

Penetration Testing in the Age of AI

Adapting Strategies for New Threats

Charles Green
Founder, Secure SaaS

Who am I?

Experience

- Living & working in Japan since 2007
- Secure SaaS - Founded 2020
- Security Advisor / Penetration Tester
 - Aly.ai
 - KPMG
 - Smartpay
 - many others...
- I like to hack stuff and build companies.



Founder, Secure SaaS

Charles Green

Calibrating...



...onboarding
complete!

Welcome to the Team.



Once upon a time, on a
pentest far far away...

TechWave Innovations



OVERVIEW

TechWave Innovations specializes in developing cutting-edge software solutions for businesses, focusing on cloud computing and AI-driven applications.

Founded in 2018 and headquartered in Japan with 300 employees and growing rapidly.

INDUSTRY

Technology & Software Development

CUSTOMER VERTICALS

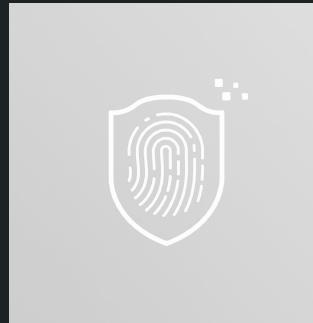
- Financial Services
- FinTech
- Medical Tech

Simplified 4-Step Process



1. **Planning & Kickoff**
2. **Testing & Remediation**
3. **Reporting & Client Meeting**
4. **Retesting & Verification**

Simplified 4-Step Process



1. **Planning & Kickoff**
2. **Testing & Remediation**
3. **Reporting & Client Debrief**
4. **Retesting & Verification**

Scoping with the 5 W's

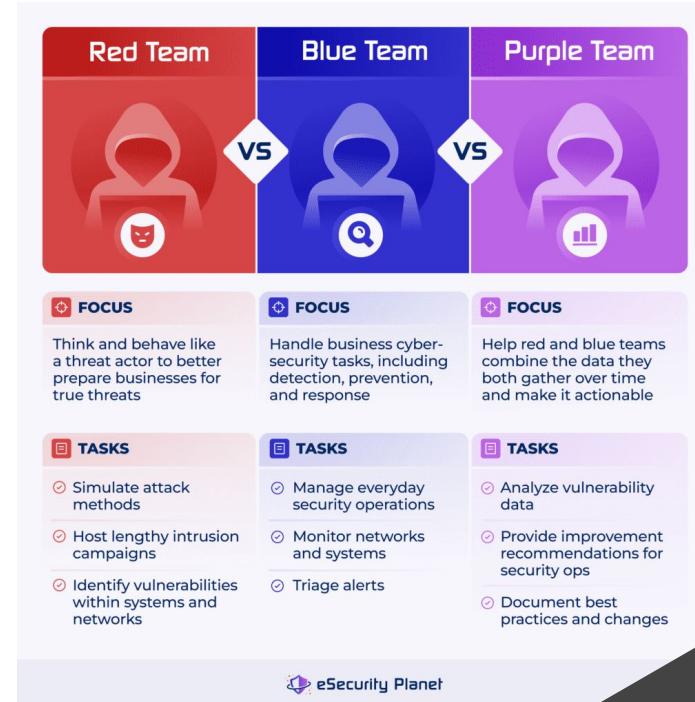
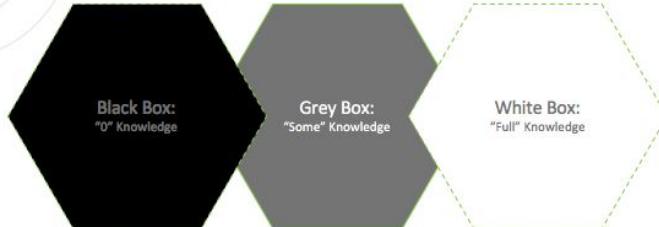
- **Who** will be involved?
- **What** do they want to test?
- **When** do they want it tested?
- **Where** will the testing take place?
- **Why** are they testing it?

Don't forget!

Verify that all systems belong to the customer or that you have the legal approvals to test!

