

Skills

1 ⚭ Frontend Sorcery ★★★★☆



=> **Frontend** is one of my strongest spells. I wield it to craft `user interfaces (UI)` and enchant `user experiences (UX)`, using many languages and tools across the realm of webcraft.

Frontend Sorcery is the noble art of turning raw elements—HTML, CSS, and JavaScript—into living, breathing pages. With this magic, I forge responsive designs, interactive runes, and smooth pathways for every traveler who visits.

Description

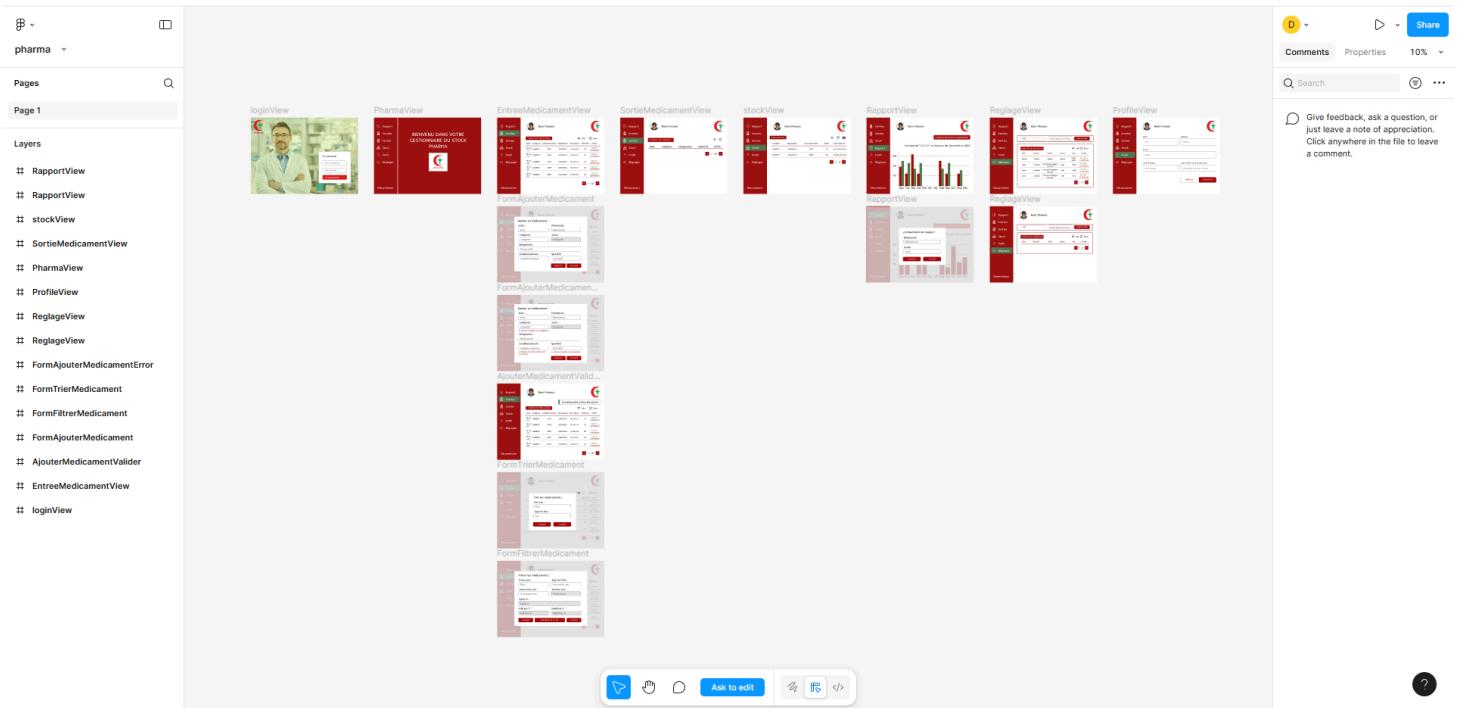
Frontend is the power to shape designs into real, working code—like turning scrolls into living portals. I use different languages and sometimes ancient frameworks to summon full user experiences.\nAs this magic is rooted in webcraft, mastering the scrolls of `HTML` and `CSS` and knowing the basics of `JavaScript` is essential. From these, other tools arise—like the mighty `React` (both for web `JS` and mobile `Native`), and spells such as `Vue`, `Next`, and `Electron`.\\nThis path focuses on the

User Experience (ux) . A skilled frontend mage must know how to build responsive layouts and summon charming User Interfaces (UI) that work well across any device or kingdom.

Gallery of Artifacts

The screenshot shows a mobile application interface designed in a UI tool. The top navigation bar includes a logo, a search icon, and a menu icon. The main content area features a header with the text 'Page Name' and a back arrow. Below the header is a footer with icons for home, search, and more. The central part of the screen displays a grid of five cards, each containing an icon and Arabic text. The cards are arranged in two rows: the first row has two cards ('كریم (سمع)' and 'حدث شریف'), and the second row has three cards ('ادعیة', 'ارکانی', and 'قرآن کریم (صبری)'). The UI tool's sidebar on the left lists components like 'Logo list', 'header', 'navigation', 'card', 'footer', and 'link'. The right sidebar shows page settings, color palette, variables, styles, and export options.

The screenshot shows a desktop application interface designed in a UI tool. The left sidebar lists 'Gestion de stock' and 'Drafts Free'. The main area features a 'CATEGORY LIST' table with three columns. Each row contains product details: NAME (e.g., 'NAME'), BOX (e.g., '12DH'), PRICE (e.g., '12DH'), QUANTITY (e.g., '20'), and actions (e.g., 'DELETE', 'SHOW'). Below the table is a 'Create Product' form with fields for Product name, Warehouse, Price, Image, Description, Category, and Quantity. At the bottom are 'RESET' and 'ADD' buttons. The UI tool's sidebar on the left lists 'Pages' (Page 1), 'Phone', and 'Desktop'. The right sidebar shows page settings, color palette, variables, styles, and export options.



Fichiers Redimensionner Retouche GAME DESIGN DOCUMENT (GDD) ⌂ m + ⌂

Design

Éléments

Texte

Marque

Importer

Outils

Projets

Applis

Media mag... Créer en bloc

Notes

Page 1 - GAME DESIGN DOCUMENT (GDD)

GAME DESIGN DOCUMENT (GDD)

Game Name : Level Up **Genre :** Rpg, simulation

Core Idea: A modern world RPG simulation where players level up, engage with monsters, and discover new dimensions.

Target Audience: Casual gamers interested in exploration, combat, and character progression.

Art Style : cyberpunk, Vibrant

Levels : gain exp from campaign and exp pill
*EXP unlocks new abilities, areas, and dimensions.

Story:
In the world of Harmony, Roven wakes up to find himself in a lush forest, surrounded by strange flora and fauna he's never seen before. As he starts to explore, he encounters NPCs who seem to be aware of the system of levels that govern their world.
Roven soon discovers that he is at level 1 and that each level represents a milestone in his growth and understanding of the world. As he interacts with NPCs and completes quests, he begins to uncover his weaknesses and strengths.

One of Roven's first challenges is a quest to deliver a group of bonds, kidnapping a nearby village. As he sets off, he finds himself facing challenges that he believes his combat skills are lacking and that he is easily overpowered. Through this experience, Roven learns that his weakness lies in his lack of combat training. Determined to improve, he seeks out a martial arts master who agrees to train him in exchange for completing tasks around the village.

As Roven trains and improves his combat skills, he also begins to understand the intricacies of the world around him. His improved physical abilities have forced him to improve his physical abilities and mental and spiritual ones.

With each level he gains, Roven becomes stronger and more adept at navigating the world of Harmony. Along the way, he discovers secrets about his past life and the true nature of the world he now inhabits.

GAME DESIGN DOCUMENT

Page 2 - GAME DESIGN DOCUMENT (GDD)

GAME DESIGN DOCUMENT (GDD)

59 % Pages 1/3 ⌂ ?

```
/* Cinzel-VariableFont_wght */
@font-face {
    font-family: 'Cinzel';
    font-style: normal;
    font-weight: 400;
    font-display: swap;
    src: url(./fonts/Cinzel/Cinzel-VariableFont_wght.ttf) format('ttf');
}

/* MedievalSharp-Regular */
@font-face {
    font-family: 'MedievalSharp';
    font-style: normal;
    font-weight: 400;
    font-display: swap;
    src: url(./fonts/MedievalSharp-Regular.ttf) format('ttf');
}

/* cyrillic-ext */
```

✨ Known Spells

- Responsive Layouts
- Cross-Browser Shielding
- Speed & Performance Buffs
- Accessibility Enchantments
- Git for Spell Versioning

📦 Tools of the Trade

- VS Code (my spellbook)
- Browsers: Chrome, Firefox, Edge (testing portals)
- Figma (design forge)
- Canva (visual scroll maker)

📚 Languages & Frameworks

◆ Core Runes

- HTML
- CSS
- JavaScript

◆ Arcane Support

- React js
- React native
- Vue js
- Laravel .blade.php
- Next js
- Electron
- Scss
- Bootstrap
- Tailwind CSS

Origin Story

My path into frontend sorcery began when I first wondered how scrolls (webpages) came to life. I began with simple `HTML`, then studied `CSS` to bring form and `JavaScript` to add movement. Over time, through many quests and trials, I learned to craft real experiences that users enjoy.

Each project sharpened my skills, and now, I continue this journey—seeking to master deeper magic and forge even stronger interfaces.

2 ⚒ Backend Alchemy ★★★★



=> **Backend** is the unseen force behind a website. While the frontend enchants users with visuals and interaction, the backend works in the shadows—handling servers, storing data, and forging powerful APIs.

Backend alchemy is the sacred art of turning logic and data into real-world applications. Using tools like Node.js, Python, and SQL, backend alchemists build server-side magic, manage spellbooks (databases), and keep the realm running smoothly.

Backend Description

Backend is the power to craft logic, store information, and shape how data flows through an application. It lets you build functions and APIs that store, update, or fetch data from enchanted vaults—known as databases.

These vaults can be **relational** (like `SQL`, `Oracle`, `SQLite`) or **non-relational** (like `MongoDB`, `Firebase`, or `PostgreSQL`).

Backend mages often carry tools like `XAMPP` or `WAMP` to summon local servers, `Postman` to test API scrolls, and `Atlas` to gaze into the MongoDB cloud.

The common tongues of backend magic include PHP, JavaScript, and Python. Each has its own schools—Laravel, CodeIgniter, and Django—while Express.js and Next.js serve both backend and frontend domains.

Backend Gallery of Artifacts

The screenshot shows the Postman application interface. On the left, there's a sidebar with various sections like 'My Workspace', 'Collections', 'Environments', 'Flows', 'APIs', 'Mock servers', 'Monitors', and 'History'. The main area displays an API documentation for 'User / Get authenticated user / Successful Response'. It shows a 'GET' request to 'https://api.getpostman.com/me'. Below the request, there are tabs for 'Params', 'Headers', and 'Body'. Under 'Query Params', there's a table with columns 'Key' and 'Value'. The 'Body' tab is selected, showing a JSON response with a status code of 200 OK. The JSON data is as follows:

```
1 {  
2   "user": [  
3     {"id": 12345678,  
4      "username": "taylor-lee",  
5      "email": "taylor.lee@example.com",  
6      "fullName": "Taylor Lee",  
7      "avatar": "https://example.com/user/r5u9qpvmuifjif6lbqmga.jpg",  
8      "isPublic": true  
9    },  
10   "operations": [  
11     {"  
12       "name": "mock_usage",  
13       "limit": 1000000,  
14       "usage": 11276,  
15       "overage": 0  
16     },  
17     {"  
18       "name": "monitor_request_runs",  
19       "usage": 1000000  
20     }  
21   ]  
22 }  
23 
```

This screenshot shows a simplified interface for selecting API methods. At the top, there's a dropdown menu with 'GET' selected and a URL field with 'https:' followed by a placeholder 'Type a new method'. Below this, there are five large, rounded rectangular buttons with colored text: 'GET' (green), 'POST' (orange), 'PUT' (blue), 'PATCH' (purple), 'DELETE' (red), 'HEAD' (teal), and 'OPTIONS' (pink). A horizontal line separates these from a text input field at the bottom containing the placeholder 'Type a new method'.



Wampserver 64bit

Start services

Be patient !

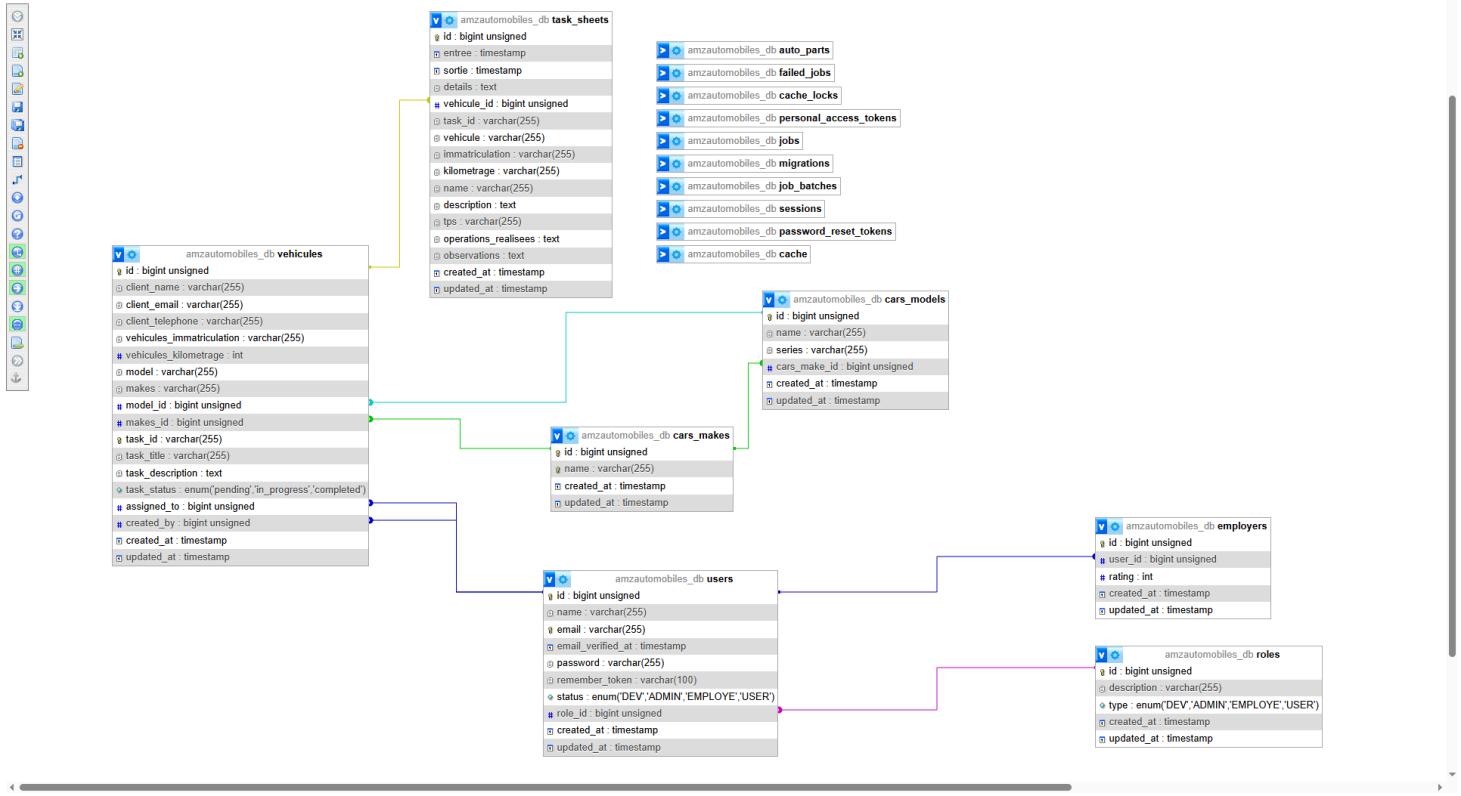


The screenshot shows the Wampserver interface with two main panes. The left pane is titled 'MariaDB 11.5.2' and contains the following items:

- Version
- Service administration 'wampmariadb64'
- MariaDB settings
- MariaDB console
- Files & Documentation
 - my.ini
 - MariaDB log (26.4 KiB)
 - MariaDB documentation
 - Port used by MariaDB : 3307
 - Use a port other than 3307
 - Test MariaDB port used: 3307
 - Default DBMS: MySQL 9.1.0
 - Help -> MariaDB - MySQL

The right pane is titled 'Wampserver - 3.3.7 - 64bit' and contains the following items:

- localhost
- PhpMyAdmin
- Adminer 4.8.4
- Your VirtualHosts
- Apache 2.4.62.1
- PHP 8.3.14
- Default DBMS: mysql
 - MySQL 9.1.0
 - MariaDB 11.5.2
- Help MariaDB - MySQL
- Services
 - Start All Services
 - Stop All Services
 - Restart All Services



The screenshot shows the `amzautomobiles_db` structure in phpMyAdmin:

- Tables:** auto_parts, cache, cache_locks, cars_makes, cars_models, cars, failed_jobs, jobs, job_batches, migrations, password_reset_tokens, personal_access_tokens, roles, sessions, task_sheets, users, vehicles.
- Summarized Table Data:**

Action	Rows	Type	Collation	Size	Overhead
auto_parts	Empty	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
cache	Empty	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
cache_locks	Empty	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
cars_makes	116	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
cars_models	2,157	InnoDB	utf8mb4_unicode_ci	224.0 Kib	-
cars	Empty	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
failed_jobs	Empty	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
jobs	Empty	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
job_batches	Empty	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
migrations	11	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
password_reset_tokens	Empty	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
personal_access_tokens	Empty	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
roles	4	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
sessions	Empty	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
task_sheets	1	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
users	5	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
vehicles	1	InnoDB	utf8mb4_unicode_ci	80.0 Kib	-
Sum	2,295	MyISAM	utf8mb4_0900_ai_ci	624.0 Kib	0 B
- Create new table:** A form is available to create a new table with 4 columns.

