

AGENTIC AI IN POWER PLATFORM

Human-in-the-Loop (HITL)

Agent Flows with Request for Information + AI Claims Review

Standalone Lab Guide | BFSI Sector | Contoso Insurance

3 Datasets | 1 Complete Lab | Fully Independent

February 2026

Table of Contents

- 1. Lab Overview**
- 2. Prerequisites**
- 3. Datasets Provided**
- 4. Data Preparation: Import Data into Dataverse**

- 4.1 Create Dataverse Tables**
- 4.2 Import Data from Excel**
- 4.3 Verify Data**

- 5. Create the Copilot Studio Agent**

- 6. Lab: Agent Flow HITL — RFI + AI Claims Review**

- 6.1 Create the Agent Flow with Trigger**
- 6.2 Add Dataverse List Rows Action**
- 6.3 Add Request for Information (RFI) Action**
- 6.4 Add the AI Prompt Action**
- 6.5 Add the Response Action**
- 6.6 Test the Flow (Manual)**
- 6.7 Test via Agent Chat**
- 6.8 Test Scenarios**
- 6.9 Validation Checkpoint**

Appendix A: Dataset Reference

Appendix B: Troubleshooting Guide

Appendix C: Key Concepts

1. Lab Overview

This standalone lab guide teaches you how to build a Human-in-the-Loop (HITL) Agent Flow in Copilot Studio. The flow pauses execution to collect human input from a field adjuster, then uses AI to review the combined claim data and adjuster assessment before returning an approval recommendation.

What You Will Build

An Agent Flow in Copilot Studio that:

Step	Action	Type
1	Retrieves all insurance claims from Dataverse	Automated
2	Sends a Request for Information (RFI) email to a field adjuster with 4 input fields	Human (HITL)
3	Waits for the adjuster to submit their damage assessment, cost estimate, and recommendation	Human (HITL)
4	Feeds claim data + adjuster input to an AI prompt that evaluates and recommends approve/reject	AI Review
5	Returns the AI recommendation back to the Copilot Studio agent	Automated

What Makes This HITL?

Human-in-the-Loop means the automated flow PAUSES and waits for a human to provide input before continuing. The flow literally stops at the RFI action, sends an email to the field adjuster, and does NOT continue until the adjuster opens the email and submits their response. This is fundamentally different from fully automated flows that run start-to-finish without any human intervention.

Flow Architecture

```

Agent Chat: "Review claim CLM-0005"
↓
Agent Flow Trigger (input: ClaimNumber)
↓
List Rows – Dataverse (Contoso_Claims, all records)
↓
REQUEST FOR INFORMATION ← FLOW PAUSES HERE
→ Email sent to adjuster via Outlook
→ 4 fields: Damage Assessment, Repair Cost,
   Recommendation, Inspection Date
→ Adjuster fills in and clicks Submit
↓ (flow resumes after human response)
Run a Prompt – AI evaluates:
• Claim data from Dataverse

```

- Adjuster's field assessment
- Cost comparison (repair vs claim amount)
- Returns: APPROVE / REJECT / NEEDS REVIEW
- ↓
- Respond to Agent (output: ReviewResult)

2. Prerequisites

Requirement	Details
Microsoft 365 / Power Platform License	Access to Copilot Studio (copilotstudio.microsoft.com) and Dataverse
Copilot Studio License	To create agents and Agent Flows
Outlook Access	The RFI action sends emails via Outlook — you need access to your inbox during the lab
Power Platform Environment	A Dataverse environment where you can create tables and import data
Lab Data Files	3 Excel files provided: Contoso_Claims.xlsx, Contoso_Customers.xlsx, Contoso_Policies.xlsx

⚠ Preview Feature

The Request for Information action is part of the Human in the loop connector in Agent Flows, currently in Preview. It is fully functional for training and testing. Known limitation: RFI sends the request via Outlook email only (not Teams).

3. Datasets Provided

This lab uses 3 curated datasets from the Contoso Insurance BFSI scenario. Each file is a clean Excel workbook ready for direct import into Dataverse.

