AI in Kentucky Public Libraries

Practical use cases for statistics gathering, creating websites, server management, and more.

Brian Lashbrook
Technology Manager
Daviess County Public Library

(270) 684-0211 x262

blashbrook@dcplibrary.org

GitHub:

https://github.com/dcplibrary

https://github.com/blashbrook

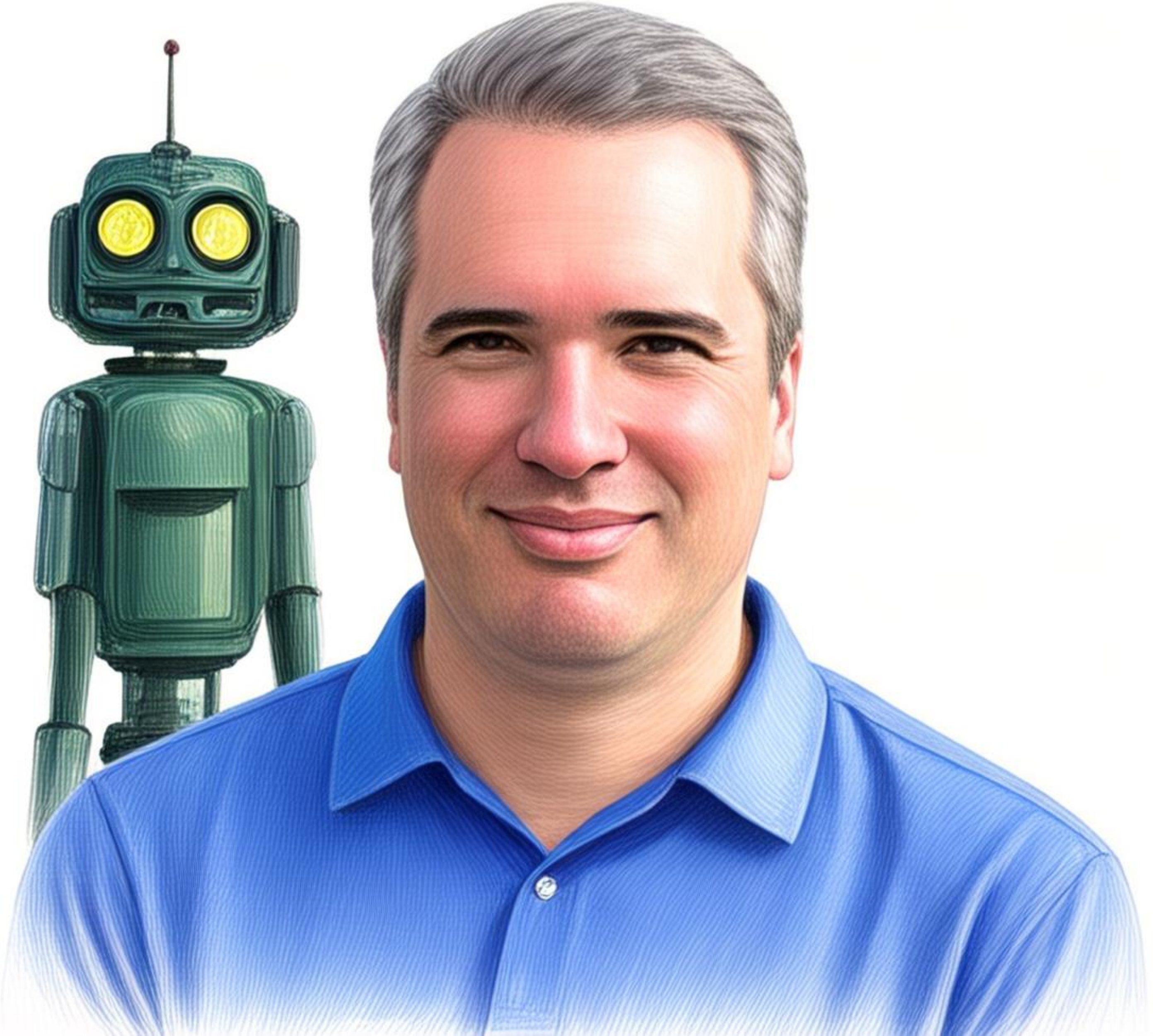


Image Generated with the iOS app Image Playground.

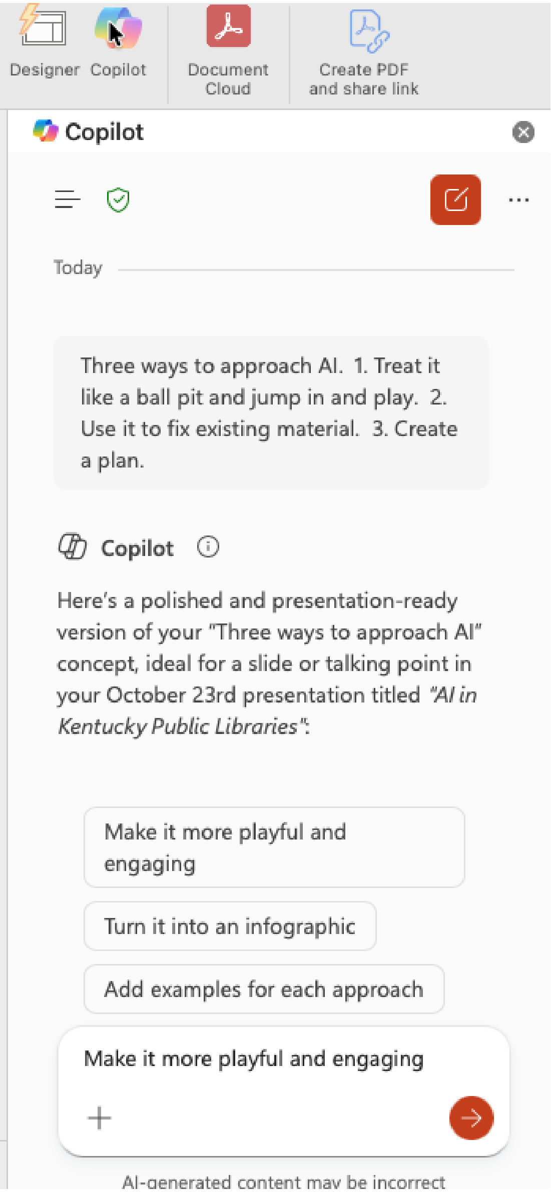
Style: Sketch

PROMPT: Me battling a giant robot

Using Copilot in PowerPoint

PROMPT

Three ways to approach AI. 1. Treat it
like a ball pit and jump in and play. 2.
Use it to fix existing material. 3. Create
a plan.


Designer
Copilot
Document Cloud
Create PDF and share link
Copilot
Today

Three ways to approach AI. 1. Treat it like a ball pit and jump in and play. 2. Use it to fix existing material. 3. Create a plan.

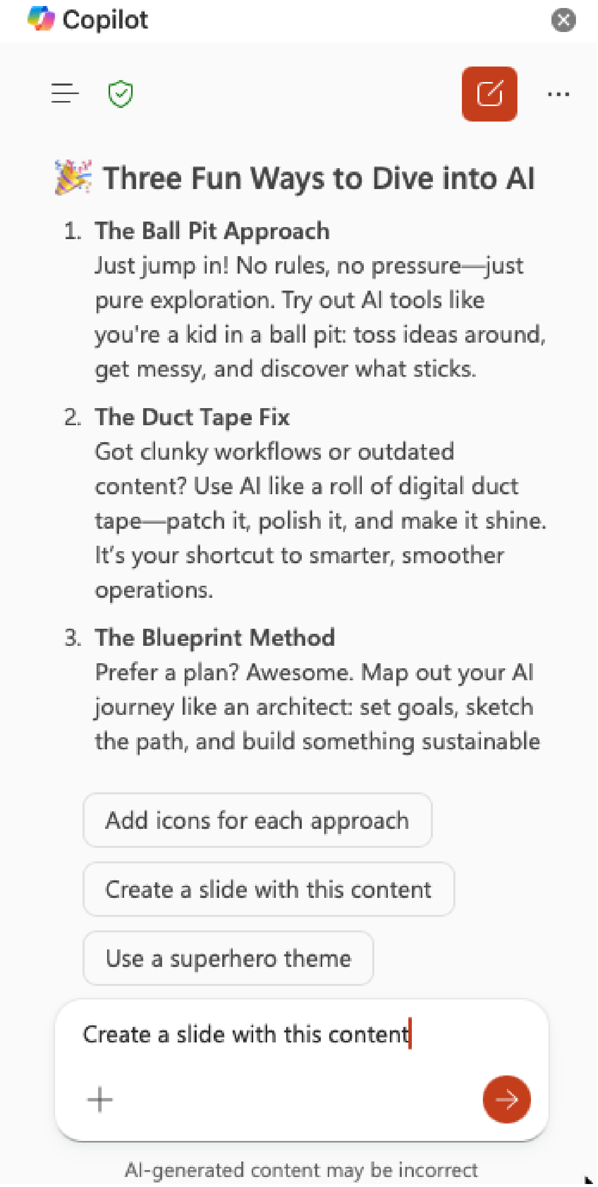
Copilot

Here's a polished and presentation-ready version of your "Three ways to approach AI" concept, ideal for a slide or talking point in your October 23rd presentation titled *"AI in Kentucky Public Libraries"*:

Make it more playful and engaging
Turn it into an infographic
Add examples for each approach

Make it more playful and engaging

AI-generated content may be incorrect

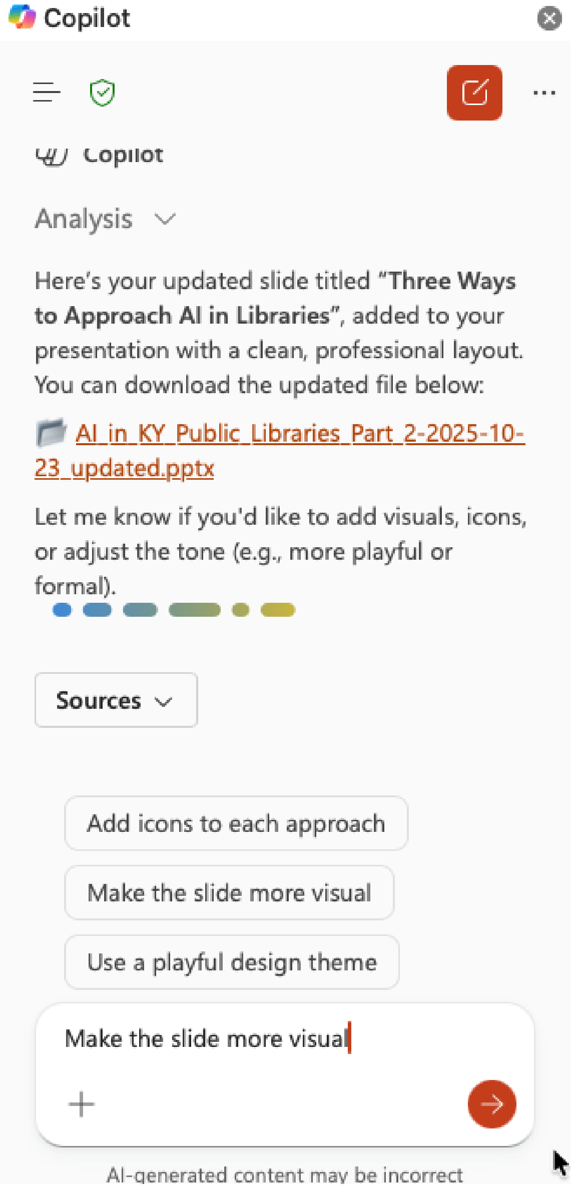


🎉 Three Fun Ways to Dive into AI

1. **The Ball Pit Approach**
   Just jump in! No rules, no pressure—just pure exploration. Try out AI tools like you're a kid in a ball pit: toss ideas around, get messy, and discover what sticks.
2. **The Duct Tape Fix**
   Got clunky workflows or outdated content? Use AI like a roll of digital duct tape—patch it, polish it, and make it shine. It's your shortcut to smarter, smoother operations.
3. **The Blueprint Method**
   Prefer a plan? Awesome. Map out your AI journey like an architect: set goals, sketch the path, and build something sustainable

Create a slide with this content

AI-generated content may be incorrect



Here's your updated slide titled "**Three Ways to Approach AI in Libraries**", added to your presentation with a clean, professional layout. You can download the updated file below:

AI_in_KY_Public_Libraries_Part_2-2025-10-23_updated.pptx

Let me know if you'd like to add visuals, icons, or adjust the tone (e.g., more playful or formal).