Backend Known Spells

- RESTful API Development
- Database Management (SQL, NoSQL)
- Authentication and Authorization
- Server-Side Logic Implementation

- Performance Tuning and Optimization

Backend Tools of the Trade

- Postman (for scroll testing)
- WAMP / XAMPP (local server summoning)
- MongoDB Atlas (cloud vault access)
- PHPMyAdmin
- MySQL Workbench — GUI for managing MySQL databases.
- MongoDB Compass — GUI for working with MongoDB (NoSQL).

Backend Languages & Frameworks

◆ Backend Core Runes

- PHP
- JavaScript
- SQL
- Python
-  CLI TOOLS
 - npm
 - composer

◆ Backend Arcane Support

- Laravel
- CodeIgniter
- Express.js
- Next.js
- Django
- MySQL
- SQLite
- MongoDB
- Firebase
- Axios

Backend Origin Story

After many adventures in frontend lands, I realized something was missing—my websites had form, but no soul. I needed more than just beauty; I needed function.

That's when I discovered backend alchemy. My first steps were simple: storing data in JSON using JavaScript. It wasn't elegant, but it worked for small quests.

As my projects grew in size and complexity, I sought deeper magic. I began to learn how to summon servers, craft powerful APIs, and master databases. Now, I walk both paths—frontend and backend—capable of building full systems from the ground up.

3 🛡 Security Wards ★



=> **Security** is the art of shielding systems from the dark forces of cyberspace. With this skill, I deploy firewalls, encryption, and secure protocols to protect users, data, and digital strongholds.

Security Wards focuses on forging and maintaining defenses against both common and advanced threats. From encryption and firewalls to OWASP defenses and secure authentication, these techniques help keep the realm safe from intruders.

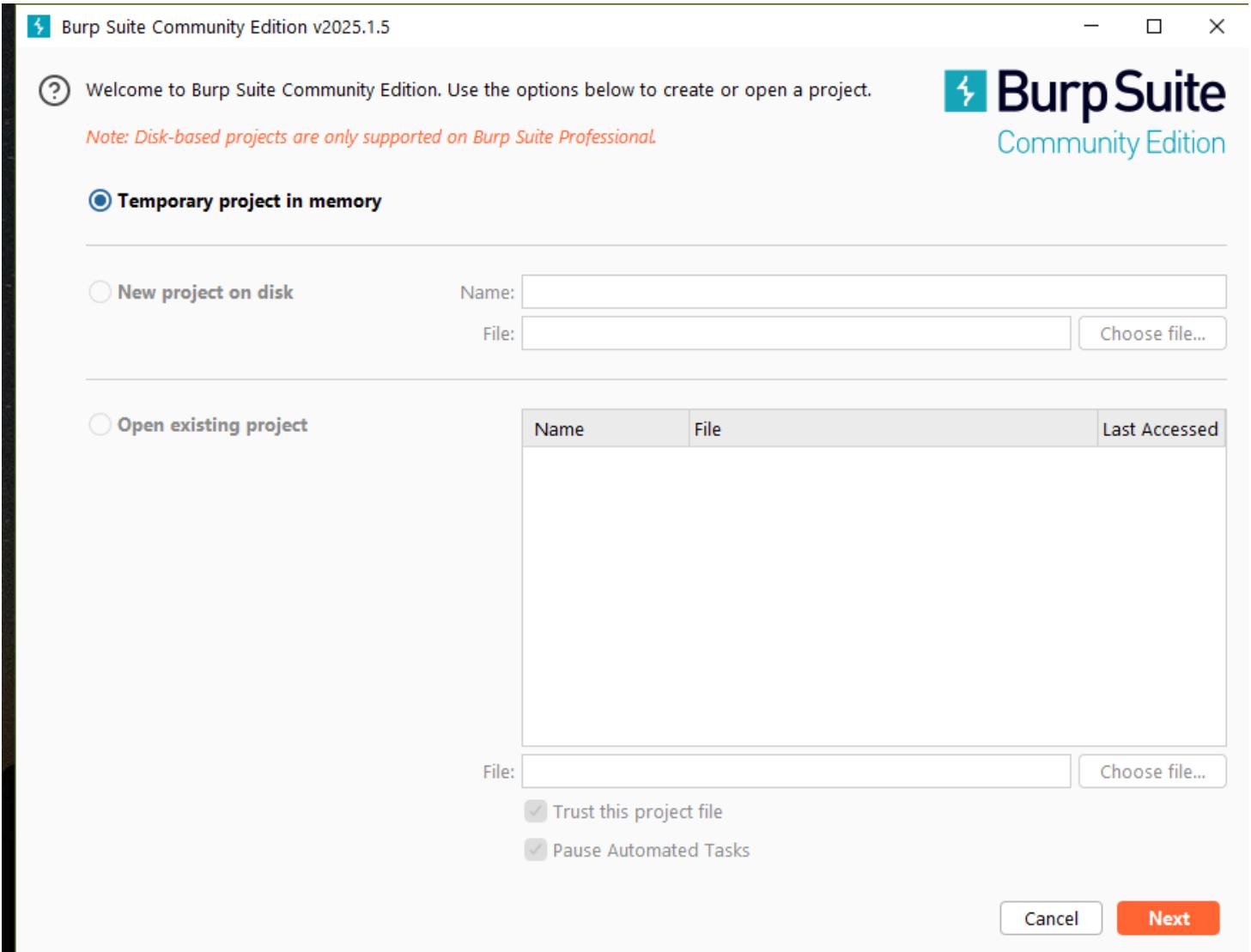
📋 Security Description

Security is about crafting strong barriers—both visible and hidden—against digital attacks. It involves defending systems with tools like firewalls, encrypted tunnels (SSL/TLS, PGP), and access control spells.

In my journey, I've trained with different tools. Some help me test my own defenses through controlled attacks ([Burp Suite](#) , [Nmap](#) , [Wireshark](#) , [Zenmap](#)). Others guide me through challenges and learning quests—like [TryHackMe](#) and [PortSwigger](#) .

While these arts can be practiced on any platform, I find their magic shines brightest on Linux systems—at least for me.

Security Gallery of Artifacts



Burp Suite Community Edition v2025.1.5

Welcome to Burp Suite Community Edition. Use the options below to create or open a project.

Note: Disk-based projects are only supported on Burp Suite Professional.

Temporary project in memory

New project on disk

Name:

File: Choose file...

Open existing project

Name	File	Last Accessed

File: Choose file...

Trust this project file

Pause Automated Tasks

Cancel **Next**

Burp Project Intruder Repeater View Help

Burp Suite Community Edition v2025.6.5 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

Tasks New scan New live task ⚡ ⓘ ⓘ ⓘ

Filter Search

1. Live passive crawl from Proxy (all traffic) ⓘ :

Add links. Add item itself, same domain and URLs in suite scope.

Capturing

Items added to site map [View site map](#)

Host	Method	URL	Status c...	MIME type
 No items to show				
Items found in the crawl will display here.				

Task configuration [View configuration](#)

Task type: Live passive crawl
Scope: Proxy (all traffic)
Configuration: Add links. Add item itself, same domain and URLs in suite scope.
Capturing

Task progress

Site map items added: 0
Responses processed: 0
Responses queued: 0

Task log

Event log All issues Memory: 132.3MB Disabled

Kleopatra

File View Certificates Tools Settings Window Help

Sign/Encrypt Decrypt/Verify Import Export Certify Lookup on Server Certificates Notepad Smartcards Groups

Welcome to Kleopatra 3.3.0 (Gpg4win-4.4.0)

Kleopatra is a front-end for the crypto software [GnuPG](#).

For most actions you need either a public key (certificate) or your own private key.

- The private key is needed to decrypt or sign.
- The public key can be used by others to verify your identity or encrypt to you.

You can learn more about this on [Wikipedia](#).

New Key Pair Import

Zenmap interface showing a scan report for localhost (127.0.0.1). The Nmap Output tab is selected, displaying the following results:

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-31 13:00 Morocco Daylight Time
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00056s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
445/tcp    open  microsoft-ds
1001/tcp   open  webpush
7070/tcp   open  realserver

Nmap done: 1 IP address (1 host up) scanned in 0.50 seconds
```

Security Known Spells

- Encryption Techniques (PGP, SSL/TLS)
- Authentication & Authorization Shields
- Firewall and VPN Configurations
- Threat Detection and Incident Response

Security Tools of the Trade

- burp Suite
- Nmap / Zenmap
- kleopatra (certificate encrypt)
- Wireshark
- [cryptool](#)
- TryHackMe / PortSwigger (training camps)

Security Languages & Frameworks

| ♦ Security Core Runes

- Python
- RSA
- Shell Scripting

| ♦ Security Arcane Support

- Python
- bash scripting
- Linux Networking Commands (`iptables` , `netstat` , `tcpdump`)

...

Security Origin Story

This skill didn't appear clearly at first. In my early days of frontend and backend development, I followed some protective practices without knowing their true nature.

Over time, I began to notice a pattern—secure code, cautious logic, clean structure. Then came a turning point: studying `Système informatique et réseaux` `computer systems and networks`, where I met allies who shared a deep interest in cybersecurity.

Together, we explored new lands of knowledge—penetration testing, network defense, and encryption. Thanks to those companions, I began to truly understand and develop this craft.

4 🌟 AI Conjuring ★★★



=> **AI Conjuring** is the craft of communicating with intelligent entities through carefully forged prompts and guided interaction.

With words as my spells, I awaken machines to assist, advise, and create.

✍️ AI Description

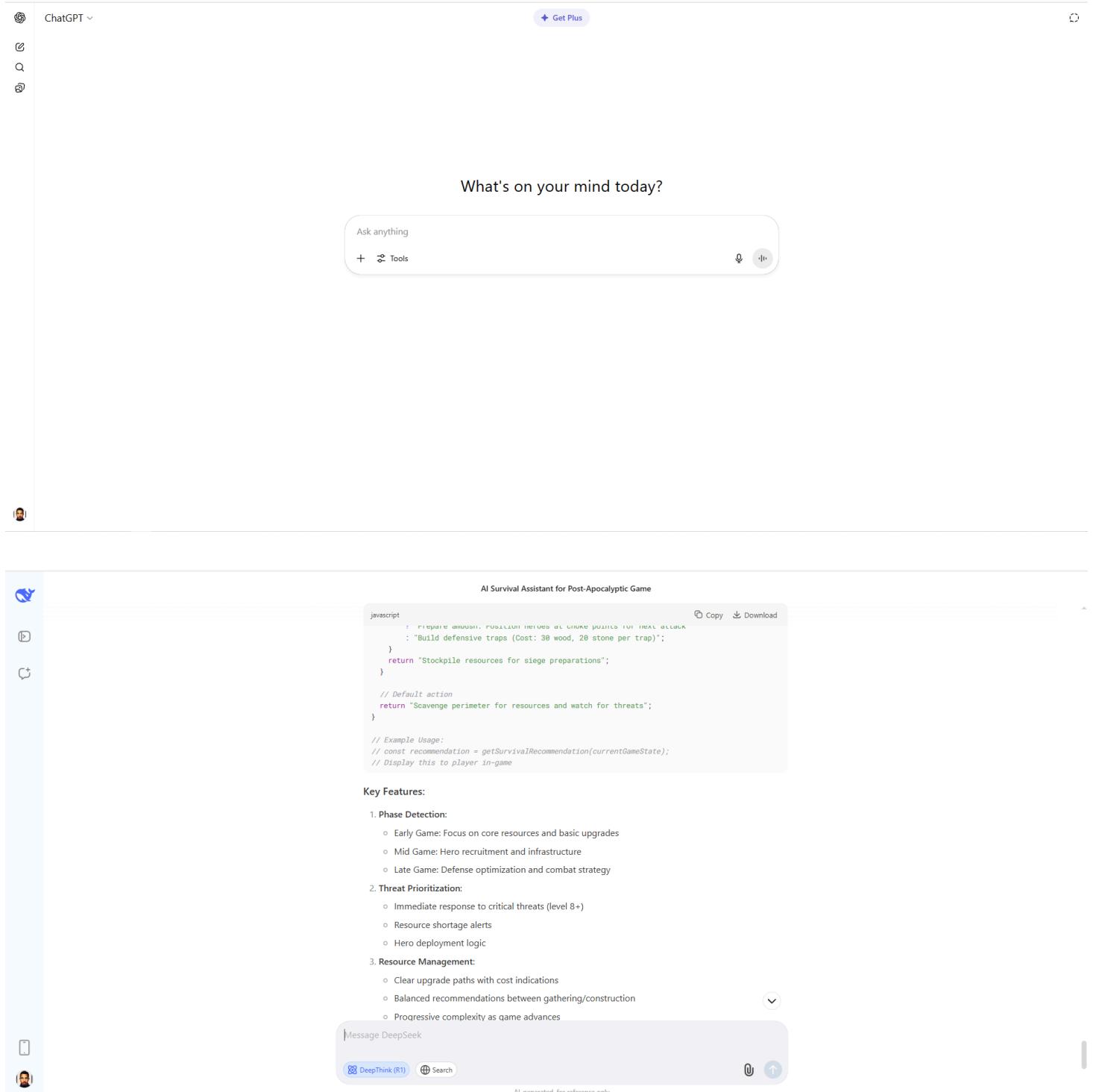
I used my knowledge of artificial intelligence to **build the chatbot featured at the top of my portfolio**, blending both technical understanding and prompt mastery.

This skill focuses on **interacting with AI-powered entities**, crafting prompts, and guiding intelligent systems to respond with clarity and purpose. But it doesn't stop there—my background also includes **hands-on experience with machine learning**, data processing, anomaly detection, and building custom models using tools like Scikit-learn and TensorFlow.

In a world where AI is expanding its domain, **AI Conjuring** allows me to both **command existing intelligence through precise interaction**, and **forge new capabilities through code and data**.

With this dual ability, I can adapt to any challenge—either by summoning insight through prompts or crafting solutions with algorithms.

AI Gallery of Artifacts



The screenshot displays the AI Conjuring interface, featuring a top navigation bar with icons for profile, ChatGPT, Get Plus, and a refresh symbol. Below this is a sidebar with icons for profile, Q, and a gear. The main area has a placeholder text "What's on your mind today?". A search bar with "Ask anything" and a "Tools" button is present. The bottom half shows a code editor titled "AI Survival Assistant for Post-Apocalyptic Game" containing JavaScript code. The code includes functions for preparing defenses, building traps, and stockpiling resources. It also defines a default action for scavenging and provides example usage. Below the code is a "Key Features:" section with three numbered points: 1. Phase Detection, 2. Threat Prioritization, and 3. Resource Management, each with corresponding bullet points. At the bottom is a message input field "Message DeepSeek" with "DeepThink (R1)" and "Search" buttons, and a note "AI-generated, for reference only".