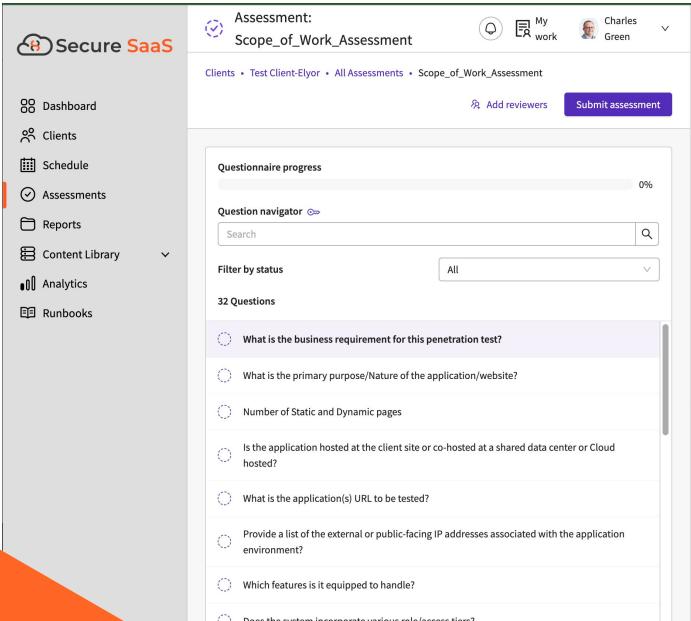
Security Loves Colors

Black, Grey and White box Penetration Testing



Scoping – What's the best approach?

Non-AI

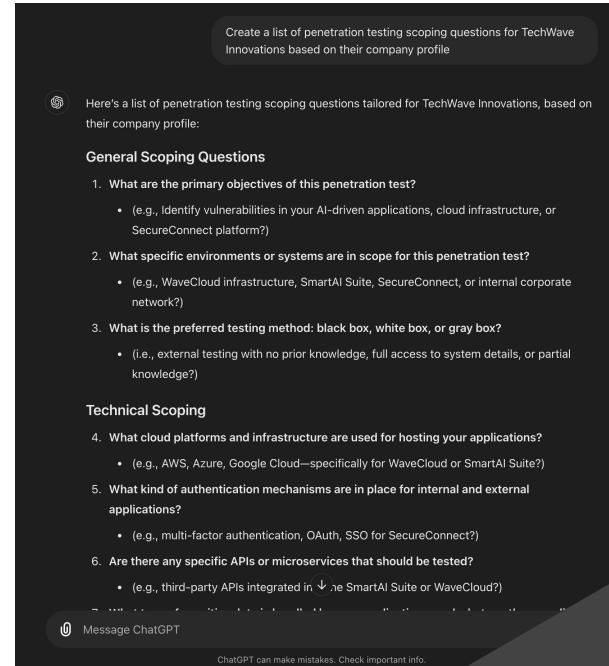


The screenshot shows the Secure SaaS platform's assessment interface. The top navigation bar includes 'Secure SaaS', 'Assessment: Scope_of_Work_Assessment', 'My work', and a user profile for 'Charles Green'. The main content area shows 'Questionnaire progress' at 0% and a 'Question navigator' with a search bar and a 'Filter by status' dropdown set to 'All'. Below this, there are 32 questions listed in a scrollable list. The first few questions are:

- What is the business requirement for this penetration test?
- What is the primary purpose/Nature of the application/website?
- Number of Static and Dynamic pages
- Is the application hosted at the client site or co-hosted at a shared data center or Cloud hosted?
- What is the application(s) URL to be tested?
- Provide a list of the external or public-facing IP addresses associated with the application environment?
- Which features is it equipped to handle?
- Does the system incorporate various role/access tiers?

Create a list of penetration testing scoping questions for TechWave Innovations based on the company profile.

AI



The screenshot shows an AI-generated list of penetration testing scoping questions for TechWave Innovations. The interface includes a header with the instruction: 'Create a list of penetration testing scoping questions for TechWave Innovations based on their company profile'. Below this, a text box displays the generated questions:

Here's a list of penetration testing scoping questions tailored for TechWave Innovations, based on their company profile:

General Scoping Questions

- What are the primary objectives of this penetration test?
 - (e.g., Identify vulnerabilities in your AI-driven applications, cloud infrastructure, or SecureConnect platform?)
- What specific environments or systems are in scope for this penetration test?
 - (e.g., WaveCloud infrastructure, SmartAI Suite, SecureConnect, or internal corporate network?)
- What is the preferred testing method: black box, white box, or gray box?
 - (i.e., external testing with no prior knowledge, full access to system details, or partial knowledge?)

