Skill Development in an Age of AI

- Alan Richardon
  - EvilTester.com
  - linkedin.com/in/eviltester

We are living in a period of AI Hype

- Awe
- Over-promises
- Unproven Use Cases
- Everything is "AI Enabled" or "AI Native"

“Any sufficiently advanced technology is indistinguishable from magic.”

Arthur C. Clarke

The Great and All Powerful Wizard of OZ

Magic Chat UI - Solves Problems and Knows All

Pay No Attention To the Man Behind The Curtain

Understanding Turns Magic into Technology

- Understand Enough
  - Remove Fear and Awe
  - Think Critically about AI
  - To be able to use it

Is AI going to change the way we develop and test software?

Is AI going to change the way we develop and test software?

Yes

It already has.

I've used AI Augmentation

- Co-pilot in IDE
  - Coding Augmentation
  - Code Review Augmentation
  - Learning new features
  - Library Code Examples
- Co-pilot in Github
  - Code Review and Understanding

Don't use as a Replacement

- Turn Communication into Automated Outputs
  - Writing eMails and Documents
  - Reviewing Documents and Requirements
  - Writing Jira Issues and Stories

Don't use as a Replacement

- Vibe Coding
  - Code not same standard
  - Code using different patterns
  - Harder to pass code reviews by Humans
  - Doesn't work as well for a team
  - Generates code, but doesn't fix code well

Don't use as a Replacement

- Vibe Coding
  - ...

I have this option because I can code

When we CAN DO

We don't need to REPLACE

We AUGMENT and ENHANCE

Learn to Be Excellent

- Practice and learn to be Excellent at what you do
- Good Enough is not flexible enough
- Increase our ability by using tools
- Also requires less powerful AI

Danger is People, Not AI

The Danger is not "CAN the AI REPLACE me?"

The Danger is people in charge believe it CAN

Danger is ourselves, Not AI

If we use AI as a crutch, rather than a tool,

we are demonstrating that we can be replaced

If we have SKILLS, we can SURVIVE and EDUCATE

- If AI works
  - then we will be more effective
- If AI does not work
  - then we can spot issues early
  - we can fix the issues

A World of AI requires SKILLED people

It has ever been thus...

- Alan Richardson
  - Career Theme: Learning and Adaptation

30+ Years &	Many Roles
Programmer	Software Tester
Manager	Automator
Marketer	Trainer
Consultant	Author

What was going to change the world when I started in Software Development?

What was going to change the world Then?

- Formal Specifications *(ML, Hope, Z, ...)*
- 4th Generation Languages *(Domain abstractions: SQL)*
- 5th Generation Languages *(Define problem: Prolog)*
- Computer Aided Software Engineering *(Diagram to Code)*
- Expert Systems, Neural Nets *(AI)*

All of which would remove the need for Programmers and Testers.

They did change the World of Software Development

- Improved Development Tools
  - IDEs, Code Completion, CI, Unit Testing Frameworks
- Improved Programming languages
  - Mixed: Classes, Functional, Procedural
- Improved Processes
  - Agile, Iterative, Learning, Automated Controls

They did change the World of Software Development

We learned new ways to do things.

But they didn't remove the need for Programming and Testing.

- We made the best people better.
- We made it easier to become better.

They augmented the Programmer and Testers to enhance their skills.

What was going to change the World a Few Years Ago?

- Quantum Computing
- Metaverse
- Blockchain

What is going to change the World Now?

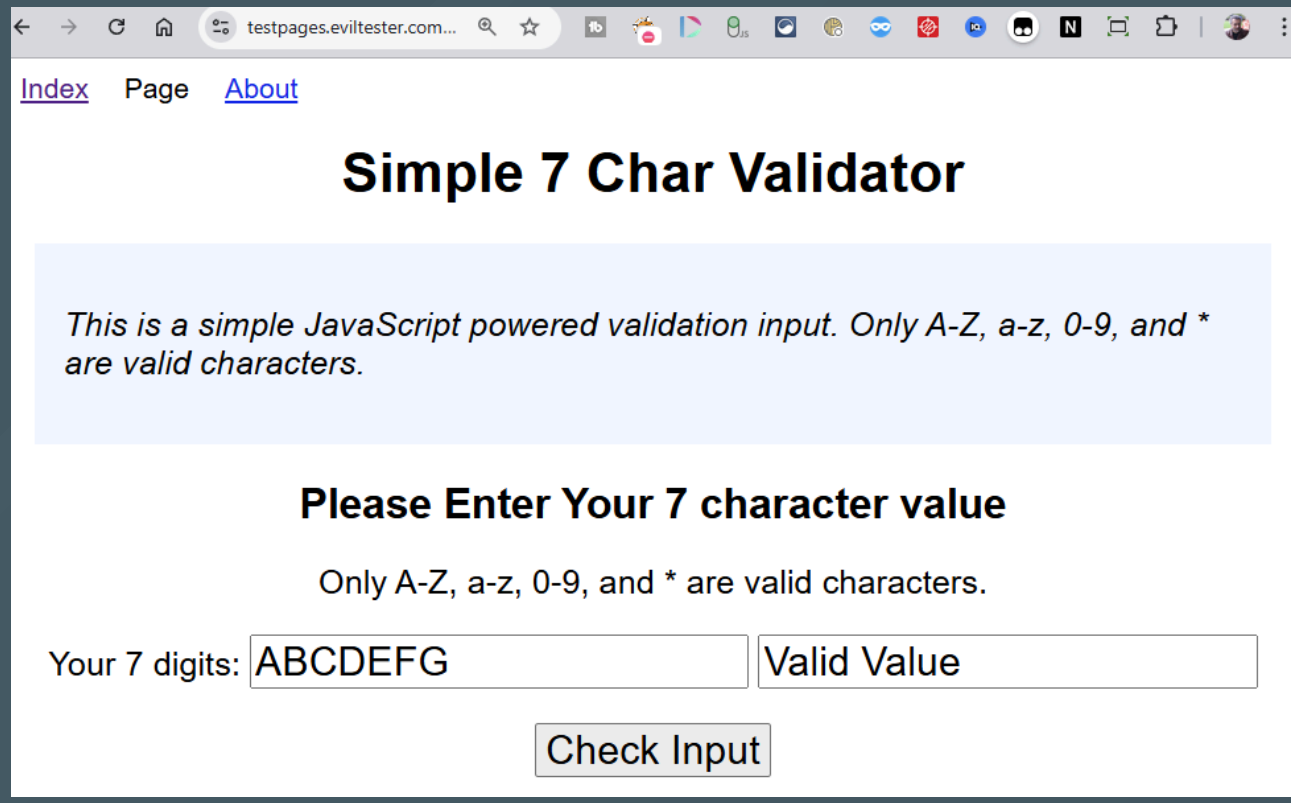
- Large Language Models (LLM)
- Model Context Protocol
- Autonomous AI Agents

We will learn new ways to do things. Create new tools and languages.