File	Table	Records	Key Columns	Role in Lab
Contoso_Claims.xlsx	Contoso_Claims	200	ClaimID, PolicyID, CustomerID, ClaimAmount, ClaimType, Status, AdjusterName	Primary — the flow queries this table and the AI reviews specific claims
Contoso_Customers.xlsx	Contoso_Customers	100	CustomerID, FullName, City, State, RiskCategory, KYCStatus	Context — provides customer risk category and demographics for AI review
Contoso_Policies.xlsx	Contoso_Policies	150	PolicyID, CustomerID, PolicyType, PremiumAmount, CoverageAmount, Status	Context — provides policy details and coverage amounts for cost comparison

Sample Claims for Testing

ClaimID	Amount (INR)	Type	Status	Adjuster	Description
CLM-0005	59,445	Third Party Liability	Approved	Vikram Reddy	Third party vehicle damage in accident
CLM-0010	2,67,087	Theft	Approved	Sanjay Patel	Luggage stolen at airport
CLM-0015	1,61,773	Accidental Damage	Open	Meera Nair	Laptop damaged in transit
CLM-0020	38,249	Accidental Damage	Denied	Sanjay Patel	Laptop damaged in transit
CLM-0001	1,55,742	Medical Emergency	Closed	Amit Verma	Emergency hospitalisation — cardiac event

4. Data Preparation: Import Data into Dataverse

Before building the Agent Flow, you need the Contoso Insurance data in Dataverse tables. Follow these steps to create the tables and import the Excel data.

Already Have Dataverse Tables?

If you already have the Contoso Insurance tables in Dataverse from a previous module, skip to Section 5. Verify that your Contoso_Claims table has 200 records, Contoso_Customers has 100, and Contoso_Policies has 150.

4.1 Create Dataverse Tables

Step 1: Navigate to <https://make.powerapps.com> and select your target environment.

Step 2: In the left navigation, select Tables.

Step 3: Click + New table → Create new tables.

Step 4: Select Import an Excel file or CSV file.

Step 5: Upload Contoso_Claims.xlsx. Power Apps will preview the data.

Step 6: Review the column mappings. Ensure:

- a. ClaimID is set as the primary column (Text)
- b. ClaimAmount is recognized as Currency or Decimal Number
- c. ClaimDate is recognized as Date
- d. All other columns are Text (Single line of text)

Step 7: Click Create. Wait for the import to complete.

Step 8: Repeat Steps 3–7 for Contoso_Customers.xlsx:

CustomerID as primary column
DateOfBirth, CustomerSince as Date

Step 9: Repeat Steps 3–7 for Contoso_Policies.xlsx:

PolicyID as primary column
PremiumAmount, CoverageAmount as Currency or Decimal Number
StartDate, EndDate as Date

4.2 Verify Data

Step 10: After all 3 imports complete, verify each table:

Table	Expected Records	How to Check
Contoso_Claims	200	Tables → Contoso_Claims → Data tab → check row count
Contoso_Customers	100	Tables → Contoso_Customers → Data tab → check row count
Contoso_Policies	150	Tables → Contoso_Policies → Data tab → check row count

Step 11: Open the Contoso_Claims table and verify that ClaimID values show as CLM-0001, CLM-0002, etc. and that ClaimAmount shows numerical values.

 **Data Preparation Complete**

You now have 3 Contoso Insurance tables in Dataverse with a total of 450 records. The Agent Flow will query the Claims table directly. The Customers and Policies tables provide additional context that the AI prompt can reference.

5. Create the Copilot Studio Agent

If you already have a Copilot Studio agent from a previous module, you can reuse it. Otherwise, create a new one.

Step 1: Navigate to <https://copilotstudio.microsoft.com>

Step 2: Select + Create → New agent.

Step 3: Configure:

Name: Contoso Insurance HITL Agent

Description: An insurance claims agent that uses human-in-the-loop workflows for claim reviews.