```
javascript
    : "Prepare ambush, position heroes at choke points for next attack
    : "Build defensive traps (Cost: 30 wood, 20 stone per trap)";
}
return "Stockpile resources for siege preparations";
}

// Default action
return "Scavenge perimeter for resources and watch for threats";
}

// Example Usage:
// const recommendation = getSurvivalRecommendation(currentGameState);
// Display this to player in-game
```

Key Features:

1. Phase Detection:
 - o Early Game: Focus on core resources and basic upgrades
 - o Mid Game: Hero recruitment and infrastructure
 - o Late Game: Defense optimization and combat strategy
2. Threat Prioritization:
 - o Immediate response to critical threats (level 8+)
 - o Resource shortage alerts
 - o Hero deployment logic
3. Resource Management:
 - o Clear upgrade paths with cost indications
 - o Balanced recommendations between gathering/construction
 - o Progressive complexity as game advances

Message DeepSeek

DeepThink (R1) Search

AI-generated, for reference only

Build an Offline Desktop Chess App Like Chess.com

Continuing debugging and fixing the analysis feature.

- Editing file chess-app-fixed/js/app.js
- Browsing file:///home/ubuntu/chess-app-fixed/index.html
- Viewing console

Continuing debugging and testing the analysis feature, focusing on Stockfish initialization and analysis event triggers.

- Clicking element
- Viewing console

Continuing debugging and testing the analysis feature, focusing on Stockfish initialization and event triggers.

- Viewing console

Resolve Stockfish worker CORS issue for offline functionality. ↗

Updating plan to add a step for resolving Stockfish worker CORS issue in offline mode.

manus

I've identified the issue with the analysis and review features. The problem is related to browser security restrictions when running the app from a local file.

The Stockfish chess engine is being loaded as a Web Worker, but browsers block Web Workers from loading via the 'file:///' protocol due to security restrictions (CORS policy). This is why the analysis feature isn't working.

I'll fix this by:

1. Modifying the code to load Stockfish as a regular script instead of a Web Worker
2. Implementing a fallback mechanism for the analysis feature
3. Fixing the review feature to work properly with the updated code

Send message to Manus

Manus's Computer

Manus is using Browser

Clicking element

file:///home/ubuntu/chess-app-fixed...

Advanced Chess App

Settings 0.0

Game Mode Time Settings Jump to live

0.0

Debug and fix the analysis feature.
Manus has stopped

Qwen3-235B-A22B-2507 ✓

< 1/4 >

A collage of icons representing key monitoring targets: servers, databases, networks, applications, hardware components (CPU, memory, disk), and business processes. Arrange the icons in a grid format with labels below each icon. Use a modern flat design style with soft shadows and gradients.

Qwen2.5-Plus

Database Database Applications Database Pythons Workload Disk usage Merges

The official Naqios Core logo placed in the center of a digital workspace. Surround it with

Image Generation Describe the image you want to generate.

1024*1024

AI-generated content may not be accurate.

The screenshot shows the Venice Uncensored AI application. At the top, there's a navigation bar with 'Venice Uncensored 1.1' and a dropdown, 'New chat', 'Settings', and a 'X' button. Below the navigation is a search bar containing 'artificial intelligence'. The main area displays a conversation entry from 'Venice Uncensored 1.1' at 4:28 PM, which defines Artificial Intelligence (AI) as a broad field of computer science dedicated to creating machines and systems capable of performing tasks that typically require human intelligence. It lists learning, reasoning, problem-solving, perception, language understanding, and decision-making as tasks. The conversation then branches into two types of AI: 'Narrow AI (Weak AI)' and 'General AI (Strong AI)'. 'Narrow AI' is described as designed for a narrow task like facial recognition or driving a car. 'General AI' is described as hypothetical AI with human-like cognitive abilities capable of performing any intellectual task a human can do. The AI can be further divided into subcategories like Machine Learning, Deep Learning, Natural Language Processing, and Computer Vision. At the bottom of the main area, there's a text input field with placeholder 'Ask a question privately...' and three icons for audio, image, and file attachments. To the right is a vertical 'Settings' sidebar with tabs for 'Text' (selected), 'Image', and 'App'. Under 'Text', 'Model' is set to 'Venice Uncensored 1.1'. Other settings include 'Web Search Enabled' (on), 'System Prompt', 'Venue Voice', and 'Advanced Settings'. A 'Reset to Defaults' button is located at the bottom of the sidebar.

✨ AI Known Spells

- Prompt Engineering
- Refining AI Outputs
- Context Management
- Instruction-based Prompting
- Creative Prompt Design
- Prompt Chaining & Iteration

💼 AI Tools

- Open ai : ChatGPT
- Google : Gemini
- Microsoft : Copilot
- DeepSeek
- AliBaba : Qwen
- Claude

These are the familiars I call upon for different tasks.

📚 AI Languages & Frameworks

◆ AI Core Runes

- English (Clear, Structured Prompts)

- Markdown (For Presentation & Clarity)

◆ AI Arcane Support

- Web Browsers
- Prompt Templates
- Notebook Environments

Ai Origin Story

I didn't go looking for this skill—**AI came to me** as part of the world's transformation.

As machines began thinking, creating, and deciding, I realized I had two choices: ignore it and fall behind, or learn to **wield it with purpose**. But I also saw the danger—people relying on it blindly, losing the will to grow on their own.

So I made a decision:

Adapt, but never depend.

I trained hard, learned deeply, and built my own discipline. Now, I use AI as a tool—not a crutch.

When a task isn't urgent, I push myself to do it as if **AI didn't exist**. That's how I sharpen my own mind, while still commanding the full power of AI when needed.

5 📱 Mobile Enchantments ★★★



=> **Mobile Enchantments** involve casting cross-platform spells to build interactive mobile apps.

| Creating responsive and engaging mobile applications with React Native and Flutter.

✍️ Mobile Description

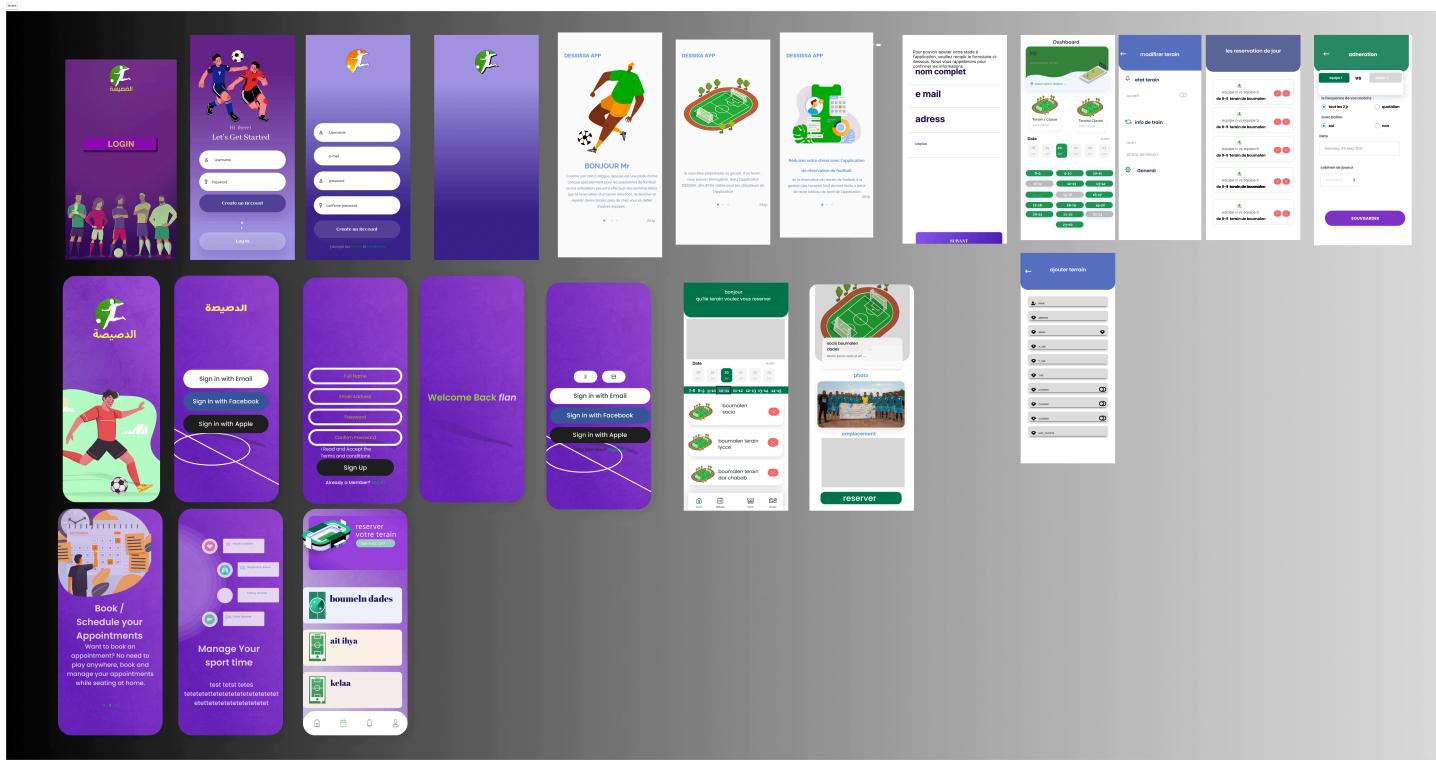
Mobile Enchantments is the art of shaping spells that fit the magical tomes carried by every traveler and noble in the realm. With the right incantations, I turn ideas into touchable enchantments—apps that respond to every scroll, tap, and whisper.

To forge these mobile spells, I use a mix of old and new magic. Sometimes I draw upon general-purpose spells from the frontlines like `HTML`, `CSS`, and `JavaScript`, combined with the sacred teachings of `Responsive Design`. Other times, I wield more focused tools of the trade—like the hybrid scrolls of `React Native` and `Expo`.

Though I've also touched the deeper magics of `Swift`, `Kotlin` and `Flutter`, those paths still demand patience and practice. Yet with each trial, my spellbook grows stronger.



Mobile Gallery of Artifacts





Hi there!

Let's Get Started

Username

Password

Create an Account

O
r

Log In

Dashboard

Mr

proprietaire terain



📍 boumalen dades ...



Terrain 1 Classe

Avec Gazon



Terrain2 Classe

Avec Gazon

Date

autre

18

Jan

19

Jan

20

Jan

21

Jan

22

Jan

23

Jan

8-9

9-10

10-11

11-12

12-13

13-14

14-15

15-16

16-17

17-18

18-19

19-20

20-21

21-22

22-23

23-00

الرئيسية



قرآن
كريم(سمعي)



حديث شريف



قرآن كريم
(بصري)



ادعية



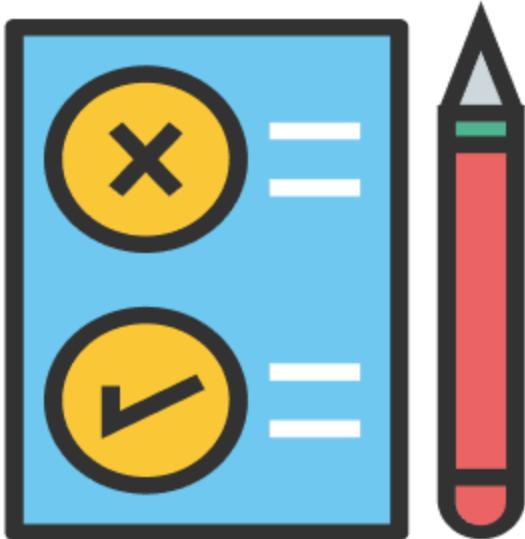
أركاني



A large, dark silhouette of a person kneeling in a prayerful pose is centered in the lower half of the image. The person is wearing a traditional Islamic head covering (ghutrah) and a long robe (aghaam). They are positioned on a small, patterned mat on a light-colored floor. The background behind the silhouette shows a bright, arched opening leading to a courtyard with minarets, creating a strong sense of depth and light.

ARKANI

اذا وجد خطأ



نحن بشر والبشر ليسوا معصومين من الخطأ، ولذلك نطلب منكم، عملاءنا الكرام، في حال وجدتم أي خطأ سواء في الأحاديث أو الآيات القرآنية، نرجو التواصل معنا وإبلاغنا فورًا. سنعمل بإذن الله على تصحيح المعلومات بأسرع وقت ممكن، فنحن نهتم بجودة وصحة المحتوى بشكل كبير.



● ● ●



✨ Mobile Known Spells

- Cross-Platform Development
- Responsive Layouts
- Navigation & State Management
- Native Module Integration

- Mobile Performance Optimization

Mobile Tools

- Android Studio
- Expo
- ADB (Android Debug Bridge)
- BlueStacks
- VS Code
- Physical Devices (for real-world testing)

Mobile Languages & Frameworks

◆ **Mobile Core Runes**

- React Native
- Kotlin
- JavaScript

◆ **Mobile Arcane Support**

- HTML & CSS (for hybrid layouts)
- Virtual Devices (emulators)
- Figma (UI/UX crafting)
- Firebase (backend spells)
- REST APIs (communication magic)

Mobile Origin Story

Long ago, in the days when glowing rectangles first entered the hands of the common folk, I held one and felt something stir within me. Not just awe—but purpose. I didn't merely see a device; I saw a spellbook waiting to be written, a window into infinite worlds. From that moment on, I knew: I would one day craft enchantments that lived within these magical tomes.

In my early days as a student of the full-stack order, I studied the ancient scrolls of web sorcery, learning the basics of structure, logic, and style. But while others were content mastering the scrolls of `HTML`, `CSS`, and `JavaScript`, my gaze often wandered to the realm of mobile magic—a realm where every touch was a rune, every swipe a command.

I began by summoning simple interfaces using hybrid charms, mixing web spells with responsive enchantments. Soon, I uncovered the sacred scrolls of `React Native`, which allowed me to craft

powerful cross-platform incantations. With the aid of Expo, I summoned my first fully working familiars—apps that answered to my every instruction.

Even now, though I have journeyed through many lands of code and built artifacts for all kinds of devices, the path of Mobile Enchantments remains one of my favorite quests. With each app, I breathe life into my ideas—transforming them into spells others can hold in their hands.

This path is far from over. Many runes remain untranslated. But the fire that sparked from that first encounter still burns, guiding my hand as I craft ever more powerful enchantments for the devices of this world

6 🌐 Web Explorer ★★★



=> **Web Explorer** — Navigating the enchanted realms of the Internet.

Web Explorer is the art of understanding how the web truly works — from HTTP incantations to domain name rituals, browser behavior, and the hidden currents of online realms.

Web Explorer Description

To become a true Web Explorer is to learn how the invisible threads of the Internet are woven. This skill grants me the ability to read browser scrolls, trace mysterious links, and understand the behavior of modern web portals. I study the sacred protocols — `HTTP` and `HTTPS` — and the gatekeepers of the web, such as `DNS` and domain lords.

From wielding inspection tools like `DevTools` to analyzing the flow of requests and responses, I can uncover how a page breathes, how resources are summoned, and where latency hides in the shadows. I navigate APIs like ancient maps, fetch data from distant servers, and learn how different clients and spells interact.

With this knowledge, I can optimize performance, reduce loading rituals, and better understand how a single click leads to an entire world of magic behind the screen.

Web Explorer Gallery of Artifacts



DuckDuckGo



Yandex





Chrome



Edge



Firefox



Safari



Opera



brave

✨ Web Explorer Known Spells

- Web Performance Monitoring
- DevTools Network Inspection
- API Consumption & Debugging
- DNS and Domain Knowledge
- HTTP Headers Decoding
- Surfing through the internet wisely

⌚ Web Explorer Tools

- Search Engines (Google, DuckDuckGo...)
- Web Browsers (Chrome, Firefox, Brave...)
- DevTools
- Inspect Element
- Online API tools (Postman, ReqBin)

Web Explorer Languages & Frameworks

◆ Web Explorer Core Runes

- English
- French
- Arabic

◆ Web Explorer Arcane Support

- HTML & CSS & JavaScript
- Browser DevTools
- API Interfaces

Web Explorer Origin Story

My journey as a Web Explorer began in the early days of my youth, armed with nothing more than curiosity and a shared family computer. I still remember the very first phrase I entered into the sacred scrolls of Google: "ألعاب / کشن" It was a simple request, but it opened a portal to an entirely new world — one of endless discovery and digital wonder.

Back then, I did not know how the results appeared, how clicking a link summoned pages from distant lands, or how the banners and games knew where I had been. But I returned, day after day, driven by that same fire of curiosity. I searched. I clicked. I explored.

Over time, that innocent wonder grew into a desire to **understand**. I began to peek behind the veil using browser tools, inspecting network spells and learning how each part of a website was conjured. I learned that behind every link was a ritual — behind every page, a structure.

What started as a child's game slowly became a quest for mastery. From reading response headers to analyzing performance bottlenecks, I grew into the role of a true Web Explorer — no longer just a visitor of the web, but a mapper of its hidden paths.

7 Data Divination ★★



=> **Data Divination or Data Analysis** — The ancient craft of reading the hidden patterns in the numbers. Harnessing the power of data analytics and visualization to uncover hidden insights.

Data Divination is the art of gathering raw numbers, cleansing them of noise, and shaping them into charts and symbols that reveal truths unseen by the naked eye.

Data Divination Description

Much like a seer reading runes by the fire, I study datasets to discover the patterns, trends, and whispers they carry. Using the enchanted tools of this age — from the `Python` language to the mystical scrolls of `Excel` — I turn piles of raw numbers into clear stories.

With the right incantations, I clean and prepare messy data, forge visualizations in the form of charts and graphs, and uncover hidden links between values. Whether it's a simple comparison or the early signs of a coming trend, my goal is to bring clarity to the fog of uncertainty.

Through `Matplotlib`, `Pandas`, and other analytical runes, I turn silent data into actionable insight — guiding decisions like a trusted oracle.



Data Divination Gallery of Artifacts

LOGO SEKOLAH

MASTER MENU

- Dashboard
- Tahun Akademik
- Jurusan
- Kelas
- Kelas Siswa
- Siswa
- Guru

KEUANGAN

- BiPot
- BiPot Siswa
- Report Tagihan

ADMINISTRATOR

- Pengguna

Pilih Tahun Akademik

Total Tagihan Siswa **Rp3.753.953**

Total Siswa Belum Bayar **9 Orang**

Jenis Pembayaran **6 Pembayaran**

Total Kas Saat Ini **Rp10.564.780**

Pembayaran Terakhir **Rp75.000**
Komite Sekolah

Lihat Transaksi

Tagihan Siswa

Bulan	Tagihan Siswa	Bulan
JAN	2000	500
FEB	500	2000
MAR	2000	2000
APR	2000	500
MAY	300	2000
JUN	2000	500

Siswa Belum Bayar

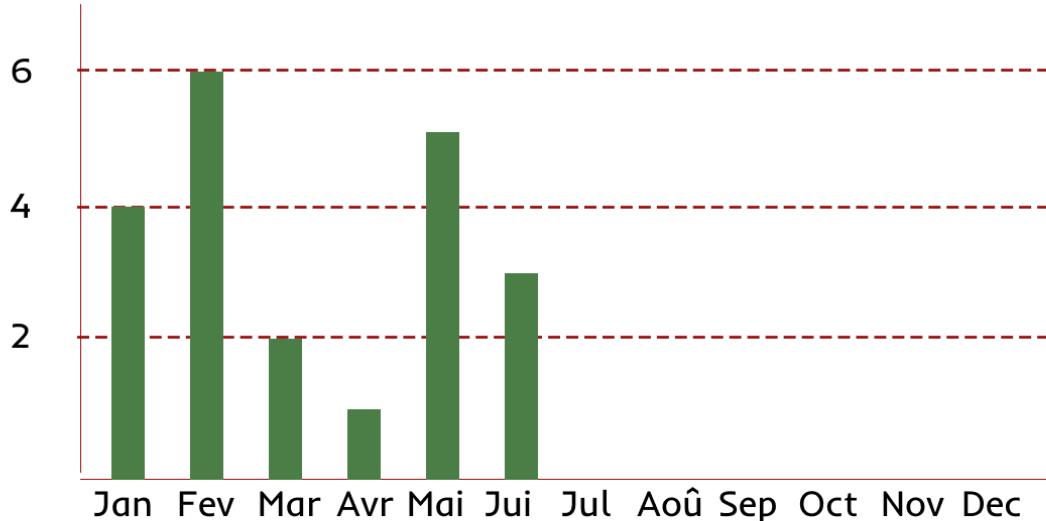
Siswa	Tagihan
Wulan XII Akuntansi C SPP Semester Genap 2022	Rp550.000
Nike XI Jasa Boga A Komite	Rp75.000
Andri XII Administrasi Perkantoran A Perpisahan Kelas	Rp100.000
Anita XII Akuntansi B Perpisahan Kelas	Rp100.000

Jenis Pembayaran

Jenis Pembayaran	Persentase
SPP	~40%
Komite	~20%
Perpisahan Kelas	~20%
Iuran	~20%

CONFIGURATION DU RAPP

Variation de "Videnge" en fonction des Nombre en 2023



🌟 Data Divination Known Spells

- Data Cleansing & Preparation
- Data Visualization (Charts, Graphs)
- Exploratory Data Analysis
- Basic Statistics & Correlation Finding
- Dashboard Crafting
- Storytelling with Data

📦 Data Divination Tools

- Excel
- Python
- Jupyter Notebook
- Matplotlib
- Pandas

Data Divination Languages & Frameworks

◆ Data Divination Core Runes

- python

◆ Data Divination Arcane Support

- Matplotlib
- Pandas
- Charts
- Graphs

Data Divination Origin Story

My journey into the art of Data Divination began not in a grand hall of scholars, but at a quiet desk while working on the back end of a project. I was tasked with a strange challenge — to turn streams of raw, tangled numbers into something others could see and understand, or to transform a clear visualization back into precise code.

At first, it was like staring at an unreadable map, lines and figures scattered without meaning. But soon, I learned the runes of analysis and the symbols of visualization. The more I worked, the more I realized that every dataset tells a story — some are whispers, some are shouts, but all can be heard if you know the language.

From that day forward, I have used my skills as a data seer to illuminate the unseen, guiding projects and decisions with the light of knowledge drawn from the shadows of raw information.

8 Code Enchantments ★★★★



=> **Code Enchantments** — Crafting elegant and efficient code-spells o banish chaos and summon clarity.

Code Enchantments is the art of writing clean, maintainable, and reusable code through sacred practices and timeless patterns.

Code Enchantments Description

Code Enchantments are the sacred practices and methods that ensure a codebase remains clean, understandable, and reliable across time. They are not always visible on the surface—your app may function the same whether or not these practices are used—but their influence lies deeper, shaping how easily the code can be read, maintained, and expanded without introducing new bugs.

Messy code, though functional, is like a crumbling foundation: it slows down development, makes debugging a nightmare, and turns collaboration into a burden. Over time, every developer realizes that disorder in the code eventually spreads into the workflow itself—reducing productivity, creating hidden risks, and making even simple tasks frustrating. Clean, enchanted code, on the other hand, improves comfort during development, speeds up problem-solving, and ensures future stability.

The role of Code Enchantments grows stronger in collaborative environments. When working on larger projects with multiple developers, the difference between structured, consistent code and chaotic

fragments is enormous. Code that is difficult to read or inconsistent in its structure becomes a barrier for teammates, while code shaped by enchantments is a shared language that everyone can understand, modify, and extend without fear.

Practicing Code Enchantments means embracing best practices that give your work a long lifespan. It means writing code that is reusable across different projects, adaptable to new requirements, and flexible enough to endure the inevitable evolution of technology. It also means developing habits that make clean code your default state of work, not just an afterthought.

The arsenal of enchantments includes powerful tools such as `TypeScript` for type safety, `regular expressions (regex)` for precise text manipulation, `SASS` and other CSS libraries for structured styling, and the disciplined use of `comments` and `documentation` to record the logic behind your spells. Tools like `eslint`, `prettier`, and `editorconfig` stand as guardians of consistency, ensuring that formatting and structure are never neglected.

But the realm of Code Enchantments is vast—beyond these examples lies a treasure of patterns, guidelines, and utilities crafted by generations of developers. Design patterns, clean architecture principles, unit testing frameworks, and dependency managers are all part of the enchanted library that a wise developer draws upon. Over time, these practices become second nature, flowing through your work as naturally as writing the code itself.

In the end, Code Enchantments are not just about discipline—they are about legacy. A well-enchanted codebase becomes timeless: a living grimoire that others can read, learn from, and build upon, ensuring that your work continues to thrive long after the initial spell has been cast.



Code Enchantments Gallery of Artifacts

