

# Automated Request Submission and Approval Workflow in Power Apps



In this case study, we explore an automated workflow designed to streamline the request submission and approval process using PowerApps and SharePoint. This workflow enhances user experience by facilitating efficient communication between users and approvers.

## Workflow Overview

### 1. User Login and Request Creation:

- When a user logs into Power Apps, they can create a request by providing essential details such as the Request Title, Amount, and Approver.
- Upon submission, an item is created in a designated SharePoint list, capturing all relevant information.

### 2. Email Notification to Approver:

- Following the creation of the item, a Power Automate flow is triggered, sending an email notification to the assigned approver.
- This email contains a direct link to the request, enabling the approver to access it quickly.

### 3. Approver Interaction:

- When the approver clicks on the link in the email, they are navigated directly to the Power Apps approver screen.
- On this screen, the approver can review the request and make decisions: they can **Accept, Reject, or Request More Information**.

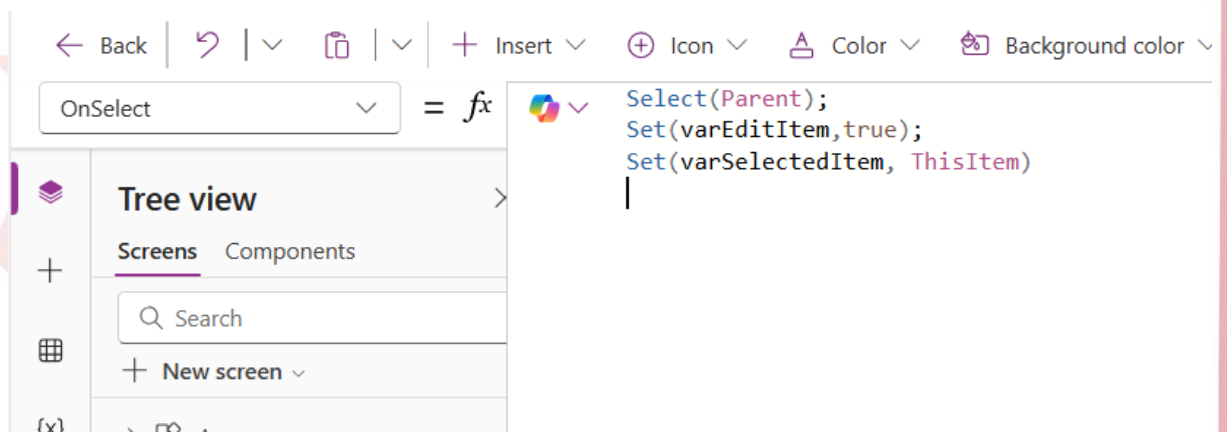
### 4. User Request Status Tracking:

- In the user interface where requests are created, users can view the status of their requests.
- If the status indicates "Need More Information," users have the option to edit their request and resubmit it for approval.
- Once a request is marked as "Approved" or "Rejected," users cannot edit it through Power Apps; instead, they have view-only access to maintain data integrity.