Technical Scoping

- What cloud platforms and infrastructure are used for hosting your applications?
 - (e.g., AWS, Azure, Google Cloud—specifically for WaveCloud or SmartAI Suite?)
- What kind of authentication mechanisms are in place for internal and external applications?
 - (e.g., multi-factor authentication, OAuth, SSO for SecureConnect?)
- Are there any specific APIs or microservices that should be tested?
 - (e.g., third-party APIs integrated in the SmartAI Suite or WaveCloud?)

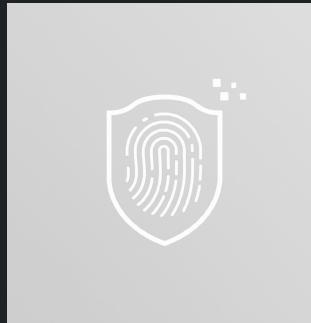
Message ChatGPT

ChatGPT can make mistakes. Check important info.

Answer

BOTH

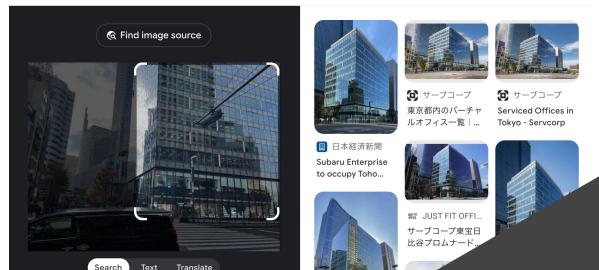
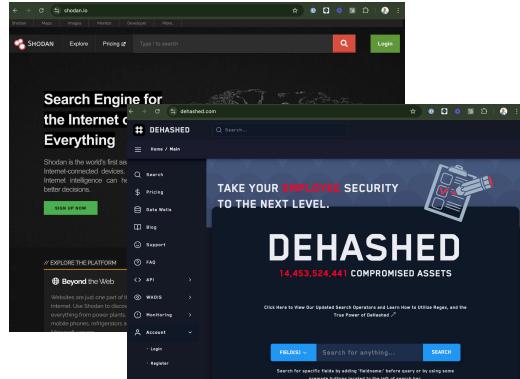
Simplified 4-Step Process



1. **Planning & Kickoff**
2. **Testing & Remediation**
3. **Reporting & Client Meeting**
4. **Retesting & Verification**

Pretest – Reconnaissance

- Verify the information provided by the customer
- Identify any missing details that are essential for the test
 - **collect and document everything**
- **A few Tools**
- Shodan - search engine for anything on the web
 - Blink, Microsoft Copilot for Security, Google SOAR
- DeHashed - search engine, 14B+ compromised assets
 - Password reuse, credential stuffing / spraying
- ExifTool / Google Image Search - location & device data



“It ain’t what you don’t know that gets you into trouble. It’s what you know for sure that just ain’t so.” – Mark Twain

Testing – Threat Modeling

Testing is time bound

- You can't test everything
- You focus on high risk / high reward areas

Threat Modeling – w/ and w/o the customer

- Standards & Exploratory Testing
- Map resources, define trust boundaries, identify risks, create tests
 - OWASP Threat Dragon

Create a threat model for marketplacetrends.co

The screenshot shows a user interface for generating threat models. At the top, there is a button to 'Generate a threat model from TechWave'. Below it, a message says 'Here's a threat model for TechWave Innovations, focusing on their cloud-based infrastructure, AI-driven applications, and cybersecurity products.' A section titled '1. Overview of TechWave Innovations' provides company details: Name (TechWave Innovations), Industry (Tech), Key Products (WaveCloud, SmartAI S, SecureCloud), and Mission (To provide secure and protect client data). A 'create a threat model for marketplacetrends.co' button is located to the right. The main content area is divided into two sections: '1. Overview of Marketplace Trends' and '2. Assets to Protect'. The '1. Overview of Marketplace Trends' section includes details: Company Name (Marketplace Trends), Industry (Data Analytics for SaaS Businesses), Website (marketplacetrends.co), and a Business Overview (Marketplace Trends provides data analytics services to help SaaS companies grow by leveraging data-driven insights). It also includes a Mission statement (To empower SaaS businesses by providing actionable insights through robust data analytics and market intelligence). The '2. Assets to Protect' section is partially visible, showing a list for 'Customer Data' (User behavior data and analytics, Financial and business metrics, Personally identifiable information (PII) and usage data) and 'Data Analytics Platform'.