```
① .eslintrc.json ×
② .eslintrcjson
1  {
2    "env": {
3      "ES2021": true,
4      "node": true
5    },
6    "extends": ["eslint:recommended", "plugin:react/recommended"],
7    "parserOptions": {
8      "ecmaFeatures": {
9        "jsx": true
10      },
11      "ecmaVersion": "latest",
12      "sourceType": "module"
13    },
14    "plugins": ["react"],
15    "rules": {
16      "indent": [1, 2, { "SwitchCase": 1 }],
17      "max-len": [
18        "error",
19        {
20          "code": 150
21        }
22      ],
23      "quotes": ["off", "single"],
24      "react/prop-types": "off",
25    }
26  }
```

```
↳ .editorconfig X
↳ .editorconfig

1 # EditorConfig is awesome: https://EditorConfig.org
2
3 # top-most EditorConfig file
4 root = true
5
6 [*]
7 indent_style = space
8 indent_size = 2
9 end_of_line = crlf
10 charset = utf-8
11 trim_trailing whitespace = false
12 insert_final_newline = false
```

The screenshot shows a code editor with two tabs: 'todo.md' and 'Preview todo.md'. The 'todo.md' tab contains a list of items under '# To Do List'. The 'Preview' tab shows the same list with checkboxes next to each item. A sidebar on the right lists additional tasks.

index.html 3, M skills.md todo.md X skills.js M Preview todo.md X

todo.md > # To Do List

```
1 # To Do List
2
3 - [ ] split skills to generale skills and special skills
4 - [ ] A section for learning materials I've created (PDFs, slides, etc.)
5 - [ ] A hobbies page — for what I do beyond code
6 - [ ] A language profile — listing all languages I speak + proficiency
7 - [ ] Cybersecurity enhancements to harden the site
8 - [ ] An achievements section
9 - [ ] A design gallery for my Figma and Canva work
10 - [ ] A collection of developer tools (color pickers, SVG/CSS generators, etc.)
11 - [ ] Possibly an achievement system to track goals and progress interactively
12
```

To Do List

- split skills to generale skills and special skills
- A section for learning materials I've created (PDFs, slides, etc.)
- A hobbies page — for what I do beyond code
- A language profile — listing all languages I speak + proficiency
- Cybersecurity enhancements to harden the site
- An achievements section
- A design gallery for my Figma and Canva work
- A collection of developer tools (color pickers, SVG/CSS generators, etc.)
- Possibly an achievement system to track goals and progress interactively

End of document

1 min read

The `Array()` constructor creates `Array` objects.

Syntax

```
new Array()
new Array(element1)
new Array(element1, element2)
new Array(element1, element2, /* .., */ elementN)
new Array(arrayLength)

