

EBOOK BÁCH KHOA TOÀN THƯ

DEEPSEEK HARNESS TOÀN TẬP NĂM 2026

Hướng dẫn thiết lập, kiến trúc Cordis, tự động hóa CSKH & sản xuất Video AI

ĐƠN GIÁ BẢN QUYỀN

199.000 VNĐ

Xuất bản chính thức trên deaitinh.com

Tác giả: Hoàng Nhật Mai – Founder Hệ sinh thái Để AI Tính

Website: deaitinh.com | Hotline/Support: Để AI Tính Hub

[GHI CHÚ]

Đây là bản Ebook Bách khoa toàn thư mở rộng chuẩn hóa về ****DeepSeek Harness**** (`dsh`) — framework và agent runtime nguồn mở thế hệ mới nhất của DeepSeek AI. Sách được thiết kế bảo mật tuyệt đối với toàn bộ dữ liệu đã được ẩn danh hóa (Anonymized), sử dụng mã giả khung (Template Pseudocode), sơ đồ tư duy trực quan và các cờ giữ chỗ chuẩn hóa.

Mục lục Ebook

- [Chương 1: Tổng quan, triết lý và khởi động với DeepSeek Harness](#chuong-1-tong-quan-triet-ly-va-khoi-dong-voi-deepseek-harness)
- [1.1 Khái niệm DeepSeek Harness và vị thế năm 2026](#11-khai-niem-deepseek-harness-va-vi-the-nam-2026)
- [1.2 So sánh DeepSeek Harness với các AI Agent Framework khác](#12-so-sanh-deepseek-harness-voi-cac-ai-agent-framework-khac)
- [1.3 Triết lý kiến trúc "Everything is a Plugin" trên Cordis Framework](#13-triet-ly-kien-truc-everything-is-a-plugin-tren-cordis-framework)
- [1.4 Chuẩn bị môi trường phần cứng và phần mềm](#14-chuan-bi-moi-truong-phan-cung-va-phan-mem)
- [1.5 Khởi chạy nhanh 60 giây với lệnh npx](#15-khoi-chay-nhanh-60-giay-voi-lenh-npx)
- [Chương 2: Thao tác giao diện Web UI và quản lý phiên làm việc](#chuong-2-thao-tac-giao-dien-web-ui-va-quan-ly-phien-lam-viec)
- [2.1 Tổng quan các khu vực chức năng trên Web UI](#21-tong-quan-cac-khu-vuc-chuc-nang-tren-web-ui)
- [2.2 Cấu hình API Key và nhà cung cấp mô hình AI](#22-cau-hinh-api-key-va-nha-cung-cap-mo-hinh-ai)
- [2.3 Quản lý các bộ Profile làm việc](#23-quan-ly-cac-bo-profile-lam-viec)
- [2.4 Nhật ký sự kiện Event-Sourced Session Log](#24-nhat-ky-su-kien-event-sourced-session-log)
- [2.5 Cơ chế cô lập an toàn OS Sandboxing](#25-co-che-co-lap-an-toan-os-sandboxing)
- [Chương 3: Thực thi dòng lệnh CLI và tự động hóa ngầm (Headless Mode)](#chuong-3-thuc-thi-dong-lenh-cli-va-tu-dong-hoa-ngam-headless-mode)
- [3.1 Cú pháp dòng lệnh CLI dsh nâng cao](#31-cu-phap-dong-lenh-cli-dsh-nang-cao)
- [3.2 Tự động hóa với cờ --profile headless](#32-tu-dong-hoa-voi-co---profile-headless)
- [3.3 Tùy biến thư mục lưu trữ qua cờ --dsh-home](#33-tuy-bien-thu-muc-luu-tru-qua-co---dsh-home)
- [3.4 Tích hợp vào quy trình CI/CD tự động](#34-tich-hop-vao-quy-trinh-cicd-tu-dong)