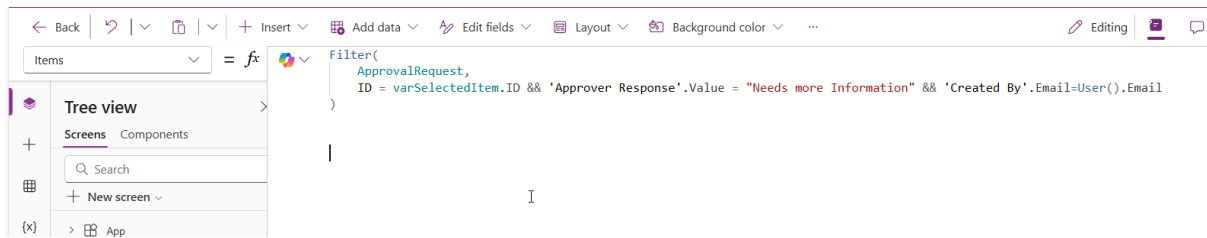
## Steps to create

- Login to Power apps

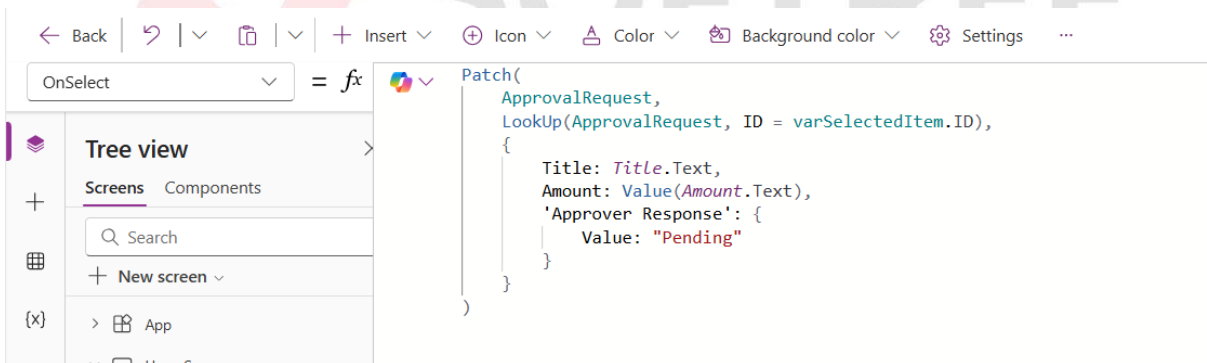
- Create a screen and name it as HomeScreen.
- In that screen add a header.
- Add a gallery to screen, here in case study gallery name is Gallery2.
- In items property of the gallery2 add `Filter(ApprovalRequest, 'Created By'.DisplayName=User().FullName)`
- This will show Approval request datasource list items those which are created by login user.
- In this case study we make the width to 620 and height to 634, X and Y to 24 and 92 respectively.
- Add 3 text box to show items field value in the gallery and make their text values as  
For Title - "Title: " & Upper(ThisItem.Title)  
For Amount- "\$ "&ThisItem.Amount  
For Approver status- ThisItem.'Approver Response'.Value
- Add an edit icon to the gallery.
- In edit icons OnSelect property provide below code



- In OnSelect of the gallery provide `Refresh(ApprovalRequest)` so that when ever we click on gallery the data source will refresh.
- In visible property of the icon provide below code  
**`If(ThisItem.'Approver Response'.Value="Needs more Information",true,false)`**
- So that the icon will be visible only if the approver response value is "Needs more information".
- In that HomeScreen add another gallery and in the items property provide the below code  
**`Filter(ApprovalRequest,ID=varSelectedItem.ID&&'Approver Response'.Value="Needs more Information"&&'Created By'.Email=User().Email)`**
- So that it will show value based on Gallery2, if the approval status is "Needs more information" and the item is created by login user.