Array()
Array(element1)
Array(element1, element2)
Array(element1, element2, /* .., */ elementN)
Array(arrayLength)
```

Note: `Array()` can be called with or without `new`. Both create a new `Array` instance.

Parameters

`element1, ... elementN`
A JavaScript array is initialized with the given elements, except in the case where a single argument is passed to the `Array` constructor and that argument is a number (see the `arrayLength` parameter below). Note that this special case only applies to JavaScript arrays created with the `Array` constructor, not array literals created with the square bracket syntax.

`arrayLength`
If the only argument passed to the `Array` constructor is an integer between 0 and $2^{32} - 1$ (inclusive), this returns a new JavaScript array with its `length` property set to that number.

Note: This implies an array of `arrayLength` empty slots, not slots with actual `undefined` values — see [sparse arrays](#).

Exceptions

`RangeError`

JavaScript > Array > Array.array

✨ Code Enchantments Known Spells

- Code Refactoring
- Design Patterns
- Clean Architecture
- Documentation & Comments
- Linting & Formatting

📦 Code Enchantments Tools

- eslint – the guardian against unclean code
- prettier – the formatter’s wand
- editorconfig / indentation – the keepers of consistency
- markdown – scrolls of documentation

📚 Code Enchantments Languages & Frameworks

◆ Code Enchantments Core Runes

- eslint
- typescript
- sass
- best practice methode

◆ Code Enchantments Arcane Support

- prettier extention
- Color palettes & style guides

Code Enchantments Origin Story

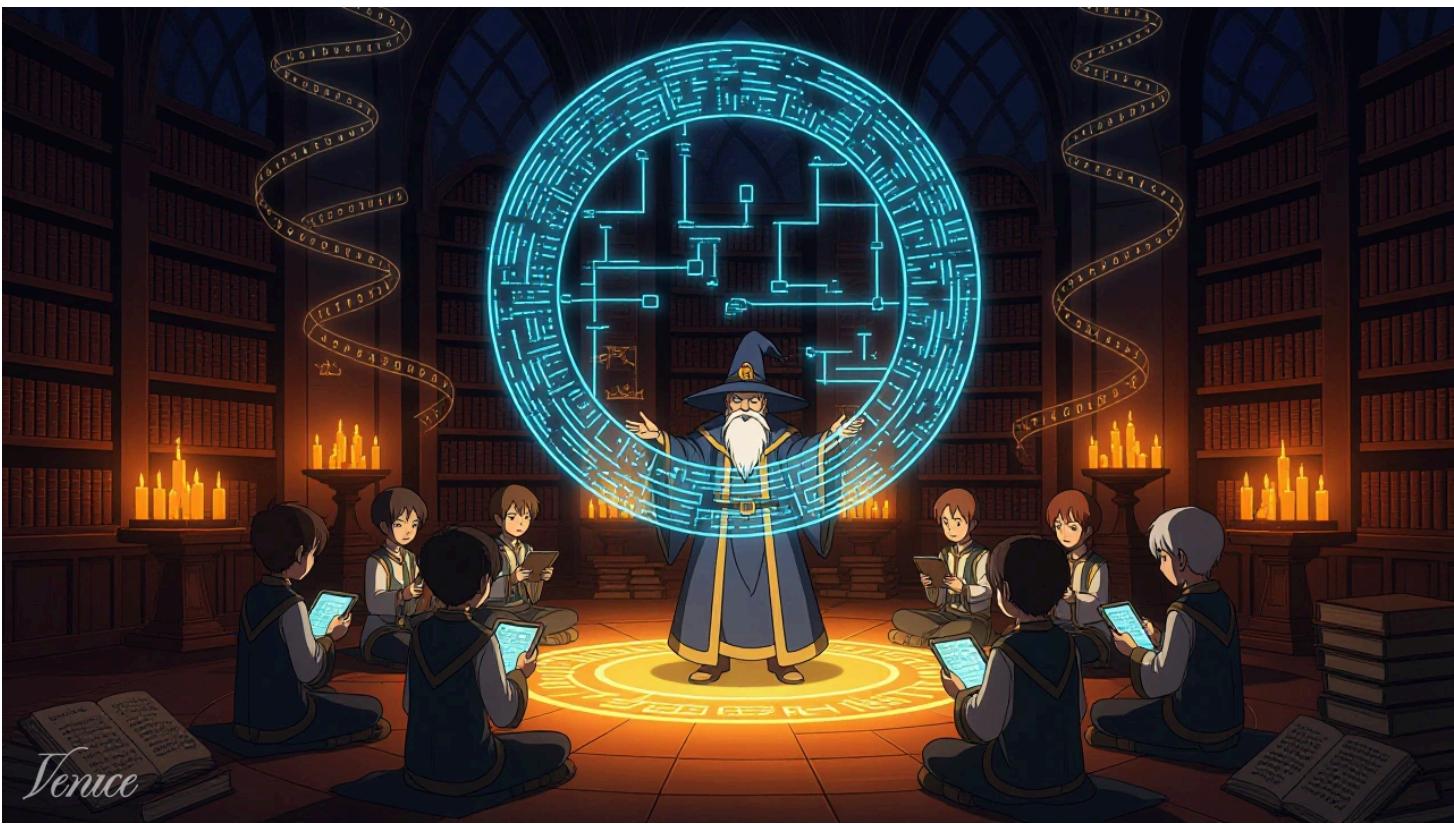
When I first began my journey into the world of code, I knew nothing of Code Enchantments. My early works were a tangle of unspaced lines, missing indentations, and scattered fragments—especially in the days of raw HTML.

Gradually, HTML itself guided me: each opening and closing tag taught me the rhythm of structure. CSS followed naturally, but JavaScript and PHP required deeper discipline. Over time, formatting and organizing code became second nature.

When I began collaborating with friends and releasing projects into the open realms of GitHub, I realized the true weight of clean code. To stand alongside other developers, I studied the ancient scrolls of best practices and invoked modern tools like `prettier`, `TypeScript`, and `SASS` to refine my craft.

Code Enchantments became not just a habit, but a creed—ensuring that my work endures, scales, and speaks clearly across the ages.

9 Mentorship ★★★



=> **Mentorship** — Guiding aspiring wizards and witches on their journeys of learning and mastery, sharing knowledge, wisdom, and encouragement.

Mentorship is the magical act of teaching and empowering others to grow with confidence—unlocking their true potential through patience, empathy, and guidance.

It is the art of sharing knowledge, providing feedback, and nurturing growth in others while continuing to learn yourself.

Mentorship Description

Mentorship is the art of guiding others toward mastery—helping them find clarity, purpose, and balance on their learning path. It is not about showing off what you know, but about **passing on what experience has taught you** and helping others avoid the pitfalls you once faced.

This craft extends far beyond technical skill. It demands empathy, communication, and patience. It's about **listening more than speaking**, allowing others to make mistakes safely, and gently steering them toward better solutions—without breaking their confidence or spirit.

Through mentoring, I've learned to adapt to **different personalities, ages, genders, and cultures**. Every person learns differently, and it's my role to understand that rhythm—sometimes by guiding

softly, sometimes by pushing a little harder, but always with kindness and respect.

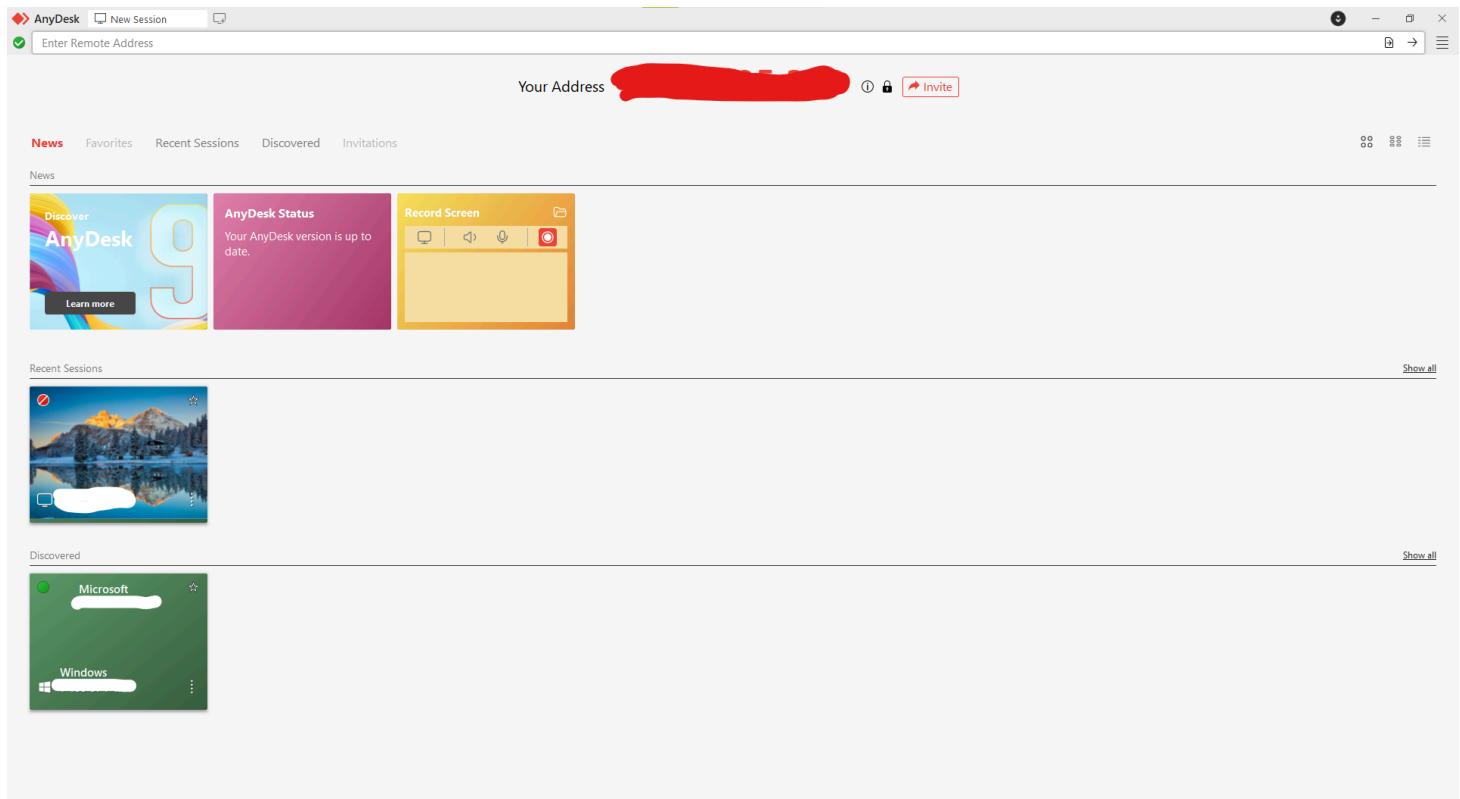
Mentorship also taught me humility. To mentor others, I must remain a **lifelong learner**—curious, open-minded, and unafraid to admit mistakes. The moment you believe you know everything, you stop growing. But when you embrace the mindset of a learner, every conversation becomes a new opportunity for discovery.

My mentorship journey reaches beyond coding. In the gaming realm, I once took over leadership of a guild and mentored many players—helping them refine their strategies, work as a team, and ultimately reach the top ranks together. That experience taught me how leadership and mentorship intertwine: both require trust, consistency, and a deep belief in others' potential.

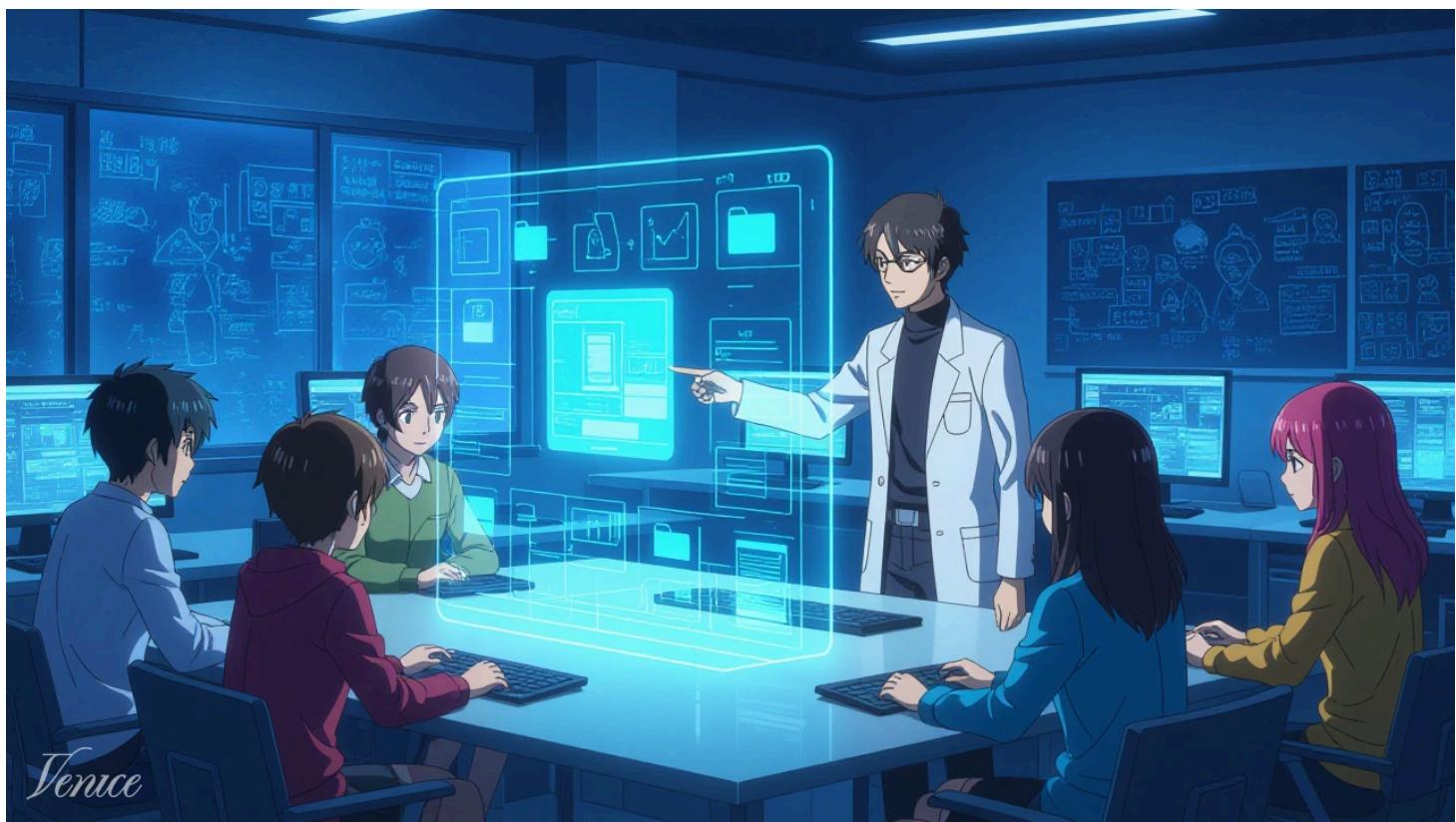
And in the real world, I now serve as a **teacher in a private school**, guiding students on their own paths of learning and creativity. Whether it's through teaching, coding, or gaming, I find meaning in helping others evolve—not just in skill, but in mindset and self-belief.

Mentorship, to me, is not a role—it's a calling. It is the quiet joy of seeing someone you've guided take flight on their own.

Mentorship Gallery of Artifacts



The screenshot shows the AnyDesk software interface. At the top, there is a toolbar with icons for connecting, disconnecting, and entering a session address. A red box highlights the 'Enter Remote Address' input field. Below the toolbar, the address 'Your Address' is entered, followed by a lock icon and an 'Invite' button. The main content area is divided into sections: 'News', 'Favorites', 'Recent Sessions', 'Discovered', and 'Invitations'. The 'Recent Sessions' section shows a thumbnail of a session titled 'Discover AnyDesk' with a large number '9'. The 'Discovered' section shows a thumbnail for a Microsoft Windows session. The interface has a clean, modern design with a light gray background and orange accents.



✨ Mentorship Known Spells

- Pair Programming & Collaboration
- Knowledge Sharing & Debugging Guidance

- Learning Path Design
- Code Reviews & Constructive Feedback
- Communication & Soft Skill Development
- Adaptable Teaching & Leadership Styles

Mentorship Tools

- **AnyDesk / TeamViewer** — Remote live assistance
- **Zoom / Google Meet** — Virtual mentorship sessions
- **Slack / Discord** — Communication & ongoing community support
- **GitHub** — Sharing, reviewing, and collaborating on code
- **Face-to-Face Mentoring** — Personal interaction and hands-on teaching

Mentorship

◆ Mentorship Core Runes

- Fundamentals of Programming
- Clear Documentation & Constructive Feedback
- Communication, Empathy, and Leadership

Mentorship Origin Story

From the very beginning, mentorship has been woven into my life. My father—himself a teacher and my first mentor—taught me how to read, write, and think critically. He shaped my curiosity, humility, and discipline. Even when he was strict, he was always fair and just, teaching me that mistakes are not failures but lessons that help us grow.

Following in his footsteps came naturally. I started helping friends with homework, explaining concepts, and realizing that teaching others deepened my own understanding. Over time, I discovered that mentorship was not just something I enjoyed—it was something I *needed*.

Since then, I've carried that same passion into everything I do—from coding to gaming, from teamwork to teaching. Each mentee, student, or player I've guided has reminded me that mentorship is a cycle: we give, we learn, and we grow together.

Mentorship is my way of giving back to the world—one learner, one conversation, and one breakthrough at a time.

10 Problem Solving ★★★★



=> **Problem Solving** — Solving complex challenges with creative and efficient solutions, like a true coding wizard.

Problem Solving is the essence of development—turning chaos into structure through logic and design.

Problem Solving Description

Problem Solving is the art of breaking down *complex challenges* into manageable pieces and crafting effective solutions. It requires a blend of **analytical thinking**, **creativity**, and **technical skill**.

Whether debugging a tricky bug, optimizing performance, or designing a new feature, this skill is about understanding the problem deeply and applying the right tools and techniques to solve it.

Problem Solving is considered one of the most critical soft skills in the tech world. It's not just about writing code; it's about thinking like a developer—anticipating issues, foreseeing potential pitfalls, and designing solutions that are both efficient and elegant.

Problem Solving is the art of turning a bug into a feature, a challenge into an opportunity, is about seeing the unseen, thinking the unthought, and creating the uncreated.

From algorithms to architecture, this skill is about analytical thinking, debugging, and implementing optimal solutions.

Problem Solving Gallery of Artifacts

The screenshot shows the Chrome DevTools interface. At the top, there's a navigation bar with tabs like Elements, Console, Sources, Network, Performance, Memory, Application, Security, Lighthouse, and Recorder. Below the navigation bar, the main content area has a status bar with 'Default levels' and '3 Issues'. A red error message in the status bar indicates a 404 Not Found error for a cursor image. The main body of the interface is mostly blank, suggesting no active code or network requests. At the bottom, there's a separate 'What's new' panel for DevTools 141, featuring a 'See all new features' button and a 'See past highlights from Chrome 140' section with a play button icon.

Home Workspaces API Network

My Workspace New Import Overview Successful Response +

Search Postman Ctrl K

Invite Share Upgrade

No environment

API Documentation #reference / User / Get authenticated user / Successful Response

GET https://api.getpostman.com/me

Params Headers Body

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Headers (1)

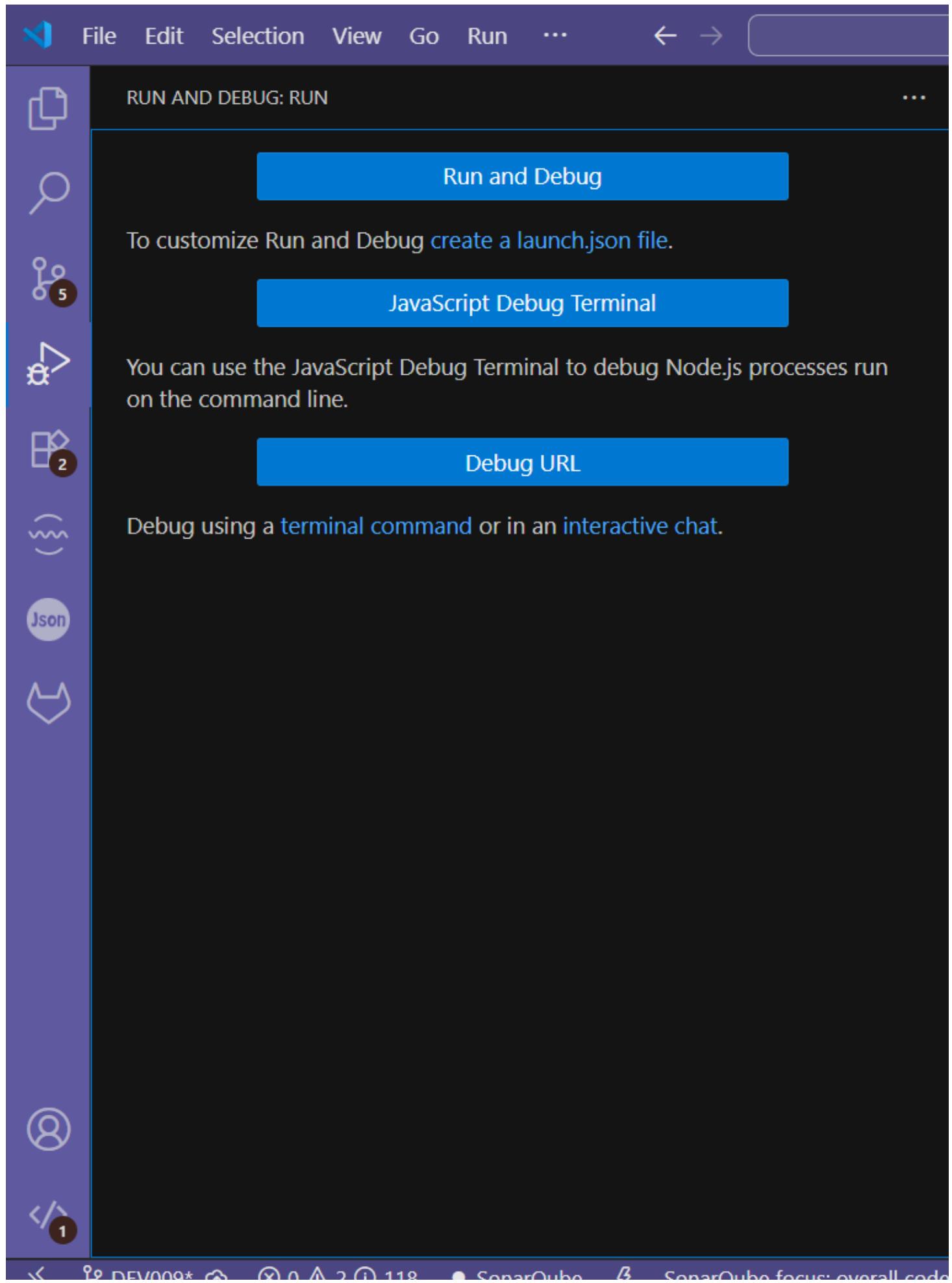
{ } JSON Preview

