**📦 Prompt Card 1 – SharePoint Task Notification via Cloud Flow**

**🪄 Magic Sentence (User Input)**

I want to automate sending reminder emails when SharePoint tasks are overdue.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 4 Simple Steps (Clean Summary)**

* **ST1**: Trigger flow every weekday at 9:00 AM
* **ST2**: Get items from SharePoint list [List Name] at [Site URL]
* **ST3**: Filter tasks where [Due Date] < today AND [Status] ≠ Completed
* **ST4**: Send email to [Assignee Email] with task summary

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Recurrence | Trigger daily | [Time] | Set recurrence schedule |
| ST2 | Get items - SharePoint | Retrieve task list | [Site URL], [List Name] | Define site + list |
| ST3 | Filter array | Find overdue tasks | [Due Date], [Status] | Customize filter expression |
| ST4 | Send Email (V2) - Outlook | Notify task owner | [Email Body] | Define email body + recipient |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that runs every weekday at 9:00 AM. It should get items from a SharePoint list at [Site URL], filter tasks where the Due Date is earlier than today and the Status is not Completed, then send an email via Outlook to the task assignees with a summary of overdue tasks."

**🧠 Notes for KAI Use:**

* ✅ Designed for **Cloud Flow using Lite connectors only**
* 🔁 [Dynamic Fields] must be filled by user
* 🔒 Always apply privacy masking before Copilot use
* 📄 Use in conjunction with KAI MASTER INSTRUCTION v8.4

✅ End of Prompt Card 1

**📦 Prompt Card 2 – SharePoint ➝ Teams Alert for New Form Entry**

**🪄 Magic Sentence (User Input)**

I want to automate notifying my team in Microsoft Teams when someone submits a Microsoft Form linked to SharePoint.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 4 Simple Steps (Clean Summary)**

* **ST1**: Trigger when a new response is submitted in Microsoft Forms
* **ST2**: Get response details using Form ID
* **ST3**: Add response to SharePoint List [List Name] at [Site URL]
* **ST4**: Post a message in Microsoft Teams [Channel Name] with key details

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Microsoft Forms Trigger | Detect form submission | [Form ID] | Select correct Form |
| ST2 | Get response details | Extract answers | [Question Fields] | Map field names |
| ST3 | Create item - SharePoint | Store record | [Site URL], [List Name] | Select SharePoint list |
| ST4 | Post message - Teams | Notify team | [Channel], [Message Content] | Define team/channel/message |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that starts when someone submits a Microsoft Form. Then get the response details, add the response into a SharePoint list at [Site URL], and post a message in Microsoft Teams [Channel Name] with the form data."

**🧠 Notes for KAI Use:**

* ✅ Designed for **Cloud Flow using Lite connectors only**
* 🔁 [Dynamic Fields] must be filled by user
* 🛡️ Microsoft Forms ID must be correct (watch for permissions)
* 🔒 Always apply privacy masking before Copilot use

✅ End of Prompt Card 2

**📦 Prompt Card 3 – Outlook Email ➝ SharePoint Log Entry**

**🪄 Magic Sentence (User Input)**

I want to automatically log incoming emails into a SharePoint list for tracking.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 4 Simple Steps (Clean Summary)**

* **ST1**: Trigger when an email arrives in Outlook (with specific subject/filter)
* **ST2**: Extract key fields: Subject, Sender, Body Preview, Date
* **ST3**: Create a new item in SharePoint list with extracted fields
* **ST4**: (Optional) Send confirmation email to self or team

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Outlook Trigger | Detect new email | [Folder], [Filter] | Choose inbox, apply rules |
| ST2 | Extract email fields | Get metadata | [Subject], [Sender], [Body Preview] | Confirm mapping |
| ST3 | Create item - SharePoint | Log into list | [Site URL], [List Name] | SharePoint config |
| ST4 | Send Email (V2) | Optional notification | [To], [Body] | Optional customization |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that triggers when a new email arrives in Outlook (with subject filter). Extract the subject, sender, and preview text, then create a new item in a SharePoint list at [Site URL] and [List Name]. Optionally, send a confirmation email to [Recipient]."