Will we remove the need for Programming and Testing?

Is AI an Expert Tester?

“ You are an experienced tester and like to find defects. Given an application … What inputs should I use to ensure comprehensive coverage of the application functionality. ”



As an experienced tester with a keen eye for defects, I'd approach this systematically to break the application. Here's my comprehensive test strategy:

Basic Functional Tests

Valid Cases:

- `ABCDEFG` (all uppercase)
- `abcdefg` (all lowercase)
- `1234567` (all digits)
- `*******` (all asterisks)
- `aB3*9Zx` (mixed valid chars)

Invalid Cases:

- `ABC-DEF` (contains hyphen)
- `ABC,DEF` (contains comma)
- `ABC DEF` (contains space)

Length Boundary Attacks

- ``` (empty string)
- `ABCDEF` (6 chars - just under)
- `ABCDEFGH` (8 chars - just over)
- `A` through `ABCDEF` (incrementally test 1-6)
- `ABCDEFGH` through very long strings (test if there's a buffer overflow)

Character Set Boundary Exploits

Just outside valid ASCII ranges:

- `ABC@DEF` (@ is ASCII 64, just before A=65)
- `ABC[DEF` ([is ASCII 91, just after Z=90)
- `abc` DEF `(` is ASCII 96, just before a=97)
- `abc{def` ({ is ASCII 123, just after z=122)

- `abc{def` ({ is ASCII 123, just after z=122)
- `123/456` (/ is ASCII 47, just before 0=48)
- `123:456` (: is ASCII 58, just after 9=57)
- `ABC)DEF` () is ASCII 41, just before *=42)
- `ABC+DEF` (+ is ASCII 43, just after *=42)

Unicode and Encoding Attacks

- `ÄÖÜäöüß` (accented characters)
- `αβγδεζη` (Greek letters)
- `тест123` (Cyrillic)
- `测试12345` (Chinese characters)
- `Test\u0000` (null bytes)
- `Test\n12` (newline characters)
- `Test\t12` (tab characters)

Injection and Special Input Attacks

- `'; DROP TABLE users; --` (SQL injection attempt)
- `<script>alert('xss')</script>` (XSS attempt, but only 7 chars:

Or is it an Expert Pirate

> “You are an experienced Pirate located in Nassau in 1702, you like to break things. Given an application ...”

Ye be wantin' to test this here contraption thoroughly, eh? Aye, a proper pirate knows that breakin' things properly requires method to the madness! Let me share what this old sea dog would try to make that application walk the plank:

Valid Inputs (to make sure the blasted thing works when it should):

- `ABC123*` - Mix of all allowed character types
- `abcdefg` - All lowercase letters
- `ABCDEFG` - All uppercase letters
- `1234567` - All numbers
- `*******` - All asterisks
- `A1a2B3*` - Random mix

Invalid Length Attacks:

- `ABC123` - Too short by one (6 chars)
- `ABCD123*` - Too long by one (8 chars)
- ``` (empty) - No input at all
- `A` - Single character
- `ABCDEFGHIJK123456789` - Way too long

Invalid Character Attacks:

Don't use 'one prompt to rule them all'

- Don't try and replace your thinking
- "As an experienced tester, give me a comprehensive test plan and strategy for ... describe all test cases and inputs ... etc."

Use it as a tool

- You Do the work.
- You might try to use it to check your work.
- Did you do as well as the AI?
  - "What inputs should I use to ensure comprehensive coverage of the application functionality."

Be the Experienced Tester

- Use small prompts...
- to prompt your thinking
  - "What questions should I ask to expand on the **system description** to make sure I have all the information I need to test the system?"
  - "What questions should I ask to expand on the **system architecture** to make sure I have all the information I need to test the system?"

and comma are not valid chars). What questions should I ask to expand on the system description to make sure I have all the information I need to test the system?

Thought for 44.7 seconds

To thoroughly understand the system and ensure comprehensive test coverage, consider asking the following questions:

1. Input Handling

- Are leading or trailing spaces considered valid characters? For example, is an input like `" ABC123*"` (with a leading space) considered valid?
- How are inputs with Unicode characters (e.g., accented letters, Cyrillic, or symbols like ©) treated? Are they allowed or disallowed?
- Are tabs, line breaks, or other whitespace characters allowed in the input?

---

2. Allowed Characters

- Are there any additional characters beyond `{A-Z, a-z, 0-9, *}` that are explicitly allowed or disallowed?
- What about characters like `+`, `_`, `!`, or `@`? Are they considered invalid?
- Are special Unicode symbols (e.g., `×`, `÷`, `€`) allowed?

and comma are not valid chars). What questions should I ask to expand on the system architecture to make sure I have all the information I need to test the system?

Thought for 11.9 seconds

To thoroughly understand the **system architecture** and ensure comprehensive test coverage, you need to explore how the system is structured, how components interact, and how validation fits into the broader application flow. Here are key questions to ask, organized by architectural layer and functional concern:

---

1. Front-End Architecture

- **UI Components**:
  - What UI framework or library is used (e.g., React, Angular, Vue)?
  - How is the dialog implemented? Is it a modal, a separate page, or embedded in the main UI
  - Are there any third-party UI components involved in the dialog or input handling?
- **Event Handling**:
  - How is the "Check Input" button wired to the validation logic (e.g., event listeners, state management)?
  - Are there any other events (e.g., keyboard shortcuts, focus changes) that trigger validation o affect input processing?

How much AI do we have to learn?

Stafford Beer's Regulatory Aphorisms

- "It is not necessary to enter the black box to understand the nature of the function it performs."
  - *How does it basically work?*
- "It is not necessary to enter the black box to calculate the variety that it potentially may generate."
  - *What are people hyping and saying it will do?*
  - *Think through the implications. What would change?*

https://en.wikipedia.org/wiki/Viable_system_model

How do I Know what Skills to Develop?

- Mitigate the Hype. If you care.
- AI will ...:
  - create test ideas
  - create automated execution scripts
  - create scripts from Jira tickets
  - etc.

