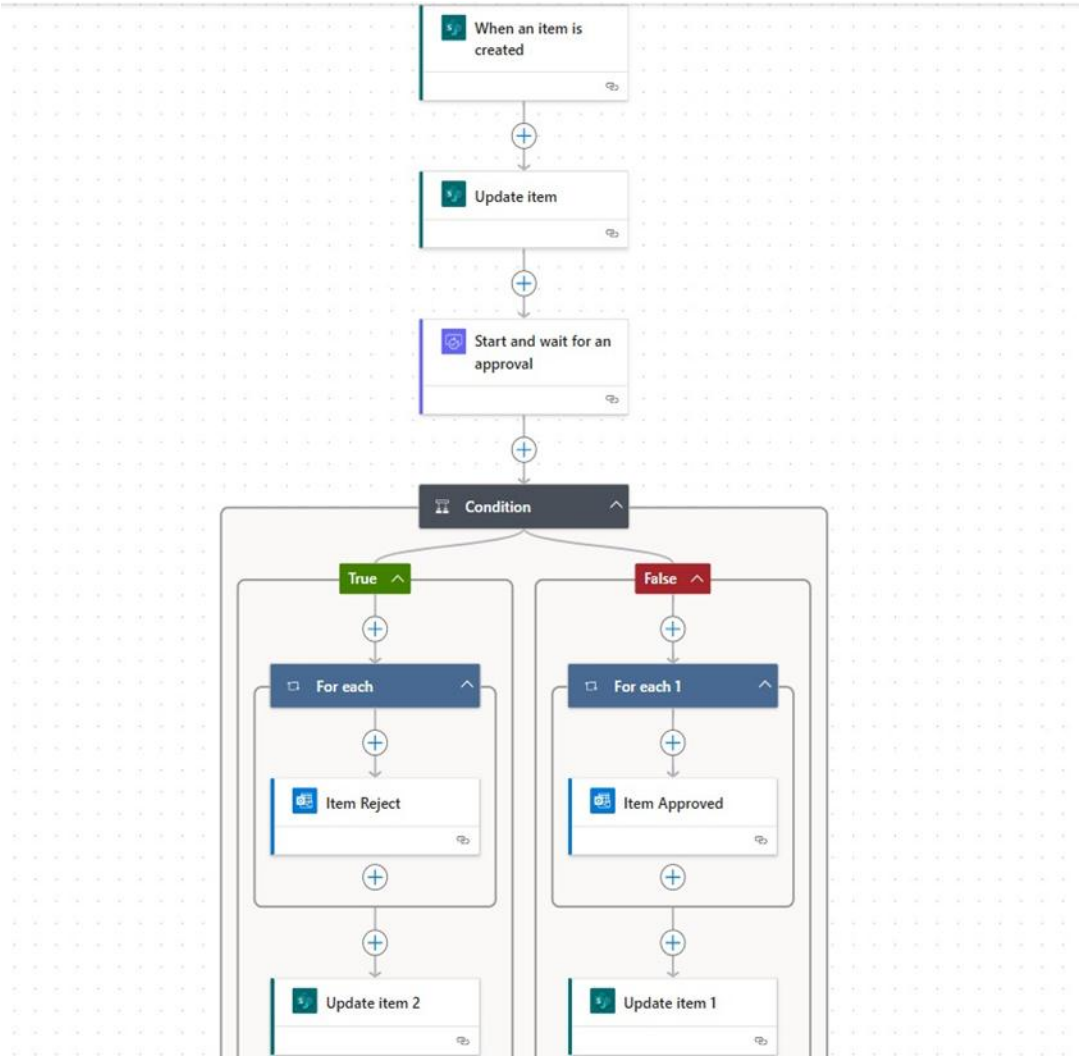

Figure 34

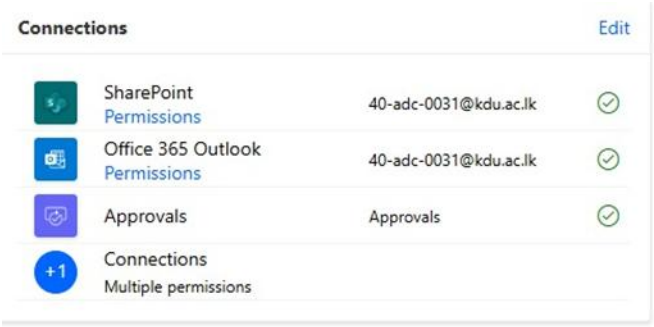

Figure 35

The leave request approval application, designed using Power Automate, follows a structured workflow consisting of three primary steps: **"When an item is created," "Update item,"** and **"Start and wait for an approval."** This flow leverages integration with **SharePoint** and **Office 365 Outlook** to streamline the leave approval process effectively.

Workflow Explanation:

i. **Step 1: Trigger - When an Item is Created**
The workflow begins when a new leave request is created and submitted by an employee. This action triggers the automation process, initializing the leave approval procedure.

ii. **Step 2: Update Item**.

At this stage, the workflow updates the corresponding SharePoint list to reflect the initiation of the leave request. This ensures the request is logged and ready for further processing.

iii. **Step 3: Start and Wait for an Approval**

The flow sends an approval request to the designated manager or approver. The manager receives a notification via Outlook and can either approve or reject the request. The flow is programmed to wait for the manager's decision before proceeding further.

Conditional Branching:

Following the third step, the process incorporates a conditional logic structure that divides into two distinct branches based on the approver's decision: **"True"** (Rejected) and **"False"** (Approved). The outcomes for each condition are handled as follows:

i. **If the Request is Rejected (Condition: True):**

- The employee receives an automated email notification via Office 365 Outlook, informing them that their leave request has been declined.
- Simultaneously, the SharePoint list is updated to reflect the rejection status, ensuring transparency and proper record-keeping.

ii. **If the Request is Approved (Condition: False):**

- The employee is notified via email, confirming that their leave request has been approved.
- The SharePoint list is updated to indicate the approval, maintaining an accurate and up-to-date record of leave requests.