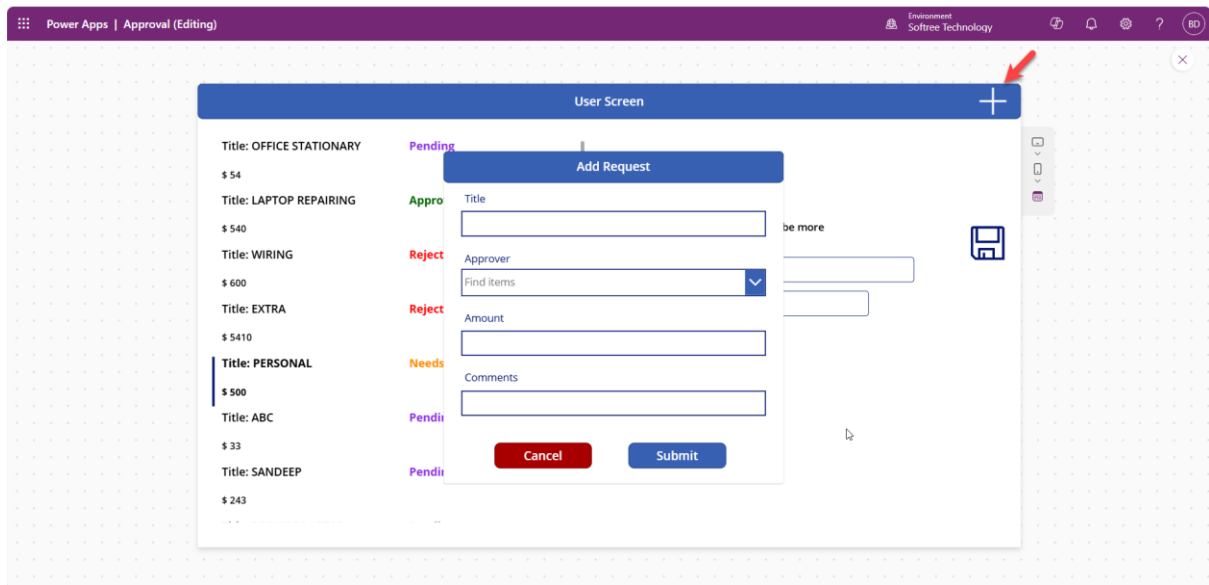


- In that gallery visible property add below code  
`If(varEditItem=true,true,false)`
- Add a text label to view comments provided by Approver. And make its Text property to **"Comments: "&Gallery2.Selected.Comments.**
- Add a label for Title header and another for Amount.
- Add two text inputs one for Title and another for amount.
- To show title in title text input's default value provide  
**LookUp(ApprovalRequest, ID = ThisItem.ID, Title)**
- In amount's text inputs default property provide  
**LookUp(ApprovalRequest, ID = ThisItem.ID, Amount)**
- Add a Save icon and in that icon's OnSelect property provide below code  
**Patch(ApprovalRequest,LookUp(ApprovalRequest,ID=varSelectedItem.ID),{Title:Title.Text,Amount:Value(Amount.Text),'Approver Response':{Value:"Pending"}})**