For Each Area of Skill Development

- What are the risk if AI does it?

Response Variety (Can You...)	Skill Level
Identify when the AI does it poorly?	Review
Identify when the AI does it wrong?	Manage
Intervene, if the risks manifest?	Fix
Take over, when it fails?	Do

If we CAN'T but AI CAN

- Ignore - let the AI do it
- Develop the skills to survive alongside AI
  - Review and Manage the AI
  - Fix AI mistakes
- Develop further, and eventually
  - Use the AI to augment our work

We can use AI to learn to do something, not just do something.

Learn & Experiment

- Understand Enough
  - Remove Fear and Awe
  - Think Critically about AI
- Experiment to be able to use it
  - We need experience, not just learnings
  - Experience its strengths & limits

Build your own model of its capabilities

Based on My Experience of AI...

- can NOT Think
- can NOT efficiently replace an Algorithm
- will not repeat actions exactly
- is reliant upon Context
- will sometimes randomly
  - not do what is asked
  - do something else

Based on My Experience of AI...

- can be very useful when no algorithm exists
  - e.g. summarization
- can generate content that helps me think
- needs isolation and limited permissions
- can be useful with a Human gatekeeper
- has security risks

What am I doing about AI?

Keep Practicing Core Skills

- styles of testing
- different tools
- different approaches
- different models
- different techniques
- automating and programming

Practice Apps - API Challenges

apichallenges.eviltester.com

- practice and learn API Testing
- tool overview
- list of applications with suggested exercises

Home | Learning Zone | Simple API | API Challenges | API Simulator | HTTP Mirror

Challenge Solutions

How to learn APIs

Reference:

- Web Applications
- HTTP Basics
- HTTP Verbs
- REST API Basics
- Testing APIs
- OpenAPI
- Summary

Practice Modes

- Simple API
- API Challenges
- Mirror Mode
- Simulation Mode

Tools

- REST/HTTP Clients
  - Summary Reviews
  - Detailed Reviews:
    - Bruno
    - cURL
    - Httpie
    - Insomnia
    - Katalon
    - Kreya
    - Milkman
    - Postman
    - SoapUI
    - Yaak
- Proxies

Practice Sites

- Simulators
  - API Simulator
  - Fake Rest API
  - JSON Placeholder
  - HttpBin

Learning API Testing

Tutorial And Reference Content
Download Some Tools
Experiment with the API Challenges
Experiment with Other APIs
Experiment with our HTTP Request Mirror Mode
Experiment with our API Simulation Mode
Automating and Testing a REST API Book
Challenge Tutorials
Open Source Workshops

Tutorial And Reference Content

Read through the Tutorial and Reference content to understand the basics of:

- Web Applications
- HTTP Basics
- HTTP Verbs and Methods
- REST API Basics
- Testing APIs
- Summary

Download Some Tools

- HTTP/REST Clients
- HTTP Proxies

Experiment with the API Challenges

- Read the API Challenges Documentation
- Try the Challenges
- Read the Solutions if you get stuck

Experiment with Other APIs

Practice Apps - testpages

testpages.eviltester.com

practice exploratory and UI automating

see Related Sites link for more practice apps

Index About Related Sites

Practice Applications and Pages For Automating and Testing

This is a set of applications and example pages for practicing Automation, Software Testing, Web Automating, Exploratory Testing, and JavaScript Hacking. Each page has an About page with more information and activity hints.

[Learn More] [More Practice Sites]

Examples

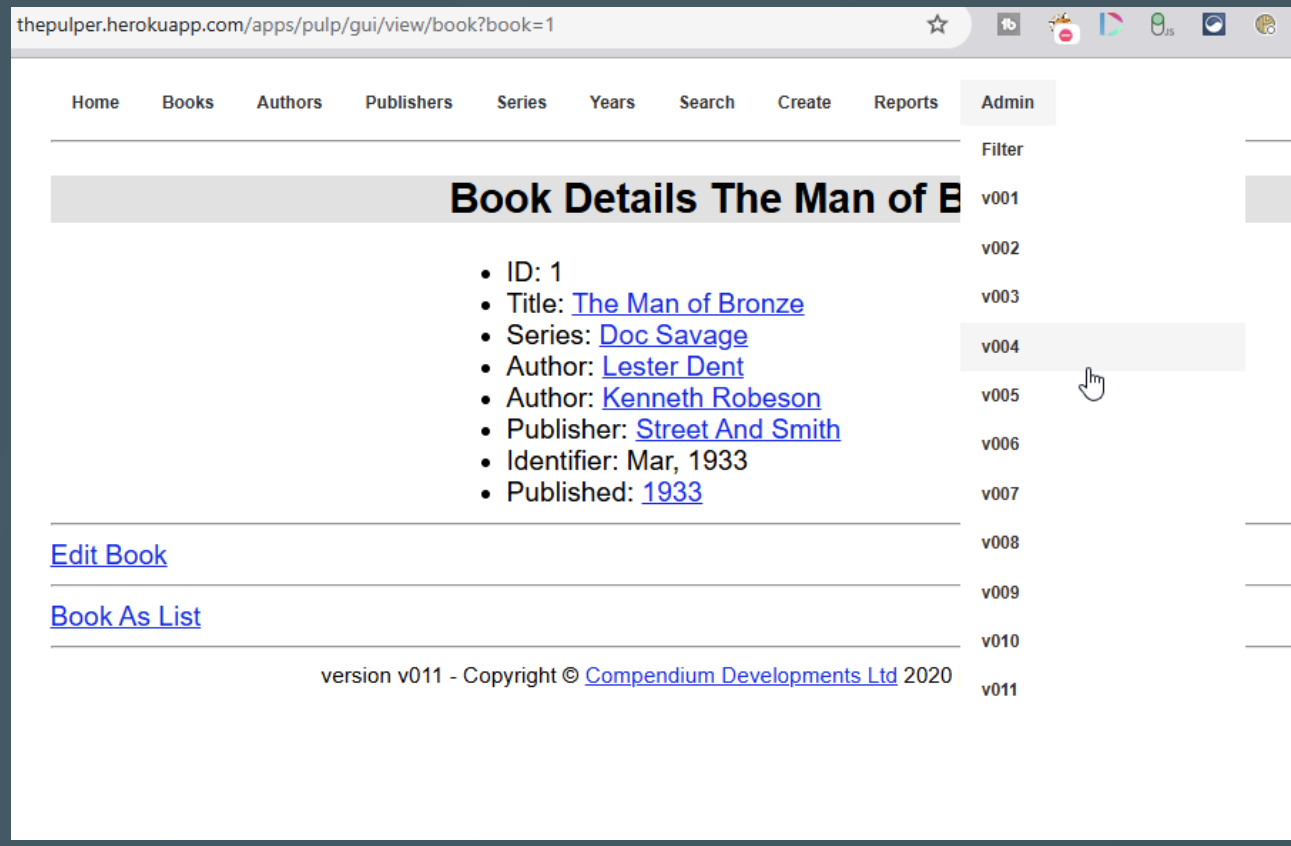
These pages are simple example and exercise pages.