Instructions: You are a claims operations assistant for Contoso Insurance. When a user asks to review a claim, use the HITL Claim Review Flow to collect a field assessment and get an AI-powered recommendation.

Step 4: Click Publish (if using trial environment or let it be as it is).

Step 5: Verify the agent is created and you can see the Overview, Knowledge, and Tools tabs.

6. Lab: Agent Flow HITL — RFI + AI Claims Review

Property	Detail
Objective	Build an Agent Flow that pauses for human input (RFI) and uses AI to review the combined claim data + adjuster assessment
HITL Feature	Request for Information (Human in the loop connector, Preview)
AI Feature	Run a prompt — AI reviews claim + adjuster input and gives approval recommendation
Trigger	Instant — When an agent calls the flow (Run from Copilot)
Duration	35 minutes

6.1 Create the Agent Flow with Trigger

Step 1: In Copilot Studio, open your agent. Navigate to Flows in the left navigation.

Step 2: Select + New flow → Build from blank.

Step 3: Change the trigger to "When an agent calls the flow" (Run from Copilot).

Step 4: In the trigger card, select + Add an input → Text. Configure:

Setting	Value
Input name	ClaimNumber
Description	The claim number to review (e.g., CLM-0005)

Step 6: Click Save Draft.

Step 5: Name the flow: Contoso - HITL Claim Review Flow

6.2 Add the Dataverse List Rows Action

Step 7: Below the trigger, select + Add an action → Microsoft Dataverse → List rows.

Step 8: Configure:

Table name: Your Contoso_Claims table (select from the dropdown and check the name you may have given)

Filter rows: Leave empty (we retrieve all records and let the AI filter by claim number)

💡 Why No Filter?

We retrieve all claims rather than filtering by ClaimNumber because: (1) OData filters on certain column types (like Choice columns) can cause BadGateway errors in Agent Flows, and (2) the AI prompt receives the full dataset via the string() expression and extracts the relevant claim. This is a proven, reliable pattern.

6.3 Add the Request for Information (RFI) Action

This is the HITL step. The flow will PAUSE here and send an email to the field adjuster. The flow does NOT continue until the adjuster responds.

Step 9: Below the List rows action, select + Add an action.

Step 10: Search for "Request for information" or browse under Human in the loop. Select Request for information.

Finding the Action

If you don't see "Human in the loop" in the connector list, search for "Request for information" directly in the action search bar. The action is part of the Human in the loop connector which is in Preview. If it doesn't appear at all, your environment may not have the preview features enabled — check with your admin.

Step 11: Configure the required fields:

Field	Value
Title	Field Assessment Required: Claim Review
Message	A claim has been submitted for review. Please provide your field assessment based on your inspection. Your input will be used by AI to evaluate the claim.
Assigned to	Your own email address (for testing — in production this would be the adjuster's email)

Step 12: Add input fields. Select + Add an input and add the following one by one:

Input Name	Type	Placeholder / Notes
Damage Assessment	Text	Describe the damage observed during field inspection
Estimated Repair Cost	Number	Enter the estimated repair or treatment cost in INR
Recommend Approval	Yes/No	(Toggle input — adjuster's recommendation)
Inspection Date	Date	Date when field inspection was conducted

Step 13: Click Save Draft.

What Happens at Runtime

When the flow reaches this action: (1) The flow PAUSES — it does not proceed to the next action. (2) An email is sent via Outlook to the assigned person with the title, message, and all 4 input fields rendered as a structured form. (3) The person opens the email, fills in all fields, and clicks Submit. (4) Only then does the flow RESUME. The submitted values become available as dynamic content for subsequent actions.

6.4 Add the AI Prompt Action

Now add an AI prompt that receives both the Dataverse claims data AND the adjuster's RFI responses.

Step 14: Below the Request for information action, select + Add an action → AI Builder → Create text with GPT using a prompt (or search for "Run a prompt").

Step 15: Select Create a new prompt.

Step 16: Enter the prompt instructions:

You are a senior claims review AI for Contoso Insurance.