**🧠 Notes for KAI Use:**

* ✅ Lite connector safe
* 🔁 Best for email-based reporting/tracking
* 🔒 Use privacy masking on email addresses before prompt use

✅ End of Prompt Card 3

**📦 Prompt Card 4 – SharePoint ➝ Auto Approval Workflow (Manager Flow)**

**🪄 Magic Sentence (User Input)**

I want to automate approval of SharePoint requests and notify the requester.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 4 Simple Steps (Clean Summary)**

* **ST1**: Trigger when a new item is added to SharePoint [List Name]
* **ST2**: Start approval process and assign to [Manager Email]
* **ST3**: Wait for approval response (Approve/Reject)
* **ST4**: Update SharePoint item + send email to requester with decision

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | SharePoint Trigger | On new request submission | [Site URL], [List Name] | Choose correct SharePoint list |
| ST2 | Start and Wait for Approval | Begin approval logic | [Approver Email] | Assign to manager |
| ST3 | Condition | Check outcome (Approve/Reject) | Response | Define follow-up logic |
| ST4 | Update item + Send Email | Notify result | [Decision], [Recipient] | Customize status + email content |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that triggers when a new item is submitted in SharePoint [List Name]. Start an approval process assigned to [Manager Email]. Once approved or rejected, update the SharePoint item and send an email to the requester with the outcome."

**🧠 Notes for KAI Use:**

* ✅ Lite connector only, standard SharePoint + Outlook + Approvals
* 🔁 Make sure Approver email field is valid and permissioned
* 🔒 Apply privacy masking for SharePoint links or names before generating

✅ End of Prompt Card 4

**📦 Prompt Card 5 – Microsoft Planner ➝ Due Date Reminder**

**🪄 Magic Sentence (User Input)**

I want to send a reminder email when Microsoft Planner tasks are due soon.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 4 Simple Steps (Clean Summary)**

* **ST1**: Trigger flow daily at 7:00 AM
* **ST2**: List tasks from Planner group [Group Name], [Plan ID]
* **ST3**: Filter tasks where [Due Date] = Today or Tomorrow
* **ST4**: Send summary email to assigned users

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Recurrence | Trigger daily | [Time] | Set time (e.g. 7:00 AM) |
| ST2 | List tasks - Planner | Get task list | [Group], [Plan ID] | Identify Planner config |
| ST3 | Filter array | Match due today or tomorrow | [Due Date] | Set filter logic |
| ST4 | Send Email (V2) | Alert assignees | [User Email], [Task Summary] | Customize recipients and message |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that runs daily at 7:00 AM. It should list all tasks from a Microsoft Planner plan [Group Name], [Plan ID], filter the tasks due today or tomorrow, and email a summary of those tasks to the assigned users."

**🧠 Notes for KAI Use:**

* ✅ Planner connectors are available in Lite
* 🔁 [Due Date] should use utcNow() + 1 logic
* 🔒 Mask group and user details before prompt export

✅ End of Prompt Card 5

**📦 Prompt Card 6 – Microsoft Forms ➝ Excel Logger (Cloud Flow)**

**🪄 Magic Sentence (User Input)**

I want to record Microsoft Forms submissions into an Excel table automatically.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 4 Simple Steps (Clean Summary)**

* **ST1**: Trigger when a new response is submitted in Microsoft Forms
* **ST2**: Get response details from form [Form ID]
* **ST3**: Add a new row to Excel Online (Business) table at [File Path]
* **ST4**: (Optional) Send confirmation to respondent

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Microsoft Forms Trigger | Detect new form entry | [Form ID] | Select correct form |
| ST2 | Get response details | Extract answers | [Field Names] | Match fields to Excel columns |
| ST3 | Add a row - Excel Online | Record data | [Excel File], [Table Name] | File path + structure |
| ST4 | Send Email (V2) | Confirm submission | [Recipient], [Message] | Optional notification content |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that starts when someone submits a Microsoft Form [Form Name]. Then get the response details and record them into a table in an Excel Online file located at [File Path]. Optionally, send a confirmation email to the user."

**🧠 Notes for KAI Use:**

* ✅ Works with Excel Online (Business) only
* 🔁 Table in Excel must be pre-created with column headers
* 🔒 Mask form names and file paths before generating prompt

✅ End of Prompt Card 6

**📦 Prompt Card 7 – SharePoint Item Expiry Reminder**

**🪄 Magic Sentence (User Input)**

I want to send a reminder before an item in SharePoint reaches its expiry date.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 4 Simple Steps (Clean Summary)**

* **ST1**: Trigger flow daily at 8:00 AM
* **ST2**: Get items from SharePoint [List Name] at [Site URL]
* **ST3**: Filter items where [Expiry Date] = Today or within next 3 days
* **ST4**: Send reminder email to [Assigned Person or Group]

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Recurrence | Run daily at 8:00 AM | [Time] | Set time/frequency |
| ST2 | Get items - SharePoint | Retrieve records | [Site URL], [List Name] | SharePoint list config |
| ST3 | Filter array | Check for soon-expiring items | [Expiry Date] | Use date comparison logic |
| ST4 | Send Email (V2) | Notify responsible person | [Email], [Item Info] | Email body + recipient |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that runs daily at 8:00 AM. It should get items from a SharePoint list at [Site URL], check if the [Expiry Date] is today or within the next 3 days, and send a reminder email to the assigned contact."