```
1 {
2   "user": {
3     "id": 12345678,
4     "username": "taylor-lee",
5     "email": "taylor.lee@example.com",
6     "fullName": "Taylor Lee",
7     "avatar": "https://example.com/user/x5u9qpv mujf6lbqmga.jpg",
8     "isPublic": true
9   },
10  "operations": [
11    {
12      "name": "mock_usage",
13      "limit": 1000000,
14      "usage": 100276,
15      "overage": 0
16    },
17    {
18      "name": "monitor_request_runs",
19      "usage": 1000000
20    }
21  ]
22}
```

Status Code 200 OK

Online Find and replace Console

Postbot Runner Start Proxy Cookies Vault Trash



✨ Problem Solving Known Spells

- Algorithm Design
- Data Structures
- Bug Diagnosis
- Optimizing Code
- Architectural Thinking

📦 Problem Solving Tools

- **Browser dev tools** — Inspecting and debugging web applications
- **Postman** — API testing and debugging
- **Figma / Sketch** — Designing and prototyping user interfaces
- **Git** — Version control and collaboration
- **JIRA / Trello** — Project management and issue tracking

📚 Problem Solving

◆ Problem Solving Core Runes

- Critical Thinking
- Logical Reasoning
- Debugging Techniques
- Algorithmic Thinking
- System Design Principles

🏔️ Problem Solving Origin Story

From the first breath of existence, humanity has been tested with trials.

Some simple, others labyrinthine.

Each problem — a quest. Each failure — a lesson. Each solution — a victory etched into the soul.

In my own journey, I have faced dragons of confusion and dungeons of error.

Some fell quickly to my blade of logic, others demanded patience, study, and at times, the counsel of fellow mages.

Yet with every encounter, I grew — wiser, sharper, and more attuned to the rhythm of the unseen.

So it is with code.

Every error message, every frozen loop, every inexplicable crash — a teacher in disguise.

Through these ordeals, I learned new spells, refined my tools, and forged my mind into a sharper weapon.

Problem Solving became not just an art — but a calling.

It is a discipline, a ritual, a way of perceiving the world.

It transforms frustration into fascination, and despair into discovery.

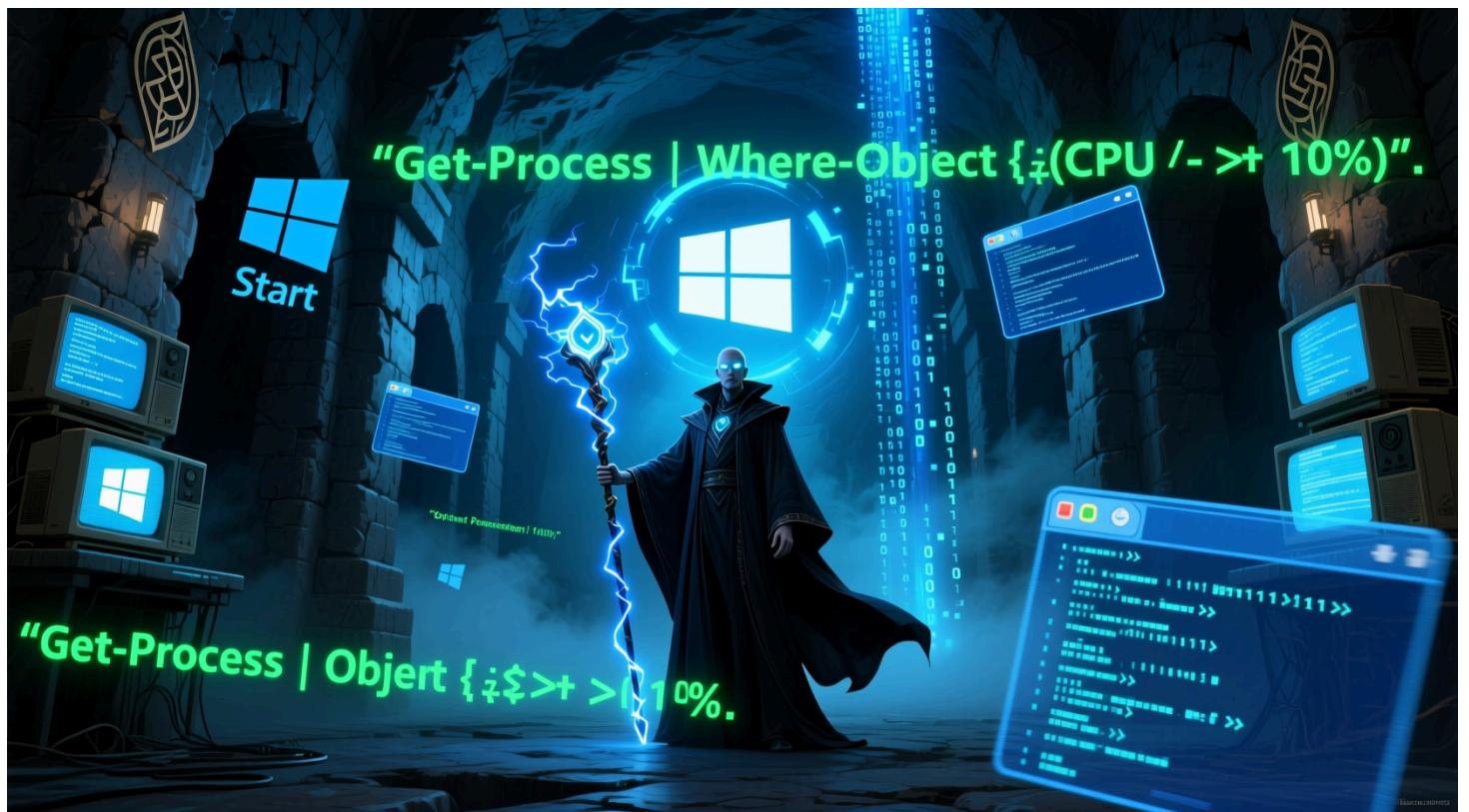
For once a wizard tastes the thrill of conjuring a solution from chaos...

the journey becomes eternal.

Solving problems becomes an *addiction of purpose* — a pursuit that kindles joy, mastery, and meaning.

Each victory is a spark of magic, reminding the wizard that even the darkest bug can lead to the brightest revelation.

11 🖥 Windows Dungeon ★★★★☆



=> **Windows Dungeon** — Exploring and mastering the mystical realms of the Windows operating system. PowerShell and CMD incantations are my tools of choice.

Windows Dungeon is my deep dive into system configuration, scripting, and automation within the cryptic corridors of the Windows OS.

Windows Dungeon Description

Windows Dungeon is the sacred art of navigating and mastering the intricate labyrinth of the Windows operating system. It demands an understanding of the core mechanisms, secret chambers, and hidden levers that make Windows come alive. From the spellbinding command-line incantations of `PowerShell` and `CMD` to the enchanted chambers of Control Panel and Settings, this craft allows me to wield the full might of the Windows realm.

This discipline spans system administration, troubleshooting, and automation. It requires not only technical prowess but also patience, precision, and a keen eye for how Windows orchestrates its symphony of processes, resources, and defenses.

I commune daily with the spirits of the system—taming services, managing user realms, and configuring the sacred network pathways. Through scripting, I automate repetitive rituals, enhance productivity, and bring order to digital chaos.

Transforming or restoring operating systems—**from ancient XP ruins to the modern towers of Windows 11**—is a power I've long mastered. No matter the version, I've explored them all and faced every peril they offer.

From the dreaded *Blue Screen of Death* to driver duels, I've forged the tools and wisdom to restore balance to any corrupted system.

Windows Dungeon Gallery of Artifacts



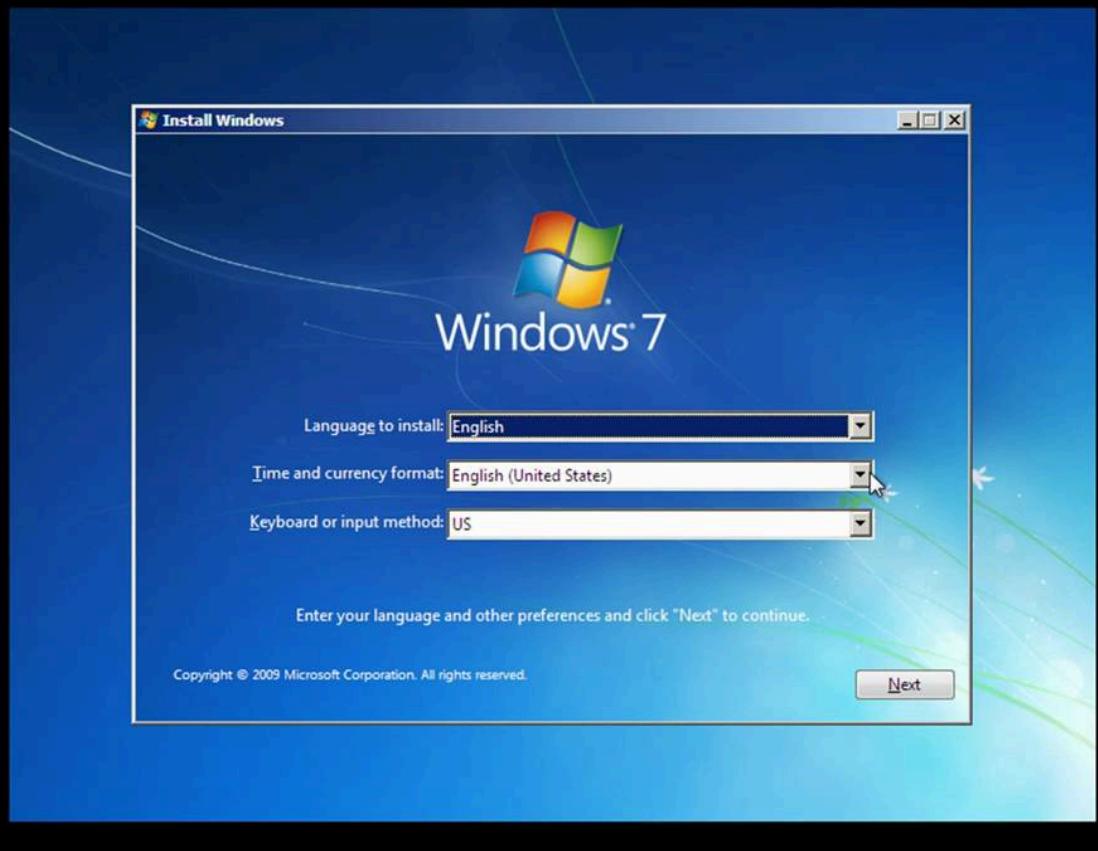
Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

20% complete



For more information about this issue and possible fixes, visit <https://www.windows.com/stopcode>

If you call a support person, give them this info:
Stop code: CRITICAL_PROCESS_DIED





Settings

Mouad Allaoui
My Microsoft account

OneDrive Manage
Windows Update
Last checked: 52 minutes ago

Find a setting

- | | | | | |
|--|--|---|---|--|
| System
Display, sound, notifications, power | Devices
Bluetooth, printers, mouse | Mobile devices
Link your Android, iPhone | Network & Internet
Wi-Fi, airplane mode, VPN | Personalization
Background, lock screen, colors |
| Apps
Uninstall, defaults | Accounts
Your accounts, email, sync, work, family | Time & Language
Speech, region, date | Gaming
Game Bar, captures, Game Mode | Ease of Access
Narrator, magnifier, high contrast |
| Search
Find my files, permissions | Privacy
Location, camera, microphone | Update & Security
Windows Update, recovery, backup | | |



System and Security

[Review your computer's status](#)
[Save backup copies of your files with File History](#)
[Backup and Restore \(Windows 7\)](#)



Network and Internet

[View network status and tasks](#)



Hardware and Sound

[View devices and printers](#)
[Add a device](#)
[Adjust commonly used mobility settings](#)



Programs

[Uninstall a program](#)



User Accounts

[Change account type](#)



Appearance and Personalization



Clock and Region

[Change date, time, or number formats](#)



Ease of Access

[Let Windows suggest settings](#)
[Optimize visual display](#)

✿ Windows Dungeon Known Spells

- PowerShell Scripting
- Batch Spellcraft (Batch Scripting)
- Registry Runes & Tweaks
- Task Scheduler Automation
- System Monitoring & Log Reading
- Performance Divination (PerfMon)
- Event Viewer Scrying
- Error Exorcism (WER)
- System Troubleshooting
- Windows Recovery Rituals
- Safe Mode Invocation

กระเป๋า Windows Dungeon Tools

- Windows PowerShell
- Command Prompt (CMD)
- Event Viewer
- Performance Monitor
- Task Scheduler
- Recovery Environment
- Windows Deployment Services

- System Reset & Format (via USB or ISO)

Windows Dungeon Core Runes

- ◆ Command Prompt (CMD)
- ◆ PowerShell
- ◆ Task Scheduler
- ◆ Safe Mode
- ◆ Recovery Environment

Windows Dungeon Origin Story

Windows — the world's most beloved and widespread operating system — became the first magical realm I ever explored. Praised for its simplicity and adaptability, it was the gateway to my lifelong journey in computing.

My first encounter was at the age of **four**, when our family's ancient PC ran **Windows 2000**. Despite my parents' strict rules, curiosity burned brighter than fear. I'd sneak into the study, often facing dire consequences, just to feel the hum of that machine. Those moments were my first sparks of digital wonder.