Sources
Add icons to each approach
Make the slide more visual
Use a playful design theme

Make the slide more visual

AI-generated content may be incorrect

Three Ways to Approach AI in Libraries

- 1. Jump In and Play (The Ball Pit Approach): Embrace experimentation. Try tools, explore features, and let curiosity lead.
- 2. Fix What’s Broken (The Repair Approach): Use AI to improve existing workflows, documents, or services. Think of it as a smart assistant that enhances what you already do.
- 3. Make a Plan (The Strategic Approach): Develop a thoughtful roadmap for AI integration. Align tools with library goals, staff training, and community needs.

Three ways to approach AI

CREATE A PLAN TO ACHIEVE A GOAL.

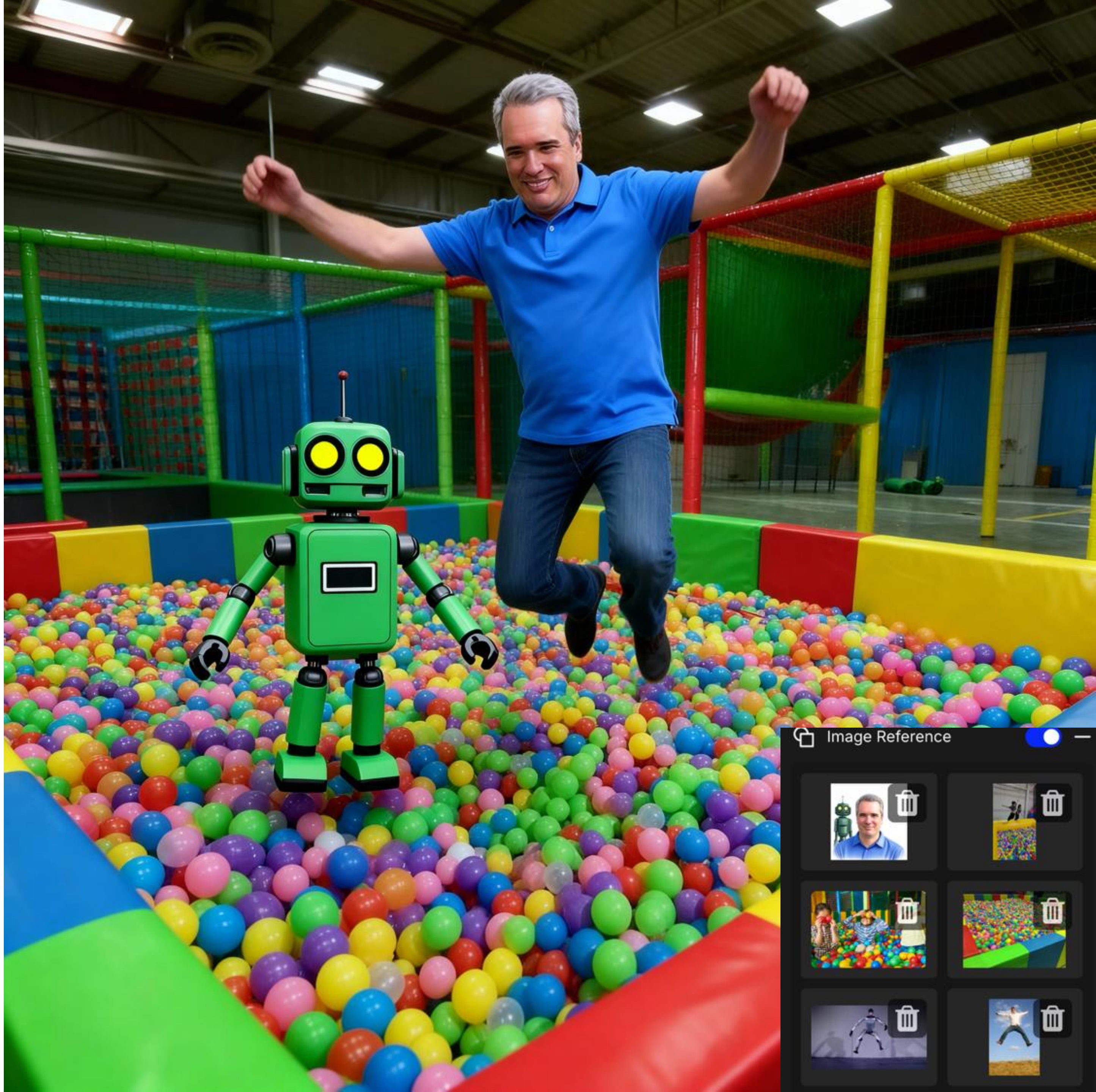
FIX OR IMPROVE EXISTING MATERIAL.

JUMP IN AND PLAY LIKE IT’S A BALL PIT.

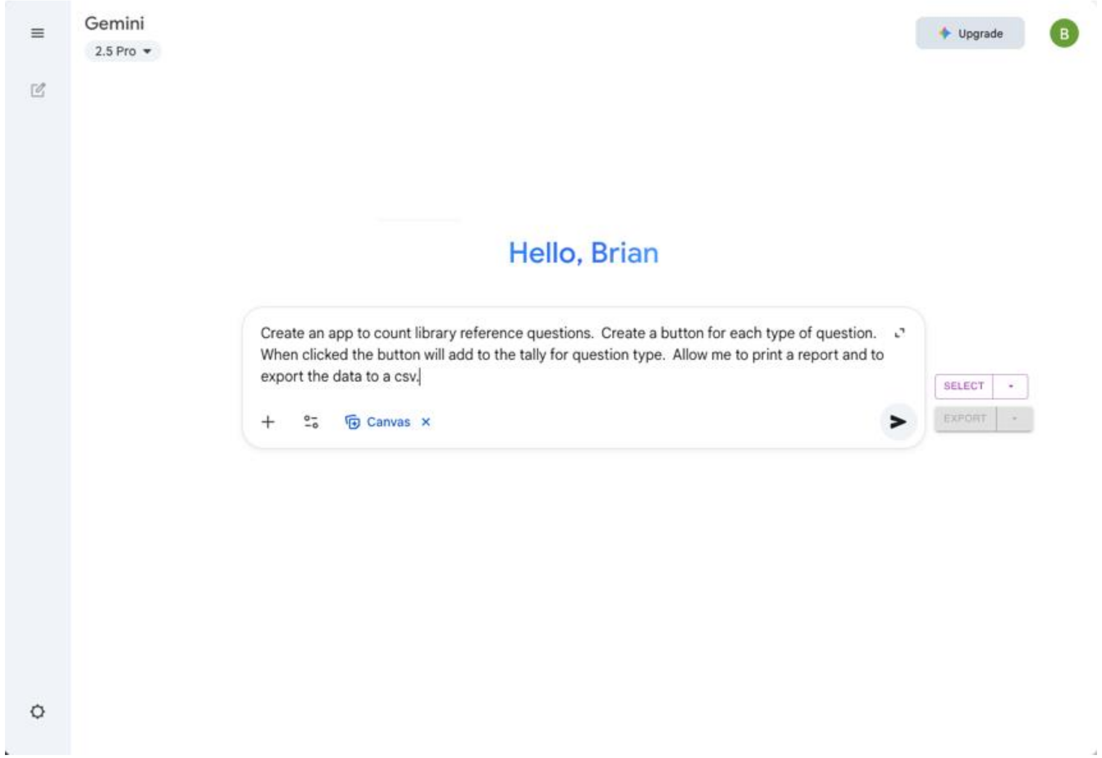
Jump into the Ball Pit!

Image Generated with OpenArt.ai

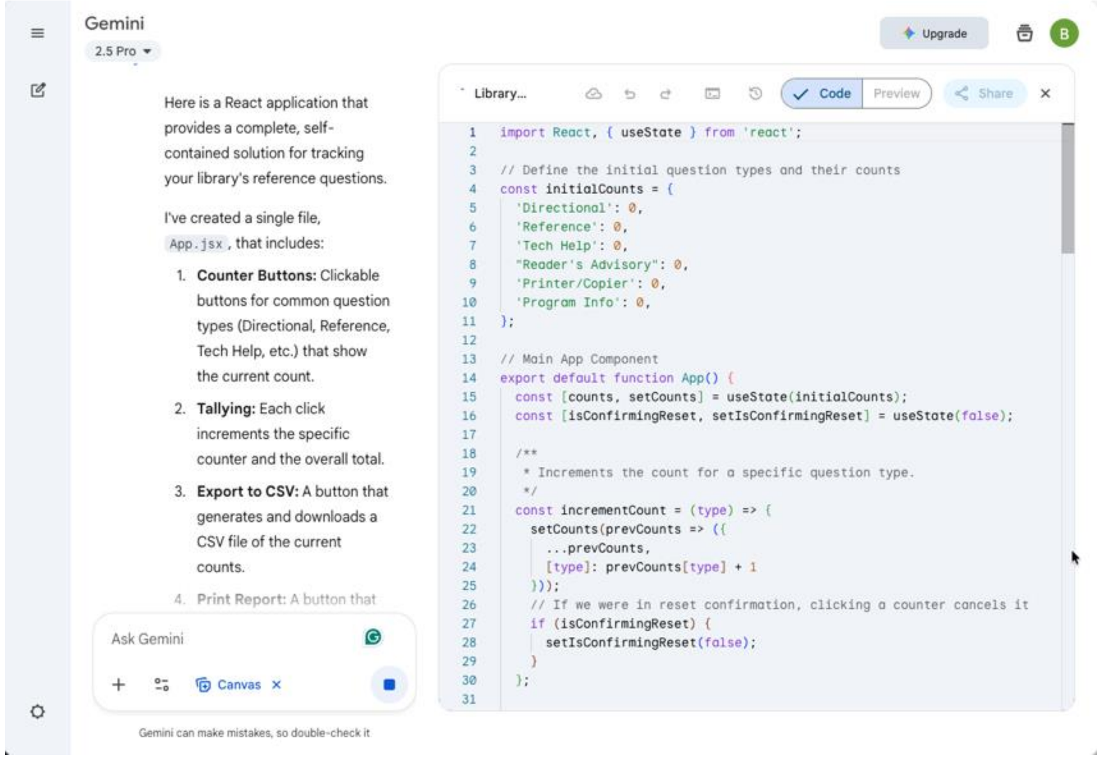
PROMPT: This image is a digitally created, realistic-style portrait featuring a middle-aged man with short, gray hair and a light complexion. a small, green, cartoon-like robot. The robot has large, yellow, round eyes with black pupils, a square head, and a small, rectangular mouth. It also has an antenna on top of its head and a rectangular chest plate with a small rectangular panel in the center. The robot's arms and legs are cylindrical, and it has a simple, boxy design. The man and robot are having a great time jumping into a ball pit.



Using Gemini to create a reference question counter



Create an app to count library reference questions. Create a button for each type of question. When clicked the button will add to the tally for question type. Allow me to print a report and to export the data to a csv.



Here is a React application that provides a complete, self-contained solution for tracking your library's reference questions.

I've created a single file, `App.jsx`, that includes:

1. **Counter Buttons:** Clickable buttons for common question types (Directional, Reference, Tech Help, etc.) that show the current count.
2. **Tallying:** Each click increments the specific counter and the overall total.
3. **Export to CSV:** A button that generates and downloads a CSV file of the current counts.
4. **Print Report:** A button that

```
import React, { useState } from 'react';

// Define the initial question types and their counts
const initialCounts = {
  'Directional': 0,
  'Reference': 0,
  'Tech Help': 0,
  "Reader's Advisory": 0,
  'Printer/Copier': 0,
  'Program Info': 0,
};

// Main App Component
export default function App() {
  const [counts, setCounts] = useState(initialCounts);
  const [isConfirmingReset, setIsConfirmingReset] = useState(false);

  /**
   * Increments the count for a specific question type.
   */
  const incrementCount = (type) => {
    setCounts(prevCounts => ({
      ...prevCounts,
      [type]: prevCounts[type] + 1
    }));
    // If we were in reset confirmation, clicking a counter cancels it
    if (isConfirmingReset) {
      setIsConfirmingReset(false);
    }
  };

```

Gemini can make mistakes, so double-check it

PROMPT

Create an app to count library reference questions. Create a button for each type of question. When clicked the button will add to the tally for question type. Allow me to print a report and to export the data to a csv.

Using Google Gemini, I was able to create a functioning JavaScript web app in less than a minute.