**🧠 Notes for KAI Use:**

* ✅ Useful for license expiry, certificate reminders, equipment due dates, etc.
* 🔁 Date filtering uses addDays(utcNow(), X) for comparison
* 🔒 Mask all URLs and emails before prompt generation

✅ End of Prompt Card 7

**📦 Prompt Card 8 – Microsoft Forms ➝ Conditional Email Routing**

**🪄 Magic Sentence (User Input)**

I want to send Microsoft Form responses to different emails based on the selected department.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 4 Simple Steps (Clean Summary)**

* **ST1**: Trigger when a new Microsoft Form is submitted
* **ST2**: Get response details from [Form ID]
* **ST3**: Evaluate the answer to [Department] field
* **ST4**: Send email to the mapped recipient based on department

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Microsoft Forms Trigger | Detect new form submission | [Form ID] | Choose the correct form |
| ST2 | Get response details | Extract data | [Department Field] | Map to condition logic |
| ST3 | Condition | Branch by selected department | [If Dept = A, B, C...] | Customize each case |
| ST4 | Send Email (V2) | Email routed to matching group | [Email Address] | Define recipients by department |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that starts when someone submits a Microsoft Form. Retrieve the department field from the response. If the department is 'IT', send an email to [IT Group]. If it's 'HR', send to [HR Group], and so on."

**🧠 Notes for KAI Use:**

* ✅ Basic branching supported in Lite version
* 🔁 Can extend to Switch Case if more than 3 departments
* 🔒 Mask all emails + department names before prompt use

✅ End of Prompt Card 8

**📦 Prompt Card 9 – Power BI Alert ➝ Microsoft Teams Notification**

**🪄 Magic Sentence (User Input)**

I want to send a Teams message when a Power BI KPI crosses a set threshold.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 4 Simple Steps (Clean Summary)**

* **ST1**: Trigger flow when a Power BI alert is activated (KPI threshold crossed)
* **ST2**: Extract alert context (value, threshold, timestamp)
* **ST3**: Format alert message with [KPI Name], [Value], [Limit]
* **ST4**: Post message to Microsoft Teams [Channel Name]

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Power BI Trigger | Detect KPI alert | [Alert Name] | Set in Power BI dashboard |
| ST2 | Parse alert details | Get KPI and value | [KPI Field] | Add label/value context |
| ST3 | Compose message | Format Teams content | [Threshold/Value] | Customize alert format |
| ST4 | Post message - Teams | Notify channel | [Team/Channel] | Select Teams destination |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that starts when a Power BI alert is triggered. Format the alert with KPI name, current value, and threshold, then post a message to a Microsoft Teams channel [Channel Name] with the details."

**🧠 Notes for KAI Use:**

* ✅ Requires Power BI alert configured on visual tile
* 🔁 Teams connector is Lite-compatible
* 🔒 Mask KPI, report name, and team name before exporting prompt

✅ End of Prompt Card 9

**📦 Prompt Card 10 – Daily Email Digest from SharePoint List**

**🪄 Magic Sentence (User Input)**

I want to send a daily email summary of newly added SharePoint items.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 4 Simple Steps (Clean Summary)**

* **ST1**: Trigger flow daily at [Time]
* **ST2**: Get items from SharePoint list where [Created Date] = Today
* **ST3**: Format results into a table or bullet list
* **ST4**: Send summary email to [Team Distribution List]

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Recurrence | Trigger daily | [Time] | Define time (e.g. 7:00 AM) |
| ST2 | Get items - SharePoint | Filter today's entries | [Created Date] | Use utcNow() date logic |
| ST3 | Select/HTML Table | Create summary view | [Fields] | Choose fields to include |
| ST4 | Send Email (V2) | Email daily digest | [Recipients] | Customize email content + target group |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that runs every day at [Time], retrieves new items added today from a SharePoint list, formats them into a summary table, and sends a daily digest email to [Distribution List]."