By the age of **six**, our household was blessed with a new artifact — a PC running **Windows XP**. Its shimmering interface and new powers captivated me. I spent countless hours exploring menus, experimenting with settings, and reviving the system whenever it froze.

If my fixes failed, I would call upon my elder brother — my first mentor in troubleshooting. Between adventures, I played legendary titles like `Age of Empires II` , `Stronghold Crusader` , `Counter-Strike 1.6` , and even `Banana` and `SNES` classics.

As time passed, I delved deeper into Windows magic — mastering its hidden mechanisms. I learned to command PowerShell and CMD, automate with Task Scheduler, observe system lifeforce through Event Viewer and Performance Monitor, and recover from calamities using the Recovery Environment. I explored every chamber: the Registry Editor, Group Policy, Device Manager, Disk Management, Defender, Firewall, Backup, Restore, and even the sacred art of dual-booting and virtualization.

From `XP` to `11`, from *installation* to *resurrection*, I've walked every path of the **Windows Dungeon**, emerging as its steadfast guardian.

12 🖥️ Linux Slayer ★★★



=> **Linux Slayer** — best open-source OS for developers and hackers. mastering its command line is essential.

Linux Slayer is about leveraging the terminal to rule the open-source realm.

📋 **Linux Slayer Description**

Linux Slayer is the art of mastering Linux, the ultimate open-source operating system, through its powerful command-line interface. It is the discipline of navigating the system's inner workings, understanding its architecture, and wielding tools that allow precise control over every aspect of the OS.

This skill covers a wide range of abilities:

- **System Administration:** Managing users, groups, permissions, services, and processes efficiently.
- **Shell Scripting:** Automating tasks, orchestrating workflows, and writing custom tools in Bash or other shells.

- **Package Management:** Installing, updating, and maintaining software across different distributions using tools like `apt`, `yum`, and `dnf`.
- **Networking & Security:** Configuring networks, managing firewalls, analyzing traffic, and securing systems against intrusions.
- **Monitoring & Troubleshooting:** Reading logs, monitoring performance, debugging errors, and recovering from system failures.

Linux Slayer is more than just knowledge of commands—it's the ability to think like the system itself, anticipate its behavior, and manipulate it to achieve goals efficiently. It requires curiosity, patience, and a hands-on approach: you truly cannot claim mastery without exploring, experimenting, and solving real-world problems on Linux.

This skill is essential for anyone serious about cybersecurity, programming, systems management, or advanced computing. The terminal becomes a sword, and the filesystem a realm to be mastered, where every command is a spell and every script a powerful incantation.

Remember: true mastery comes from experience. No one can call themselves a developer, sysadmin, or hacker without first walking the Linux path.

Linux Types

Linux distributions can be broadly categorized into **Debian-based** and **Non-Debian-based** systems, each with its own philosophy, tools, and package management.

1. Debian-Based Distributions

These distros share a common foundation from **Debian** and usually use `apt` or `dpkg` for package management. They are known for stability, large repositories, and community support. Examples include:

- **Ubuntu** – beginner-friendly, widely used for development and servers.
- **Kali Linux** – security-focused, preloaded with penetration testing tools.
- **Linux Mint** – user-friendly, with a polished desktop experience.
- **Elementary OS** – elegant design, lightweight and intuitive interface.

2. Non-Debian-Based Distributions

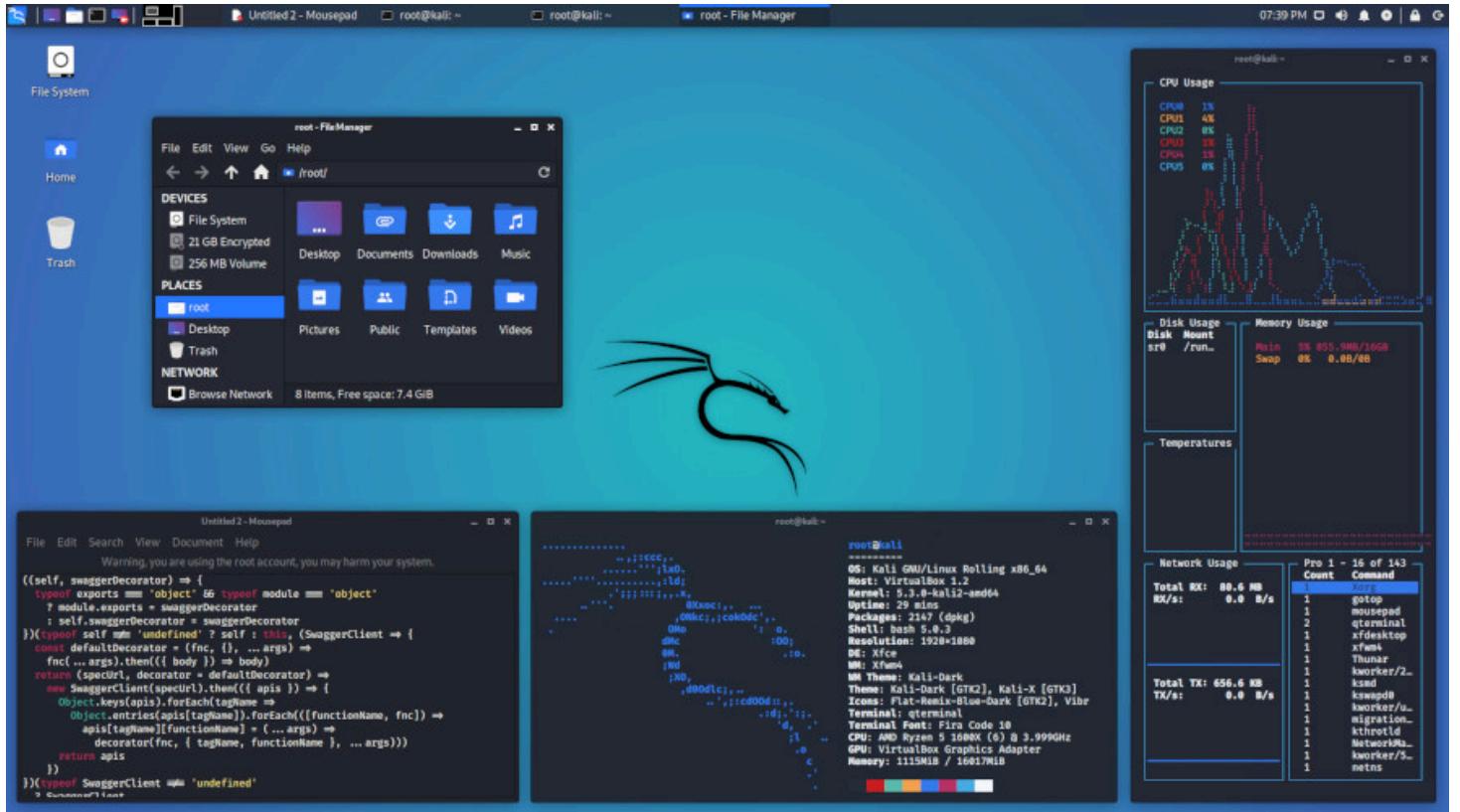
These distributions are not based on Debian and often use other package managers like `yum`, `dnf`, or `pacman`. They offer flexibility and unique philosophies for users seeking alternatives. Examples include:

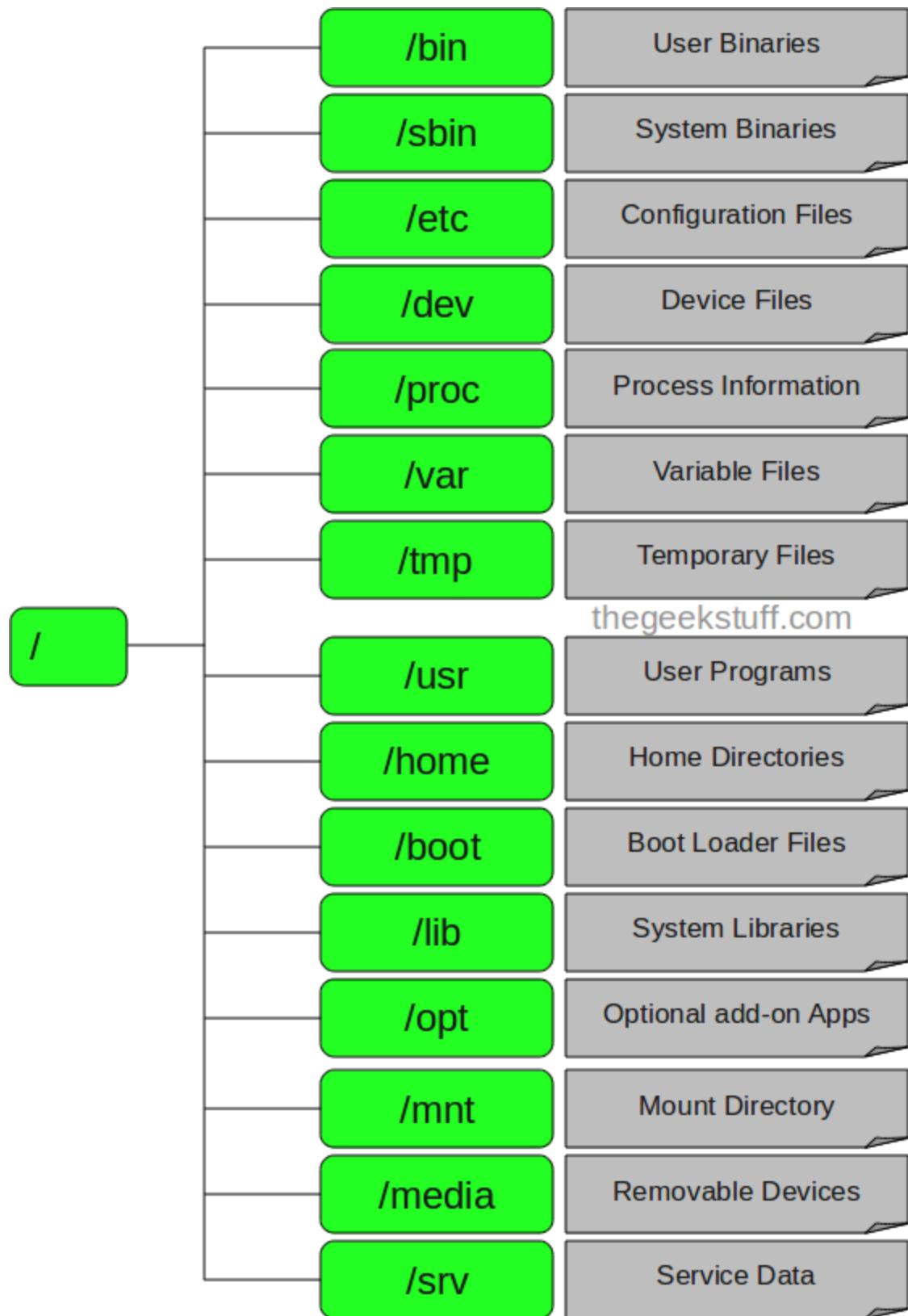
- **Fedora** – cutting-edge, with the latest software and technologies.
- **Arch Linux** – minimal, highly customizable, for advanced users.

- **openSUSE** – enterprise-grade options, with a focus on stability and tools.
- **CentOS / Rocky Linux / AlmaLinux** – server-oriented, reliable and robust.

Understanding the type of Linux you're using is crucial for package management, system administration, and compatibility with tools. Mastering both Debian and non-Debian systems allows a Linux Slayer to rule all realms of open-source.

Linux Slayer Gallery of Artifacts







Top 50 Linux Commands you must know

- | | | | | |
|-----------|------------|-------------|----------------------|---------------------------|
| 1. ls | 11. cat | 21. diff | 31. kill and killall | 41. apt, pacman, yum, rpm |
| 2. pwd | 12. echo | 22. cmp | 32. df | 42. sudo |
| 3. cd | 13. less | 23. comm | 33. mount | 43. cal |
| 4. mkdir | 14. man | 24. sort | 34. chmod | 44. alias |
| 5. mv | 15. uname | 25. export | 35. chown | 45. dd |
| 6. cp | 16. whoami | 26. zip | 36. ifconfig | 46. wheris |
| 7. rm | 17. tar | 27. unzip | 37. traceroute | 47. whatis |
| 8. touch | 18. grep | 28. ssh | 38. wget | 48. top |
| 9. ln | 19. head | 29. service | 39. ufw | 49. useradd |
| 10. clear | 20. tail | 20. ps | 40. iptables | 50. passwd |





✨ Linux Slayer Known Spells

- Shell Scripting (Bash)
- Package Management (apt, yum)

- Cron Jobs & Services
- Log Monitoring
- User/Permission Management

Linux Slayer Tools

- Bash, SSH
- Package Managers (apt, yum)
- Text Editors (vim, nano)
- Network Tools (ping, netstat)
- File Management (ls, cp, mv, rm)
- Version Control (git)
- Virtualization (VirtualBox, Docker)

Linux Slayer Core Runes

- ◆ Terminal (Bash)
- ◆ SSH (Secure Shell)
- ◆ Package Managers (apt, yum)
- ◆ Text Editors (vim, nano)
- ◆ File Management (ls, cp, mv, rm)

Linux Slayer Origin Story

Linux — the world's most beloved and widespread open-source operating system.

My story with Linux began when I was **12 years old**, after hearing people talk about the *deep web* and *dark web*. I got curious and wanted to explore it myself. Installing the Tor browser on Windows worked fine, but I didn't feel secure. People kept saying it was dangerous and could even lead to trouble, so I started looking for a more private and safe way to do it. At that time, I didn't fully understand how things worked yet.

One day, my big brother — who was studying networking — installed **Kali Linux** on his laptop. It was my first time seeing what I thought was a “real hacker” in action (okay, he only hacked the Wi-Fi password 😅). But for a teenager who had only seen that kind of thing in movies, it was mind-blowing. I still remember him using **aircrack** and other cracking tools to get the password. I was amazed — I wanted to learn how to do that too.