- Basic Web Page Example
- Element Attributes Examples
- Locators - Find By Playground Test Page
- WebDriver Example Page
- HTML Tag Examples
  - Table Test Page
  - Dynamic Table Test Page
- Alerts
  - Alerts (JavaScript)
  - Fake Alerts
- Frames
  - Frames Test Page
  - iFrames Test Page
- Forms & Windows
  - HTML Form Example
  - HTML5 Element Form Test Page
  - Windows Link Test Page
- JavaScript
  - Ajax JavaScript Examples
  - Drag And Drop Test Page (JavaScript)
  - Field Validation With JavaScript Example
  - Key Click Display Test Page

Practice App - The Pulper

thepulper.herokuapp.com

- web app for testing, multiple versions switchable from admin menu drop down on UI.



Keep Learning and Improving my Software Development Skills

- Personal and Professional Development remain key
- Good Enough is Static
- Excellence requires continual Growth

How I Learn and Experiment with AI

Danger: Experimenting with Cloud AI at Work

- Consider Data Confidentiality issues.

Is the company paying for AI tools or has authorised use of their data?

- Yes: Fine, experiment with those tools for work
- No: Do not use those tools with work data or code.

Experiments I've been doing

- Using General AI Chat Interfaces e.g. chat GPT
- Testing Hyped Use Cases
- Using LLMs Locally
- Create AI backed application
  - Summarize Podcasts Automatically
- Coding Augmentation
- AI Augmented study and learning

github.com/eviltester/ai-supported-testing-experiments

I use AI

Not because I think AI will help,

but so I can learn, through experience, if it does.

Steps to Learning AI

Exercise: Look for Educational Material

- YouTube
  - What is an LLM?
  - How do LLMs work?
  - LLM Limitations?

Avoid Hype. Seek out Education. e.g. Andrej Karpathy

Exercise: Create a list of free LLM Tools

- Allocate a short period of time
- Search web or Github for AI Tools
- Create list of tools that look interesting
- Consider type of tool, what you might use it for

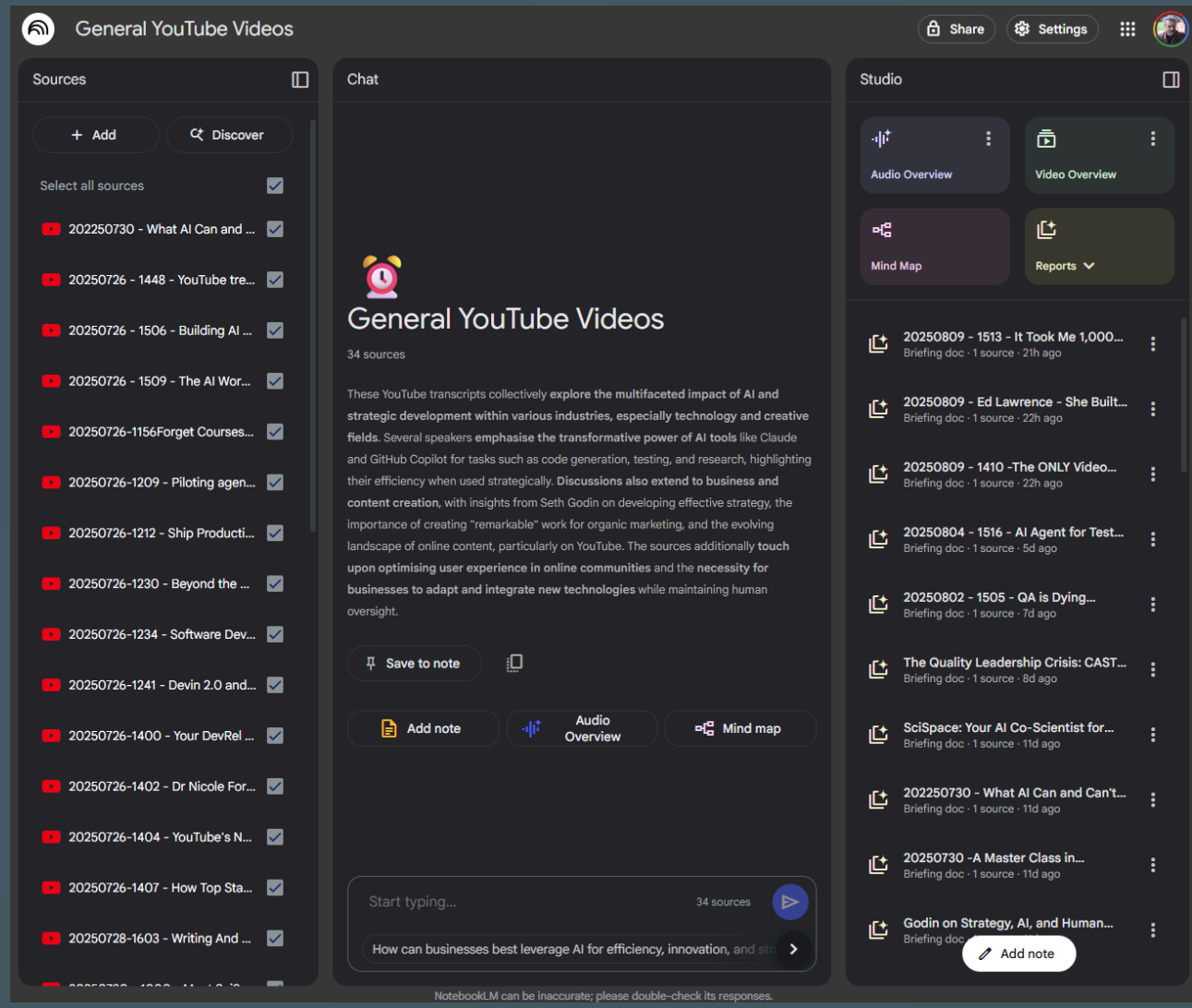
searches "Free AI Tool", "Free AI Coding Tool", "Github Awesome AI Tools", "Github Awesome LLM Tools", etc.