**🧠 Notes for KAI Use:**

* ✅ Designed for team visibility / coordination
* 🔁 Combine Get Items + Filter Query for Created = Today
* 🔒 Ensure SharePoint path and emails are masked before output

✅ End of Prompt Card 10

**📦Prompt Card 11 – Email Attachment ➝ Extract ➝ Email + Excel Log**

**🪄 Magic Sentence (User Input)**

I want to extract data from PDF email attachments and use it in an email and Excel log.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**  
(⚠️ Assumes PDF extraction is handled via pre-trained AI model or connected parser)

**🧱 5 Simple Steps (Clean Summary)**

* **ST1**: Trigger when a new email arrives (with filters)
* **ST2**: Check if email contains attachments + match PDF file
* **ST3**: Extract data from PDF using [AI Builder Model] or custom parser
* **ST4**: Send extracted data by email (reusing subject from original email)
* **ST5**: Add a row in Excel Online table with the extracted content

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Outlook Trigger | Detect new filtered email | [From, Subject Filter] | Email rule setup |
| ST2 | Condition + Get Attachments | Check for PDF presence | [.pdf] | Filter attachment type |
| ST3 | Parse PDF | Extract structured data | [Trained Model ID] | Must configure extraction method |
| ST4 | Send Email (V2) | Email extracted text | [Original Subject], [Extracted Text] | Format message |
| ST5 | Add a row - Excel Online | Log to table | [Excel File], [Table Columns] | Setup file + mapping |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that starts when an email arrives from a specific sender with a subject filter. If the email has a PDF attachment, extract data using a pre-trained model or parser. Then, email the extracted content using the original email subject and also log the content as a new row in Excel Online."

**🧠 Notes for KAI Use:**

* ✅ Common use case in reporting, invoicing, log extraction
* 🔁 PDF parsing assumes pre-trained logic (e.g. AI Builder Form Processor or 3rd-party tool)
* 🔒 Mask sender address, file paths, and subject keywords in public prompt
* ❗ If no AI Builder, advise user to configure external parsing step manually

✅ End of Prompt Card 11

**📦 Prompt Card 12 – Critical Defect Email ➝ Excel Compilation + Script**

**🪄 Magic Sentence (User Input)**

I want to extract structured data from critical defect emails and log it into Excel with a script triggered at the end.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 6 Simple Steps (Clean Summary)**

* **ST1**: Trigger flow when a new Outlook email arrives with subject Machine Shutdown Control Notification [Critical Alarm]
* **ST2**: Convert email HTML body into plain text
* **ST3**: Parse and slice key data (e.g., MC, Lot No, Defect, Date) using index logic
* **ST4**: Add parsed data to Excel Table 1: CriticalDefectSummary
* **ST5**: Log full message and date to Excel Table 2: MessageLog
* **ST6**: Run Office Script to update formatting or post-processing

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Outlook Trigger | Detect critical alarm email | [Subject Filter] | Email subject: [Critical Alarm] |
| ST2 | HTML ➝ Text | Clean up body content | [Email Body] | Use conversion service connector |
| ST3 | Text Parser (Expression) | Slice values using index/slice | [MC, Lot, Defect, Op] | Update string logic |
| ST4 | Add row - Excel Online | Save summary entry | [Excel File Path], [Table 1] | Table for defect summary |
| ST5 | Add row - Excel Online | Save full message log | [Excel File Path], [Table 2] | Table for archive/logs |
| ST6 | Run Office Script | Finalize Excel updates | [Script ID] | Script for formatting/post-process |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that triggers when a new Outlook email arrives with subject containing [Critical Alarm]. Convert the email body to plain text, extract key fields using text functions (MC, Lot No, Defect, Date), log them into Excel Table 1, and also log the full message into Excel Table 2. After that, run an Office Script to finalize the file."