- [Chương 4: Lập trình phát triển Custom Plugin trên Cordis Framework](#chuong-4-lap-trinh-phat-trien-custom-plugin-tren-cordis-framework)
- [4.1 Cấu trúc mã nguồn một Plugin chuẩn TypeScript](#41-cau-truc-ma-nguon-mot-plugin-chuan-typescript)
- [4.2 Đăng ký công cụ mới (Tool Registry) cho Agent](#42-dang-ky-cong-cu-moi-tool-registry-cho-agent)
- [4.3 Lắng nghe sự kiện và xử lý vòng đời Plugin](#43-lang-nghe-su-kien-va-xu-ly-vong-doi-plugin)
- [4.4 Nạp Plugin và tệp vá cấu hình --patch](#44-nap-plugin-va-tep-va-cau-hinh---patch)
- [Chương 5: Case study 1 – Tự động hóa giáo trình và kịch bản CSKH doanh nghiệp](#chuong-5-case-study-1--tu-dong-hoa-giao-trinh-va-kich-ban-cskh-doanh-nghiep)
- [5.1 Đặt bài toán nghiệp vụ CSKH chuẩn hóa](#51-dat-bai-toan-nghiep-vu-cskh-chuan-hoa)
- [5.2 Thiết lập luồng xử lý Agent tự động hóa tài liệu](#52-thiet-lap-luong-xu-ly-agent-tu-dong-hoa-tai-lieu)
- [5.3 Chuẩn hóa kho kịch bản tư vấn đa kênh (Call, Chat, Zalo, Email)](#53-chuan-hoa-kho-kich-ban-tu-van-da-kenh-call-chat-zalo-email)
- [Chương 6: Case study 2 – Dây chuyền sản xuất Video AI tự động bằng Python & FFmpeg](#chuong-6-case-study-2--day-chuyen-san-xuat-video-ai-tu-dong-bang-python--ffmpeg)
- [6.1 Kiến trúc hệ thống sản xuất video tự động](#61-kien-truc-he-thong-san-xuat-video-tu-dong)
- [6.2 Khung mã nguồn Python gọi Voice API sinh giọng thoại](#62-khung-ma-nguon-python-goi-voice-api-sinh-giong-thoai)
- [6.3 Script ghép khung hình và render video hoàn chỉnh qua FFmpeg](#63-script-ghep-khung-hinh-va-render-video-hoan-chinh-qua-ffmpeg)
- [Chương 7: Kỹ thuật khắc phục sự cố hệ thống (Troubleshooting Guide)](#chuong-7-ky-thuat-khac-phuc-su-co-he-thong-troubleshooting-guide)
- [7.1 Xử lý lỗi npx bị chặn bởi Execution Policy trên Windows PowerShell](#71-xu-ly-loi-npx-bi-chan-boy-execution-policy-tren-windows-powershell)
- [7.2 Xử lý lỗi trùng cổng EADDRINUSE: address already in use 127.0.0.1:3080](#72-xu-ly-loi-trung-cong-eaddrinuse-address-already-in-use-1270013080)
- [7.3 Khắc phục sự cố hết credit API Key và cách gia hạn](#73-khac-phuc-su-co-het-credit-api-key-va-cach-gia-han)
- [Chương 8: Bộ tài liệu kiểm thử thực chiến (Testing Playbook)](#chuong-8-bo-tai-lieu-kiem-thu-thuc-chien-testing-playbook)
- [Chương 9: Hướng dẫn chụp ảnh minh họa chi tiết](#chuong-9-huong-dan-chup-anh-minh-hoa-chi-tiet)
- [Chương 10: Kịch bản video khóa học thực hành đỉnh kèm](#chuong-10-kich-ban-video-khoa-hoc-thuc-hanh-dinh-kem)

Chương 1: Tổng quan, triết lý và khởi động với DeepSeek Harness

1.1 Khái niệm DeepSeek Harness và vị thế năm 2026

Trong kỷ nguyên phát triển bùng nổ của trí tuệ nhân tạo năm 2026, rào cản lớn nhất không còn nằm ở khả năng của mô hình ngôn ngữ lớn (LLM), mà nằm ở hạ tầng khung dựng (scaffolding) cho Agent. Một mô hình AI dù thông minh đến đâu cũng không thể tự hoạt động hiệu quả nếu thiếu hệ thống quản lý ngữ cảnh, công cụ thao tác tập tin và môi trường thực thi an toàn.