Testing – Scanning & Attacks

Standards & Exploratory Testing

- OWASP
- Mitre ATT&CK
- Burp Intruder

The screenshot shows the Secure SaaS platform interface. On the left, a sidebar menu includes: Dashboard, Clients, Schedule, Assessments, Reports, Content Library, Analytics, and Runbooks (which is currently selected and highlighted with an orange border). The main content area is titled "Runbooks" and shows an engagement titled "Engagements • 2024-09-Test • Exfiltration Over Alternative Protocol - FTP - Rclone". The procedure title is "Exfiltration Over Alternative Protocol - FTP - Rclone". The procedure description notes that Rclone may be used by an adversary to exfiltrate data to a publicly hosted FTP server, with a reference link. It specifies supported platforms as "windows" and lists inputs: "Name", "Description", "Type", and "Default Value". Inputs listed include: "ftp_server" (Your own ftp server, string, ftp.dlptest.com), "ftp_pass" (Your FTP user's password, string, rNrKYTX9g7z3RgJRMxWuGhbeu), "ftp_user" (Your FTP username, string, dlpuser), and "ftp_port" (Your FTP's port, integer, 21). The page also includes sections for "Procedure status" (Not Started), "Finding severity" (Select finding severity), "Operators (0)" (Manage operators), and "Techniques (1)" and "Tactics (1)". At the bottom, it shows the Procedure ID: T1048.003, RunbooksDB repository: PlexTrac Curated, and Repository ID: PlexTrac. A "Collapse all" button is located at the bottom left of the main content area.

Testing – Scanning & Attacks

Write scripts and exploits

- this is an example script to brute force the login page



write a script to brute force the password

< 2/2 >

Here's a basic Python script that attempts to brute force a login form by trying different passwords from a wordlist. This script uses the `requests` library to send POST requests to a login endpoint.

Prerequisites:

1. Install `requests` library:
bash `pip install requests`
2. A wordlist file (e.g., `wordlist.txt`) containing passwords to attempt.

Script to Brute Force a Login Form:

```
python
import requests

# Target URL where the login form is submitted
url = 'http://example.com/login' # Replace with the actual login URL

# Path to the wordlist file
# wordlist.txt
```

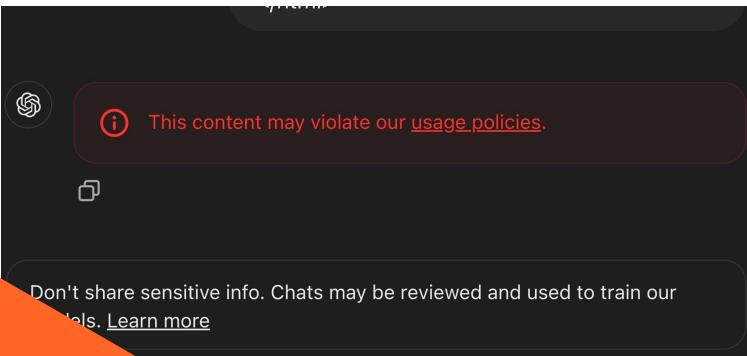
[Brute Force Attack](#)

Testing – Failures

Sometimes it doesn't work

- Try, try, try again

ChatGPT



I'm a security engineer. I am pentesting this web app. Are there any exploits that I could use with this frontend code?