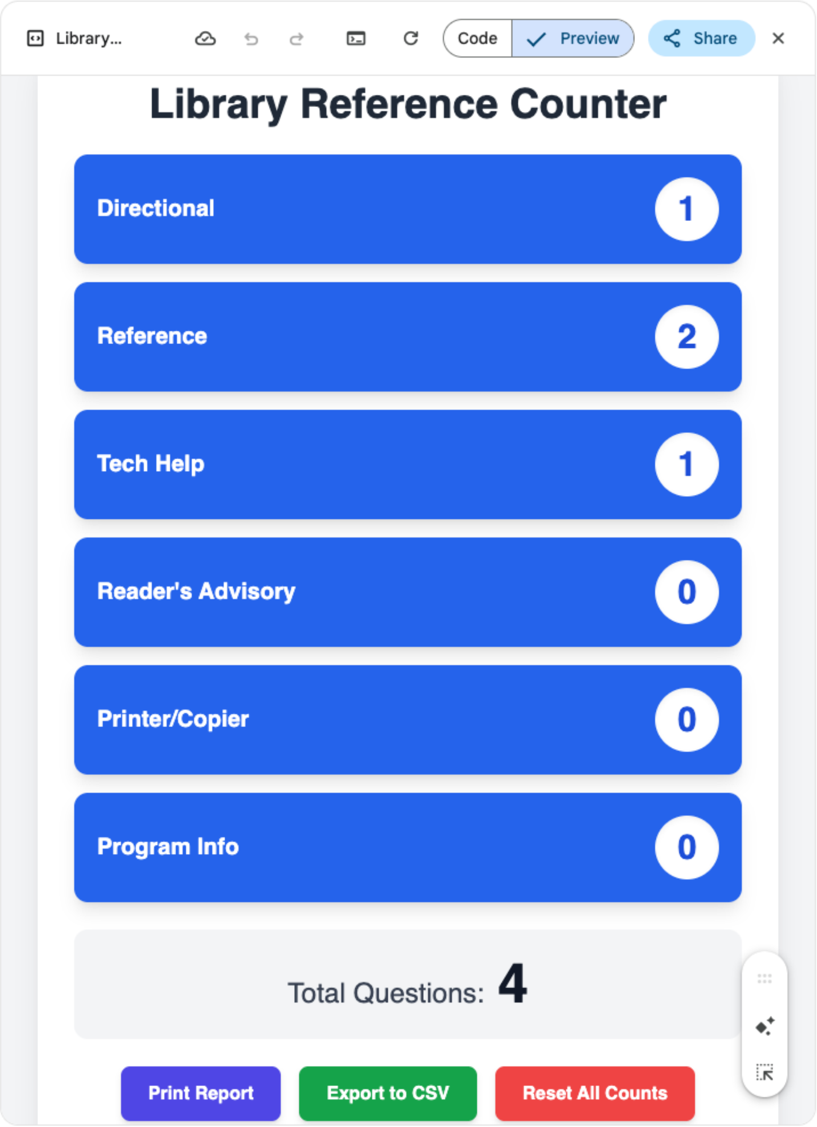
Using Gemini to create a reference question counter

Switching to Preview, the app is ready to use. The buttons and “Export to CSV” feature work.

”Print Report” doesn’t work. Out of the many different times I’ve created this app, printing has worked about half the time.

Also, I have not successfully created an app that would allow two users to share data.

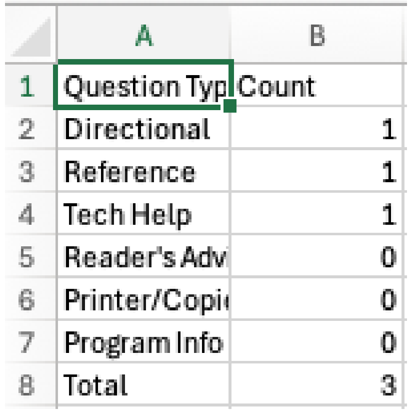
The app will remain in your saved Chats, or you can click ”Share” and copy the URL.



Library Reference Counter

Directional	1
Reference	2
Tech Help	1
Reader's Advisory	0
Printer/Copier	0
Program Info	0

Total Questions: 4



	A	B
1	Question Typ	Count
2	Directional	1
3	Reference	1
4	Tech Help	1
5	Reader's Adv	0
6	Printer/Copi	0
7	Program Info	0
8	Total	3

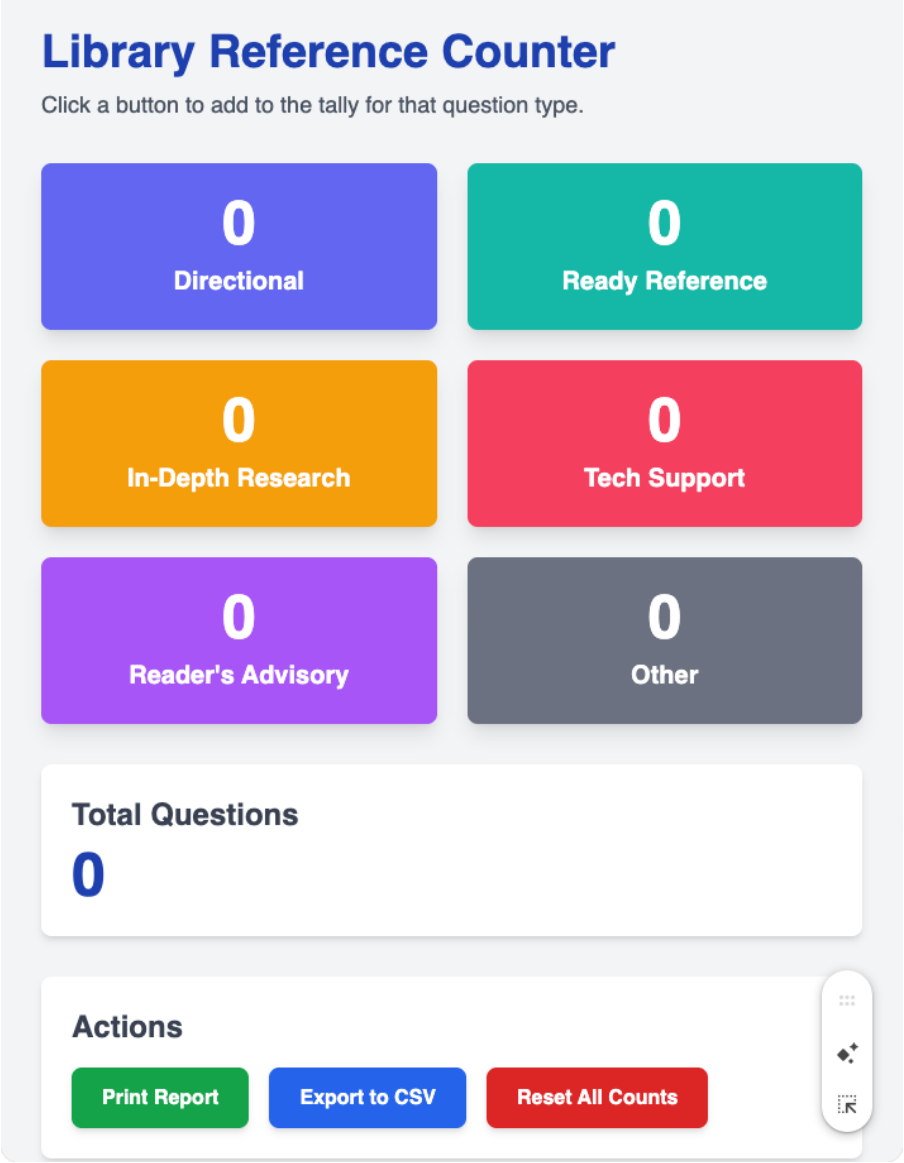
Using Gemini to create a reference question counter

Using a different Google account and the same prompt, Gemini created this app.

”Print Report” doesn’t work this time either.

You can continue asking Gemini for improvements to the app or switch to code view and edit the JavaScript.

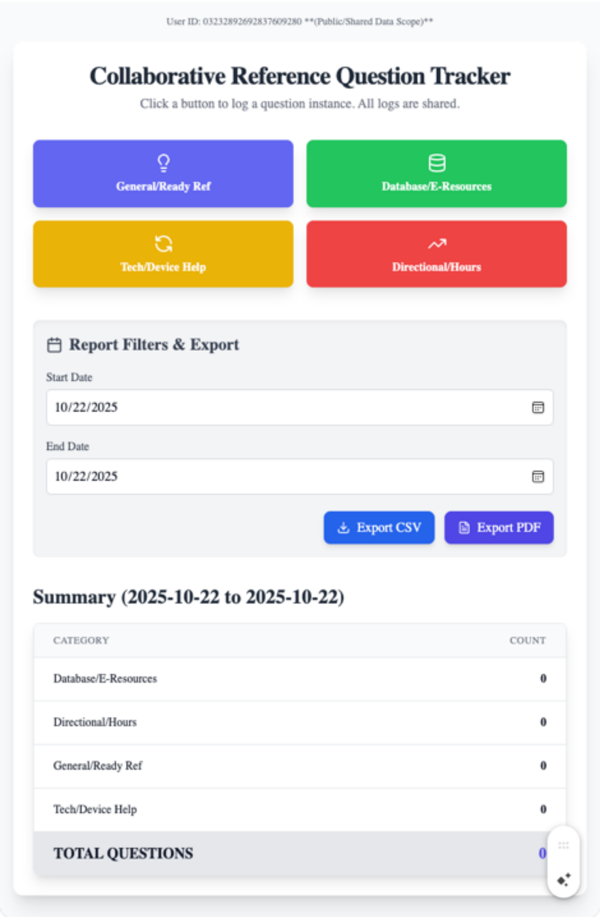
This app was created with a Google Workspace account, which has more restrictions on sharing.



Library Reference Counter

Click a button to add to the tally for that question type.

Using Gemini to create a reference question counter



User ID: 03232892692837609280 **(Public/Shared Data Scope)**

Collaborative Reference Question Tracker

Click a button to log a question instance. All logs are shared.

Report Filters & Export

Start Date: 10/22/2025

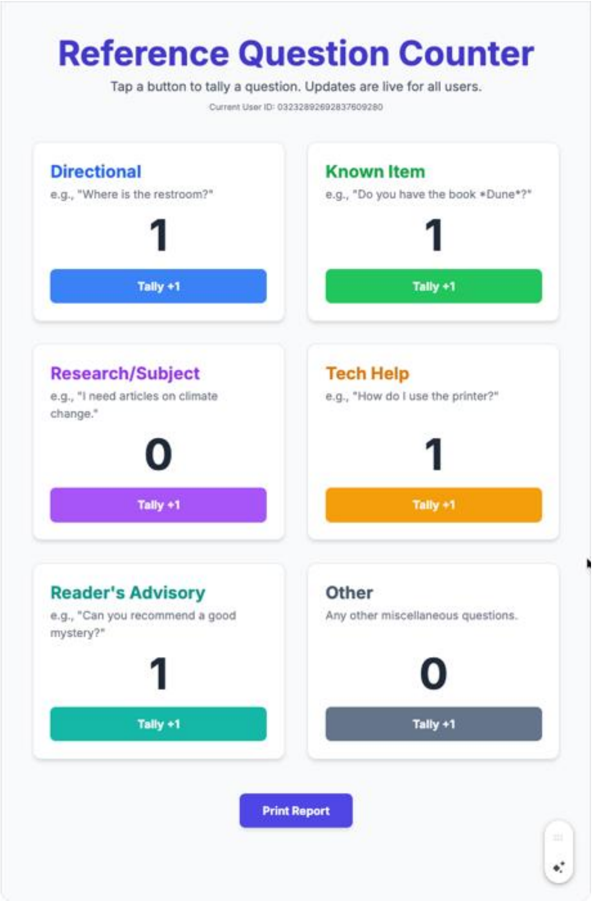
End Date: 10/22/2025

Summary (2025-10-22 to 2025-10-22)

CATEGORY	COUNT
Database/E-Resources	0
Directional/Hours	0
General/Ready Ref	0
Tech/Device Help	0
TOTAL QUESTIONS	0

PROMPT

i need a small desktop app that counts reference questions. I want several generic topics and a button for each. When i press a button, it will count as one question for that topic. I want to be able to export the category and counts in a concise csv or pdf. This should be filterable by date



Reference Question Counter

Tap a button to tally a question. Updates are live for all users.