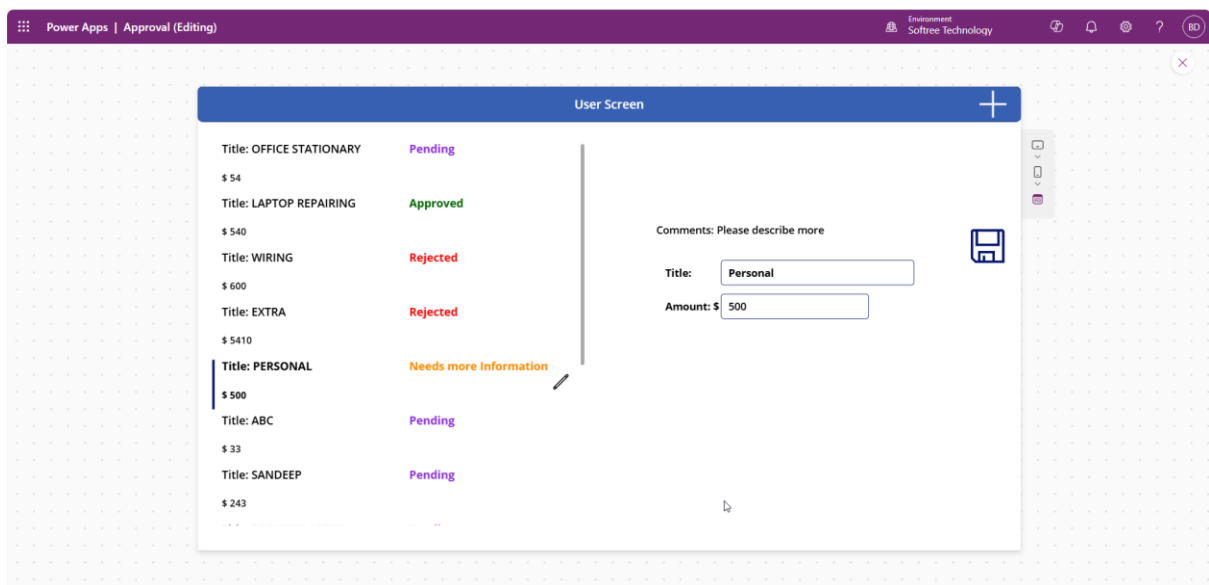


- To add a new request, add an + icon to the header. And make its onselect property to **Set(varFormVisible,true)**
- Add a non-responsive container to the screen. Make the X property of the container to 408 and Y property to 112.
- Set width and height of the container to 564 and 550.
- Add a form in the container and set its data source as Approval Request.
- Add a button for submit and make its onselect property to  
**SubmitForm(Form2);**  
**Set(varFormVisible,false)**
- Add another button and make its onselect property to  
**ResetForm(Form2);**  
**Set(varFormVisible,false)**

- When a user clicks on + icon the form will look like below by using which user can create request.



- And the Home screen will look like below



- Now we will create power automate flow.
- For that first create a instant cloud flow and provide trigger as when an item is created. Provide site url and list name.

**When an item is created**

\* Site Address: POWERPLATFORM -  
 https://s.XXXXXXXXXXn/sites/POWERPLATFORM

\* List Name: ApprovalRequest

Show advanced options

- Add another action Get items and provide site URL and list name and Id from previous action

**Get item**

\* Site Address: POWERPLATFORM -  
 https://XXXXXXXXXXjint.com/sites/POWERPLATFORM

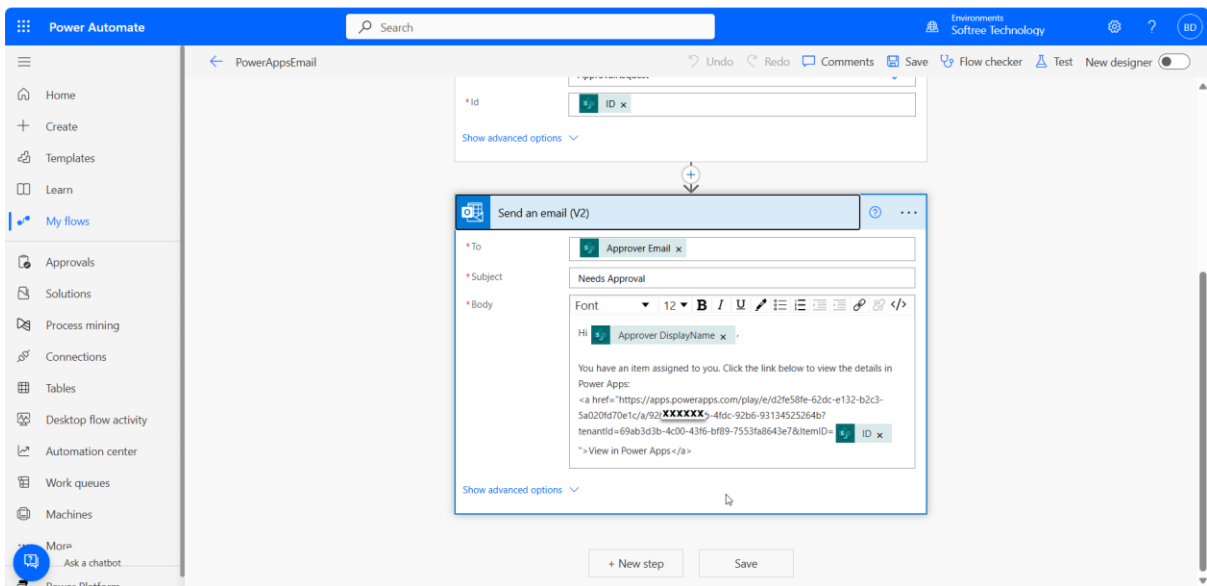
\* List Name: ApprovalRequest

\* Id: ID x

Show advanced options

- Add another action Send an Email V2. In to provide get item's Approver Email, Subject as Need Approved, In Body provide the information like below.