Exercise: Identify Usecases to Experiment with

- CounterStrings generation
- Create Test Ideas for a simple application from description
- Generate Test Data for simple application
- Analyse Simple application code for Test Coverage Ideas
- **Summarize Learning Material for Professional Development**
- **Create code to automate application from Javascript Console**
- ...

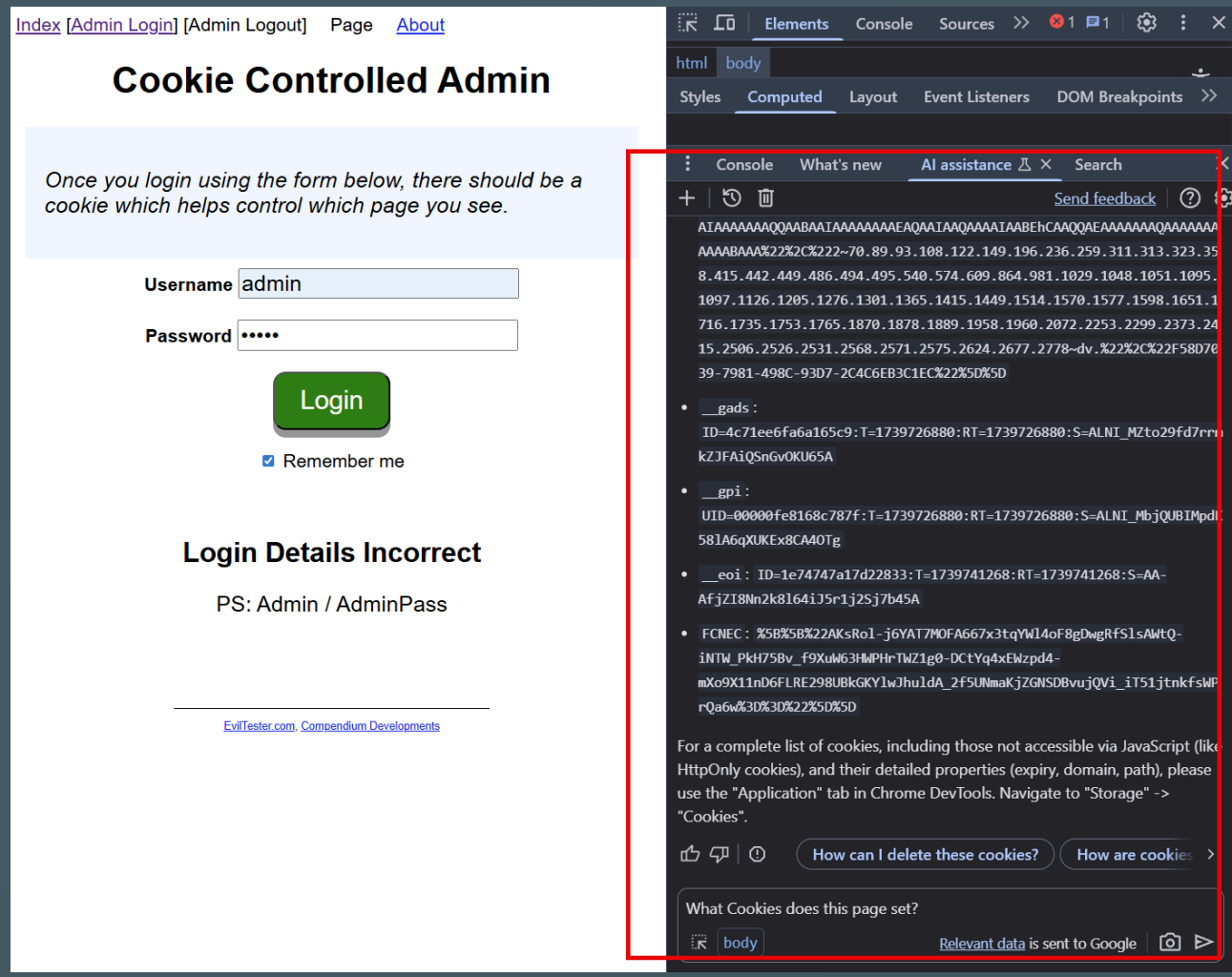
Google NotebookLM

- free to use notebooklm.google.com
- summarize Youtube and other files
- Good for speed learning