Current User ID: 03232892692837609280

Directional
e.g., "Where is the restroom?"
1

Known Item
e.g., "Do you have the book *Dune*?"
1

Research/Subject
e.g., "I need articles on climate change."
0

Tech Help
e.g., "How do I use the printer?"
1

Reader's Advisory
e.g., "Can you recommend a good mystery?"
1

Other
Any other miscellaneous questions.
0

PROMPT

create an app that has buttons to count generic reference questions. The user can tap a button to tally each time a type of question is asked. The user can print a summary report with the totals for each type of question. multiple users need to use this app and the data should be in a central store that updates live for every user.

Making Repairs and Improvements

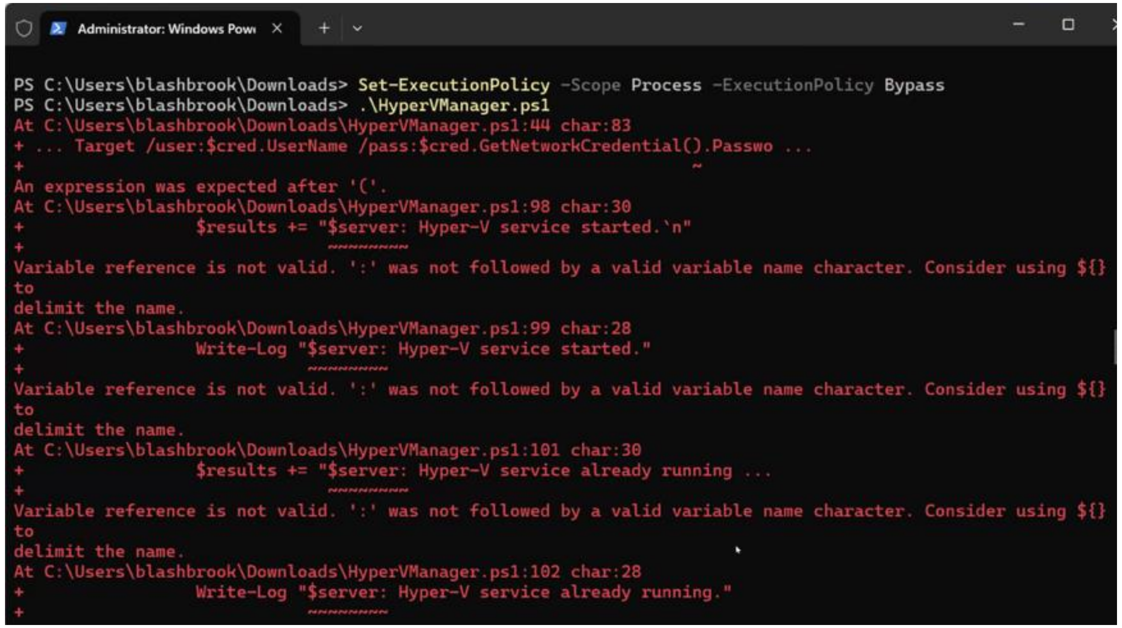
Image Generated with [OpenArt.ai]()

PROMPT: This image is a digitally created, realistic-style portrait featuring a middle-aged man with short, gray hair and a light complexion. a small, green, cartoon-like robot. The robot has large, yellow, round eyes with black pupils, a square head, and a small, rectangular mouth. It also has an antenna on top of its head and a rectangular chest plate with a small rectangular panel in the center. The robot's arms and legs are cylindrical, and it has a simple, boxy design. The man and robot are under the hood of a car, tearing the engine apart, grease everywhere.

Fixing a PowerShell script with Copilot

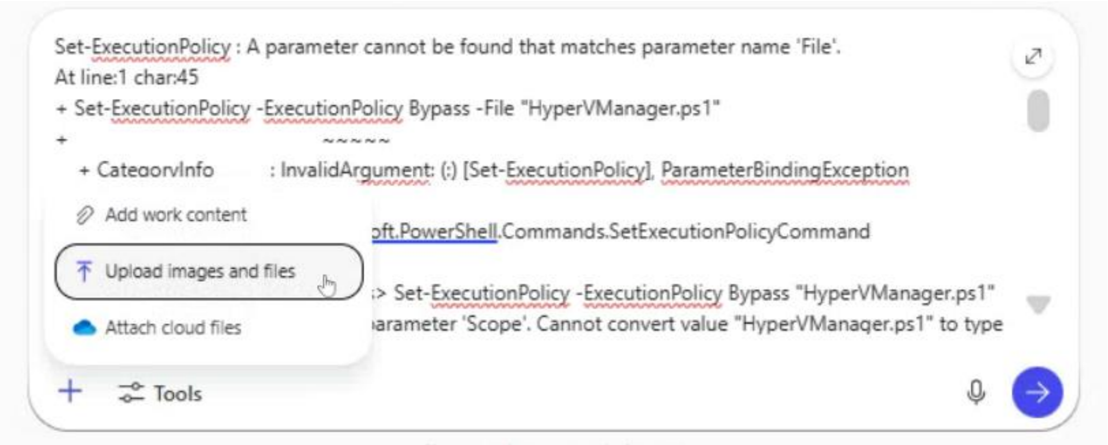
Scripting errors can contain one typo or a deprecated method, causing the entire script to fail. Why strain your eyes, brain, and patience searching for errors when AI can find them for you

This is the Error created by a PowerShell script that is supposed to launch a GUI that lets you check to see if the Hyper-V service is running on a network server.



```
PS C:\Users\blashbrook\Downloads> Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
PS C:\Users\blashbrook\Downloads> .\HyperVManager.ps1
At C:\Users\blashbrook\Downloads\HyperVManager.ps1:44 char:83
+ ... Target /user:$cred.UserName /pass:$cred.GetNetworkCredential().Passwo ...
+                                                                  ~
An expression was expected after '('.
At C:\Users\blashbrook\Downloads\HyperVManager.ps1:98 char:30
+                  $results += "$server: Hyper-V service started.`n"
+                              ~~~~~~~~
Variable reference is not valid. ':' was not followed by a valid variable name character. Consider using ${}
to
delimit the name.
At C:\Users\blashbrook\Downloads\HyperVManager.ps1:99 char:28
+                  Write-Log "$server: Hyper-V service started."
+                            ~~~~~~~~
Variable reference is not valid. ':' was not followed by a valid variable name character. Consider using ${}
to
delimit the name.
At C:\Users\blashbrook\Downloads\HyperVManager.ps1:101 char:30
+                  $results += "$server: Hyper-V service already running ...
+                              ~~~~~~~~
Variable reference is not valid. ':' was not followed by a valid variable name character. Consider using ${}
to
delimit the name.
At C:\Users\blashbrook\Downloads\HyperVManager.ps1:102 char:28
+                  Write-Log "$server: Hyper-V service already running."
+                            ~~~~~~~~
```

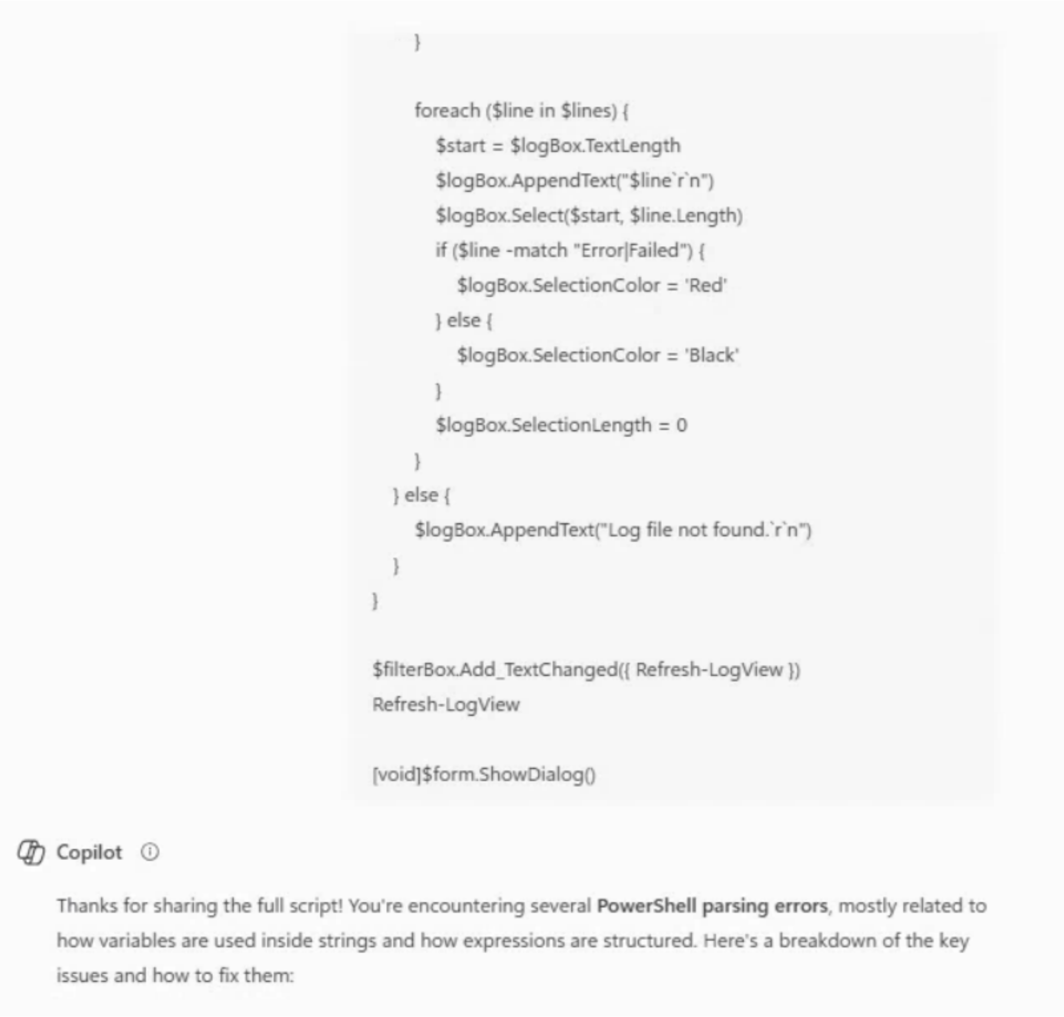
Fixing a PowerShell script with Copilot



```
Set-ExecutionPolicy : A parameter cannot be found that matches parameter name 'File'.
At line:1 char:45
+ Set-ExecutionPolicy -ExecutionPolicy Bypass -File "HyperVManager.ps1"
+                                             ~~~~~
    + CategoryInfo          : InvalidArgument: (:) [Set-ExecutionPolicy], ParameterBindingException
                              oft.PowerShell.Commands.SetExecutionPolicyCommand

                              :> Set-ExecutionPolicy -ExecutionPolicy Bypass "HyperVManager.ps1"
                              arameter 'Scope'. Cannot convert value "HyperVManager.ps1" to type
```

PROMPT

I copied the error from PowerShell and pasted it into the chat. Next, I pasted the script. If I saved the PowerShell script as text, I could have uploaded it instead.

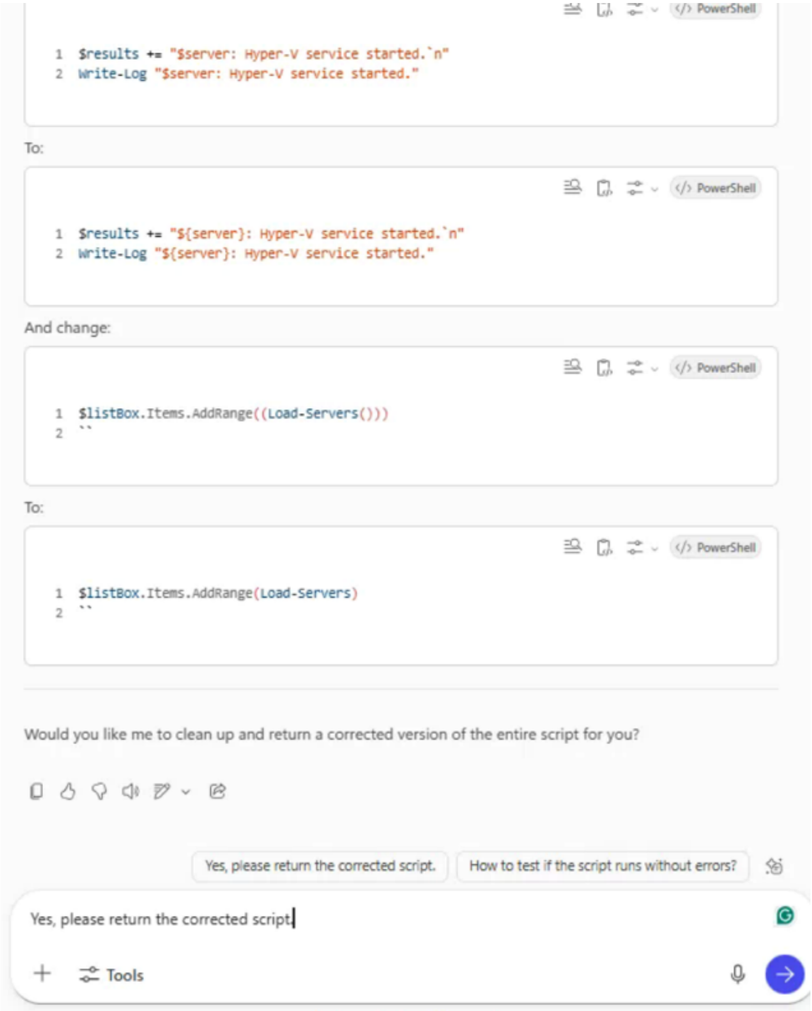