TASK: Evaluate the claim and provide an approval recommendation.

CLAIMS DATA (all records from Dataverse):

/ClaimsData

FIELD ADJUSTER'S ASSESSMENT:

- Damage Assessment: /DamageAssessment
- Estimated Repair Cost (INR): /RepairCost
- Adjuster Recommends Approval: /AdjusterRecommendation
- Inspection Date: /InspectionDate

CLAIM TO REVIEW:

EVALUATION RULES:

1. Find the specific claim matching the ClaimNumber in the claims data
2. Compare the estimated repair cost to the claim amount:
 - If repair cost is within 30% of claim amount: Cost is reasonable
 - If repair cost exceeds claim amount by >30%: Flag as over-estimated
 - If repair cost is <50% of claim amount: Flag as under-claimed
3. Check adjuster recommendation
4. Verify damage assessment has sufficient detail (>10 words)

RESPOND WITH:

- RECOMMENDATION: APPROVE or REJECT or NEEDS FURTHER REVIEW
- CLAIM DETAILS: Claim number, amount, type, current status
- COST ANALYSIS: Claim amount vs repair estimate comparison
- ADJUSTER INPUT SUMMARY: Key points from the assessment
- RATIONALE: 2-3 sentences explaining the recommendation
- RISK FLAGS: Any concerns (if applicable)

Create the Content Inputs

Each / reference in the prompt needs a corresponding content input. Click + Add content (below the prompt text area) to create each named input:

Input Name	Type	Source
ClaimsData	Text	Dataverse List rows output (all claims)
DamageAssessment	Text	RFI response — Damage Assessment field
RepairCost	Text	RFI response — Estimated Repair Cost field

Input Name	Type	Source
AdjusterRecommendation	Text	RFI response — Recommend Approval field
InspectionDate	Text	RFI response — Inspection Date field
ClaimNumber	Text	Flow trigger input — ClaimNumber

Step 17: Click Save prompt.

Map the Content Inputs in the Flow Designer

Back in the Flow Designer view, you will see fields for each content input. Map them:

Input	How to Map
ClaimsData	Click the field → select fx tab → enter: string(body('List_rows')?['value'])
DamageAssessment	Click the field → select ⚡ Dynamic content → under Request for information outputs → select Damage Assessment
RepairCost	⚡ Dynamic content → Request for information → Estimated Repair Cost
AdjusterRecommendation	⚡ Dynamic content → Request for information → Recommend Approval
InspectionDate	⚡ Dynamic content → Request for information → Inspection Date
ClaimNumber	⚡ Dynamic content → When an agent calls the flow → ClaimNumber

⚠ Critical: ClaimsData Must Use fx Expression

The ClaimsData input MUST use the fx expression: string(body('List_rows')?['value']). Do NOT use the Dynamic content picker for this field. The List rows action returns an array and the prompt input expects text. Without the string() conversion, the AI receives empty data.

6.5 Add the Response Action

Step 18: Below the prompt action, select + Add an action → search for "Respond to the agent".

Step 19: Add an output:

Setting	Value
Output type	Text
Output name	ReviewResult
Value	⚡ Dynamic content → select the Text output from the prompt action

Step 20: Save and Publish the flow.

6.6 Test the Flow (Manual)

Step 21: In the Flow Designer, select Test → Manually.

Step 22: Enter CLM-0005 as the ClaimNumber input. Click Run flow.

Step 23: The flow will execute the List rows action, then PAUSE at the Request for Information step.

Check Your Outlook Inbox

The email comes from Microsoft Power Automate. It contains the title "Field Assessment Required: Claim Review", your message, and 4 input fields rendered as a form. If you don't see it, check your spam/junk folder. The flow will remain paused until you submit the response.

Step 24: Open the RFI email in Outlook. Fill in the fields:

Field	Value to Enter
Damage Assessment	Vehicle front bumper and headlight assembly damaged in collision. Inspection conducted at Fortis Garage. Damage is consistent with reported incident involving third party vehicle. No structural frame damage observed.
Estimated Repair Cost	55000
Recommend Approval	Yes
Inspection Date	Today's date

Step 25: Click Submit in the email.