**🧠 Notes for KAI Use:**

* ✅ Logic slicing assumes consistent email format (indexOf + slice)
* ✅ Excel actions use Excel Online (Business) – no desktop PAD required
* 🔁 Office Script must be preconfigured with workbook + script ID
* 🔒 Mask email subjects and Excel paths before generating prompt

# ✅ End of Prompt Card 12 📦 Prompt Card 13 – Daily Excel Script ➝ Conditional Email Notification

**🪄 Magic Sentence (User Input)**

I want to send a daily email if a result from an Excel script is found — otherwise send a 'no alert' message.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 5 Simple Steps (Clean Summary)**

* **ST1**: Trigger flow daily at [Time]
* **ST2**: Run Office Script on Excel file in OneDrive ➝ get result
* **ST3**: Evaluate if result field (e.g., notifyname) is not blank
* **ST4a**: If true ➝ Send alert email with result data (e.g., machine affected)
* **ST4b**: Else ➝ Send email saying “No notification info” with friendly image

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Recurrence | Trigger daily | [Time] | Choose time (e.g., 7:00 AM) |
| ST2 | Run Office Script (Excel Online) | Generate result from file | [File Path], [Script ID] | Excel in OneDrive + output field |
| ST3 | Condition | Check if result ≠ blank | [notifyname] | Decision control logic |
| ST4a | Send Email (V2) | Notify if critical found | [Subject], [Body with Info] | Email recipient & alert content |
| ST4b | Send Email (V2) | Friendly no-info reply | [Subject], [Image] | Optional emoji/image feedback |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that runs daily. It triggers an Excel Online script stored in OneDrive. If the script returns a value (notifyname), send an email with the alert content. If not, send a different email saying no notification is needed with a friendly icon."

**🧠 Notes for KAI Use:**

* ✅ Office Script must return a field (e.g., notifyname) for evaluation
* ✅ Excel file must be in OneDrive or SharePoint with permission
* 🔁 Supports automated machine-critical alerting
* 🔒 Always mask filenames and recipient emails before public prompt

✅ End of Prompt Card 13

**📦 Prompt Card 14 – SharePoint ➝ Approval Alert Based on Condition**

**🪄 Magic Sentence (User Input)**

I want to notify a specific person only if a SharePoint request meets a certain condition.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 4 Simple Steps (Clean Summary)**

* **ST1**: Trigger flow when a new item is added to SharePoint List [List Name]
* **ST2**: Evaluate a specific field (e.g., if Amount > 5000 or Priority = High)
* **ST3**: If the condition is met ➝ send an alert or approval request
* **ST4**: (Optional) Log the event or update SharePoint field

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | SharePoint Trigger | Detect new item | [Site URL], [List Name] | Setup trigger site/list |
| ST2 | Condition | Evaluate input field | [Amount], [Priority] | Define logical condition |
| ST3 | Send Email or Approval | Notify relevant user | [Email], [Message] | Setup routing & message logic |
| ST4 | Update Item (optional) | Mark item as alerted/logged | [Status Field] | Optional tracking update |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that triggers when a new item is added to a SharePoint list. If the value of [Amount] is greater than 5000 or [Priority] equals 'High', send an approval or alert email to [Manager]. Optionally update the item with status 'Notified'."

**🧠 Notes for KAI Use:**

* ✅ Useful for finance, safety alerts, urgent IT tickets, etc.
* 🔁 You can add multiple condition branches for different routing
* 🔒 Mask SharePoint URLs and internal emails before sharing prompt

# ✅ End of Prompt Card 14

# 📦 Prompt Card 15 – Auto Weekly Summary by Department

**🪄 Magic Sentence (User Input)**

I want to send a weekly email summary grouped by department based on data in SharePoint.

**🧭 Recommended Flow Type**

**Cloud Flow (Lite Compatible)**

**🧱 5 Simple Steps (Clean Summary)**

* **ST1**: Trigger flow every Monday at 8:00 AM
* **ST2**: Get items from SharePoint list [List Name] filtered for past 7 days
* **ST3**: Group data by department and aggregate result (count, % or sum)
* **ST4**: Format summary into HTML table or bullet list
* **ST5**: Send formatted email to [Department Heads or Distribution List]

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Recurrence | Trigger weekly | [Time] | Set to Monday 8:00 AM |
| ST2 | Get items - SharePoint | Retrieve last 7 days | [Created Date] | Use addDays(utcNow(),-7) logic |
| ST3 | Select + Group (manual) | Group by department | [Dept Field] | Use expressions or compose logic |
| ST4 | Compose / HTML Table | Format for email | [Summary Rows] | Choose fields to display |
| ST5 | Send Email (V2) | Send weekly digest | [To], [Body] | Email target group + message |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that runs every Monday at 8:00 AM. It should get items from a SharePoint list added in the past 7 days, group them by department, format a summary of each department’s submission count, and email the report to department heads."