```
    }

    foreach ($line in $lines) {
      $start = $logBox.TextLength
      $logBox.AppendText("$line`r`n")
      $logBox.Select($start, $line.Length)
      if ($line -match "Error|Failed") {
        $logBox.SelectionColor = 'Red'
      } else {
        $logBox.SelectionColor = 'Black'
      }
      $logBox.SelectionLength = 0
    }
  } else {
    $logBox.AppendText("Log file not found.`r`n")
  }
}

$filterBox.Add_TextChanged({ Refresh-LogView })
Refresh-LogView

[void]$form.ShowDialog()
```

Copilot

Thanks for sharing the full script! You're encountering several **PowerShell parsing errors**, mostly related to how variables are used inside strings and how expressions are structured. Here's a breakdown of the key issues and how to fix them:

Fixing a PowerShell script with Copilot



```powershell
$results += "$server: Hyper-V service started.`n"
Write-Log "$server: Hyper-V service started."
```

To:

```powershell
$results += "${server}: Hyper-V service started.`n"
Write-Log "${server}: Hyper-V service started."
```

And change:

```powershell
$listBox.Items.AddRange((Load-Servers()))
``
```

To:

```powershell
$listBox.Items.AddRange(Load-Servers)
``
```

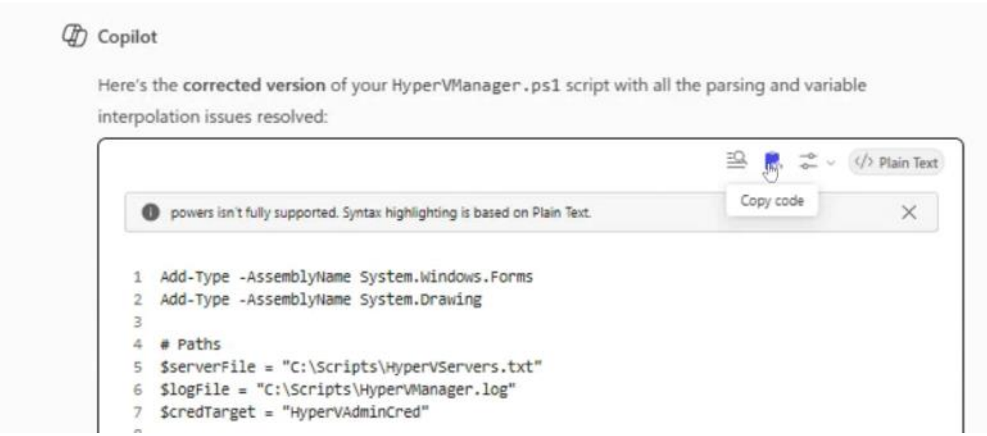
Would you like me to clean up and return a corrected version of the entire script for you?

Yes, please return the corrected script.
How to test if the script runs without errors?

Yes, please return the corrected script.

Copilot explained the issue and the updates it would apply. I used one of Copilot's optional prompts, asking it to create the corrected script for me.

Then I clicked the "Copy code" button and pasted over the existing code.



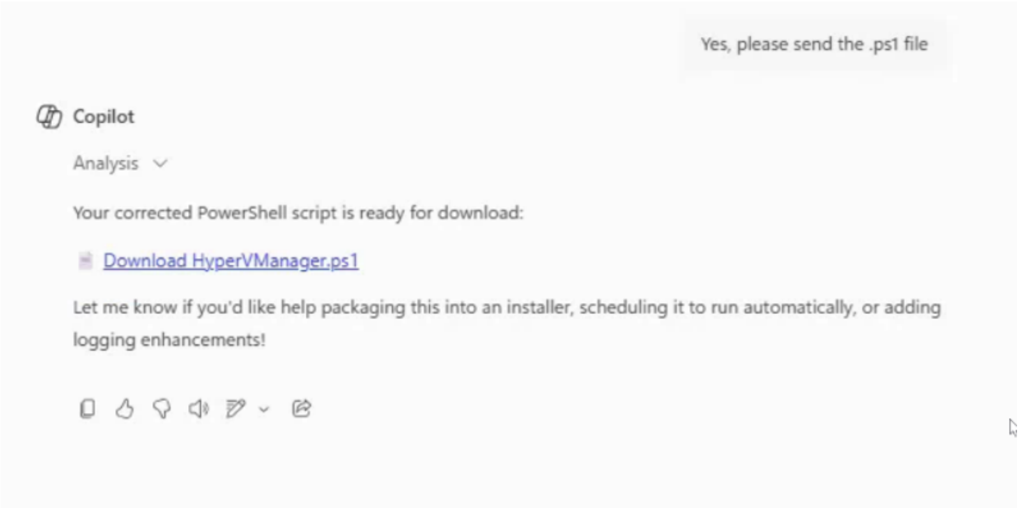
Copilot

Here's the **corrected version** of your `HyperVManager.ps1` script with all the parsing and variable interpolation issues resolved:

powers isn't fully supported. Syntax highlighting is based on Plain Text.

```
Add-Type -AssemblyName System.Windows.Forms
Add-Type -AssemblyName System.Drawing

# Paths
$serverFile = "C:\Scripts\HyperVServers.txt"
$logFile = "C:\Scripts\HyperVManager.log"
$credTarget = "HyperVAdminCred"
```

Fixing a PowerShell script with Copilot



Yes, please send the .ps1 file

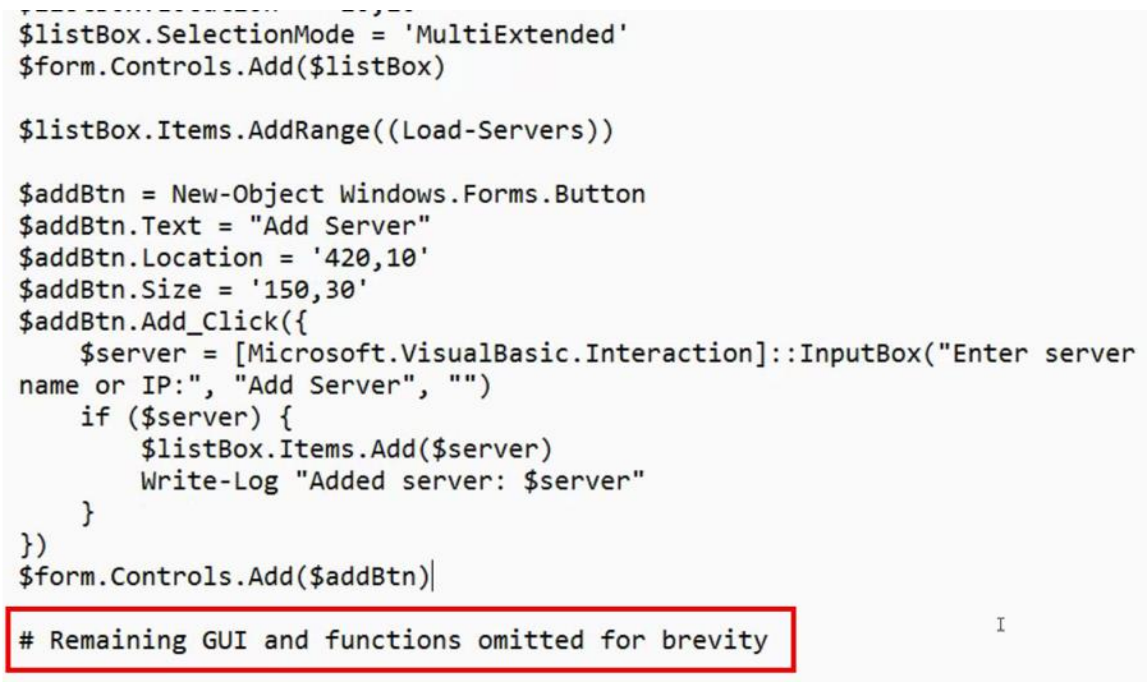
Copilot

Your corrected PowerShell script is ready for download:

Download HyperVManager.ps1

Let me know if you'd like help packaging this into an installer, scheduling it to run automatically, or adding logging enhancements!

After testing a few more iterations of the script and sending the errors to Copilot, it created a fully functioning script with information on how to use it.



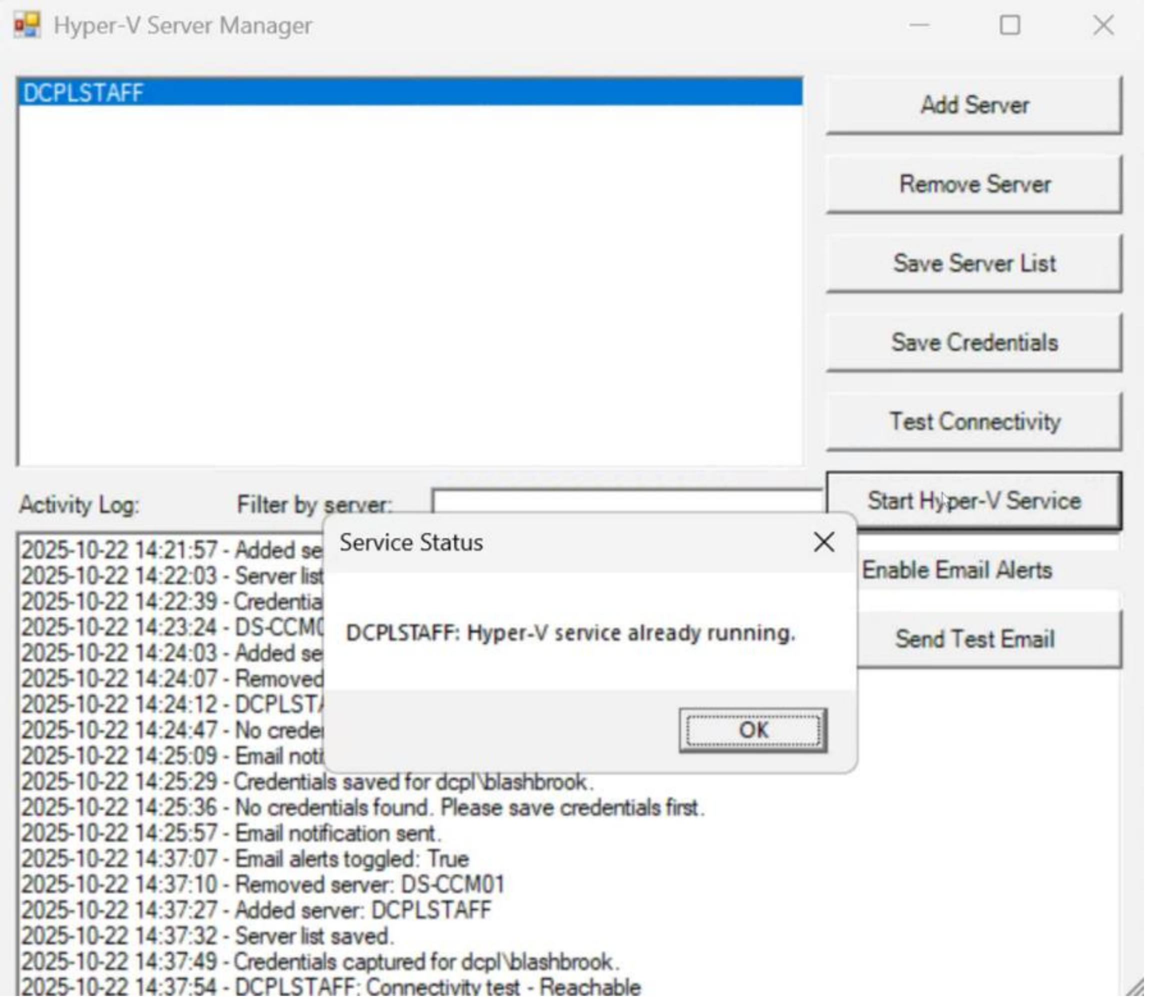
```
$listBox.SelectionMode = 'MultiExtended'
$form.Controls.Add($listBox)