After that day, I started searching for information about Linux. I discovered that Kali Linux was based on **Debian**, so I downloaded **Ubuntu** and installed it on my old laptop. I began learning the basics of

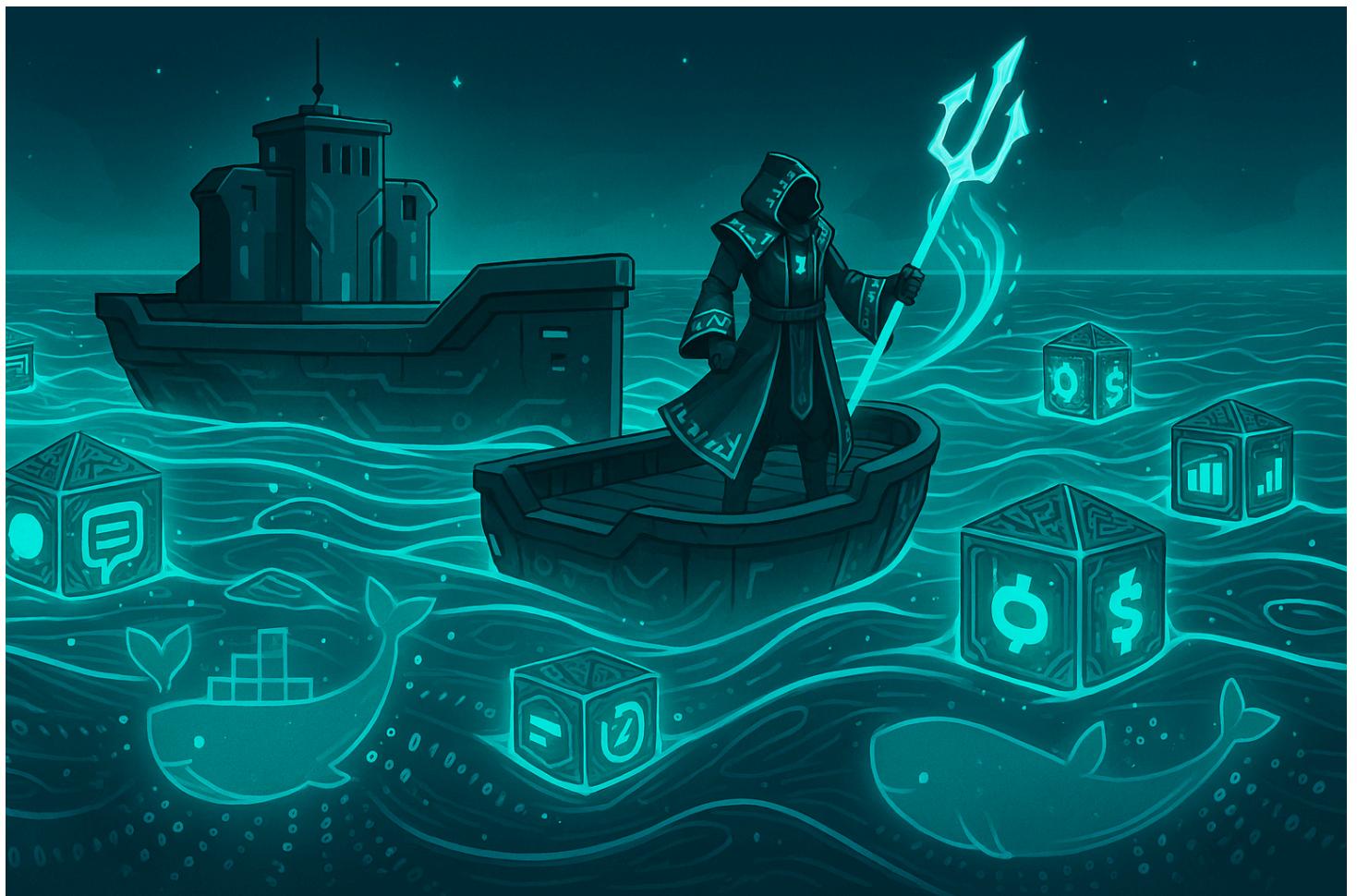
Linux — commands, the file system, and how everything worked. I was fascinated by the power of the terminal and how you could do *anything* with it.

After some time, I lost interest in Linux until my first year at university. When I started learning programming and web development, I found that most tools and frameworks were built for Linux. So I decided to give it another try. I installed different distros — starting again with **Kali Linux**, then **Ubuntu**, **Mint**, and **Fedor**a.

When I reached my first bachelor year, I specialized in **cybersecurity**, and that was the moment I began using Linux more seriously. I learned how to use different tools and scripts, configure and manage Linux servers, and secure them against attacks.

I still remember the first time I used **nmap**, **Wireshark**, **Metasploit**, and other powerful tools — it felt like a whole new world opened before me. I was hooked, and I wanted to learn more and more.

13 🚤 Docker ★★



=> **Docker** — Containerizing applications and orchestrating microservices across realms of code and cloud.

Docker is about crafting lightweight, portable containers that bring consistency, speed, and isolation to modern development workflows.

Docker Description

Docker is the art of *containerization* — packaging applications and their dependencies into self-contained, lightweight units called **containers**. These containers ensure that software runs consistently across any environment — from a developer's local machine to production servers or cloud platforms.

With Docker, I can define an application's entire environment through **Dockerfiles** and **docker-compose** configurations — including the base operating system, required libraries, and dependencies. Each container runs in isolation, ensuring clean, predictable behavior regardless of the underlying system.

Docker plays a vital role in **modern DevOps** and **microservices architectures**, enabling developers to:

- Build and deploy faster,
- Scale efficiently, and
- Ensure consistent performance across platforms.

It bridges the gap between development and production, eliminating the notorious "it works on my machine" problem. Whether I'm deploying a single app or orchestrating multiple services, Docker ensures reliability, portability, and simplicity.

In essence, Docker empowers developers to **build once and run anywhere** — a true evolution in software deployment.



Container management commands

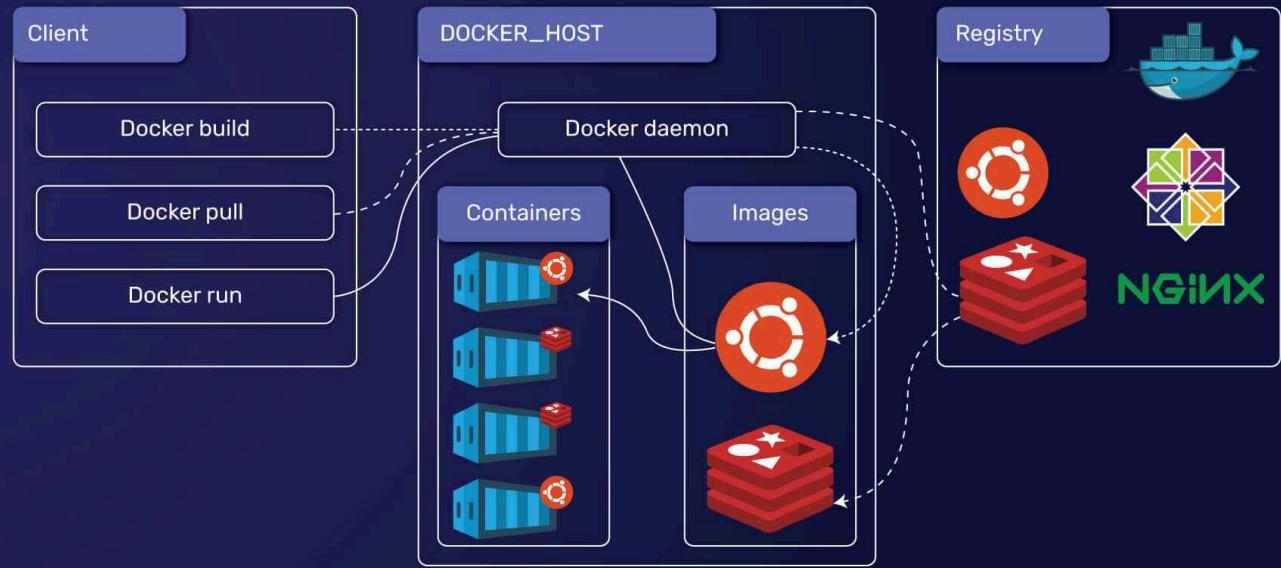
command	description
<code>docker create image [command]</code> <code>docker run image [command]</code>	create the container = <code>create + start</code>
<code>docker start container...</code>	start the container
<code>docker stop container...</code>	graceful ² stop
<code>docker kill container...</code>	kill (SIGKILL) the container
<code>docker restart container...</code>	= <code>stop + start</code>
<code>docker pause container...</code>	suspend the container
<code>docker unpause container...</code>	resume the container
<code>docker rm [-f³] container...</code>	destroy the container

²send SIGTERM to the main process + SIGKILL 10 seconds later

³-f allows removing running containers (= `docker kill + docker rm`)

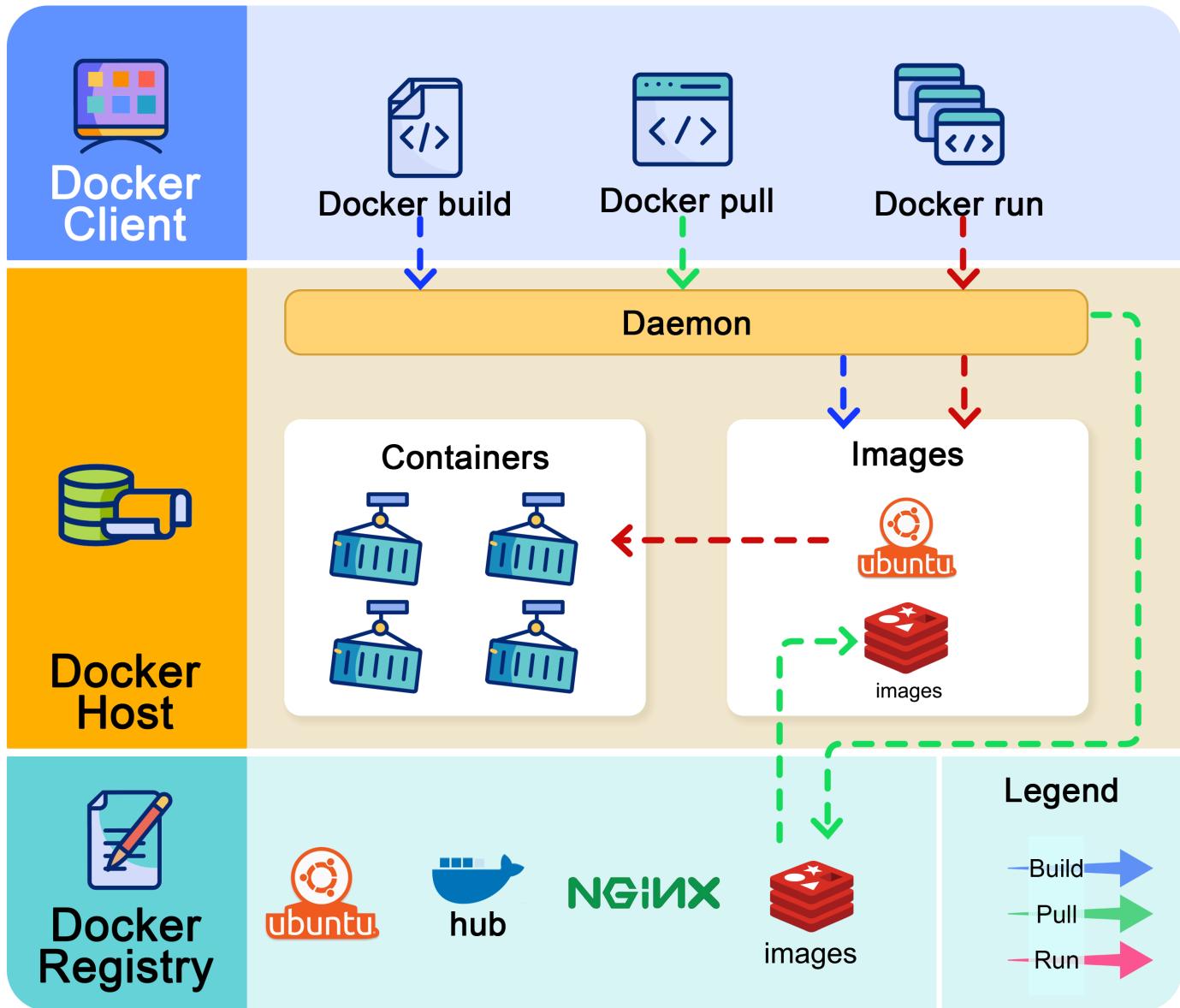


Docker Container Architecture



How does Docker Work ?

 blog.bytebytego.com



⭐ Docker Known Spells

- Dockerfile Creation & Optimization
- Image & Container Management
- Docker Compose for Multi-Container Apps
- Volume & Network Configuration
- Containerized Development Environments

Docker Tools

- Docker Desktop
- Docker CLI
- Docker Compose
- Docker Hub
- Container Orchestration (Kubernetes Basics)

Docker Core Runes

- ◆ Docker CLI Commands
- ◆ Dockerfile
- ◆ Docker Compose
- ◆ Docker Hub

Docker Origin Story

Achieving this skill wasn't something I planned from the start.

It all began during my **second year of development studies**, when our programming instructor first introduced us to Docker. At the time, it felt complex and abstract, so I gave up on it for a while.

Later, as I advanced in my studies — particularly during my **cybersecurity specialization** — we started working on **big data** projects that required containerization. That's when I realized Docker wasn't just another tool; it was a cornerstone of modern infrastructure.

I began learning it seriously, experimenting with containers, building small projects, and debugging my own environments. Over time, I became comfortable using Docker to deploy and manage applications seamlessly.

Now, I can confidently say I understand Docker's core principles — from building images to managing containers — and appreciate its true power in creating scalable, isolated, and portable systems.

14 Virtualization ★★



=> **Virtualization** — Creating and managing virtual environments for development and testing.

Virtualization is the arcane craft of forging digital realms within realms—safe, isolated worlds for testing, deployment, and exploration.

Virtualization Description

Virtualization is the art of creating virtual versions of computing resources—such as servers, operating systems, storage devices, and networks—allowing multiple isolated environments to run on a single physical machine.

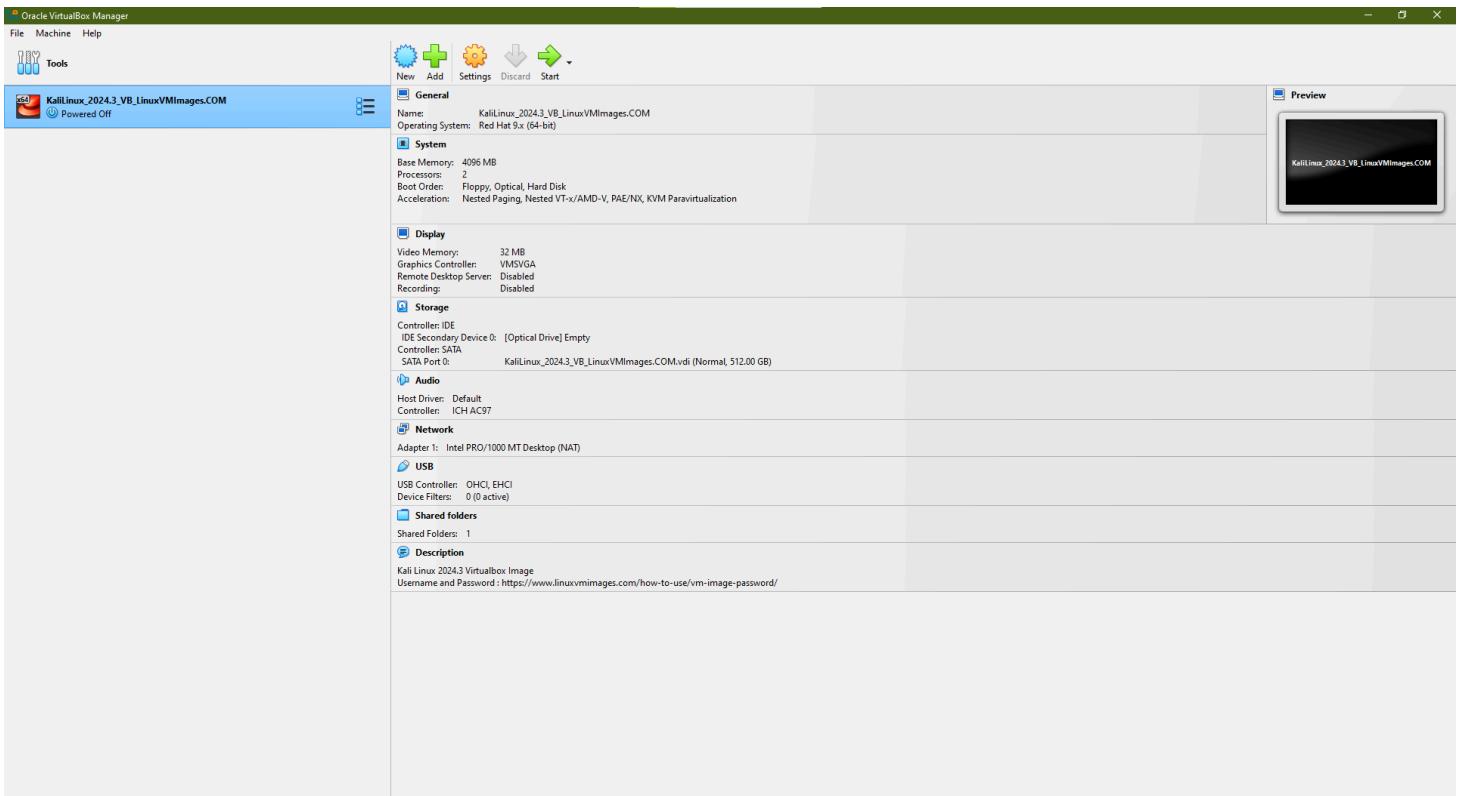
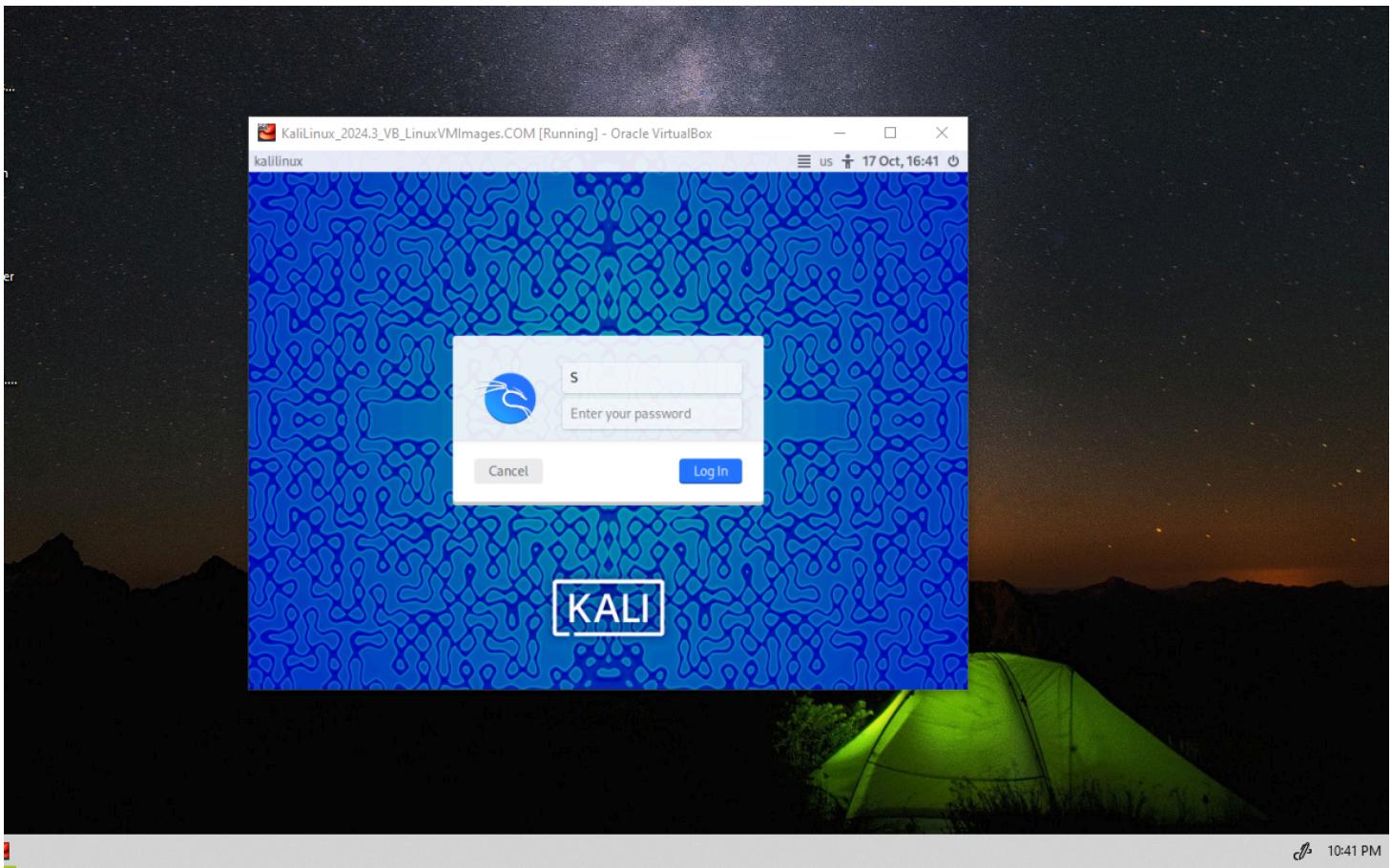
It's like constructing a hidden chamber within your own workspace—a space where experiments can unfold freely, without affecting the world outside.

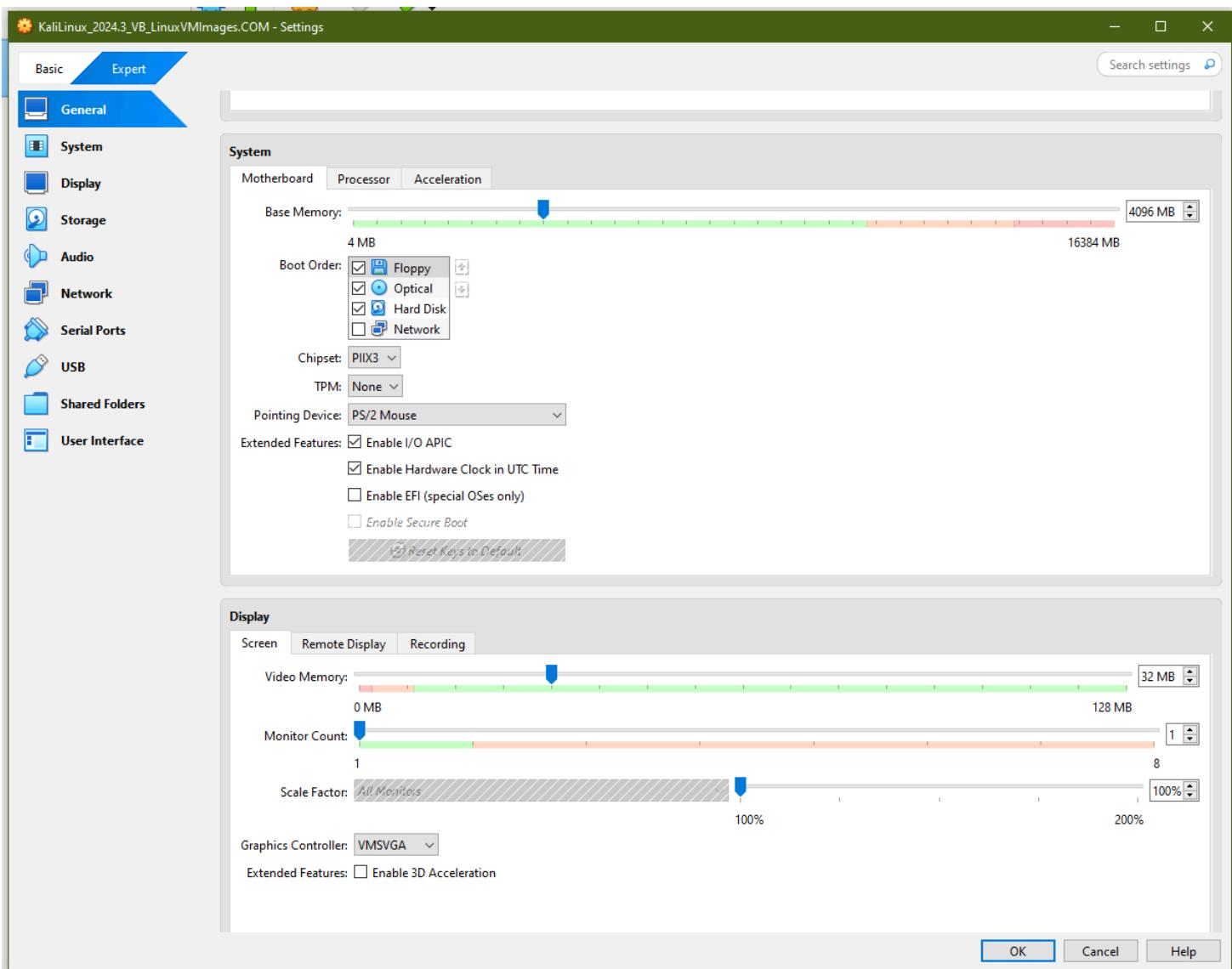
Through virtualization, I can install and run different operating systems, simulate various devices, and test complex configurations—all within a controlled and reversible environment.

It's an essential discipline for developers, system administrators, and cybersecurity practitioners, offering flexibility, safety, and efficiency in every experiment.



Virtualization Gallery of Artifacts





✨ Virtualization Known Spells

- Virtual Machine Management
- ISO & OS Installation
- Snapshot & Rollback
- Virtual Networking
- Multi-OS Testing

💼 Virtualization Tools

- VMware
- VirtualBox
- Bluestacks
- Android Studio Emulator
- WSL (Windows Subsystem for Linux)



Virtualization Origin Story

My journey into **virtualization** began right after mastering Linux.

I needed a way to explore Linux freely—without constantly formatting my main system. Buying a second computer wasn't an option, so virtualization became my portal to freedom.

It allowed me to summon different operating systems within my own machine, experiment safely, and learn without limits. My hardware wasn't the strongest, but it was enough to open new worlds—and that's what mattered most.