DeepSeek Harness (dòng lệnh CLI là dsh) được DeepSeek AI trình làng vào tháng 8/2026 như một giải pháp toàn diện cho bài toán này. Đóng vai trò là một Agent Runtime nguồn mở, DeepSeek Harness cung cấp môi trường thực thi lý tưởng giúp chuyển hóa mô hình ngôn ngữ thô thành các Agent phần mềm thực thụ.

Các đặc điểm cốt lõi của DeepSeek Harness:

- Tối ưu hóa token bằng Prefix Caching: Giảm tới 60–80% chi phí gọi API nhờ khả năng tự động lưu trữ và tái sử dụng tiền tố ngữ cảnh (context prefix).
- Hệ thống nhật ký minh bạch (Event-Sourced Session Log): Ghi lại mọi sự kiện theo dạng append-only, cho phép xem lại toàn bộ chuỗi suy nghĩ và hành động của Agent.
- Cơ chế an toàn cô lập cấp OS: Tích hợp bộ cô lập `bwrap` trên Linux và `Seatbelt` trên macOS, đảm bảo Agent không bao giờ can thiệp trái phép vào hệ thống host.

1.2 So sánh DeepSeek Harness với các AI Agent Framework khác

| Tiêu chí | DeepSeek Harness (dsh) | LangChain / LangGraph | AutoGen | Antigravity Agent |

| :--- | :--- | :--- | :--- | :--- |

| Kiến trúc cốt lõi | Plugin-based (Cordis) | Graph-based (DAG) | Multi-agent conversation | Custom Skill System |

| Môi trường thực thi | Tích hợp sẵn Web UI & Headless CLI | Cần tự xây dựng UI | Cần tự dựng UI | Tích hợp sẵn Antigravity IDE |

| Bảo mật Sandbox | Cô lập cấp hệ điều hành (bwrap/Seatbelt) | Phụ thuộc Docker | Phụ thuộc Docker | Môi trường hệ thống an toàn |

| Khả năng đổi mô hình | Model Agnostic (40+ providers) | Hỗ trợ nhiều LLM | Hỗ trợ nhiều LLM | Đa dạng mô hình tích hợp |

| Độ phức tạp cài đặt | Cực kỳ đơn giản (npx chạy ngay) | Phức tạp, nhiều dependency | Phức tạp | Đã cài sẵn |

1.3 Triết lý kiến trúc "Everything is a Plugin" trên Cordis Framework

Khác biệt lớn nhất về mặt thiết kế của DeepSeek Harness so với các framework khác nằm ở việc ứng dụng Cordis Framework. Tất cả mọi thành phần cấu thành hệ thống — từ adapter kết nối mô hình AI, bộ công cụ đọc ghi tập tin, trình chạy terminal cho đến bản thân giao diện người dùng — đều được đóng gói dưới dạng các plugin độc lập.

```
flowchart TD
    subgraph Core ["Cordis Core Framework Runtime"]
        Ctx["Context Lifecycle Manager (ctx)"]
        LogBus["Event-Sourced Session Bus"]
    end

    subgraph Adapters ["Model & Execution Plugins"]
        ModelPlugin["Model Adapter Plugin\n(DeepSeek / OpenAI / Ollama)"]
        FSPlugin["Filesystem Tools Plugin"]
        TermPlugin["Terminal Execution Plugin"]
        SandboxPlugin["OS Sandbox Plugin\n(bwrap / Seatbelt)"]
    end

    subgraph UILayer ["User Interface Plugins"]
        WebUI["Web Interface (Port 3080)"]
        HeadlessCLI["Headless CLI Runner"]
    end

    Ctx --> Adapters
    Ctx --> UILayer
    Adapters --> LogBus
```

1.4 Chuẩn bị môi trường phần cứng và phần mềm

Để khởi chạy và phát triển trên DeepSeek Harness, bạn cần chuẩn bị:

1. Yêu cầu Node.js: Phiên bản Node.js `^22.19.0` hoặc `>=24.0.0`.

- Kiểm tra phiên bản Node.js hiện tại:

```
node -v
```

2. Quyền truy cập API Key: Chuẩn bị API Key từ DeepSeek Platform (`sk-...`) hoặc bất kỳ nhà cung cấp tương thích (OpenAI, Anthropic, Ollama).