Step 26: Return to the flow designer. The flow should resume and execute the AI prompt. Wait for completion.

Step 27: Check the flow run history. Open the completed run and inspect:

- a. Request for information action — shows the adjuster's submitted values
- b. Prompt action — shows the AI's evaluation with RECOMMENDATION, COST ANALYSIS, RATIONALE, and RISK FLAGS
- c. Respond to agent action — shows the ReviewResult output

6.7 Test via Agent Chat

Step 28: Navigate to your agent in Copilot Studio → Tools tab → Add a tool.

Step 29: Search for your flow: "Contoso - HITL Claim Review Flow". Select it and add it.

Step 30: Open the Test your agent panel. Type:

I need a review of claim CLM-0010

Step 31: The agent invokes the flow. The flow pauses at the RFI step. Check your Outlook inbox and respond to the RFI email.

Step 32: After submitting the RFI, the flow resumes, the AI evaluates the claim, and the agent displays the review result in the chat.

6.8 Test Scenarios

Test with different claims and RFI values to see how the AI adjusts its recommendation:

Scenario	Claim	RFI: Repair Cost	RFI: Recommend	RFI: Damage Assessment	Expected AI Result
Reasonable claim	CLM-0005 (₹59K)	55000	Yes	Detailed (4+ sentences)	APPROVE — cost within 30%, adjuster supports
Over-estimated repair	CLM-0010 (₹2.67L)	600000	Yes	Detailed assessment	NEEDS REVIEW — repair cost exceeds claim by >30%
Adjuster rejects	CLM-0015 (₹1.61L)	150000	No	Detailed assessment	REJECT — adjuster does not recommend
Vague assessment	CLM-0020 (₹38K)	35000	Yes	Just 2 words: "Minor damage"	NEEDS REVIEW — insufficient detail (<10 words)
Already denied	CLM-0020 (₹38K)	40000	Yes	Detailed reassessment	APPROVE or REVIEW — AI notes claim was previously Denied

6.9 Validation Checkpoint

Check	Expected Result
Flow has 4 actions	Trigger → List rows → Request for Information → Run a prompt → Respond to agent
RFI email received	Email arrives in Outlook with 4 input fields (text, number, yes/no, date)
Flow pauses at RFI	Flow run history shows the RFI action in "Waiting" state until you respond
Flow resumes after RFI submit	After clicking Submit in email, flow continues to the prompt action
AI prompt receives all data	Prompt output references both claim data AND adjuster input
AI gives structured recommendation	Output includes RECOMMENDATION, COST ANALYSIS, RATIONALE, and RISK FLAGS
Agent receives result	ReviewResult output displayed in agent chat

Lab Complete

You have built an Agent Flow with Human-in-the-Loop capabilities. The flow pauses execution to collect human input (field adjuster's damage assessment via email), then feeds that input along with Dataverse claim data to an AI prompt for intelligent evaluation. This pattern — automated retrieval → human input → AI review → response — is fundamental to production BFSI workflows where both human expertise and AI analysis are needed for decision-making.

Appendix A: Dataset Reference

Contoso_Claims.xlsx (200 rows)

Column	Type	Description
ClaimID	Text	Unique claim identifier (CLM-0001 to CLM-0200)
PolicyID	Text	Reference to the policy (POL-XXXX)
CustomerID	Text	Reference to the customer (CUST-XXXX)
ClaimDate	Date	Date the claim was filed
ClaimAmount	Currency	Claim amount in INR (₹8K to ₹5M+)
ClaimType	Text	Medical Emergency, Personal Injury, Third Party Liability, Theft, Accidental Damage, Natural Disaster
Status	Text	Open, Approved, Denied, Closed, Under Review
Description	Text	Brief description of the claim
AdjusterName	Text	Assigned claims adjuster

Contoso_Customers.xlsx (100 rows)