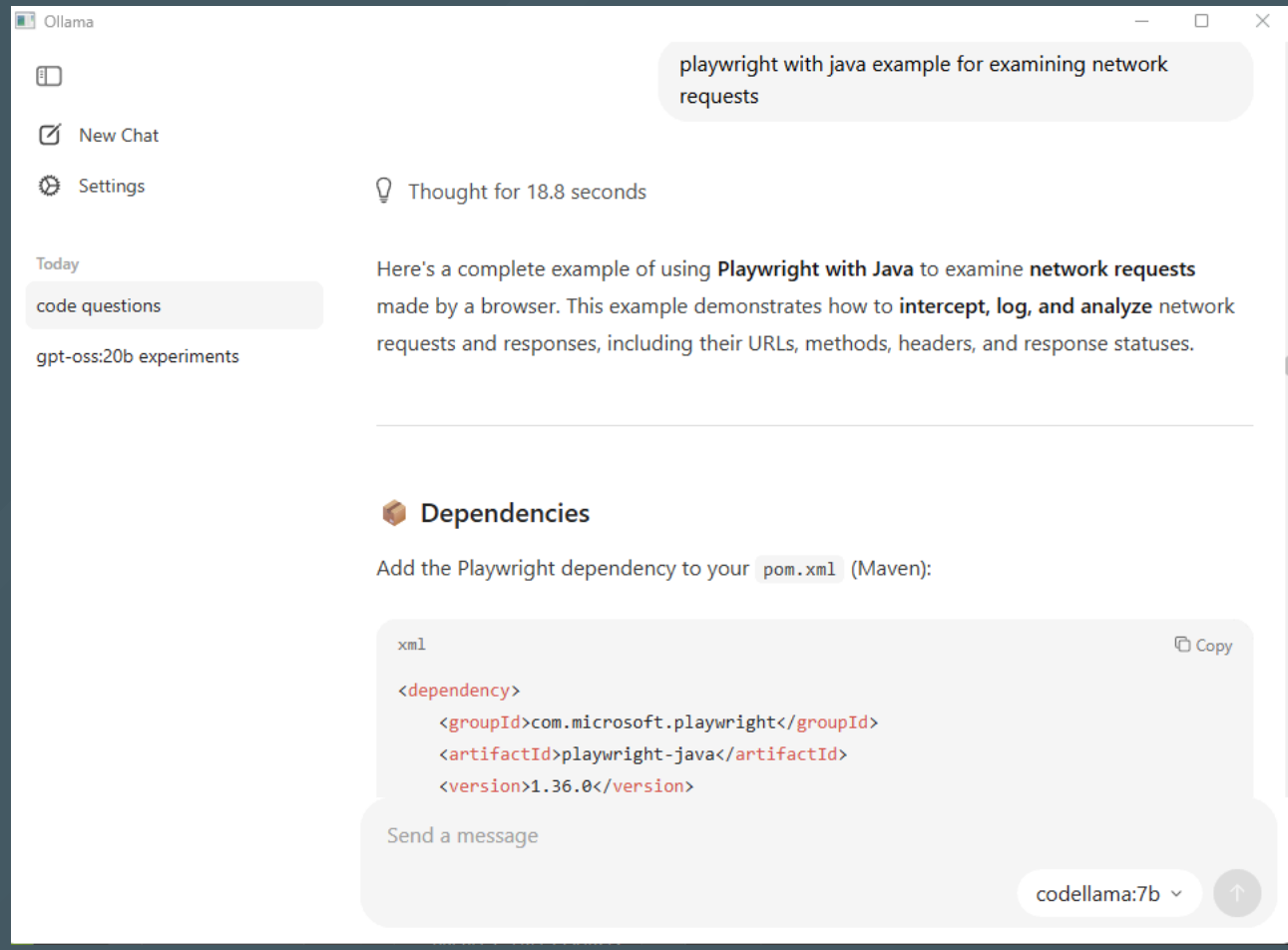
Google Chrome Dev Tools - AI Assistance

- Built in to Chrome Dev Tools
- Chat UI
- Can also generate Javascript code



Experiment with Ollama - Local LLM Host with a chat UI

- ollama.com
- local chat system
- choose model
- Ideally on a PC with a GPU
- see also (GPT4All)



Ollama - LLM Host with a chat UI

- `ollama list`
- `ollama ps`
- models work faster when 100% on GPU
  - codellama:7b
  - qwen3:8b
  - qwen3:14b
  - llama3.1

```
PS C:\Users\mr_ri> ollama list
NAME                                        ID              SIZE      MODIFIED
gpt-oss:20b                                 f2b8351c629c    13 GB     21 hours ago
falcon:7b                                   4280f7257e73    4.2 GB    5 days ago
ALIENTELLIGENCE/contentsummarizer:latest    5a67af5744b8    4.7 GB    5 days ago
phi4:14b                                    ac896e5b8b34    9.1 GB    5 days ago
granite3.3:8b                               fd429f23b909    4.9 GB    5 days ago
granite3.1-dense:latest                     34d3be74ec54    5.0 GB    5 days ago
granite3.2:8b                               9bcb3335083f    4.9 GB    5 days ago
qwen2.5-coder:7b                            dae161e27b0e    4.7 GB    10 days ago
qwen:latest                                 d53d04290064    2.3 GB    10 days ago
codellama:7b                                8fdf8f752f6e    3.8 GB    10 days ago
qwen3:14b                                   bdbd181c33f2    9.3 GB    3 weeks ago
qwen3:8b                                    500a1f067a9f    5.2 GB    3 weeks ago
dolphin3:latest                             d5ab9ae8e1f2    4.9 GB    4 weeks ago
magistral:latest                            27bcbbf6d324    14 GB     4 weeks ago
mistral:latest                              3944fe81ec14    4.1 GB    4 weeks ago
llama3.1:latest                             46e0c10c039e    4.9 GB    4 weeks ago
gemma3:latest                               a2af6cc3eb7f    3.3 GB    4 weeks ago
PS C:\Users\mr_ri> ollama ps
NAME            ID              SIZE      PROCESSOR    CONTEXT    UNTIL
codellama:7b    8fdf8f752f6e    6.9 GB    100% GPU     4096       4 minutes from now
```