1.5 Khởi chạy nhanh 60 giây với lệnh npx

Bạn không cần thực hiện clone repository hay biên dịch mã nguồn phức tạp. Chỉ cần mở Terminal và gõ:

```
npx @deepseek-ai/dsh web
```

Ngay lập tức, trình quản lý sẽ tải bản dựng runtime mới nhất và khởi chạy dịch vụ Web UI tại địa chỉ <http://127.0.0.1:3080>.

Chương 2: Thao tác giao diện Web UI và quản lý phiên làm việc

2.1 Tổng quan các khu vực chức năng trên Web UI

Khi truy cập vào <http://127.0.0.1:3080>, giao diện quản trị hiện ra bao gồm 4 khu vực chính:

```
+-----+
| [1] Header Bar: Logo DeepSeek Harness | Active Profile Selector | Settings (Gear) |
+-----+
| [2] Sidebar Left | [3] Main Agent Execution Workspace Panel |
| - Sessions History | User Prompt Input |
| - Plugin Manager | ----- |
| * fs-tools | Live Event Log (Timeline format): |
| * terminal-tools | [10:14:02] > Reading package.json ... OK |
| * os-sandbox | [10:14:05] > Executing `npm test` inside Sandbox ... ERR |
| - API Credentials | [10:14:09] > Applying patch to src/math.ts ... FIXED |
| | Agent Final Answer Display |
+-----+
| [4] Quick Action Footer Input Bar | [Send Action] |
+-----+
```

2.2 Cấu hình API Key và nhà cung cấp mô hình AI

Để kết nối DeepSeek Harness với mô hình AI:

1. Nhấn vào biểu tượng Settings (Gear) ở góc trên bên phải màn hình.
2. Tại mục Provider Credentials, chọn `DeepSeek` (hoặc `OpenAI`, `Anthropic`, `Ollama`).
3. Dán API Key của bạn vào ô `API Key` và chọn mô hình mặc định (ví dụ: `deepseek-chat` hoặc `deepseek-reasoner`).
4. Nhấn Save Configuration.

2.3 Quản lý các bộ Profile làm việc

DeepSeek Harness cung cấp khái niệm Profile giúp nhóm các plugin và cấu hình phù hợp với từng mục đích công việc:

- Web Profile (`web`): Bật đầy đủ giao diện đồ họa Web UI và các công cụ trực quan.
- Coding Profile (`coding`): Tập trung tối đa các công cụ đọc/ghi tệp, phân tích AST code và chạy test.
- Headless Profile (`headless`): Tắt giao diện đồ họa, tối ưu cho chạy tự động dòng lệnh.

2.4 Nhật ký sự kiện Event-Sourced Session Log

Mọi thao tác của Agent đều được lưu lại dưới dạng Event Log. Dữ liệu nhật ký được lưu tại thư mục mặc định `~/dsh/sessions/` dưới định dạng JSON Lines (.jsonl). Tính năng này mang lại lợi ích:

- Minh bạch hoàn toàn từng bước suy nghĩ của AI.
- Dễ dàng khôi phục phiên làm việc bị gián đoạn.

- Đảm bảo khả năng audit an toàn thông tin.

2.5 Cơ chế cô lập an toàn OS Sandboxing

Khi Agent cần thực thi các lệnh dòng lệnh (Shell command), DeepSeek Harness sẽ tự động đẩy lệnh đó vào môi trường Sandbox:

- Trên Linux: Sử dụng `bwrap` (Bubblewrap) tạo không gian tên (namespace) cô lập.
- Trên macOS: Sử dụng cơ chế `Seatbelt` hạn chế quyền truy cập đĩa cứng và mạng.

Chương 3: Thực thi dòng lệnh CLI và tự động hóa ngầm (Headless Mode)

3.1 Cú pháp dòng lệnh CLI dsh nâng cao

Bên cạnh giao diện Web UI, công cụ dòng lệnh dsh cung cấp đầy đủ cờ lệnh điều khiển:

```
# Cú pháp tổng quát
dsh [--profile <name>] [--patch <filepath>] [--dsh-home <dir>] <command|prompt>
```

Các cờ lệnh phổ biến:

- `--profile`: Chỉ định profile làm việc (mặc định là `web`).
- `--patch`: Nạp tệp và cấu hình tùy biến YAML/JSON.
- `--dump-config`: In toàn bộ cây cấu hình plugin hiện tại ra màn hình.
- `--dsh-home`: Đổi thư mục lưu trữ cấu hình và session log.