[https://apps.powerapps.com/play/e/d322fe58fe-62dc-e132-b2c3-5a020fd70e1c/a/XXxXXxXXxXX-2f1b-4f4ddc-92b6-93134525264b?tenantId=69ab3d3b-4c00-43f6-bf89-7553fa8643e7&ItemID=@{outputs\('Get item'\)\['body/ID'\]}](https://apps.powerapps.com/play/e/d322fe58fe-62dc-e132-b2c3-5a020fd70e1c/a/XXxXXxXXxXX-2f1b-4f4ddc-92b6-93134525264b?tenantId=69ab3d3b-4c00-43f6-bf89-7553fa8643e7&ItemID=@{outputs('Get item')['body/ID']})

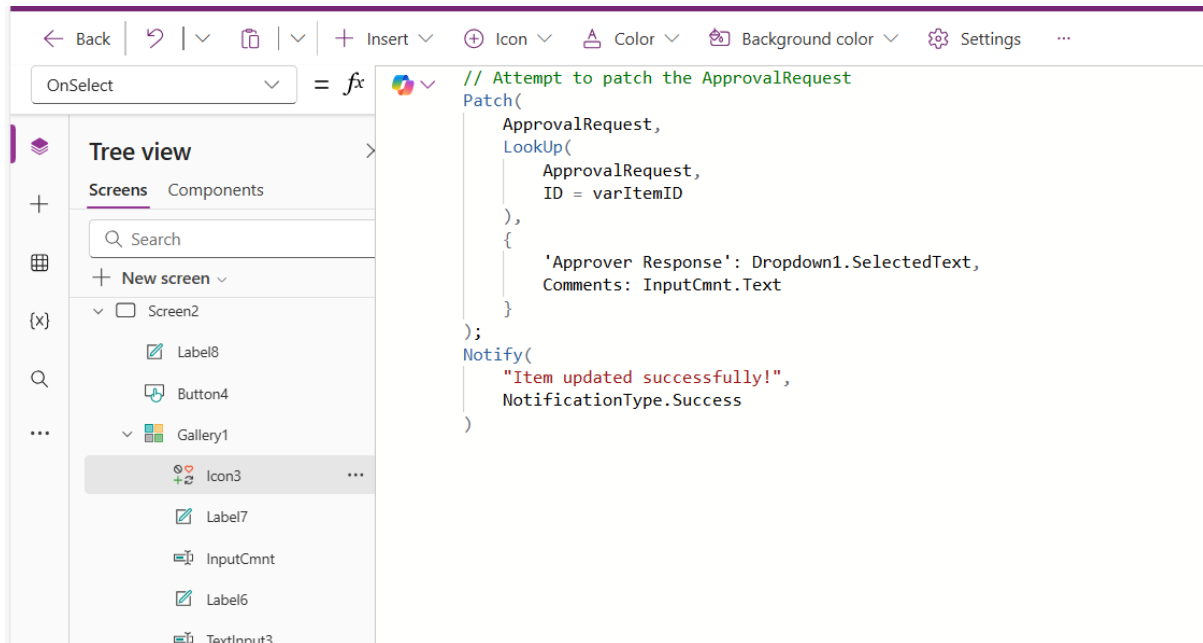


- So that this will send mail body with deep link to the item.

