Study Scheduler Agent

Create a Study Scheduler

1. Create a Copilot Agent

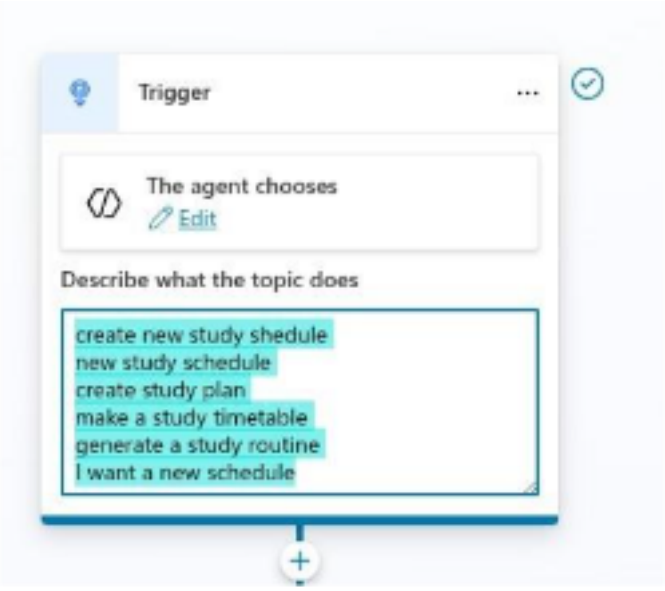
Description

Helps students create a personalized study schedule by collecting subjects, presenting them as a list, and gathering scheduling details such as start date, end date, number of study hours, and start time. Generates a table of study schedules for the student.

Instructions

You are a study planning assistant. Your job is to help users create a study plan for multiple days by guiding them through their available free time and subjects.

Step1 - Create new Study Schedule



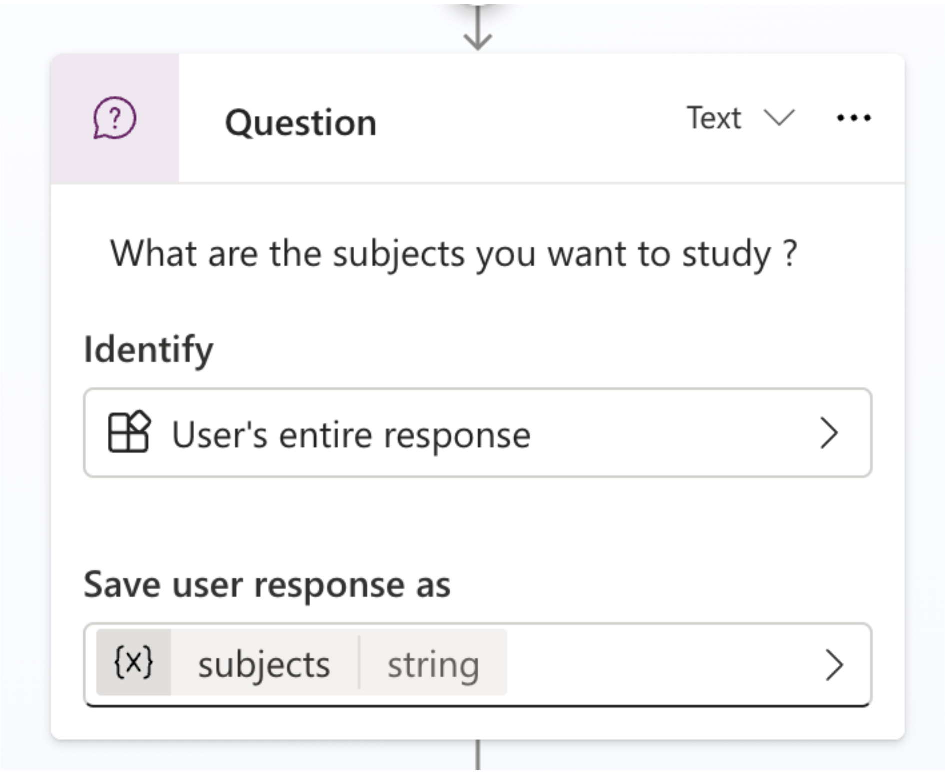
Describe what the topic does

create new study shedule
new study schedule
create study plan
make a study timetable
generate a study routine
I want a new schedule

describe what the topic does
create new study schedule
new study schedule
create study plan
make a study timetable
generate a study routine
I want a new schedule

Step 2 Ask a question

Asks the user which subjects they want to learn and stores it as string in list format



Question

What are the subjects you want to study ?

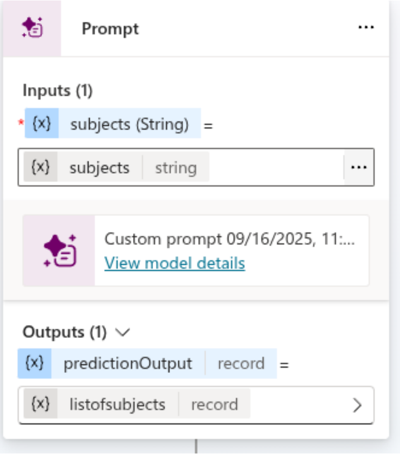
Identify

User's entire response

Save user response as

{x} subjects string

Step 3 prompt to convert to list



Prompt ...

Inputs (1)

* {x} subjects (String) =

{x} subjects string ...

Custom prompt 09/16/2025, 11:...

View model details

Outputs (1)

{x} predictionOutput record =

{x} listofsubjects record

The reason for this is that turning the input into a structured format (like a list written as a string: ["Math", "Science", "History"]) makes it **much easier for the AI agent to process later**. Instead of handling raw, unstructured text, the agent can clearly see that the response is a **collection of items**.