3.2 Tự động hóa với cờ --profile headless

Để thực hiện tác vụ chạy ngầm một lần (One-off task) và tự động thoát khi hoàn thành:

```
npx @deepseek-ai/dsh --profile headless "Quét toàn bộ dự án, tìm các file .ts có lỗi type và tạo file fix-log.md"
```

3.3 Tùy biến thư mục lưu trữ qua cờ --dsh-home

Nếu bạn muốn lưu trữ dữ liệu làm việc của Agent vào một ổ đĩa riêng hoặc thư mục dự án:

```
npx @deepseek-ai/dsh --dsh-home D:\DeepSeekWorkspace\Data web
```

3.4 Tích hợp vào quy trình CI/CD tự động

Ví dụ tích hợp lệnh thực thi DeepSeek Harness vào tệp github-action.yml:

```
name: DeepSeek Automated Code Audit

on: [push, pull_request]

jobs:
  audit:
```

```

runs-on: ubuntu-latest
steps:
  - uses: actions/checkout@v4
  - uses: actions/setup-node@v4
    with:
      node-version: '22'
  - name: Run DeepSeek Harness Audit
    env:
      DEEPSEEK_API_KEY: ${ secrets.DEEPSEEK_API_KEY }
    run: |
      npx @deepseek-ai/dsh --profile headless "Check security vulnerabilities in src/ and output audit.md"

```

Chương 4: Lập trình phát triển Custom Plugin trên Cordis Framework

4.1 Cấu trúc mã nguồn một Plugin chuẩn TypeScript

Một plugin trong DeepSeek Harness bao gồm một tên định danh (name) và một hàm khởi tạo (apply):

```

import type { Context } from "@deepseek-ai/cordis";

export const name = "custom-slack-plugin";

export function apply(ctx: Context) {
  ctx.on("ready", () => {
    console.log("[Plugin Loader] Khởi tạo thành công Slack Plugin!");
  });
}

```

4.2 Đăng ký công cụ mới (Tool Registry) cho Agent

Để cung cấp thêm một công cụ mới cho Agent sử dụng:

```

import type { Context } from "@deepseek-ai/cordis";

export const name = "custom-database-tool";

export function apply(ctx: Context) {
  ctx.plugin("tool-registry", {
    name: "query_user_stats",
    description: "Truy vấn thống kê người dùng mới từ cơ sở dữ liệu",
    async execute(params: { startDate: string; endDate: string }) {
      const stats = {
        totalUsers: 1420,
        activeUsers: 890,
        period: `${params.startDate} -> ${params.endDate}`
      };
      return { success: true, data: stats };
    }
  });
}

```

Chương 5: Case study 1 – Tự động hóa giáo trình và kịch bản CSKH doanh nghiệp

5.1 Đặt bài toán nghiệp vụ CSKH chuẩn hóa

Một doanh nghiệp dịch vụ kỹ thuật cần xây dựng bộ tài liệu đào tạo chuẩn hóa cho 2 đối tượng chính:

- Đối tượng A (Nhân viên CSKH / Tổng đài viên): Đạt chuẩn KPI, SLA, FCR, CSAT trên 4 kênh giao tiếp (Tổng đài, Chat Zalo, Fanpage comment, Email).
- Đối tượng B (Quản lý phòng CSKH / Team Leader): Quản lý case tồn, rà soát chất lượng QA, báo cáo Power BI và quản trị rủi ro truyền thông.

5.2 Thiết lập luồng xử lý Agent tự động hóa tài liệu

DeepSeek Harness đọc dữ liệu thô từ các file CSV nghiệp vụ chuẩn hóa, tự động biên soạn thành bài viết chuẩn hóa chương-trình-đào-tạo-CSKH.md.