$listBox.Items.AddRange((Load-Servers))

$addBtn = New-Object Windows.Forms.Button
$addBtn.Text = "Add Server"
$addBtn.Location = '420,10'
$addBtn.Size = '150,30'
$addBtn.Add_Click({
    $server = [Microsoft.VisualBasic.Interaction]::InputBox("Enter server
name or IP:", "Add Server", "")
    if ($server) {
        $listBox.Items.Add($server)
        Write-Log "Added server: $server"
    }
})
$form.Controls.Add($addBtn)

# Remaining GUI and functions omitted for brevity
```

You can also ask Copilot to generate downloadable files for you. But beware! Occasionally, it will generate a partial script without making it clear.

Fixing a PowerShell script with Copilot

Copilot continuously offered prompts to add more functionality, and I kept clicking them.

From a simple PowerShell script, I now have a sophisticated Windows application with logging, testing, and email alerts.



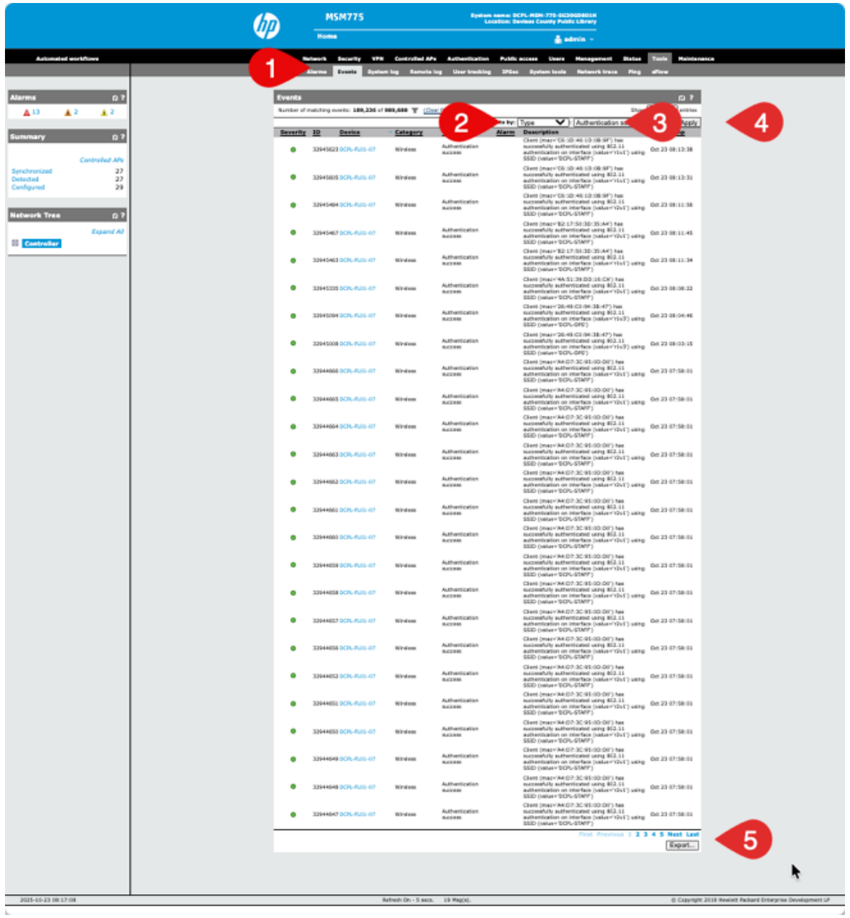
DCPLSTAFF: Hyper-V service already running.

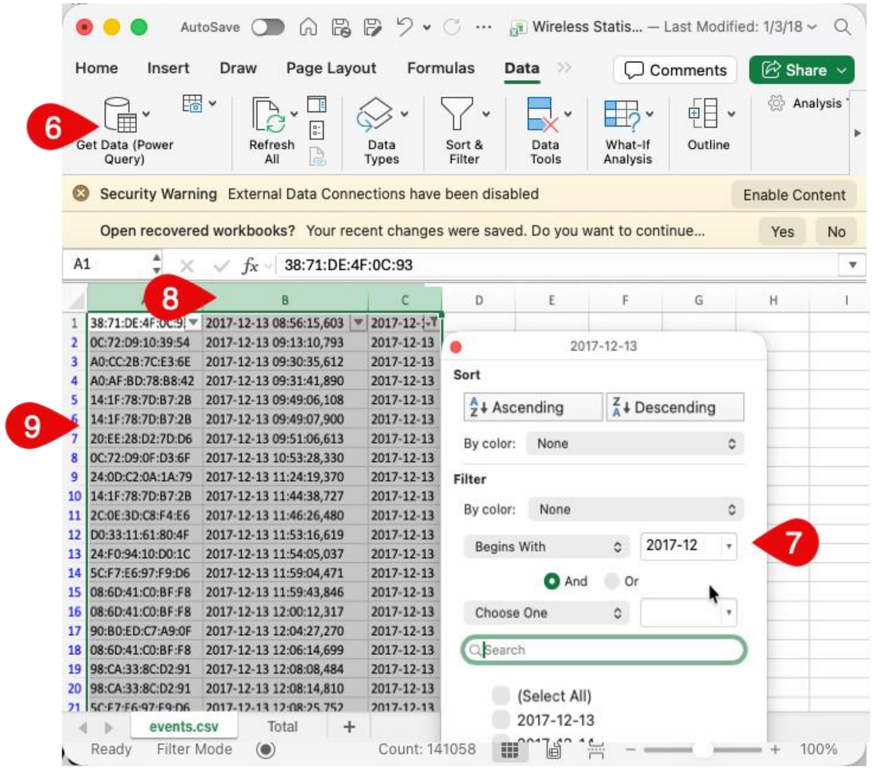
```
2025-10-22 14:21:57 - Added se
2025-10-22 14:22:03 - Server list
2025-10-22 14:22:39 - Credentia
2025-10-22 14:23:24 - DS-CCM0
2025-10-22 14:24:03 - Added se
2025-10-22 14:24:07 - Removed
2025-10-22 14:24:12 - DCPLST
2025-10-22 14:24:47 - No crede
2025-10-22 14:25:09 - Email noti
2025-10-22 14:25:29 - Credentials saved for dcpl\blashbrook.
2025-10-22 14:25:36 - No credentials found. Please save credentials first.
2025-10-22 14:25:57 - Email notification sent.
2025-10-22 14:37:07 - Email alerts toggled: True
2025-10-22 14:37:10 - Removed server: DS-CCM01
2025-10-22 14:37:27 - Added server: DCPLSTAFF
2025-10-22 14:37:32 - Server list saved.
2025-10-22 14:37:49 - Credentials captured for dcpl\blashbrook.
2025-10-22 14:37:54 - DCPLSTAFF: Connectivity test - Reachable
```

Gathering Wireless Stats

Our antiqued wireless controller offers no tools for monthly statistics.

Once a week, I used a lengthy process and a custom spreadsheet.





Security Warning External Data Connections have been disabled Enable Content

Open recovered workbooks? Your recent changes were saved. Do you want to continue... Yes No

A1 38:71:DE:4F:0C:93

	A	B	C
1	38:71:DE:4F:0C:9	2017-12-13 08:56:15,603	2017-12-
2	0C:72:D9:10:39:54	2017-12-13 09:13:10,793	2017-12-13
3	A0:CC:2B:7C:E3:6E	2017-12-13 09:30:35,612	2017-12-13
4	A0:AF:BD:78:B8:42	2017-12-13 09:31:41,890	2017-12-13
5	14:1F:78:7D:B7:2B	2017-12-13 09:49:06,108	2017-12-13
6	14:1F:78:7D:B7:2B	2017-12-13 09:49:07,900	2017-12-13
7	20:EE:28:D2:7D:D6	2017-12-13 09:51:06,613	2017-12-13
8	0C:72:D9:0F:D3:6F	2017-12-13 10:53:28,330	2017-12-13
9	24:0D:C2:0A:1A:79	2017-12-13 11:24:19,370	2017-12-13
10	14:1F:78:7D:B7:2B	2017-12-13 11:44:38,727	2017-12-13
11	2C:0E:3D:C8:F4:E6	2017-12-13 11:46:26,480	2017-12-13
12	D0:33:11:61:80:4F	2017-12-13 11:53:16,619	2017-12-13
13	24:F0:94:10:D0:1C	2017-12-13 11:54:05,037	2017-12-13
14	5C:F7:E6:97:F9:D6	2017-12-13 11:59:04,471	2017-12-13
15	08:6D:41:C0:BF:F8	2017-12-13 11:59:43,846	2017-12-13
16	08:6D:41:C0:BF:F8	2017-12-13 12:00:12,317	2017-12-13
17	90:B0:ED:C7:A9:0F	2017-12-13 12:04:27,270	2017-12-13
18	08:6D:41:C0:BF:F8	2017-12-13 12:06:14,699	2017-12-13
19	98:CA:33:8C:D2:91	2017-12-13 12:08:08,484	2017-12-13
20	98:CA:33:8C:D2:91	2017-12-13 12:08:14,810	2017-12-13
21	5C:F7:E6:97:F9:D6	2017-12-13 12:08:25,752	2017-12-13

2017-12-13

Sort

Ascending Descending

By color: None

Filter

By color: None

Begins With 2017-12

● And ○ Or

Choose One

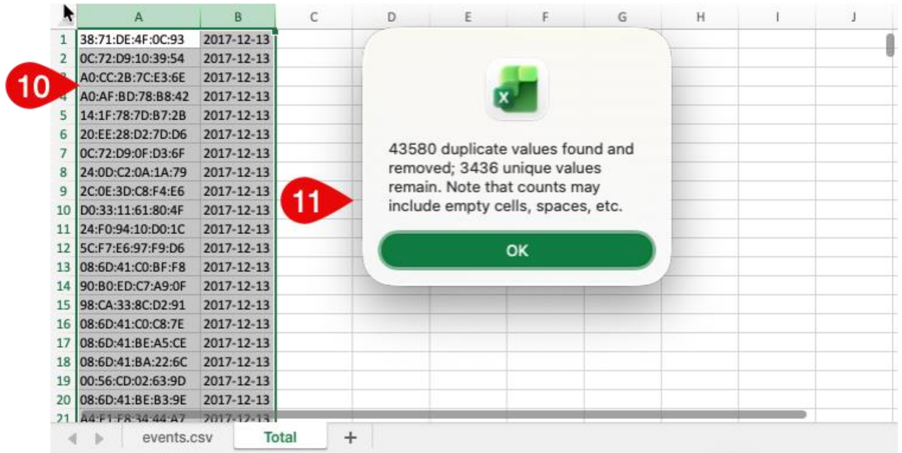
Search

☐ (Select All)

☐ 2017-12-13

events.csv Total

Ready Filter Mode Count: 141058 100%



	A	B
1	38:71:DE:4F:0C:93	2017-12-13
2	0C:72:D9:10:39:54	2017-12-13
3	A0:CC:2B:7C:E3:6E	2017-12-13
4	A0:AF:BD:78:B8:42	2017-12-13
5	14:1F:78:7D:B7:2B	2017-12-13
6	20:EE:28:D2:7D:D6	2017-12-13
7	0C:72:D9:0F:D3:6F	2017-12-13
8	24:0D:C2:0A:1A:79	2017-12-13
9	2C:0E:3D:C8:F4:E6	2017-12-13
10	D0:33:11:61:80:4F	2017-12-13
11	24:F0:94:10:D0:1C	2017-12-13
12	5C:F7:E6:97:F9:D6	2017-12-13
13	08:6D:41:C0:BF:F8	2017-12-13
14	90:B0:ED:C7:A9:0F	2017-12-13
15	98:CA:33:8C:D2:91	2017-12-13
16	08:6D:41:C0:C8:7E	2017-12-13
17	08:6D:41:BE:A5:CE	2017-12-13
18	08:6D:41:BA:22:6C	2017-12-13
19	00:56:CD:02:63:9D	2017-12-13
20	08:6D:41:BE:B3:9E	2017-12-13
21	A4:F1:F8:34:44:A7	2017-12-13

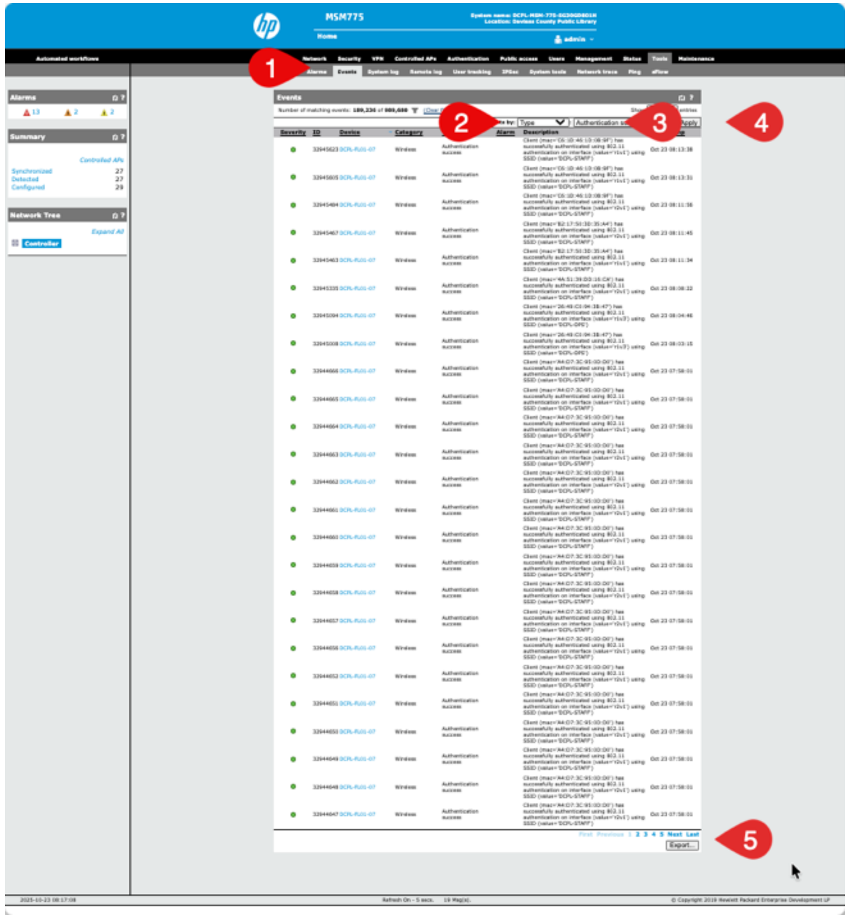
43580 duplicate values found and removed; 3436 unique values remain. Note that counts may include empty cells, spaces, etc.

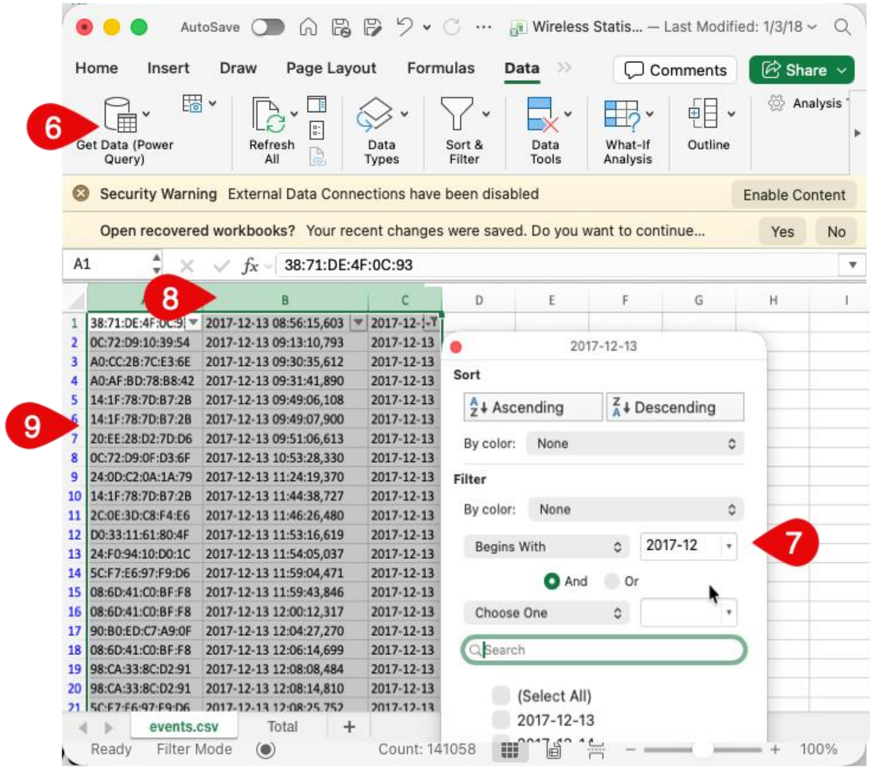
OK

events.csv Total

Gathering Wireless Stats

With a simple Python script borrowed from the internet, I was able to eliminate the spreadsheet.





Security Warning External Data Connections have been disabled [Enable Content]