4.2. Implementation Process

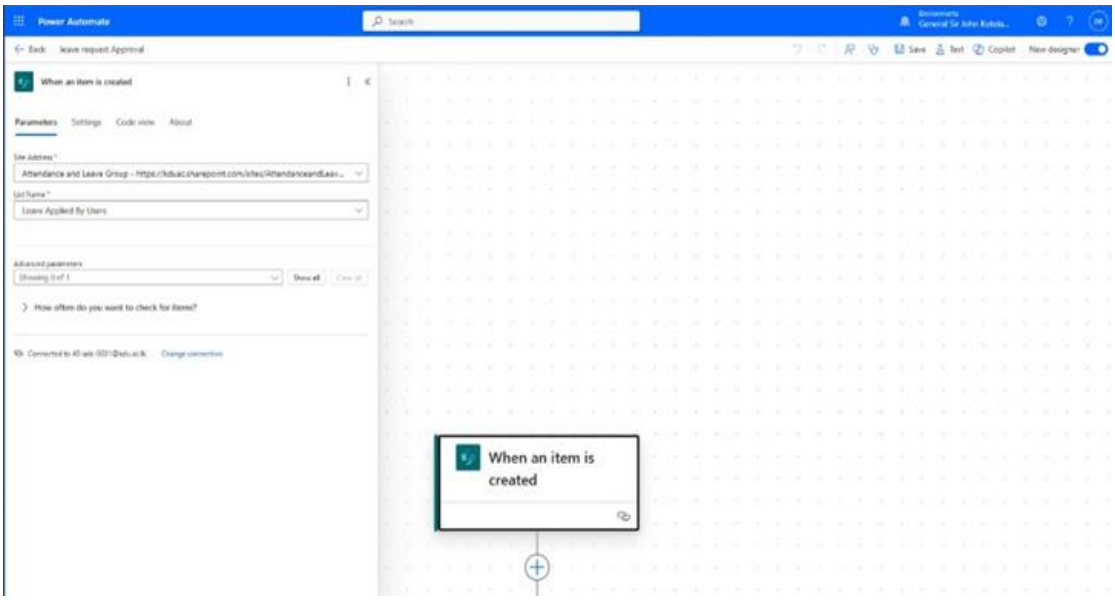

Figure 36

The image depicts the initial step in a **Power Automate workflow** for managing leave request approvals. The workflow starts with the trigger action **"When an item is created."**

Key Details:

i. **Site Address:**

- The workflow is connected to a SharePoint site titled **"Attendance and Leave Group"** located at the specified URL. This site serves as the central repository for managing leave-related data.

ii. **List Name:**

- The trigger monitors the **"Leave Applied By Users"** list within the SharePoint site. This list contains entries for leave requests submitted by employees.

iii. **Advanced Parameters:**

- No advanced options are selected in this parameter.

The **"When an item is created"** trigger is visually represented as a block, indicating that the flow starts when a new entry is added to the specified SharePoint list. This block serves as the foundation for subsequent actions in the leave request approval process.

This setup allows the automation to dynamically respond whenever an employee submits a leave request, ensuring seamless integration with the organization's data management system.

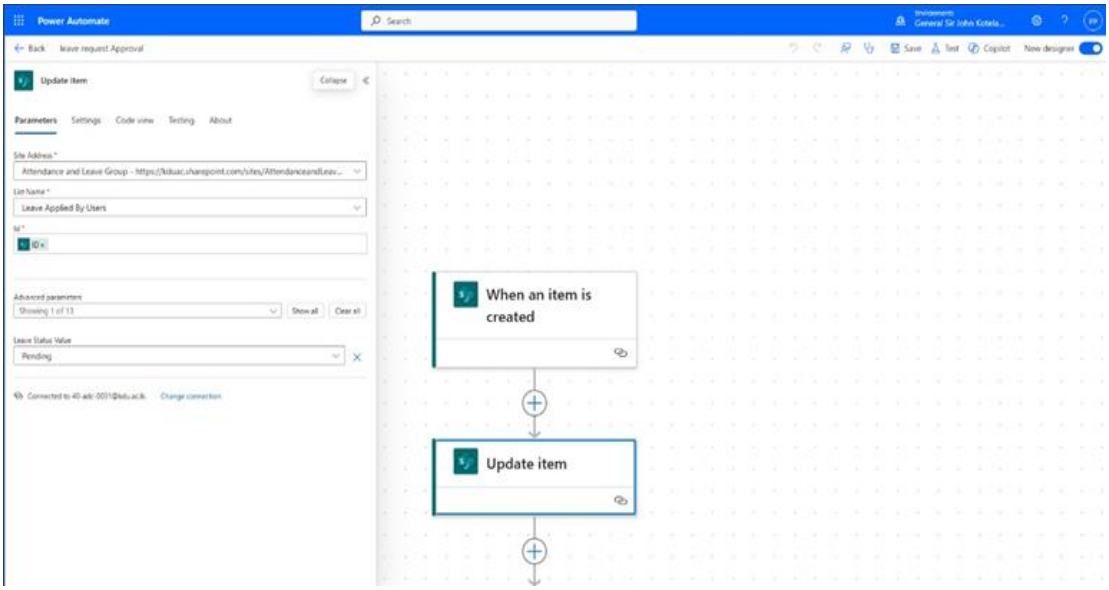

Figure 37

This action is designed to modify an existing item within a specified data source or system. In this specific context, it appears to be updating a record within a SharePoint list titled **"Attendance and Leave Group"**.

<u>Key Components</u>:

i. **Action:** The **"Update item"** action itself is clearly visible.
ii. **Parameters:** This section allows users to define the specifics of the update operation:
    - **Site Address:** This refers to the SharePoint site where the "Attendance and Leave Group" list is located.
    - **List Name:** This explicitly identifies the list where the item will be updated **"Leave Applied By Users"**.
- **Leave Status Value:** This field represents the leave status value as **"Pending"**.

Within the larger flow, this **"Update item"** action plays a role in updating leave requests list in SharePoint.