**🧠 Notes for KAI Use:**

* ✅ Good for RPA adoption tracking, HR logs, maintenance trends
* 🔁 Grouping logic may use loop or expression workaround (if not using premium connector)
* 🔒 Mask SharePoint URL and emails before public sharing

✅ End of Prompt Card 15

**📦 Prompt Card 16 – AI Hub: Scan Invoice PDF ➝ Excel + Email Approval**

**🪄 Magic Sentence (User Input)**

I want to extract fields from invoice PDFs and log them in Excel, then email for approval.

**🧭 Recommended Flow Type**

**Cloud Flow (AI Builder + Excel Online)**

**🧱 5 Simple Steps (Clean Summary)**

* **ST1**: Trigger flow when a new invoice email arrives with PDF attached
* **ST2**: Use AI Builder Form Processing model to extract key invoice fields (e.g., Vendor, Date, Amount)
* **ST3**: Add extracted data into Excel Online table [InvoiceLog.xlsx]
* **ST4**: Format summary with invoice info
* **ST5**: Send approval email with Accept/Reject buttons

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Outlook Trigger | Detect new invoice email | [From, Subject] | Set filter criteria |
| ST2 | AI Builder – Form Processor | Extract structured fields from PDF | [Model ID] | Use trained invoice model |
| ST3 | Add row – Excel Online | Record invoice data | [Excel File], [Table] | Setup target file/columns |
| ST4 | Compose / Format | Create readable summary | [Vendor, Amount, Date] | Format for approver clarity |
| ST5 | Send Approval Email | Trigger Approve/Reject | [To], [Approval Options] | Set approver + logic |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that starts when an invoice email with PDF is received. Use an AI Builder model to extract fields like Vendor, Amount, Date. Save the results to an Excel Online file, format a summary, and send an approval email with Approve/Reject buttons."

**🧠 Notes for KAI Use:**

* ⚠️ Requires AI Builder credits and pre-trained Form Processing model
* ✅ Excel and Approval connectors are Lite-compatible
* 🔁 Common for finance, AP automation, or vendor PO matching
* 🔒 Mask vendor names, amounts, and file paths before reuse

✅ End of Prompt Card 16

**📦 Prompt Card 17 – SharePoint ➝ Scan Latest Invoice PDF ➝ Excel + Approval**

**🪄 Magic Sentence (User Input)**

I want to extract fields from the latest invoice PDF uploaded in SharePoint and log them in Excel, then email for approval.

**🧭 Recommended Flow Type**

**Cloud Flow (AI Builder + SharePoint + Excel Online)**

**🧱 6 Simple Steps (Clean Summary)**

* **ST1**: Trigger flow on a fixed schedule (e.g., every 15 minutes)
* **ST2**: List files from SharePoint folder [Invoices/Incoming]
* **ST3**: Filter to get only .pdf files and sort by Created or Modified date descending
* **ST4**: Use AI Builder Form Processor to extract invoice fields from the latest file (Vendor, Date, Amount)
* **ST5**: Add extracted data into Excel table [InvoiceLog.xlsx]
* **ST6**: Send summary + approval email to [Approver Email]

**🔄 Smart LEGO Workflow Table**

| **Line** | **Action/Connector** | **Purpose** | **Dynamic Field** | **What User Must Customize** |
| --- | --- | --- | --- | --- |
| ST1 | Recurrence | Trigger every X mins | [Time] | Set interval (e.g. 15 min) |
| ST2 | Get files - SharePoint | Detect uploaded PDFs | [Library], [Folder Path] | SharePoint config |
| ST3 | Filter + Sort | Pick latest .pdf file | [File Extension], [Created] | Apply filter/sort logic |
| ST4 | Form Processor (AI Builder) | Extract data fields | [AI Model ID] | Trained invoice model |
| ST5 | Add row – Excel Online | Save to Invoice Table | [Excel File], [Table Name] | Set up table headers |
| ST6 | Send Email (V2) | Approval message | [Approver], [Summary] | Customize recipients + message |

**💬 Copilot/ChatGPT Prompt (Paste-Ready)**

"Create a cloud flow that runs every 15 minutes. It checks a SharePoint folder for the latest uploaded PDF invoice, uses AI Builder to extract fields (Vendor, Date, Amount), logs the values into Excel, and sends a summary email for approval."

**🧠 Notes for KAI Use:**

* ✅ Sorting by Created or Modified column ensures latest file detection
* ✅ AI Builder must be pre-trained and licensed
* 🔁 This flow supports both invoice logging and quick verification
* 🔒 Always mask file names, customer, and approver data before publishing

✅ End of Prompt Card 17