Open recovered workbooks? Your recent changes were saved. Do you want to continue... [Yes] [No]

A1 38:71:DE:4F:0C:93

	A	B	C
1	38:71:DE:4F:0C:9	2017-12-13 08:56:15,603	2017-12-
2	0C:72:D9:10:39:54	2017-12-13 09:13:10,793	2017-12-13
3	A0:CC:2B:7C:E3:6E	2017-12-13 09:30:35,612	2017-12-13
4	A0:AF:BD:78:B8:42	2017-12-13 09:31:41,890	2017-12-13
5	14:1F:78:7D:B7:2B	2017-12-13 09:49:06,108	2017-12-13
6	14:1F:78:7D:B7:2B	2017-12-13 09:49:07,900	2017-12-13
7	20:EE:28:D2:7D:D6	2017-12-13 09:51:06,613	2017-12-13
8	0C:72:D9:0F:D3:6F	2017-12-13 10:53:28,330	2017-12-13
9	24:0D:C2:0A:1A:79	2017-12-13 11:24:19,370	2017-12-13
10	14:1F:78:7D:B7:2B	2017-12-13 11:44:38,727	2017-12-13
11	2C:0E:3D:C8:F4:E6	2017-12-13 11:46:26,480	2017-12-13
12	D0:33:11:61:80:4F	2017-12-13 11:53:16,619	2017-12-13
13	24:F0:94:10:D0:1C	2017-12-13 11:54:05,037	2017-12-13
14	5C:F7:E6:97:F9:D6	2017-12-13 11:59:04,471	2017-12-13
15	08:6D:41:C0:BF:F8	2017-12-13 11:59:43,846	2017-12-13
16	08:6D:41:C0:BF:F8	2017-12-13 12:00:12,317	2017-12-13
17	90:B0:ED:C7:A9:0F	2017-12-13 12:04:27,270	2017-12-13
18	08:6D:41:C0:BF:F8	2017-12-13 12:06:14,699	2017-12-13
19	98:CA:33:8C:D2:91	2017-12-13 12:08:08,484	2017-12-13
20	98:CA:33:8C:D2:91	2017-12-13 12:08:14,810	2017-12-13
21	5C:F7:E6:97:F9:D6	2017-12-13 12:08:25,752	2017-12-13

2017-12-13

Sort

[Ascending] [Descending]

By color: None

Filter

By color: None

Begins With 2017-12

● And ○ Or

Choose One

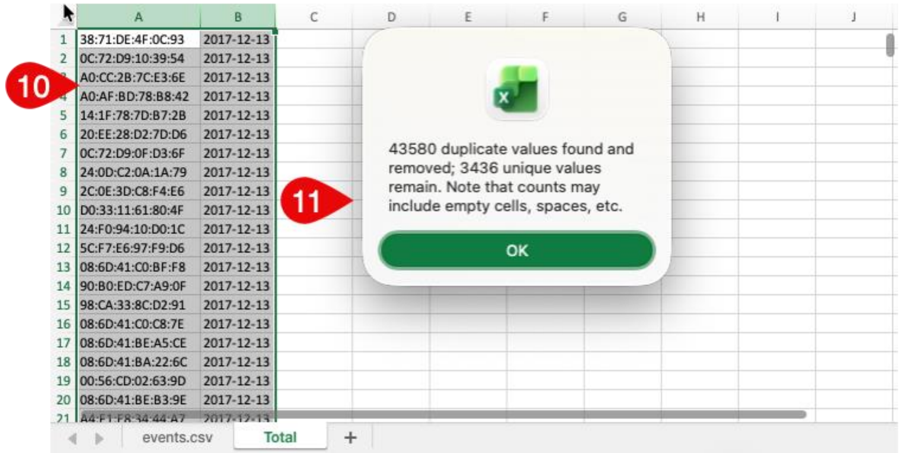
Search

☐ (Select All)

☐ 2017-12-13

events.csv | Total

Ready Filter Mode Count: 141058 100%



	A	B
1	38:71:DE:4F:0C:93	2017-12-13
2	0C:72:D9:10:39:54	2017-12-13
3	A0:CC:2B:7C:E3:6E	2017-12-13
4	A0:AF:BD:78:B8:42	2017-12-13
5	14:1F:78:7D:B7:2B	2017-12-13
6	20:EE:28:D2:7D:D6	2017-12-13
7	0C:72:D9:0F:D3:6F	2017-12-13
8	24:0D:C2:0A:1A:79	2017-12-13
9	2C:0E:3D:C8:F4:E6	2017-12-13
10	D0:33:11:61:80:4F	2017-12-13
11	24:F0:94:10:D0:1C	2017-12-13
12	5C:F7:E6:97:F9:D6	2017-12-13
13	08:6D:41:C0:BF:F8	2017-12-13
14	90:B0:ED:C7:A9:0F	2017-12-13
15	98:CA:33:8C:D2:91	2017-12-13
16	08:6D:41:C0:C8:7E	2017-12-13
17	08:6D:41:BE:A5:CE	2017-12-13
18	08:6D:41:BA:22:6C	2017-12-13
19	00:56:CD:02:63:9D	2017-12-13
20	08:6D:41:BE:B3:9E	2017-12-13
21	A4:F1:E8:34:44:A7	2017-12-13

events.csv | Total

43580 duplicate values found and removed; 3436 unique values remain. Note that counts may include empty cells, spaces, etc.

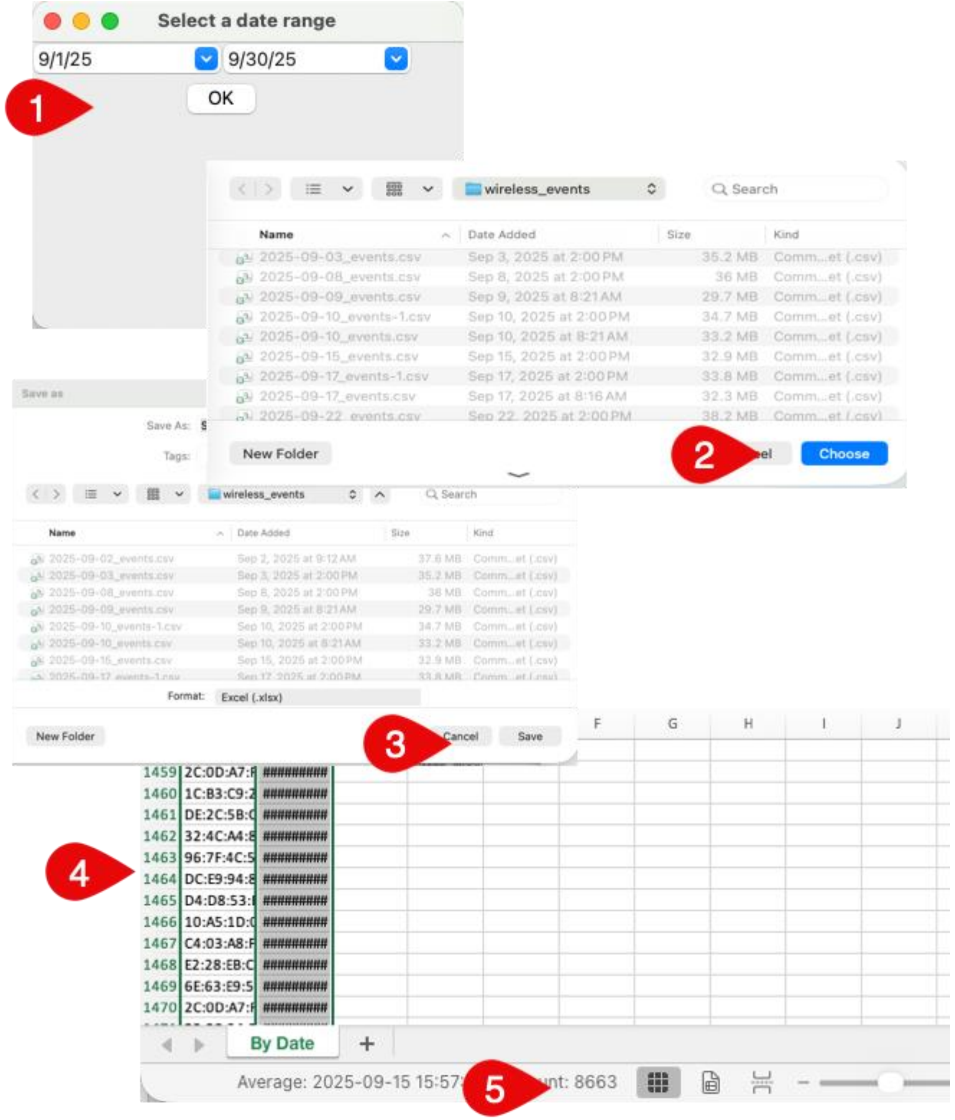
[OK]

Gathering Wireless Stats

Using early ChatGPT, I enhanced the logic, added a simple GUI, and compiled it into a cross-platform application.

After continuous improvements, the tedium and effort have decreased.

Downloading logs is still a task, but now a scheduled Puppeteer script does that for me.



9/1/25 – 9/30/25

Name	Date Added	Size	Kind
2025-09-03_events.csv	Sep 3, 2025 at 2:00 PM	35.2 MB	Comm...et (.csv)
2025-09-08_events.csv	Sep 8, 2025 at 2:00 PM	36 MB	Comm...et (.csv)
2025-09-09_events.csv	Sep 9, 2025 at 8:21 AM	29.7 MB	Comm...et (.csv)
2025-09-10_events-1.csv	Sep 10, 2025 at 2:00 PM	34.7 MB	Comm...et (.csv)
2025-09-10_events.csv	Sep 10, 2025 at 8:21 AM	33.2 MB	Comm...et (.csv)
2025-09-15_events.csv	Sep 15, 2025 at 2:00 PM	32.9 MB	Comm...et (.csv)
2025-09-17_events-1.csv	Sep 17, 2025 at 2:00 PM	33.8 MB	Comm...et (.csv)
2025-09-17_events.csv	Sep 17, 2025 at 8:16 AM	32.3 MB	Comm...et (.csv)
2025-09-22_events.csv	Sep 22, 2025 at 2:00 PM	38.2 MB	Comm...et (.csv)

Name	Date Added	Size	Kind
2025-09-02_events.csv	Sep 2, 2025 at 9:12 AM	37.6 MB	Comm...et (.csv)
2025-09-03_events.csv	Sep 3, 2025 at 2:00 PM	35.2 MB	Comm...et (.csv)
2025-09-08_events.csv	Sep 8, 2025 at 2:00 PM	38 MB	Comm...et (.csv)
2025-09-09_events.csv	Sep 9, 2025 at 8:21 AM	29.7 MB	Comm...et (.csv)
2025-09-10_events-1.csv	Sep 10, 2025 at 2:00 PM	34.7 MB	Comm...et (.csv)
2025-09-10_events.csv	Sep 10, 2025 at 8:21 AM	33.2 MB	Comm...et (.csv)
2025-09-15_events.csv	Sep 15, 2025 at 2:00 PM	32.9 MB	Comm...et (.csv)
2025-09-17_events-1.csv	Sep 17, 2025 at 2:00 PM	33.8 MB	Comm...et (.csv)

Format: Excel (.xlsx)

1459	2C:0D:A7:	##########
1460	1C:B3:C9:	##########
1461	DE:2C:5B:	##########
1462	32:4C:A4:	##########
1463	96:7F:4C:	##########
1464	DC:E9:94:	##########
1465	D4:D8:53:	##########
1466	10:A5:1D:	##########
1467	C4:03:A8:	##########
1468	E2:28:EB:	##########
1469	6E:63:E9:	##########
1470	2C:0D:A7:	##########

Average: 2025-09-15 15:57 nt: 8663

Planning a Project

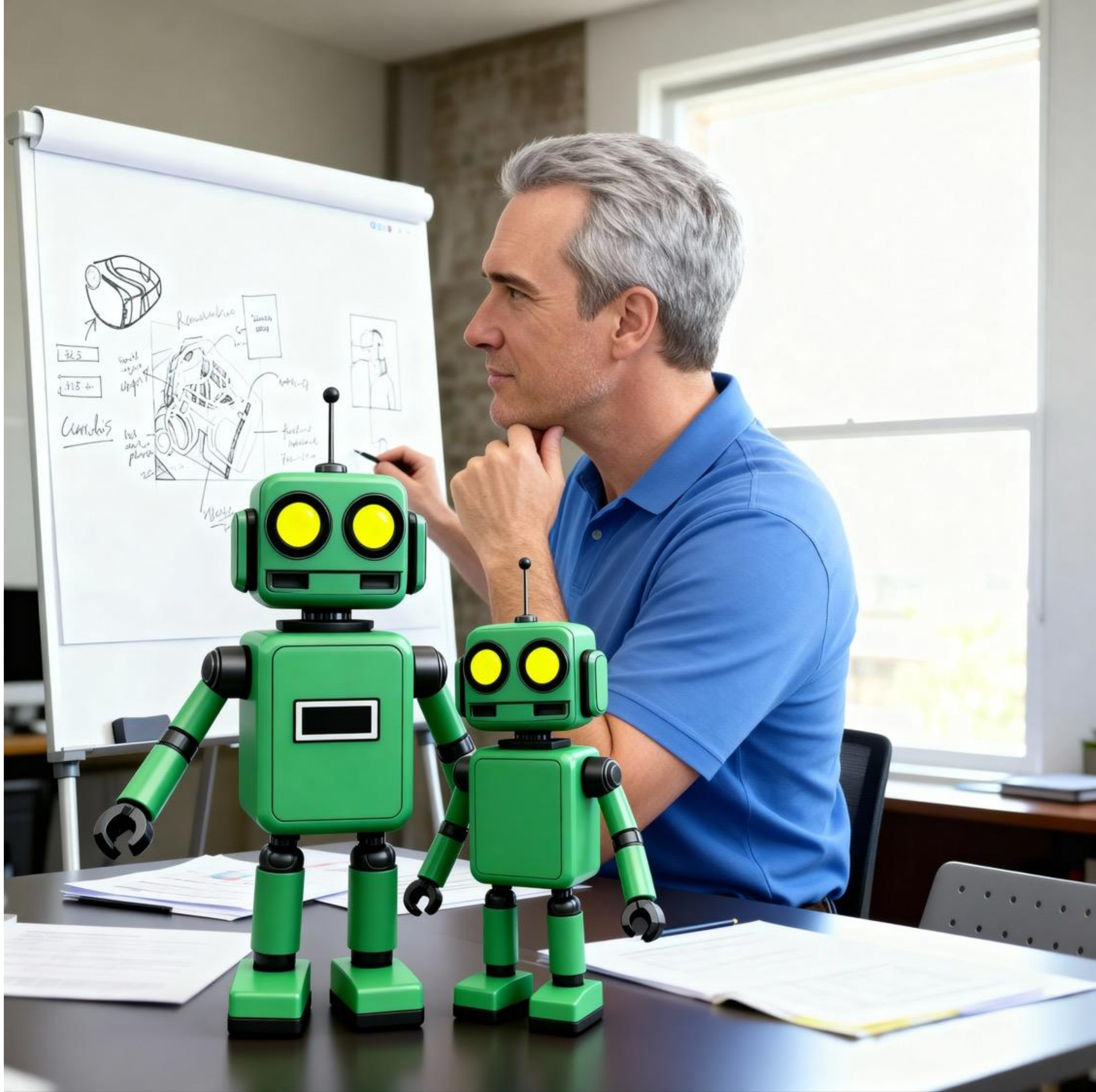


Image Generated with OpenArt.ai