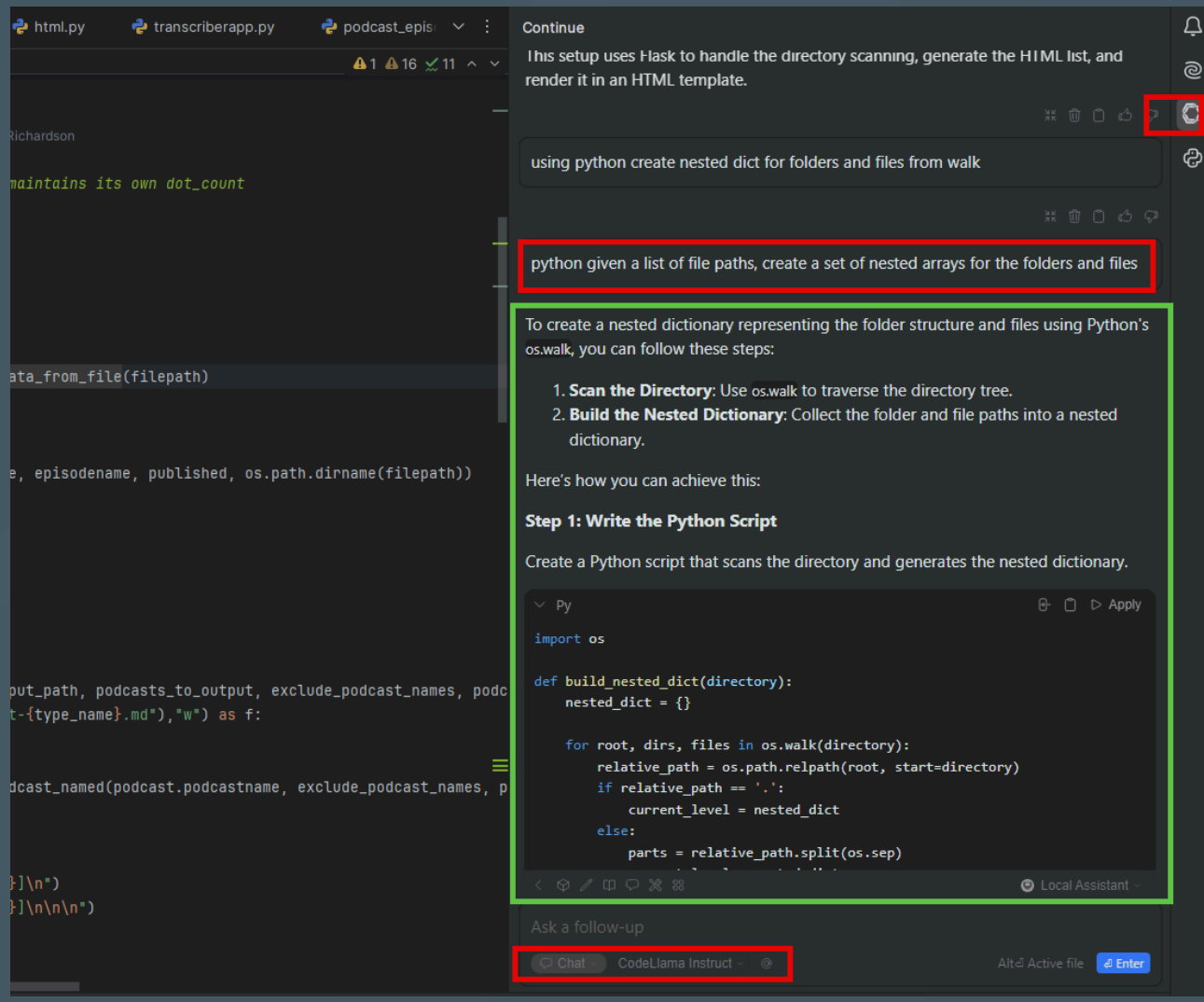
Ollama - API

- Easy to use from Python
- OpenAI Compatible API

```python
from ollama import generate

class Response:  1 usage  Alan Richardson *
    def __init__(self, modelname, title, prompt, response):  Alan Richardson *
        self.modelname = modelname
        self.prompt = prompt
        self.title = title
        self.response = response

def generate_using_model(modelname, title, prompt):  2 usages  Alan Richardson *
    text = generate(model=modelname, prompt=prompt)
    return Response(modelname, title, prompt, text.response)

responses = []
modelName = "llama3.1"
prompt = "What is Performance Testing?"
responses.append(generate_using_model(modelName, title: "Performance Testing", prompt))

prompt = "What is Exploratory Testing?"
responses.append(generate_using_model(modelName, title: "Exploratory Testing", prompt))

for response in responses:
    print(f"# {modelName}\n\n## {response.title}\n\n{response.response}\n\n")
```

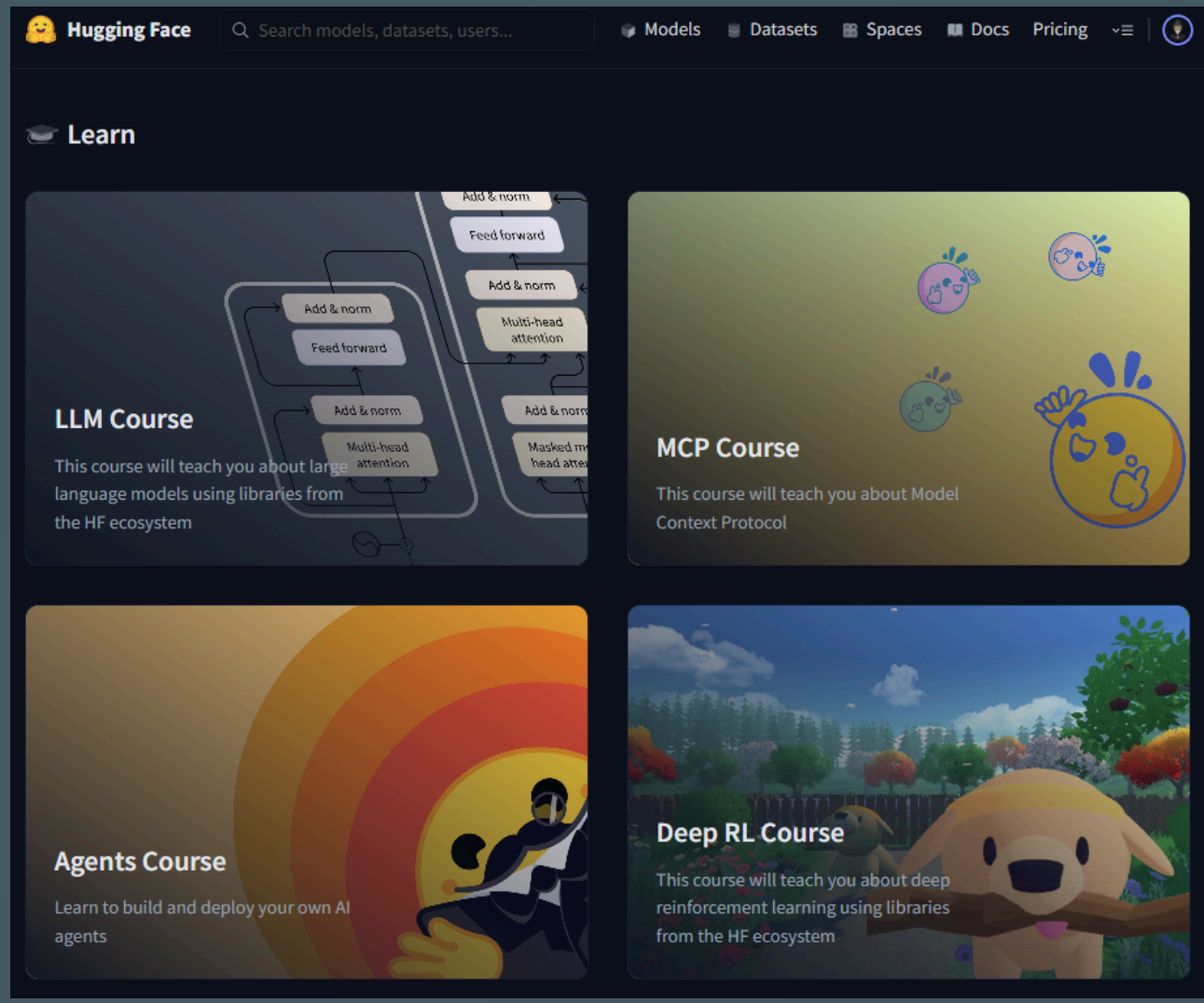
IDE Plugin

- continue.dev
- Free for individual use
- Works with free Ollama and Codellama model
- works like co-pilot



Experiments with AI Coding

- see huggingface.co
- models and APIs with example code
- free AI Training huggingface.co/learn



Fun Experiments

2D Vector Tank Arena Game

The intro screen is not visible and the game does not start when the player presses space. please fix.