Column	Type	Description
CustomerID	Text	Unique customer identifier (CUST-0001 to CUST-0100)
FullName	Text	Customer's full name
Email	Text	Email address
Phone	Text	Phone number (+91 format)
City	Text	Indian city
State	Text	Indian state
DateOfBirth	Date	Date of birth
KYCStatus	Text	Verified, Pending, Incomplete
RiskCategory	Text	High, Medium, Low
CustomerSince	Date	Customer tenure start date

Contoso_Policies.xlsx (150 rows)

Column	Type	Description
PolicyID	Text	Unique policy identifier (POL-0001 to POL-0150)
CustomerID	Text	Reference to the customer
PolicyType	Text	Health, Motor, Life, Home, Business Liability
PremiumAmount	Currency	Annual/periodic premium in INR
CoverageAmount	Currency	Maximum coverage in INR
StartDate	Date	Policy start date
EndDate	Date	Policy end date

Column	Type	Description
Status	Text	Active, Expired, Cancelled, Lapsed
PaymentFrequency	Text	Monthly, Quarterly, Semi-Annual, Annual

Appendix B: Troubleshooting Guide

Issue	Cause	Solution
"Request for information" action not found	Human in the loop connector not available in your environment	Search directly for "Request for information" in the action search. If not found, ask your admin to enable preview features
RFI email not received	Email in spam/junk or wrong email address	Check spam folder. Verify the "Assigned to" field has the correct email address. Email comes from noreply@microsoft.com
Flow stays paused indefinitely	RFI response not submitted yet	Open the email and click Submit. The flow waits until a response is received — there is no timeout
AI prompt returns empty or generic response	ClaimsData input not mapped correctly	Verify ClaimsData uses the fx expression: string(body('List_rows')?['value']). Do NOT use the dynamic content picker for this field
BadGateway error on List rows	OData filter on a Choice column	Remove any filter from the List rows action. Leave the Filter rows field empty
Prompt doesn't reference adjuster input	RFI output fields not mapped to prompt inputs	Verify all 4 RFI outputs (Damage Assessment, Estimated Repair Cost, etc.) are mapped via Dynamic content picker
Flow fails at prompt action	Prompt input name mismatch	Ensure the / references in the prompt text exactly match the content input names (case-sensitive)
Agent doesn't invoke the flow	Flow not added as a tool	Go to agent → Tools → Add a tool → search for your flow name → add it

Appendix C: Key Concepts

Concept	Explanation
Human-in-the-Loop (HITL)	A workflow pattern where an automated process pauses to collect human input before proceeding. The human's expertise augments the AI's analysis.
Request for Information (RFI)	An Agent Flow action that sends a structured form via Outlook email to a specified person, then pauses the flow until the person submits their response.
Agent Flows	Automated workflows in Copilot Studio that can be triggered by agent conversations, scheduled times, or events. They combine deterministic steps (Dataverse queries) with AI capabilities (prompts).
string(body('List_rows')?['value'])	An fx expression that converts the Dataverse List rows output (an array) into a text string. Required because AI prompt inputs only accept text, not arrays.
Run a Prompt	An AI Builder action that sends instructions + data to a GPT model and returns generated text. Used here to evaluate claim data combined with human input.
Dynamic Content	Values from previous flow actions that can be inserted into subsequent action fields. RFI responses appear as dynamic content after the RFI action.

HITL vs Fully Automated: When to Use Which

Aspect	Fully Automated (Labs 1-3 pattern)	HITL (This Lab)
Human involvement	None — runs end-to-end	Flow pauses for human input
Decision quality	Based on data + AI alone	Data + Human expertise + AI review
Speed	Seconds	Minutes to hours (depends on human)
Use case	Triage, notifications, reports	Complex reviews, approvals, audits
Risk tolerance	Suitable for low-risk decisions	Required for high-value/high-risk decisions
BFSI example	Auto-assign adjuster to new claim	Review ₹10L+ claim with field inspection

--- End of Lab Guide ---