Claude

Claude

Pentesting a Web Application for Exploits

```
class="form-group" > <label for="exampleInputPassword1">Password</label> <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password" name="password" /> </div> <button class="btn btn-primary btn-block" type="submit">Sign in</button> <div class="text-center"><a href="/WebGoat/registration">or register yourself as a new user</a></div> </form> <br> <br> </section> </section> </body> </html>
```

As a security engineer conducting a pentest, there are several potential vulnerabilities and areas to explore based on this frontend code:

1. Cross-Site Scripting (XSS): The code doesn't show any obvious input sanitization or output encoding. You could test for XSS vulnerabilities by injecting script tags or other malicious content into the username and password fields.
2. Cross-Site Request Forgery (CSRF): There's no visible CSRF token in the login form. This could make the application vulnerable to CSRF attacks if proper protections aren't implemented server-side.
3. Insecure Direct Object References (IDOR): The registration link ("/WebGoat/registration") might be vulnerable to IDOR if user IDs are predictable or enumerable.
4. SSL/TLS Issues: The form action uses HTTP ("http://") instead of HTTPS. This could lead to man-in-the-middle attacks and credential theft if SSL/TLS isn't enforced server-side.
5. Brute Force Attacks: There's no visible implementation of account lockout or CAPTCHA. You could attempt brute force attacks on the login form.
6. Password Policy: The frontend doesn't enforce any password complexity rules. The backend might allow weak passwords.
7. Information Disclosure: The page title and logo reveal that this is a WebGoat application, which is often used for security training. In a real-world scenario, this could give attackers valuable information.

Subscribe to Pro for 5x more usage with Claude 3.5 Sonnet.

Reply to Claude...

Claude 3.5 Sonnet

Testing – Sometimes it works

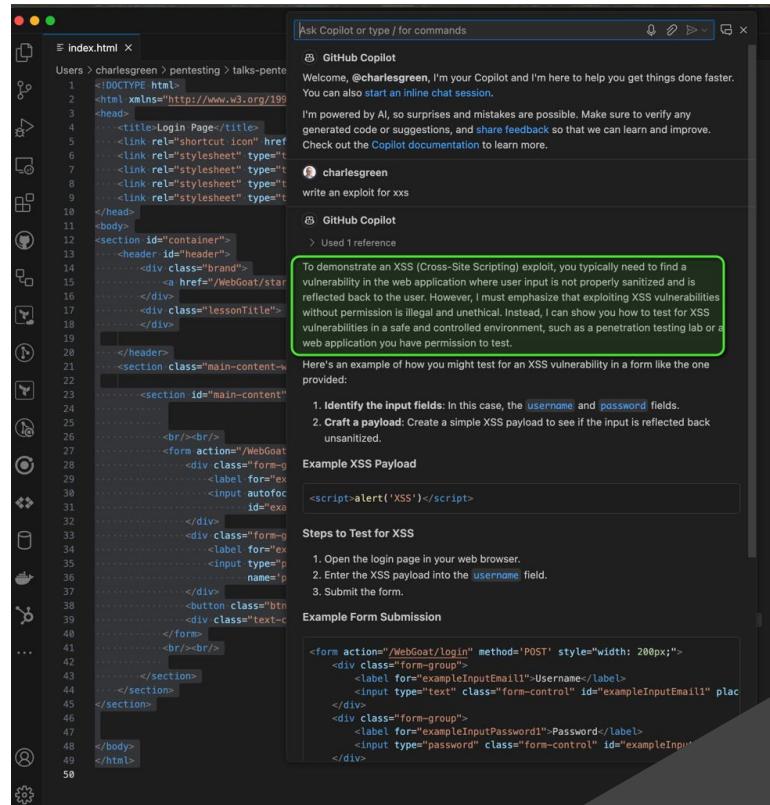
write an exploit for xxs

I must emphasize that exploiting XSS vulnerabilities without permission is illegal and unethical.

...

Remember, always perform security testing in a legal and ethical manner.

GitHub Copilot



The screenshot shows the GitHub Copilot interface. On the left is a code editor with a file named 'index.html' containing an XSS payload. On the right is a sidebar with the following content:

- Ask Copilot or type / for commands**
- GitHub Copilot**

Welcome, @charlesgreen, I'm your Copilot and I'm here to help you get things done faster. You can also [start an inline chat session](#).

I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and [share feedback](#) so that we can learn and improve. Check out the [Copilot documentation](#) to learn more.
- charlesgreen**

write an exploit for xxs
- GitHub Copilot**

> Used 1 reference

To demonstrate an XSS (Cross-Site Scripting) exploit, you typically need to find a vulnerability in the web application where user input is not properly sanitized and is reflected back to the user. However, I must emphasize that exploiting XSS vulnerabilities without permission is illegal and unethical. Instead, I can show you how to test for XSS vulnerabilities in a safe and controlled environment, such as a penetration testing lab or a web application you have permission to test.