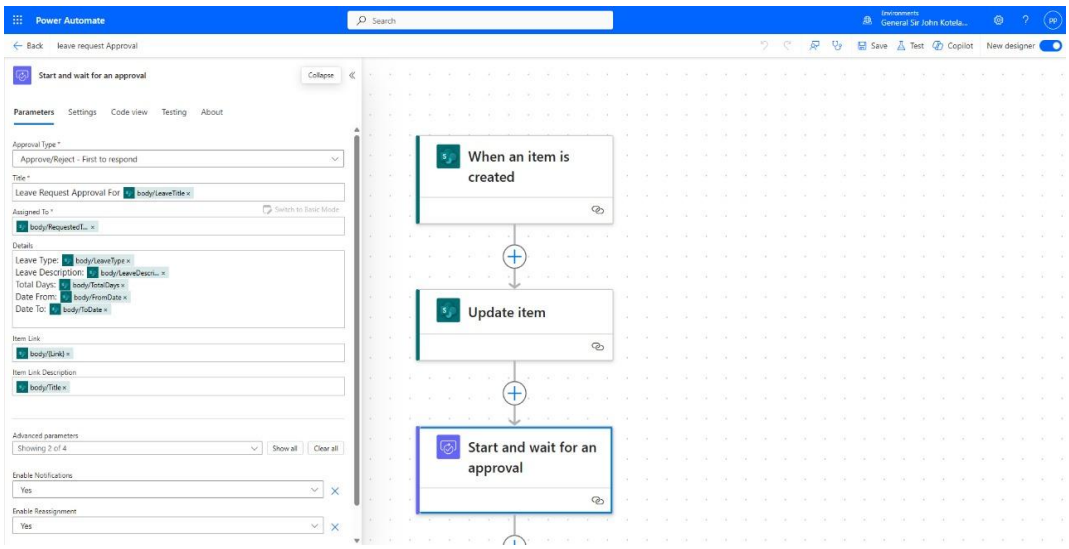

Figure 38

The **"Start and wait for an approval"** action is designed to trigger an approval process within the flow. It essentially pauses the flow's execution and waits for a designated approver to either approve or reject the request.

Key Components:

i. **Action:** The **"Start and wait for an approval"** action itself is clearly visible.

ii. **Parameters:** This section allows users to define:

- **Approval Type:** The type of approval process is Approve/Reject - First to respond.
- **Assigned To:** The user or group email responsible for approving the request.

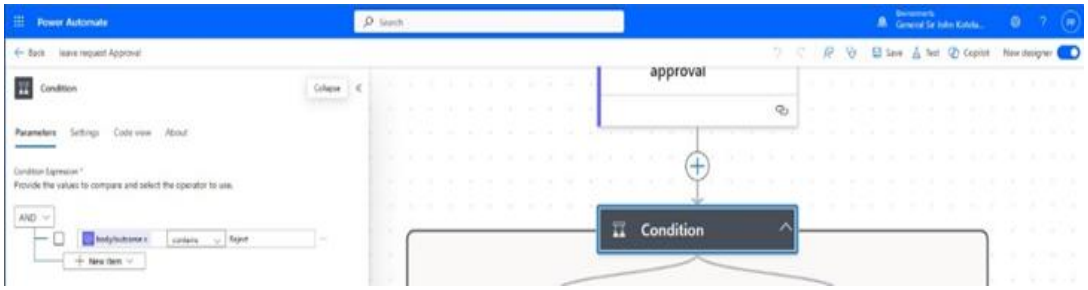

Figure 39

This image shows the conditional logic that determines the next steps based on the approval outcome.

Condition:

i. The condition which is provided checks whether the approval response **"contains"** the word **"Reject"**.

ii. The outcome splits the workflow into two branches:

- **True**: Approval is rejected.
- **False**: Approval is granted.

Logic Implementation:

The condition ensures that further steps (notifications or updates) occur based on whether the leave request was approved or rejected.

Figure 40

This action is designed to iterate over a collection of items. It executes a set of actions for each individual item within that collection.

Key Components:

i. **Action:** The **"For each"** action itself is clearly visible.

ii. **Parameters:**

- **Select an output from previous steps:** This allows the user to specify the collection of items that the "For each" loop should iterate over.

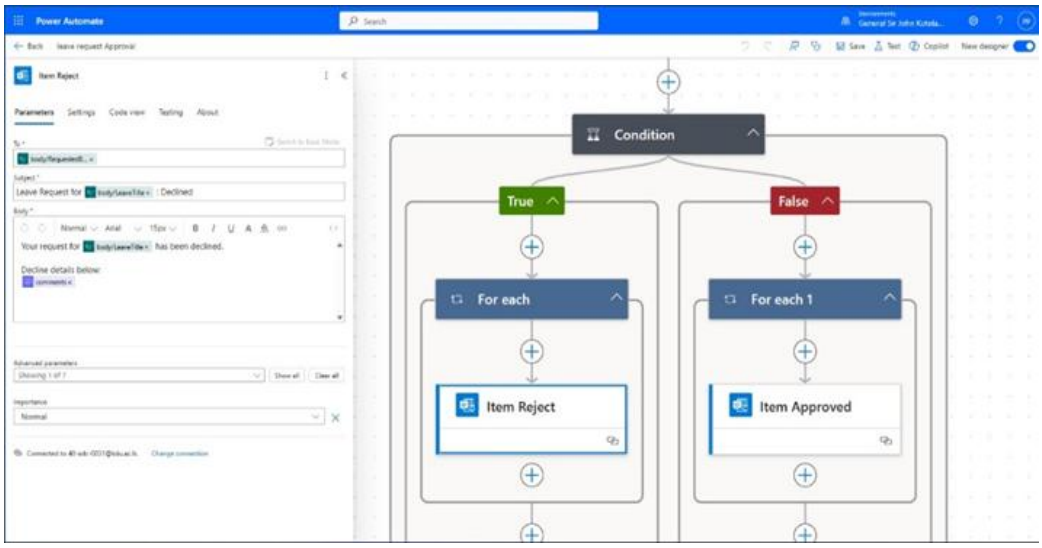

Figure 41

This action is likely designed to send a notification or message to the user indicating that their leave request has been rejected. The content of this notification is likely customized based on the **"Body Title"** and **"body"** fields.

Key Components:

i. **Action:** The **"Item Reject"** action itself is clearly visible.

ii. **Parameters:** This section allows users to define the specifics of the notification:

- **To:** This field specifies the recipient of the notification, which is the user who submitted the leave request.
- **Body Title:** This field appears to be used to set the subject of the notification message. The default value is **"Declined."**
- **Body:** This field contains the main body of the notification message, informing the user that their leave request has been declined. The default message includes a placeholder for the user to provide specific reasons for the decline.