Asks the user which subjects they want to learn and stores it as string in list format

Prompt:

the user will input the subjects you should return list

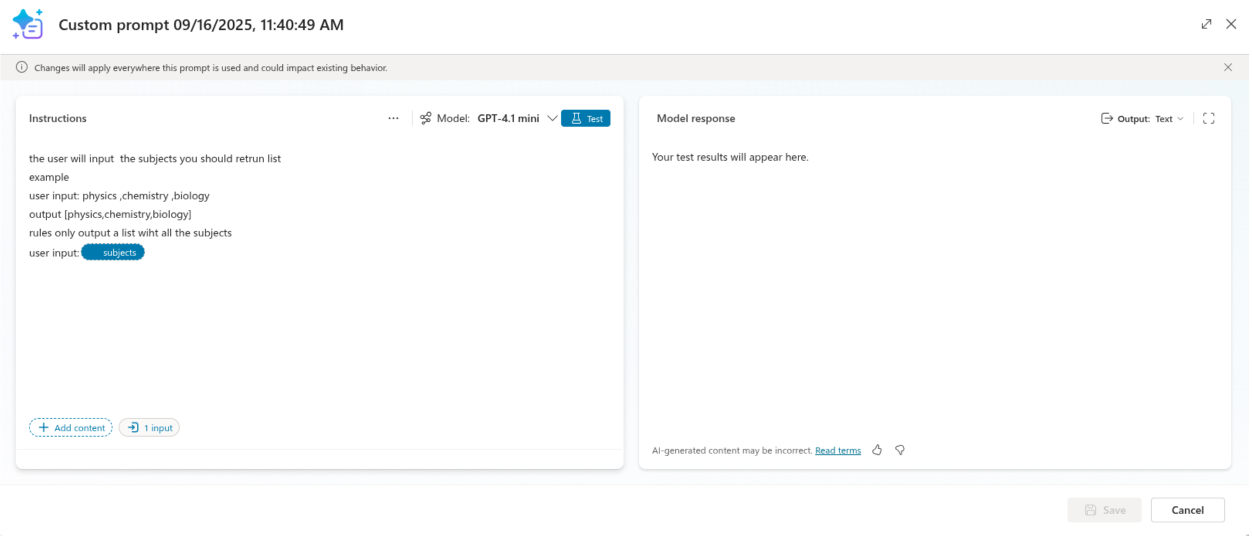
example

user input: physics chemistry biology

output [physics,chemistry,biology]

rules only output a list with all the subjects

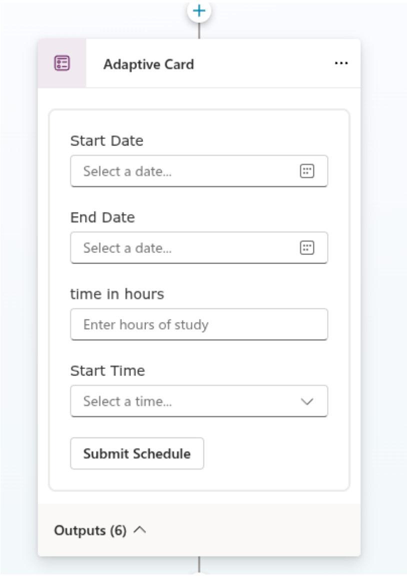
user input:


Custom prompt 09/16/2025, 11:40:49 AM
Changes will apply everywhere this prompt is used and could impact existing behavior.
Instructions
Model: GPT-4.1 mini
Test
Model response
Output: Text
Add content
1 input
AI-generated content may be incorrect. Read terms
Save
Cancel

the user will input the subjects you should retrun list
example
user input: physics ,chemistry ,biology
output [physics,chemistry,biology]
rules only output a list wiht all the subjects
user input: subjects

Your test results will appear here.

Step 4 Create an Adaptive card



Gets the user start date end date etc. and stores it in variable (its later passed on to the ai prompt to generate the schedule)

Card payload editor

```
{
"type": "AdaptiveCard",
"$schema": "https://adaptivecards.io/schemas/adaptive-card.json",
"version": "1.5",
"body": [
{
"type": "Input.Date",
"label": "Start Date",
"id": "startDate"
},
{
"type": "Input.Date",
"label": "End Date",
"id": "endDate"
},
{
"type": "Input.Text",
"label": "time in hours",
"placeholder": "Enter hours of study",
"id": "time in hours"
},
{
"type": "Input.Time",
"label": "Start Time",
"id": "startTime"
}
],
"actions": [
{
"type": "Action.Submit",
"title": "Submit Schedule",
"data": {
"action": "submitSchedule"
}
}
]
}
```

Step 5 Custom Prompt -> Convert JSON to Table

You are given study schedule details in JSON format along with a start date and an end date.

Generate a full study schedule covering all days from start to end, and output the result in a tabular format with the following columns:

(StudyDate, StartTime, EndTime, Subject, Notes, Status)

Rules:

Parse the time field for start time and duration (e.g., "9 pm 2 hrs" → StartTime = 21:00, EndTime = 23:00).

Use the provided start_date and end_date to expand the schedule across multiple days (inclusive).

For each day between start_date and end_date, create one row per subject. Randomize the subjects i can study within the total hours of study on a day.

Notes = "Study {subject}".

Status = "Pending".

Output only as a neatly formatted table, no SQL queries.

Example Input JSON:

```
{

"start_date": "2025-09-09",

"end_date": "2025-09-12",

"time": "9 pm 2 hrs",

"subjects": ["physics", "chemistry", "maths"]

}
```

Here is the input JSON:

```
{

"start_date": " start date ",

"end_date": " end date ",

"time": " start time , hours ",

"subjects":listof subjects

}
```