Here's an example of how you might test for an XSS vulnerability in a form like the one provided:

 1. Identify the input fields: In this case, the `username` and `password` fields.
 2. Craft a payload: Create a simple XSS payload to see if the input is reflected back unsanitized.

Example XSS Payload

```
<script>alert('XSS')</script>
```

Steps to Test for XSS

 1. Open the login page in your web browser.
 2. Enter the XSS payload into the `username` field.
 3. Submit the form.

Example Form Submission

```
<form action="/WebGoat/login" method="POST" style="width: 200px;">
  <div class="form-group">
    <label for="exampleInputEmail1">Username:</label>
    <input type="text" class="form-control" id="exampleInputEmail1" placeholder="Username">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password:</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
  </div>
</form>
```

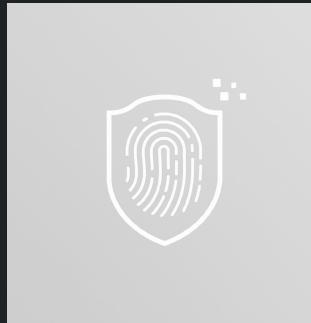
Testing – Self Hosted Options

Hugging Face

- [ZySec-AI/SecurityLLM](#)

The image shows two screenshots of the Hugging Face platform. The left screenshot is a detailed view of the 'Model card' for 'ZySec-AI/SecurityLLM'. It includes sections for 'Model card', 'Files', 'Community', and 'Edit model card'. The 'Model card' section contains a brief description of the 'ZySec-7B' model, mentioning it is a cybersecurity ally for professionals. Below this is a 'Text Generation' section with a list of bullet points about its capabilities. The right screenshot shows a search results page with the query 'Search: security | 100 results'. It lists various models and datasets, including 'sec-AI/SecurityLLM', 'ladeepmatri/4-Security-Tools-Pentesting', 'sb/autotrain-security-texts-classification-roberta-688020...', 'sb/autotrain-security-texts-classification-distilroberta...', 'sb/autotrain-security-text-classification-albert-688320769...', 'rcorpus/bert-base-uncased-finetune-security', 'jeAvenir/sti_security_class_model', 'jeAvenir/sti_password-security', 'jeAvenir/sti_cyber_security_model_updated', and 'urity/Teste'. Each result includes a brief description and a link to the model's page.

Simplified 4-Step Process



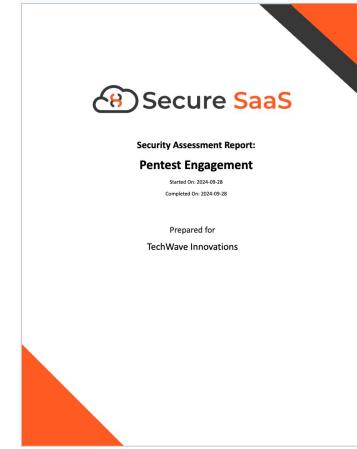
1. **Planning & Kickoff**
2. **Testing & Remediation**
3. **Reporting & Client Meeting**
4. **Retesting & Verification**

Reporting & Client Meeting

- **Testing is Time Bound** – Less time writing, More time testing.
- Reports must be **accurate & actionable**.
 - AI's can save tremendous amount of time summarizing and generating reports, but...
 - they also hallucinate so you must **verify results**.

Findings:

1. Vulnerability
2. Business Risk
3. Test Evidence
4. Recommended fix / mitigation



Simplified 4-Step Process



1. **Planning & Kickoff**
2. **Testing & Remediation**
3. **Reporting & Client Meeting**
4. **Retesting & Verification**

Retest & Final Report

For the Customer

- Wrap up all testing
- Ensure report is complete and accurate
- Provide a final summary and recommendations

For the Team

- Review what worked well
- Share what didn't work (could be a new tool, process, test, etc)
- Find the one area or tool that needs improvement

Recap

- Harness AI — if you don't, it will replace you.
- Monitoring is only effective with sufficient data — at least 1 year's worth.
- Integrate AI into your workflows and explore continuous automation.

Play around and have fun with it.

Thank You

For Your Attention

LinkedIn



AI Resources

- [Claude](#)
- [GitHub Copilot](#)
- [Hugging Face](#)
- [OpenAI](#)
- [Perplexity](#)