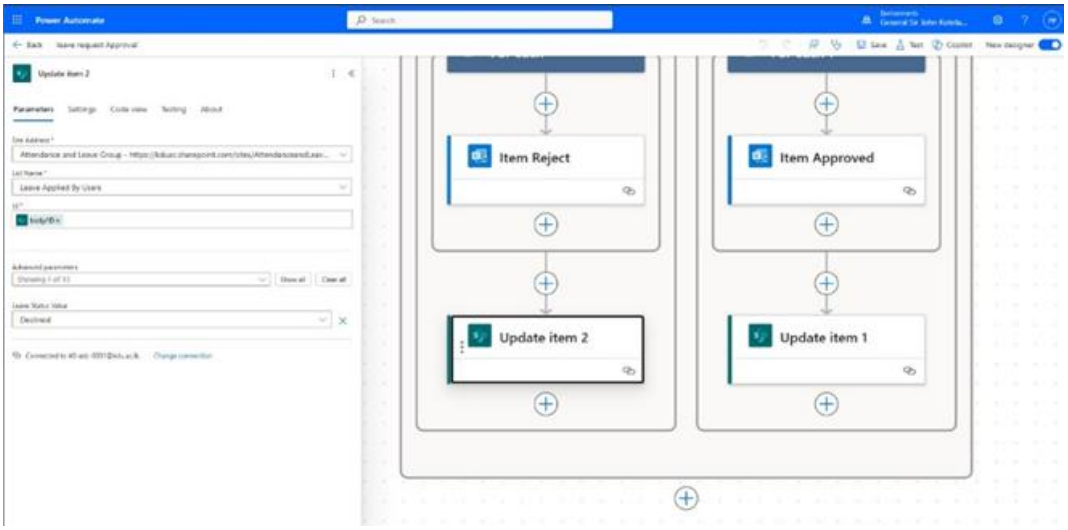

Figure 42

This action is designed to modify an existing item within a specified data source or system. In this specific context, it appears to be updating a record within a SharePoint list titled **"Attendance and Leave Group"**.

Key Components:

i. **Action:** The **"Update item"** action itself is clearly visible.

ii. **Parameters:** This section allows users to define the specifics of the update operation:

- **Site Address:** This refers to the SharePoint site where the **"Attendance and Leave Group"** list is located.
- **List Name:** This identifies the list where the item will be updated ("**Leave Applied By Users** ").

iii. **Advanced Parameters:** This section provides additional customization options for the update action, such as:

- **Leave Status Value:** This field represents the current status of the leave as **"Decline"**. The update action modifies this value in the SharePoint List to reflect changes in the leave request's status.

Within the larger flow, this **"Update item 2"** action plays a role in managing leave requests. It might be triggered when a new leave request is created or when the status of an existing request changes. By updating the **"Leave Applied By Users"** field, the flow ensures that the system accurately records who has submitted a leave request.

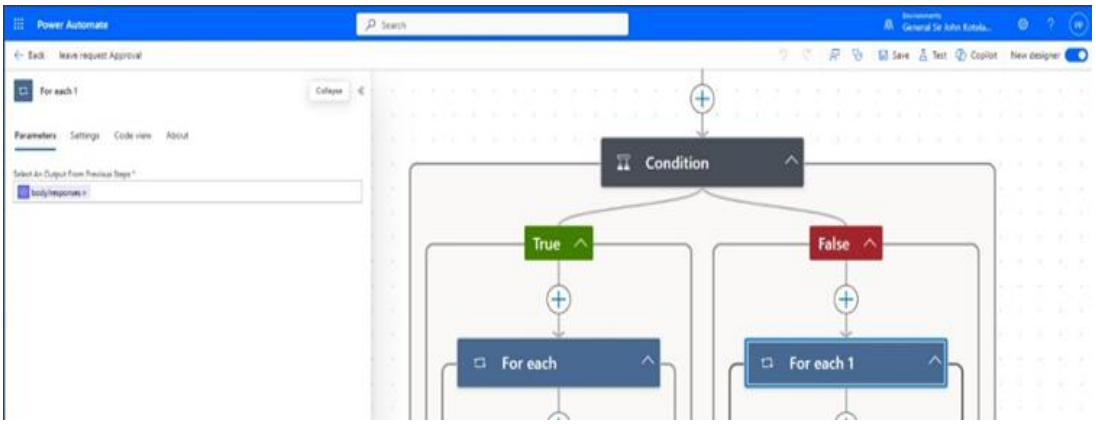

Figure 43

This action is designed to iterate over a collection of items. It executes a set of actions for each individual item within that collection.

<u>Key Components</u>:

i. **Action:** The **"For each 1"** action itself is clearly visible.

ii. **Parameters:**

- **Select an output from previous steps:** This allows the user to specify the collection of items that the **"For each 1"** loop should iterate over. This collection could be an array, a list, or any other data structure that contains multiple elements.

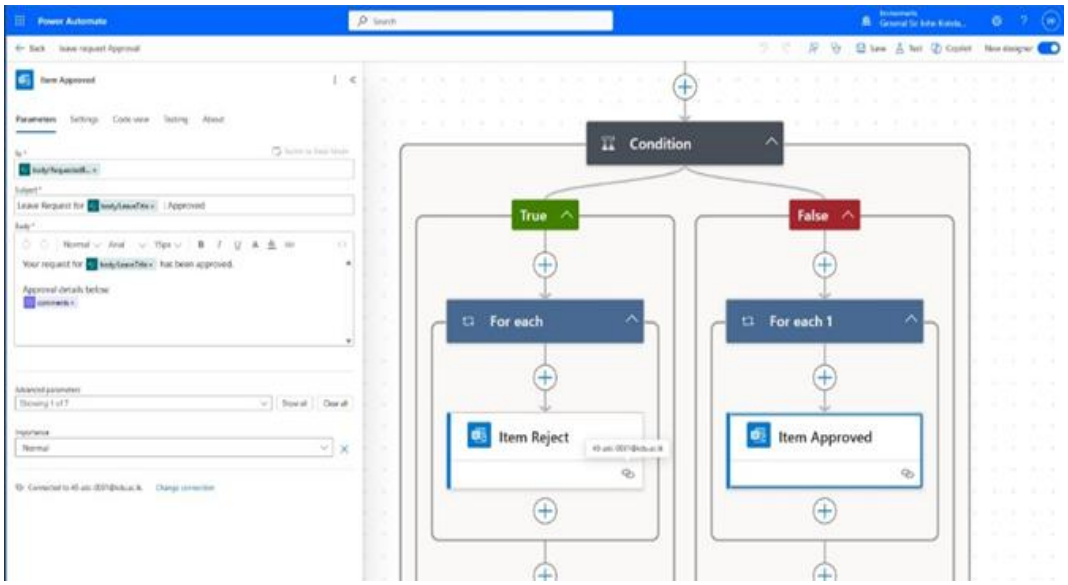

Figure 44

This action is designed to modify an existing item within a specified data source or system. In this specific context, it appears to be updating a record within a SharePoint list titled **"Attendance and Leave Group."**

Key Components:

i. **Action:** The **"Update item"** action itself is clearly visible.

ii. **Parameters:** This section allows users to define the specifics of the update operation:

- **Site Address:** This refers to the SharePoint site where the **"Attendance and Leave Group"** list is located.
- **List Name:** This explicitly identifies the list where the item will be updated **("Leave Applied By Users ")**.

iii. **Advanced Parameters:** This section provides additional customization options for the update action, such as:

- **Leave Status Value:** This field represents the current status of the leave as **"Approved"**. The update action modifies this value in the SharePoint List to reflect changes in the leave request's status.