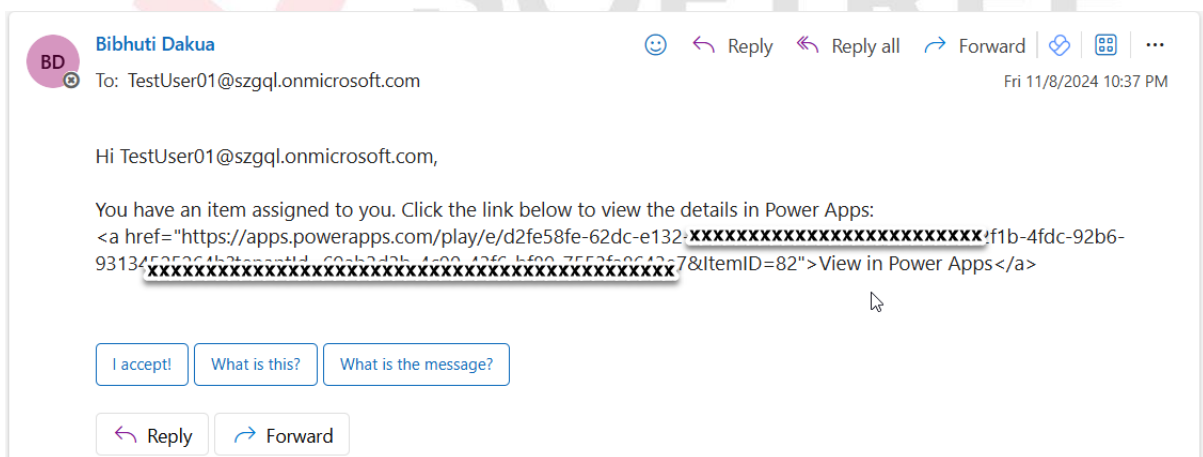
### Now we will create screen for Approver

- Before that first provide below code in App on start.  
**Set(varItemID, Value(Param("ItemID")));**
- It will first store the value which comes in the URL in a global variable.
- In app start screen provide **Switch(Param("ItemID"),Blank(),HomeScreen,Screen2)**
- So that if URL having parameter then it will navigate to Screen2.
- Add a gallery to the screen . So that approver can view the item and edit the status and can provide comments to the item.
- For that gallery's item property provide below code  
**Filter(ApprovalRequest,Approver.Email=User().Email && ID=varItemID)**
- Add three text input, one for Title, one for viewing amount and another for viewing Created by.
- And in every property provide like  
ThisItem.Title,ThisItem.Created by, ThisItem.Amount.
- And make the edit property of the input text box to view.
- Add a dropdown control to change approver status.
- Add below code to its item property  
**Choices([@ApprovalRequest]. 'Approver Response')**
- In default property of the dropdown provide  
**ThisItem. 'Approver Response'.Value**
- Add another input text to the gallery and make its visible property to below code  
**If(Dropdown1.Selected.Value = "Needs more Information", true, false)**
- Add a save icon to the gallery and in icons OnSelect property provide below code  
**Patch(ApprovalRequest,LookUp(ApprovalRequest,ID=varItemID),{'Approver**