flowchart TD

```
RawCSV["File nghiệp vụ thô (CSV)"] --> DSHAgent["DeepSeek Harness Agent Scaffolding"]
DSHAgent --> ModuleA["Giai đoạn 1: Onboarding (Tuần 1-2)\n- A1. Giới thiệu Doanh nghiệp\n- A2. Kiến thức sản phẩm\n- A3. Hệ thống công cụ"]
DSHAgent --> ModuleB["Giai đoạn 2: Core (Tuần 3-8)\n- B1. Phân loại kênh\n- B2. Tư vấn nghiệp vụ\n- B3. Quy trình 5 bước xử lý case"]
DSHAgent --> ModuleC["Giai đoạn 3: Ongoing (Hàng tháng)\n- C1. Đào tạo lại chính sách\n- C2. Case bank phân tích\n- C3. Coaching 1-1"]
```

Chương 6: Case study 2 – Dây chuyền sản xuất Video AI tự động bằng Python & FFmpeg

6.1 Kiến trúc hệ thống sản xuất video tự động

Dựa trên mô hình tự động hóa tổng quát, DeepSeek Harness điều khiển toàn bộ dây chuyền sản xuất video tự động không cần can thiệp phần mềm dựng phim.

flowchart LR

```
Script["Kịch bản script.txt"] --> PythonTTS["Python tts.py\n(Voice API)"]
PythonTTS --> AudioFiles["Các file thoại seg0.mp3, seg1.mp3..."]
AudioFiles --> PythonImg["Python images.py\n(Tải ảnh minh họa)"]
PythonImg --> FFmpegBuild["Script build.ps1 / FFmpeg\n(Ghép video & burn.ps1 tạo phụ đề)"]
FFmpegBuild --> FinalMP4["File video hoàn chỉnh output_video.mp4"]
```

6.2 Khung mã nguồn Python gọi Voice API sinh giọng thoại (`tts.py`)

Mã nguồn Python dạng template chuẩn hóa (đã ẩn danh hóa API Key) giúp tự động chuyển kịch bản chữ thành giọng thoại:

```
import os
import requests
```

```
# Lấy API Key từ biến môi trường để bảo mật
API_KEY = os.environ.get("VOICE_SERVICE_API_KEY", "<YOUR_API_KEY_HERE>")
VOICE_ID = "<YOUR_VOICE_ID_HERE>"

def generate_speech(text, output_filename):
    url = f"https://api.elevenlabs.io/v1/text-to-speech/{VOICE_ID}"
    headers = {
        "Accept": "audio/mpeg",
        "Content-Type": "application/json",
        "xi-api-key": API_KEY
    }
    data = {
        "text": text,
        "model_id": "eleven_multilingual_v2",
        "voice_settings": {"stability": 0.5, "similarity_boost": 0.75}
    }
    response = requests.post(url, json=data, headers=headers)
    if response.status_code == 200:
        with open(output_filename, "wb") as f:
            f.write(response.content)
        print(f"[OK] Đã tạo file âm thanh: {output_filename}")

# Tự động đọc kịch bản và sinh từng phân đoạn
if __name__ == "__main__":
    if os.path.exists("script.txt"):
        with open("script.txt", "r", encoding="utf-8") as f:
            lines = f.readlines()
        for idx, line in enumerate(lines):
            if line.strip():
                generate_speech(line.strip(), f"./audio/seg{idx}.mp3")
```

6.3 Script ghép khung hình và render video hoàn chỉnh qua FFmpeg (`build.ps1`)

Lệnh PowerShell mẫu thực thi FFmpeg để tạo video hoàn chỉnh:

```
# build.ps1 - Ghép các phân đoạn hình ảnh và âm thanh thành video base
ffmpeg -y -loop 1 -i ./img/img0.jpg -i ./audio/seg0.mp3 -c:v libx264 -tune stillimage -c:a aac -b:a 192k -pix_fmt yuv420p -shortest ./work/seg0.mp4
ffmpeg -y -loop 1 -i ./img/img1.jpg -i ./audio/seg1.mp3 -c:v libx264 -tune stillimage -c:a aac -b:a 192k -pix_fmt yuv420p -shortest ./work/seg1.mp4

# Burn phụ đề chuẩn tiếng Việt vào video cuối
ffmpeg -y -i ./work/base.mp4 -vf "subtitles=./work/subs.srt:force_style='Fontname=Segoe UI,FontSize=18,PrimaryColour=&H00FFFFFF,OutlineColour=&H00000000'" ./out/output_video.mp4
```

Chương 7: Kỹ thuật khắc phục sự cố hệ thống (Troubleshooting Guide)

7.1 Xử lý lỗi npx bị chặn bởi Execution Policy trên Windows PowerShell

- Triệu chứng lỗi:

npx : File C:\Program Files\nodejs\npx.ps1 cannot be loaded because running scripts is disabled on this system.