Within the larger flow, this **"Update item 2"** action plays a role in managing leave requests. It might be triggered when a new leave request is created or when the status of an existing request changes. By updating the **"Leave Applied By Users"** field, the flow ensures that the system accurately records who has submitted a leave request.

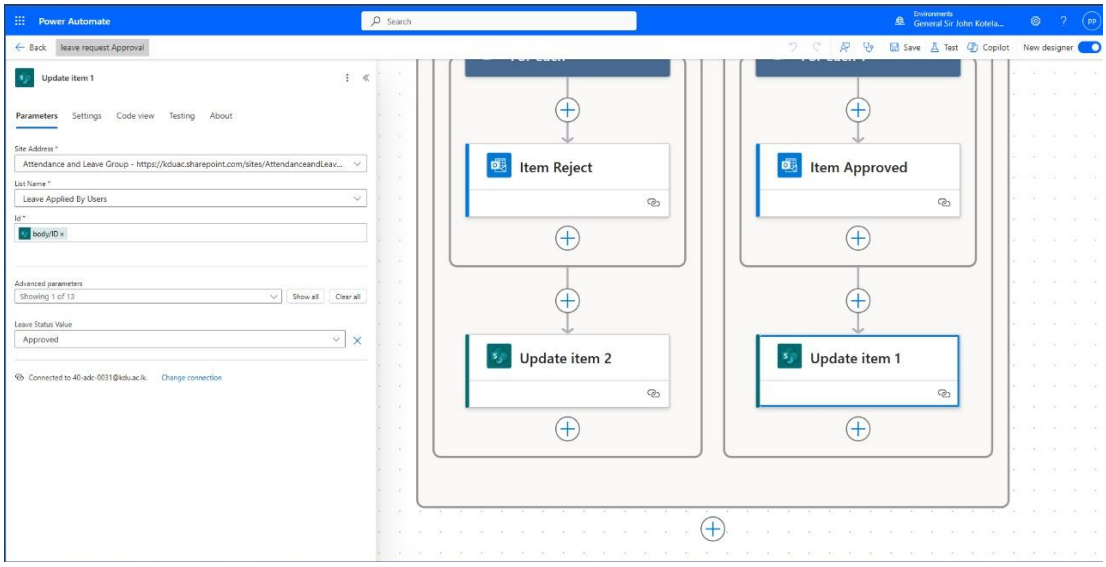

Figure 45

This action is designed to modify an existing item within a specified data source or system. In this specific context, it appears to be updating a record within a SharePoint list titled **"Attendance and Leave Group."**

Key Components:

i. **Action:** The **"Update item 1"** action itself is clearly visible.

ii. **Parameters:** This section allows users to define the specifics of the update operation:

- **Site Address:** This refers to the SharePoint site where the **"Attendance and Leave Group"** list is located.
- **List Name:** This explicitly identifies the list where the item will be updated **("Leave Applied By Users ")**.

iii. **Advanced Parameters:** This section provides additional customization options for the update action, such as:

- **Leave Status Value:** This field represents the current status of the leave as **"Approved"**. The update action modifies this value in the SharePoint List to reflect changes in the leave request's status.

Within the larger flow, this **"Update item 1"** action plays a role in managing leave requests. It might be triggered when a new leave request is created or when the status of an existing request changes. By updating the **"Leave Applied By Users"** field, the flow ensures that the system accurately records who has submitted a leave request.

4.3. Testing and Troubleshooting

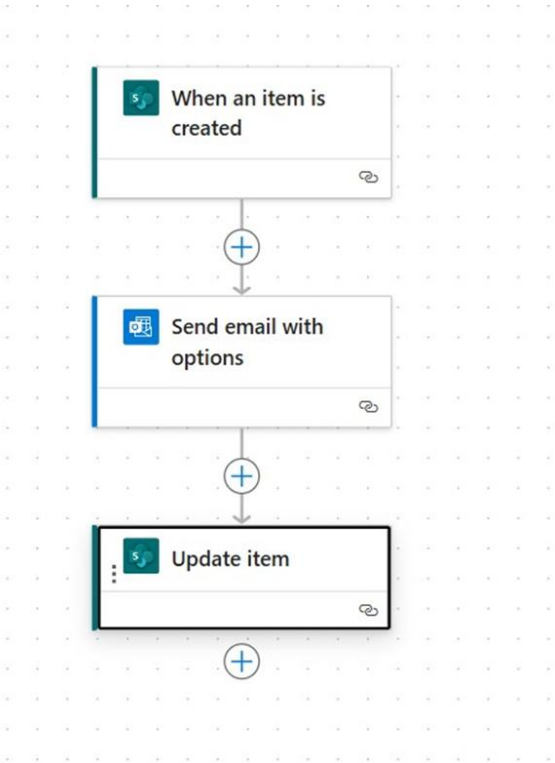

Figure 46

During the initial testing phase, the workflow depicted in the diagram was the first to be implemented. The process begins with the creation of an item, followed by an automated email notification requesting approval for the submitted request. However, a key issue was identified: once the request was approved, the SharePoint system was updated, but the relevant employee did not receive a notification email indicating the approval status.

To address this issue, we adopted the approach illustrated in ***Figure 20***. This updated process introduced an additional step to ensure that once a request is approved, an automated email notification is also sent to the relevant employee. This improvement not only enhances communication but also ensures that all stakeholders are promptly informed about the status of the request.

4.4. Results and Findings

Email-Based Request Approval Flow

Initially, a request form is submitted to the designated requester via email through Outlook. The requester reviews the form and approves or declines the request. If the request is approved, the originator of the request reviews it again, adds relevant comments, and sends a follow-up email back to the requester. The requester then finalizes the decision to approve or decline the request. This email-based workflow ensures traceability and maintains a formal record of communications.