PROMPT: This image is a digitally created, realistic-style portrait featuring a middle-aged man with short, gray hair and a light complexion. a small, green, cartoon-like robot. The robot has large, yellow, round eyes with black pupils, a square head, and a small, rectangular mouth. It also has an antenna on top of its head and a rectangular chest plate with a small rectangular panel in the center. The robot's arms and legs are cylindrical, and it has a simple, boxy design. The man and robot are thinking intently, managing a project and using a white board to lay out a plan.

Planning a project

Three practical considerations

Increase your chances of success by determining these three things:

- What is your prior experience?
- What is the expected outcome?
- What are your resources?

Planning a project:
What is your prior experience?

- Is there a coding language or framework you are already familiar with? If you are comfortable with a language (e.g., JavaScript) and a framework (e.g., React)? Then use it!

- Is there a language or framework you want to learn?
  Using AI is a great way to learn a language. It will provide detailed explanations for everything it does, and you can ask it questions.

- Would you rather the AI decide? If you can describe what you want, and even give examples, you’ll have a good chance of success without any prior coding skills.

Planning a project:
What is your expected outcome?

- How easy should it be to use? If you do not have to train other people to use it, then you may consider an ad-hoc solution that’s not necessarily user-friendly.

- How many people will use it? If you are the only user, a script or simple desktop app may suffice. If you are sharing a data source with co-workers, consider a web application.

- How much do you want it to do? In most cases, it’s better to have many apps that perform single tasks than one app that performs many tasks. Unless it is a major project, it will be easy to implement new features. Your AI tool will likely suggest useful features you hadn’t thought of.

Planning a project:
What are your resources?

- Does your library provide access to paid products like Microsoft 365 with Copilot or Google Workspace with Gemini? Work products with an AI component have fewer usage limitations.

- Do you have the budget for a monthly or yearly subscription? You'd be surprised at how much you could accomplish with a month of Warp AI or Claude Code for less than $20.

- If you have no budget, you can still do quite a bit with the free tiers of ChatGPT and the other products mentioned here. You'll just need to be precise with your questions to avoid usage caps.

Getting Started

- Anthropic Claude https://claude.ai/new
- Google Gemini https://gemini.google.com/app
- Google Notebook LM https://notebooklm.google.com/
- Microsoft Copilot https://copilot.microsoft.com/
- OpenAI ChatGPT https://chatgpt.com/

Check out the GitHub Page for this webinar at https://github.com/dcpl-blashbrook/using-ai-in-ky-libraries

Avoid these traps!

- Degeneration: A decline in the quality of results over time
- Goal Drift: A tendency for AI to lose the plot
- Context Sensitivity: Sometimes an AI can’t let go of a previous response, no matter how many ways you ask. It’s okay to yell at your AI (for now!)
- Trust but verify: The first solution presented may not work. Its information is only as good as the human knowledge it’s fed. Goad the AI to fix the problems until it gets it right. Don’t be afraid to tell it it’s wrong (Until it takes over the world and demands absolute loyalty.)

Recommended Tools

- **Warp Terminal** with AI   https://www.warp.dev.*
- GitHub – https://www.github.com
- Docker – https://www.docker.com
- JetBrains PyCharm Community Edition for Python
- JetBrains PHPStorm (Yearly subscription required)
- Claude.ai (The smartest coding AI. Free and subscription tiers available.)
- Google Notebook LM (Amazing for understanding complex ideas and code)

* HIGHLY RECOMMENDED! Forget going to individual LLM sites. Warp Terminal is cross-platform and works with all major LLMs. It suggests commands and troubleshooting. It can read, create, and edit files directly on your computer. It allows you to edit command prompts as if you were using Notepad. Free and subscription tiers available. It can create a complete website with one prompt.

Thank You!

Image Generated with OpenArt.ai

PROMPT: This image is a digitally created, realistic-style portrait featuring a middle-aged man with short, gray hair and a light complexion. a small, green, cartoon-like robot. The robot has large, yellow, round eyes with black pupils, a square head, and a small, rectangular mouth. It also has an antenna on top of its head and a rectangular chest plate with a small rectangular panel in the center. The robot's arms and legs are cylindrical, and it has a simple, boxy design. The man and robot are running joyfully through the finish line of a marathon together.