- Nguyên nhân: Windows PowerShell chặn thực thi file `.ps1` theo chính sách bảo mật mặc định.
- Khắc phục triệt để:
- *Cách 1 (Chạy qua CMD):* Sử dụng cú pháp `cmd.exe /c "npx @deepseek-ai/dsh web"`
- *Cách 2 (Bypass PowerShell):* Sử dụng cú pháp `powershell -ExecutionPolicy Bypass -Command "npx @deepseek-ai/dsh web"`
- *Cách 3 (Dùng npx.cmd):* Gõ trực tiếp `npx.cmd @deepseek-ai/dsh web`

7.2 Xử lý lỗi trùng cổng EADDRINUSE: address already in use 127.0.0.1:3080

- Triệu chứng lỗi:

Error: listen EADDRINUSE: address already in use 127.0.0.1:3080

- Nguyên nhân: Đã có một tiến trình `dsh` hoặc `node.exe` đang chạy ẩn và chiếm dụng cổng 3080.
- Khắc phục:
- Không cần chạy lại lệnh `npx` mới. Tiến trình trước đó đã khởi chạy thành công, bạn chỉ cần mở trình duyệt và truy cập thẳng link: `http://127.0.0.1:3080`.
- Nếu muốn tắt tiến trình cũ để restart: Mở Task Manager hoặc gõ lệnh PowerShell: `Stop-Process -Name node -Force`.

7.3 Khắc phục sự cố hết credit API Key và cách gia hạn

- Triệu chứng: Agent dừng làm việc và báo lỗi kết nối API hoặc từ chối phản hồi.
 - Bản chất: DeepSeek Harness là phần mềm mã nguồn mở miễn phí 100%. Việc hết credit là do tài khoản API Key nhà cung cấp mô hình bị hết số dư.
 - Cách gia hạn:
1. Đăng nhập trang quản lý API Key chính hãng:
 - DeepSeek Platform: platform.deepseek.com -> vào mục Top up để nạp thêm tiền.
 - OpenAI Platform: platform.openai.com
 2. Mở giao diện `http://127.0.0.1:3080` -> vào Settings (Bánh răng) -> dán API Key mới -> nhấn Save.

Chương 8: Bộ tài liệu kiểm thử thực chiến (Testing Playbook)

Bảng dưới đây là tập hợp các lệnh và prompt mẫu đã kiểm thử thành công 100% trên môi trường làm việc thực tế:

STT	Tác vụ kiểm thử	Cú pháp lệnh Terminal / Giao diện	Prompt mẫu copy-paste	Kết quả kỳ vọng
1	Mở giao diện Web	`npx.cmd @deepseek-ai/dsh web`	*(Cấu hình API Key trong Settings)*	Mở Web UI tại `http://127.0.0.1:3080`
2	Phân tích file CSV	Web UI	"Đọc tệp `ngniepvu.csv` và tạo file tóm tắt kịch bản xử lý khiếu nại"	Xuất file Markdown tóm tắt nghiệp vụ CSKH
3	Thao tác sản xuất Video	Terminal	`python tts.py`	Tự động sinh file thoại MP3 từ kịch bản
4	Render video FFmpeg	PowerShell	`powershell ./build.ps1`	Xuất file video MP4 hoàn chỉnh kèm phụ đề
5	Dump cây cấu hình	`npx @deepseek-ai/dsh --dump-config`	*(Không có prompt)*	In toàn bộ danh sách plugin JSON ra console

Chương 9: Hướng dẫn chụp ảnh minh họa chi tiết

Để cuốn Ebook hoàn thiện về mặt thị giác, dưới đây là danh sách 8 ảnh chụp màn hình thực tế cần chuẩn bị:

- Ảnh 1 (Trang bìa Ebook): Ảnh thiết kế bìa Ebook với tiêu đề "Bách khoa toàn thư DeepSeek Harness toàn tập 2026", phối màu xanh mờ slate và xanh cyan.
- Ảnh 2 (Màn hình khởi động Terminal): Chụp màn hình Terminal khi gõ lệnh ``powershell -ExecutionPolicy Bypass -Command "npx @deepseek-ai/dsh web"``. Khoanh đỏ URL ``http://127.0.0.1:3080``.
- Ảnh 3 (Giao diện Web UI tổng quan): Chụp toàn cảnh màn hình Web UI. Vẽ khung chữ nhật đánh dấu 4 khu vực: Header, Sidebar, Workspace và Input bar.
- Ảnh 4 (Cấu hình API Key): Chụp pop-up Settings nơi điền API Key. Khoanh đỏ menu chọn Provider ``DeepSeek``.
- Ảnh 5 (Case study CSKH Doanh nghiệp): Chụp màn hình VS Code mở file ``chương-trình-đào-tạo-CSKH.md`` do DeepSeek Harness tạo ra.
- Ảnh 6 (Case study Video Automation): Chụp màn hình thư mục làm việc chứa tệp ``output_video.mp4`` và các phân đoạn MP3.
- Ảnh 7 (Khắc phục lỗi EADDRINUSE): Chụp màn hình báo lỗi cổng 3080 và mũi tên hướng dẫn mở trực tiếp trình duyệt ``http://127.0.0.1:3080``.
- Ảnh 8 (Mã nguồn Python TTS): Chụp màn hình VS Code hiển thị tệp ``tts.py`` gọi Voice API.

Chương 10: Kịch bản video khóa học thực hành đính kèm

Bộ Ebook đi kèm 5 kịch bản video khóa học thao tác thực hành (Screencast) dành cho người xem thích học qua video:

Video 1: Hướng dẫn cài đặt và khắc phục lỗi khởi chạy DeepSeek Harness

- Thời lượng: 3 phút.
- Lời thoại: "Chào mừng bạn đến với khóa học DeepSeek Harness. Trong video đầu tiên này, tôi sẽ hướng dẫn bạn cách khởi chạy Agent và xử lý triệt để lỗi Execution Policy trên Windows..."
- Thao tác màn hình: Mở Terminal, gõ `powershell -ExecutionPolicy Bypass -Command "npx @deepseek-ai/dsh web"`, mở trình duyệt truy cập `http://127.0.0.1:3080`.

Video 2: Thiết lập API Key tại platform.deepseek.com và chọn Profile

- Thời lượng: 3 phút.
- Lời thoại: "Hướng dẫn tạo API Key trên DeepSeek Platform, kiểm tra số dư credit và chọn Profile làm việc phù hợp..."
- Thao tác màn hình: Truy cập `platform.deepseek.com`, tạo key mới, dán vào Settings của Web UI.

Video 3: Case study thực tế: Tự động hóa tài liệu CSKH doanh nghiệp

- Thời lượng: 5 phút.
- Lời thoại: "Trong video này, chúng ta sẽ xem cách DeepSeek Harness đọc file CSV nghiệp vụ và tự biên soạn bộ tài liệu đào tạo CSKH chuyên nghiệp..."
- Thao tác màn hình: Trình chiếu các bước Agent đọc tệp CSV, tổng hợp thông tin và xuất file Markdown `chương-trình-đào-tạo-CSKH.md`.

Video 4: Case study thực tế: Tự động hóa sản xuất Video ngắn với Python & FFmpeg

- Thời lượng: 6 phút.
- Lời thoại: "Hướng dẫn tự động hóa dây chuyền sản xuất video ngắn: từ kịch bản văn bản đến sinh giọng nói AI và render video hoàn chỉnh qua FFmpeg..."
- Thao tác màn hình: Chạy script Python `tts.py`, kiểm tra các file MP3 và mở video sản phẩm `output_video.mp4`.

Video 5: Hướng dẫn xử lý sự cố trùng cổng 3080 và quản lý phiên làm việc

- Thời lượng: 4 phút.

- Lời thoại: "Hướng dẫn xử lý lỗi EADDRINUSE khi cổng 3080 đã chạy ngầm và cách khôi phục phiên làm việc cũ..."
- Thao tác màn hình: Kiểm tra danh sách tiến độ, mở lại giao diện trình duyệt và xem lịch sử Event Log.