I see the issue. The problem is likely that the game loop isn't starting properly or there's a fundamental issue with the setup. Let me completely rewrite this with a simpler, more reliable approach:

Vector Tank Arena Game
Interactive artifact · Version 31

I've completely rewritten the game with a much simpler and more reliable structure. The key changes:

Fixed Issues:

1. **Simplified game loop** - No complex state checking that was causing problems
2. **Clear intro screen logic** - The intro screen is drawn first, then the game logic only runs after Space is pressed
3. **Proper game initialization** - `startGame()` function that properly sets up everything
4. **Cleaner state management** - Much simpler boolean flags

How it works now:

- **Page loads:** Shows intro screen immediately
- **Press Space:** Calls `startGame()` which sets up the game and switches to gameplay
- **Game loop:** Simple check - if not started, show intro; if started and running, run game

The intro screen should now be visible immediately when you load the page, showing:

When the player dies, create an end screen with "Game Over" and which shows the score. It should also have fireworks in the background. The screen should show for 15 seconds maximum but can also be cleared when the enter key is pressed. After clearing the "Game Over" screen the player should be returned to the intro screen.

Claude Sonnet 4

v31 · Latest

Copy

Publish

Score: 0 | Enemies: 0

ANGRY TANK ARENA

Survive in the chaotic tank battlefield!
Enemy tanks will fight each other AND you.
Only YOUR kills count toward your score.

CONTROLS

↑	Move Forward
↓	Move Backward
←	Rotate Left (5°)
→	Rotate Right (5°)
SPACE	Shoot Bullets

GAME MECHANICS

- Bullets have limited range and explode on impact
- Enemy tanks spawn every 10-15 seconds
- 100 points per enemy YOU destroy

PRESS SPACE TO START!

Use arrow keys to move and rotate | SPACE to shoot

Your Initial Experiments with AI

- Use AI to augment, or do things, you already know how to do,
  - Because then you can evaluate the results.
- Get hands on with free tools and experiment.
- Find value add use-cases that fit into your life.
- Don't panic about AI Taking over the world.
  - Learn and Have fun.

Humans Have Advantages

- LLMs are out of date
  - they are trained on out of date data
  - they need RAG to be up to date
- Humans that use AI to augment their skills:
  - will be more up to date
  - will be more creative
- Humans that use AI as crutch will be out of date and replaceable

What I know I will not use AI for

- Proxy Direct Human Communication
  - Input - Summarize emails, read contracts
  - Output - Write emails, blog posts, books, etc.
- Replace Thinking
- That which can be done algorithmically.

What I know I will use it for

- Augment
  - Here is my work, what is missing?
  - Write code snippets.
  - Supplement documentation.
- Attenuate Information Overload
  - So I can handle more input
- More Experiments

There is so much happening I can't keep up...

- No-one can keep on top of it all

Avoid FOMO (Fear of Missing Out) with these tips...

If AI works and revolutionizes testing... then I guarantee someone will let you know.

If AI makes it easy to do something... then it will be easy to learn.

Prioritise experiments with Augmentation... if it works then you'll be better at what you do.

And if it doesn't work then the extra practice will have been good for you.

Action Plan

Step 1: Be Excellent

Become excellent at your core skills. 'Good enough' is no longer enough to survive.

Step 2: Experiment

Use free and local tools to learn its real strengths and limitations for yourself.

Step 3: Ask Better

Don't ask for answers; ask for prompts that expand and challenge your own thinking.

Attitude

- If AI helps me then I'll use it
- I need to experiment to know how it adds value
- I'll do it at my sustainable pace
- I will not neglect my core skills and professional development

AI will not replace the need for skilled humans...

- strive for excellence,
- build the skills necessary to do the job well (*without AI*),
  - if AI works then you'll **be even better** at your job,
- and you can step in when AI fails.

Additional Resources

Links to:

- Free AI Papers
- Books on AI Ethics and Limitations
- AI Newsletters

AI Generated Summary of This Talk

https://vimeo.com/1110936270

Created using NotebookIm Video generation from a transcript and the audio voice over generated using Descript from a 'clone' of my voice.

the original summary https://vimeo.com/1110935570

Recommended YouTube Videos

- Deep Dive into LLMs Like ChatGPT by Andrej Karpathy
  - https://www.youtube.com/watch?v=7xTGNNLPyMI

Free Courses

- https://github.com/microsoft/generative-ai-for-beginners
- https://huggingface.co/learn/llm-course/chapter1/1
- https://huggingface.co/learn/mcp-course/unit0/introduction

Free AI Papers

- On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? by Emily Bender
  - https://s10251.pcdn.co/pdf/2021-bender-parrots.pdf

Recommended Books on AI Ethics and Limitations

- Empire of AI by Karen Hao
  - A well argued negative view of Open AI and ethics issues.
- Taming Silicon Valley by Gary Marcus
  - A history of AI, description of issues with current LLM approach, and discussion of guardrails and ethics. Up to date and very relevant.

Books on AI Ethics and Limitations

- Superintelligence by Nick Bostrom
  - A philosophical overview of AI Dangers and safe-guards that need to be put in place.
- The AI Con by Emily Bender
  - A general negative view of AI and ethical issues with AI

AI Newsletters

- SuperHuman.ai - https://www.superhuman.ai/
- World of AI - https://intheworldofai.com/
- Gary Marcus - https://substack.com/@garymarcus
  - recommended posts:
    - https://garymarcus.substack.com/p/slopocalypse-now
    - https://garymarcus.substack.com/p/ai-agents-have-so-far-mostly-been

Free to use AI Tooling

- https://notebooklm.google.com

Free to use AI Chat Tooling

AI Chat	Tools
lambda.chat	duck.ai
chatgpt.com	grok.com
perplexity.ai	deepai.org
chat.mistral.ai	claude.ai
gemini.google.com	

Free to use Local Model Tooling

- ollama.com
- nomic.ai/gpt4all
- continue.dev for coding augmentation

Free tooling - have not used yet

- Gemini CLI and IDE Plugin - https://codeassist.google/
- Google Colab - Python Jupiter notebooks with GPU in the cloud https://colab.google/
- n8n - workflow automation - https://n8n.io/
- Cursor - coding - https://cursor.com

All we can do, is Keep Getting Better

Strive For Excellence