Step 01:

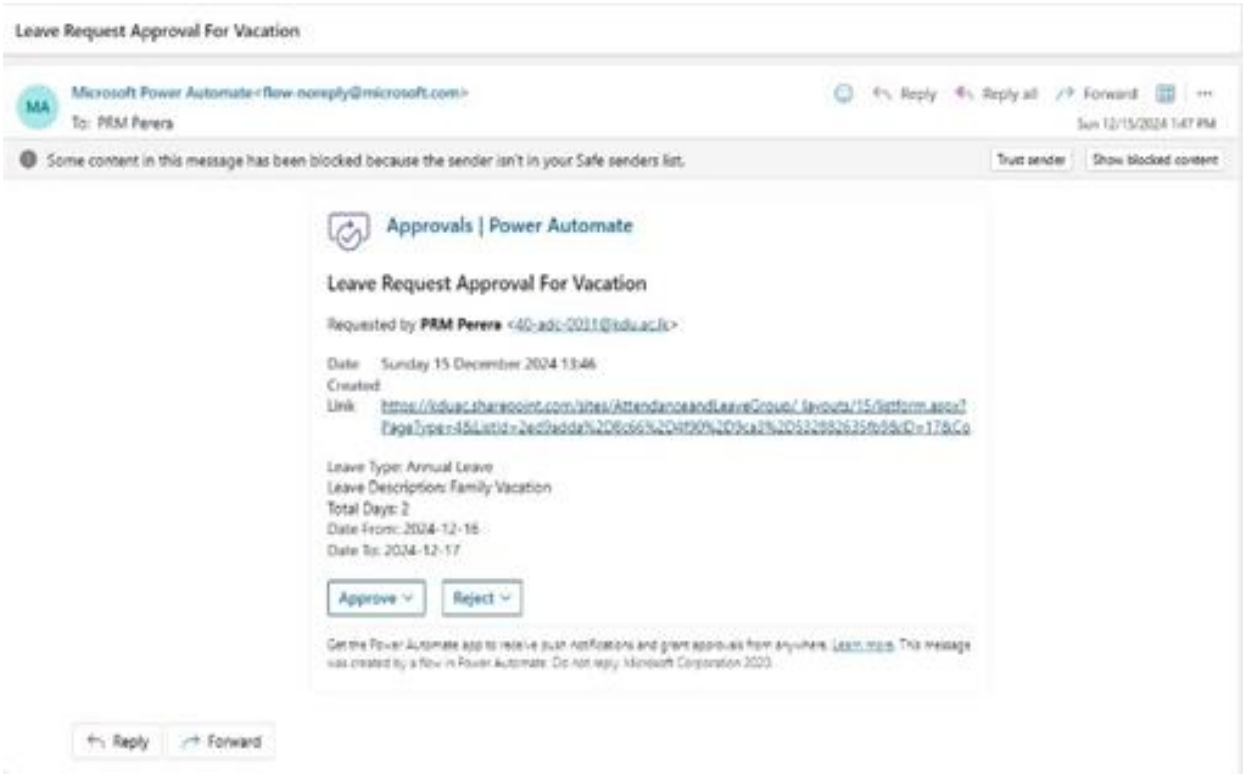

Figure 47

Step 02:

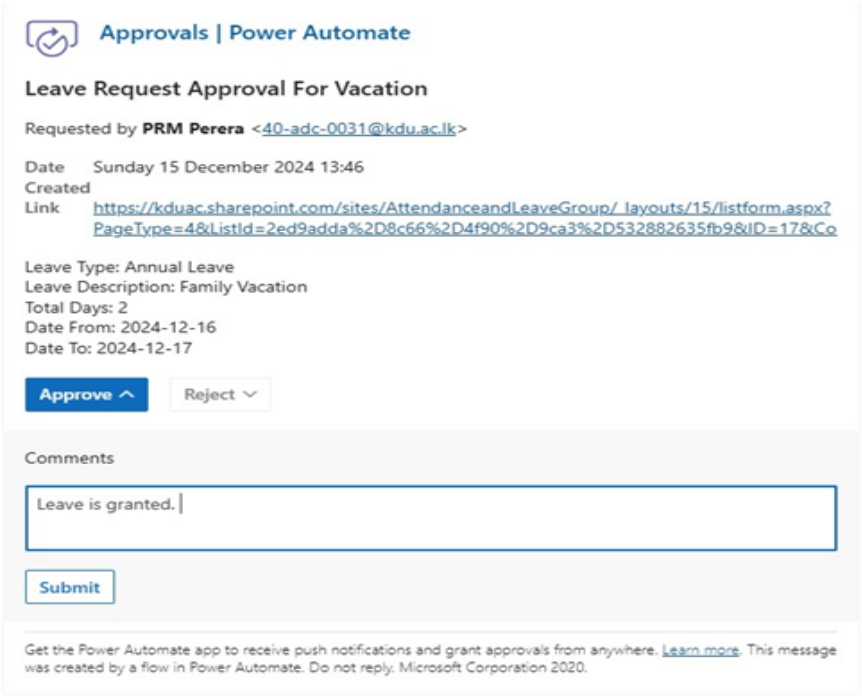

Figure 48

Step 03:

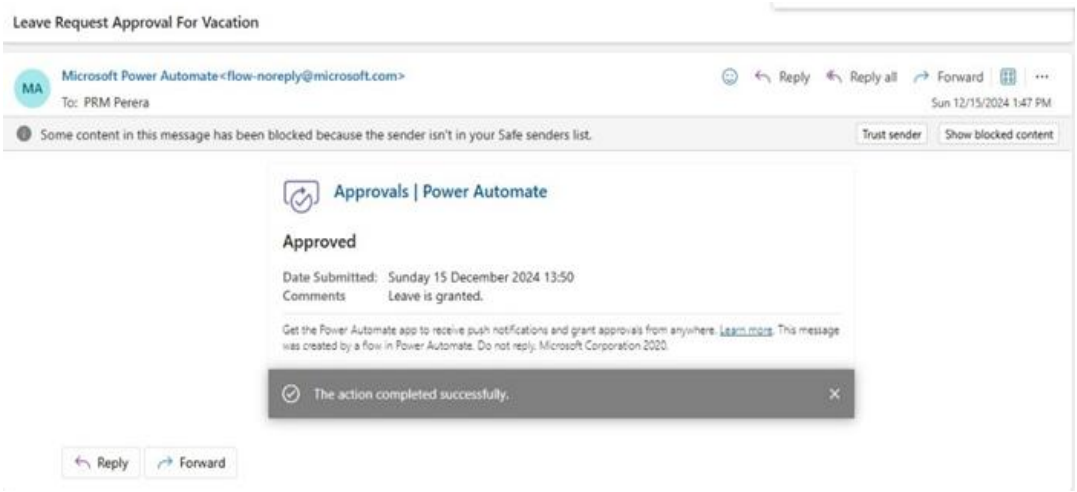

Figure 49

Step 04:

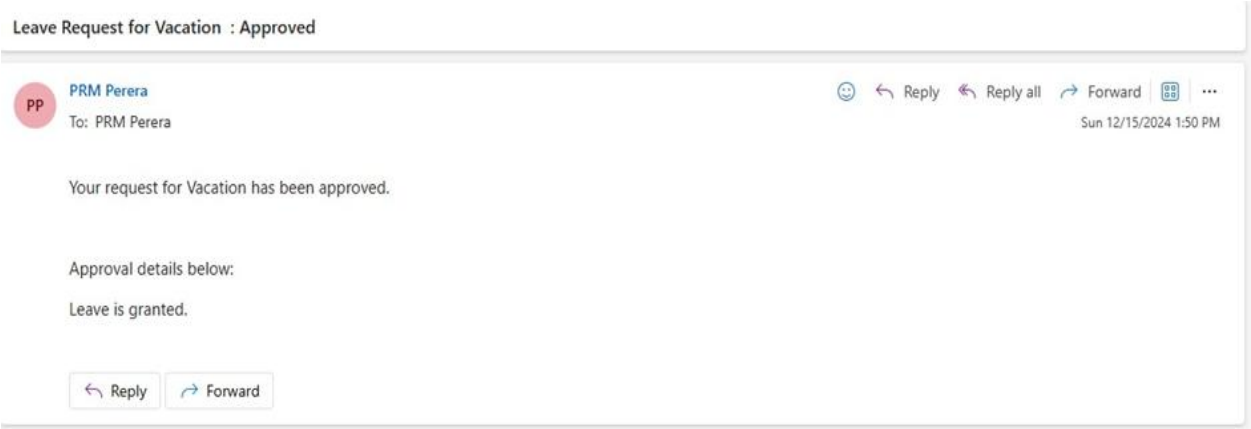

Figure 50

Integration with Teams for Notifications

In the second option, the approval process includes an integration with Teams. Notifications regarding the request's status (approved or rejected) are sent directly to Teams. This feature enhances real-time communication by providing employees with immediate updates on the status of their requests, eliminating delays caused by email-only communication.

Step 01:

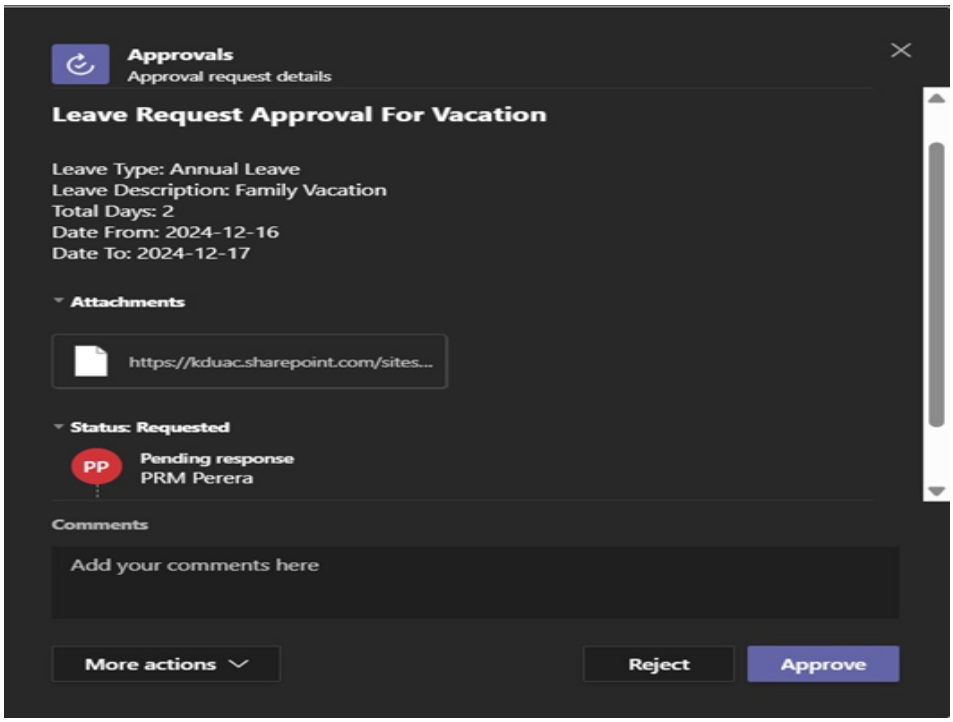

Figure 51

Step 02:

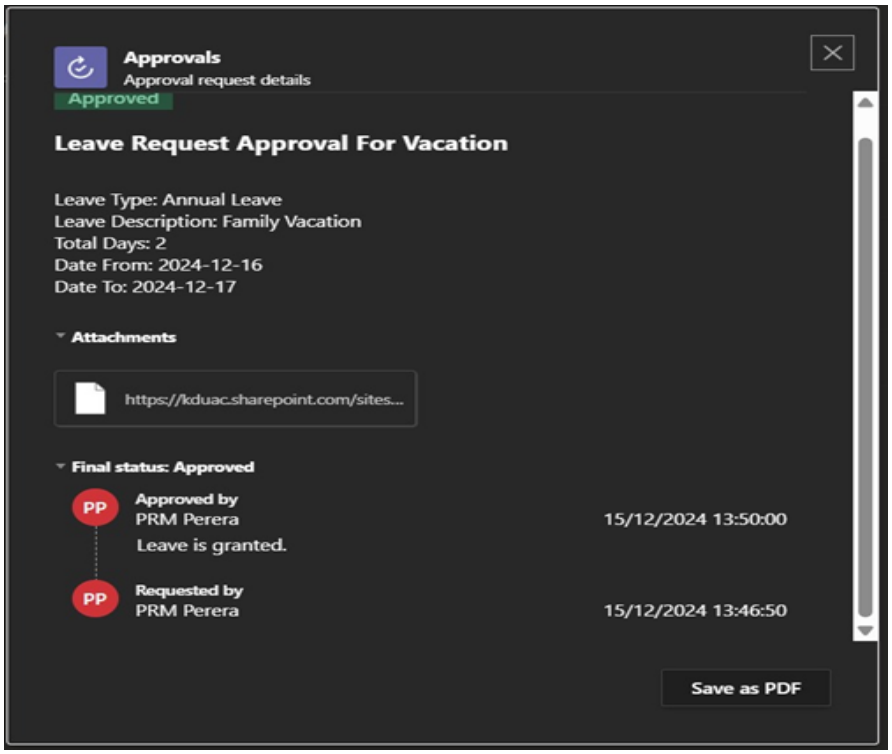

Figure 52

Step 03:

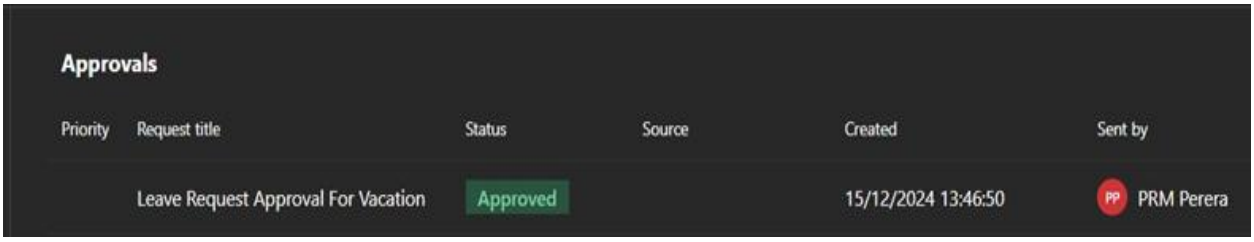

Figure 53

5. Conclusion

The Leave Request Application developed using Power App has successfully streamlined the leave request process for both employees and managers. With its automated workflows, responsive design, and real-time updates, the app has enhanced operational efficiency and user experience. Although there are some limitations, such as the lack of delegation functionality for managers, the app has provided significant improvements in how leave requests are handled within the organization. Future improvements and additional features will further optimize the leave management process.

The leave request approval application built with Power Automate streamlines processes through automation, ensuring efficiency, transparency, and timely communication while reducing manual effort and enabling future enhancements like advanced analytics and HR system integration.

6. Appendix

***Figure 01*:** Home Screen of the Employee Leave Request App.
***Figure 02*:** Create a Request Screen of the Employee Leave Request App.
***Figure 03*:** My Leave Requests Screen of the Employee Leave Request App.
***Figure 04*:** My Leave Balance Screen of the Employee Leave Request App.
***Figure 05*:** Company Holidays Screen of the Employee Leave Request App.
***Figure 06*:** About Screen of the Employee Leave Request App.
***Figure 07*:** Manager Dashboard Screen of the Employee Leave Request App.
***Figure 08*:** Balance List SharePoint List.
***Figure 09*:** Company Holidays SharePoint List.
***Figure 10*:** Leave Applied by Users SharePoint List.
***Figure 11*:** The code for the login buttons in the Home Screen of the app.
***Figure 12*:** Header of the app.
***Figure 13*:** Navigation bar of the app.
***Figure 14*:** *NavigationItem* code for the Navigation bar of the app.
***Figure 15*:** *NavigationItem* code for the Navigation bar of the app.
***Figure 16*:** The code to show the Items on the Navigation design.
***Figure 17*:** The code to select the item from the designed Navigation.
***Figure 18*:** Using the *ClearCollect* function to navigate between titles and target screens.
***Figure 19*:** The code which is used to switch between employee and manager's view which is based on the *NavigationView* value.
***Figure 20*:** The code filter to the gallery that provides the request status types.
***Figure 21*:** The code which refreshes the MyLeaveColl collection based on the selected filter (All, Pending, etc.), showing leave requests specific to the logged-in user.
***Figure 22*:** The code which shows the output "No Record Found" when there is no data available.
***Figure 23*:** The code which shows the gallery in the Company Holidays Screen.
***Figure 24*:** The code which shows the items in the gallery of the Company Holidays Screen.
***Figure 25*:** The code which connects the Balance List SharePoint.
***Figure 26*:** The code for the two date pickers.
***Figure 27*:** The code for the default date.
***Figure 28*:** The code for the OnChange.
***Figure 29*:** The code for the Select Approver.
***Figure 30*:** The code to display the name of the approver.
***Figure 31 (a)*:** The code for the submit button.
***Figure 31 (b)*:** The code for the submit button.
Figure 32: The code for the Leave Detail.
Figure 33: The code for the Leave Detail.
***Figure 33*: "**When an Item is Created" trigger of the automated workflow – primary step of the workflow.
***Figure 34*:** "Update Item" – the second step of the workflow.
***Figure 35*:** "Start and Wait for an Approval" – the third step of the workflow.

7. References

1. Bulb Digital (2022). *Get Feedback From Your Users - Approvals vs. Email with Options*. [online] YouTube. Available at: https://www.youtube.com/watch?v=8zRdDqHxgE0 [Accessed 15 Dec. 2024].

2. Reza Dorrani (2021). *Run Power Automate flow from SharePoint - For a Selected Item trigger*. [online] YouTube. Available at: https://www.youtube.com/watch?v=2Hp9CTd8zQU [Accessed 15 Dec. 2024].

3. Reza Dorrani (2022a). *How to build Power Automate Approval Workflows for SharePoint | Step by Step Tutorial*. [online] YouTube. Available at: https://www.youtube.com/watch?v=tiLyIL6Q2Wk [Accessed 15 Dec. 2024].

4. Reza Dorrani (2022b). *Power Apps Button Control for Beginners | OnSelect, Style, Properties*. [online] YouTube. Available at: https://www.youtube.com/watch?v=REN93Y-kME8 [Accessed 15 Dec. 2024].

5. Time and date (2024). Climate & Weather Averages in Freeport, Texas, USA. *Timeanddate.com*. [online] doi:https://doi.org/1004254/tadcom_300x600.

6. Tips By Anil (2021). *Leave Application Request in PowerApps | leave management system PowerApps - Part 1*. [online] YouTube. Available at: https://www.youtube.com/watch?v=ouB3ebUzQJ4 [Accessed 13 Dec. 2024].

7. Vacation Tracker. (2022). *Sri Lanka Leave Laws & Holidays*. [online] Available at: https://vacationtracker.io/leave-laws/asia/sri-lanka/#maternity-and-paternity-leave [Accessed 15 Dec. 2024].

8. Veronique Lengelle | MVP (2023). *Find out how to PATCH the SharePoint Item ID as a LOOKUP column for optimized efficiency!* [online] YouTube. Available at: https://www.youtube.com/watch?v=BhBgLyWwKoc [Accessed 15 Dec. 2024].