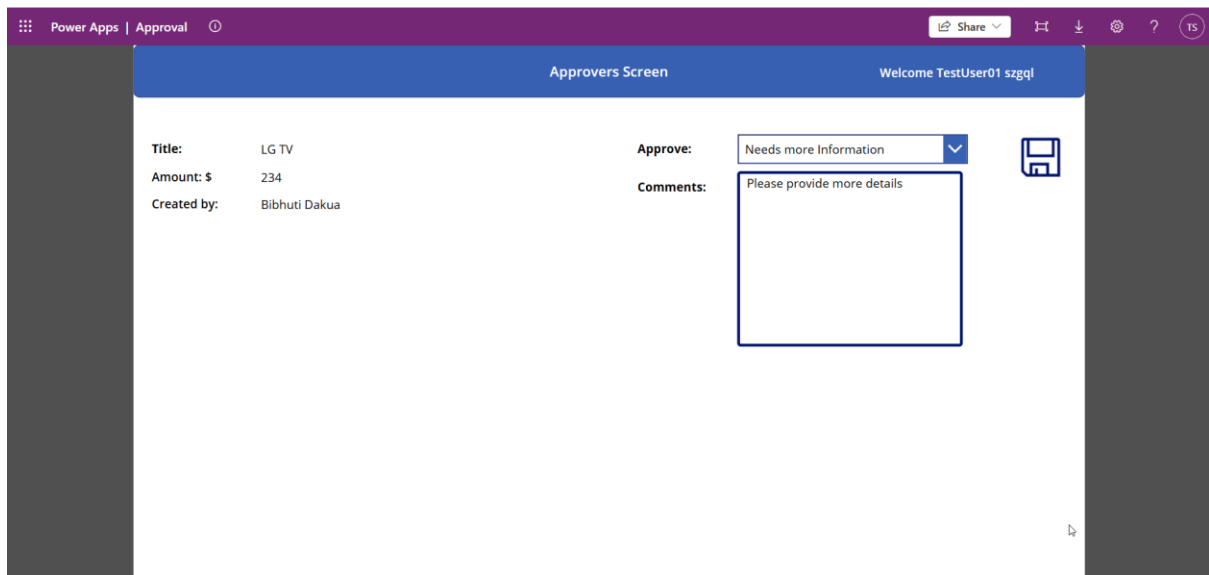
**Response':Dropdown1.SelectedText,Comments:InputCmnt.Text});Notify("Item updated successfully!",NotificationType.Success)**



**Approver will get email like below image when an item is created in SharePoint list.**



When the approver open the provided link the below screen will appear.



Here approver can Approve, reject or Ask for more information by choosing Needs more information and Save.

When the approver save the item after approve or reject or after asking more information the item will update on SharePoint list.

And if the approver choose Needs more information then the comment section will visible. The items in the SharePoint are look like below image

The screenshot shows a SharePoint list titled 'ApprovalRequest'. The table contains the following data:

Title	Approver	Approver Response	Comments	Amount	ID
LG TV	TestUser01@szggl.onmicrosoft.com	Pending		₹234.00	82
Office stationary	Bibhuti Dakua	Pending		₹54.00	62
Laptop repairing	Bibhuti Dakua	Approved		₹540.00	63
Wiring	Bibhuti Dakua	Rejected		₹600.00	64
Extra	TestUser01@szggl.onmicrosoft.com	Rejected		₹5,410.00	65
Personal	TestUser01@szggl.onmicrosoft.com	Needs more Information	Please describe more	₹500.00	66
ABC	Sandeep G	Pending	Aur do	₹33.00	67
Sandeep	TestUser01@szggl.onmicrosoft.com	Pending		₹243.00	68
Require laptop		Pending		₹65.00	78
EQWR	TestUser01@szggl.onmicrosoft.com	Pending		₹65.00	79
ABCDER	TestUser01@szggl.onmicrosoft.com	Pending		₹545.00	80
Gallodu require	TestUser01@szggl.onmicrosoft.com	Approved		₹432.00	81

## Benefits of this approach

**Enhanced Efficiency:** Automating notifications and approvals reduces manual follow-ups and speeds up the decision-making process.



**Improved User Experience:** Users can easily track their requests and respond promptly if additional information is required.

**Clear Communication:** Direct links in emails facilitate quick access for approvers, ensuring timely reviews.

## **Conclusion**

This automated request submission and approval workflow demonstrates how integrating Power Apps with SharePoint and Power Automate can significantly enhance operational efficiency. By providing users with clear visibility into their requests and enabling approvers to act swiftly, organizations can foster better collaboration and streamline their processes. Feel free to adjust any sections or details to better fit your specific project context!



**SOFTREE**  
**TECHNOLOGY**