



ADRIAN LV

Independent Developer
Founder of DuskRain

CONTACT

prdusk.com@gmail.com
duskrain.cn
github.com/ETO-ze

EDUCATION

Civil Aviation University of China

Undergraduate studies
Aircraft Propulsion
Engineering

Sep 2023 - Jun 2027
Expected graduation: Jun 2027

TECHNICAL SKILLS

Languages

Python, JavaScript, SQL
HTML/CSS, PowerShell

Web and backend

Vue 3, Vite, FastAPI, SQLite
REST APIs, OAuth

Desktop tools

PyInstaller, Windows utilities
USB CDC/UART
Device discovery and recovery

Raspberry Pi / embedded

Raspberry Pi 4B development
ESP32, MicroPython, I2C
OLED/LED, EasyEDA

SOFTWARE ENGINEERING

New Grad 2027 | Python & Full-Stack Development

PROFILE

Engineering undergraduate graduating in June 2027, targeting entry-level software engineering roles. Builds web applications, Windows utilities, and device integrations with Python, Vue 3, FastAPI, Raspberry Pi, and ESP32.

Founded DuskRain as a personal application platform. Project work covers implementation, deployment, account ownership, device recovery, and reproducible data workflows.

INDEPENDENT DEVELOPMENT

DuskRain | Founder and Independent Developer

2025 - Present | Personal technology platform

- Build and maintain personal application services, covering requirements, front-end and back-end implementation, deployment, releases, and ongoing maintenance.
- Configure access controls, reverse proxies, and runtime environments; separate public source from private credentials, configuration, and application data.

SELECTED PROJECTS

DuskRain Food Map

Full-stack dining and review application

Vue 3, Vite, FastAPI, SQLite, OAuth, Docker

github.com/ETO-ze/duskrain-food-map

- Built a dining database with map browsing, filters, mobile views, and an author workspace, combining AMap and Google POI data with locally managed ratings, photos, and Markdown reviews.
- Implemented visitor, author, and administrator permissions with stable account IDs for record ownership; added invitations, activation, password management, and Google/GitHub OAuth binding.
- Used Argon2id password hashing and OAuth state/PKCE; handled GCJ-02 and WGS84 coordinates and deployed behind a reverse proxy with Docker, separating runtime data and secrets from published source.

Codex OLED Status Display

Desktop software and embedded device integration

Python, MicroPython, RP2040-Zero, USB CDC, EasyEDA

github.com/ETO-ze/codex-oled-status-display

- Designed an OLED carrier PCB and developed MicroPython firmware to render task states, animations, and LED indicators on an RP2040-Zero and SH1106 display.
- Built a Python bridge that converts local Codex Desktop events into USB CDC messages; implemented serial discovery, reconnection, and state retransmission after device disconnects.
- Delivered a Windows preview, portable application, and installation/startup workflows; limited device messages to status fields and activity categories, excluding prompts, response bodies, and tool arguments.

ADRIAN LV

Projects and Additional
Experience

DATA AND EXPERIMENTS

Libraries and models

TensorFlow, NumPy, SciPy
XGBoost, Random Forest
HistGradientBoosting

Workflows

Data preprocessing
Model-training workflows
CLI/YAML configuration
Backtesting and evaluation
Signal windowing and
reconstruction

Integration

Third-party map and data APIs
Google/GitHub OAuth
Role permissions and record
ownership

Deployment

Linux, Windows, Docker
Docker Compose, Git/GitHub
Nginx/OpenResty
Reverse proxies

OTHER APPOINTMENT

Chengdu Miaoyiyou Technology Co., Ltd.

Supervisor
Corporate governance
appointment

HONORS

Third Prize, Provincial Round
12th East Money Cup
National College Student
Financial Challenge

LANGUAGES

Mandarin Chinese: Native
English: CET-4
(College English Test, Band 4)

PROJECT EXPERIENCE

Desktop Systems | Data Workflows | Applied ML Tooling

HoloCubic Hub

Windows companion tools and firmware stability patches

Python, ESP32, USB, LVGL, PyInstaller
github.com/ETO-ze/holocubic-hub

- Developed desktop tooling and firmware patches for the HoloCubic/AIO ecosystem, unifying device configuration, Codex status bridging, and background operation in one Windows executable.
- Diagnosed access to destroyed LVGL objects during application transitions and adjusted surface recreation; corrected the order of readiness and first-frame acknowledgments in screen sharing.
- Corrected stale Wi-Fi connection state, serialized device operations to reduce serial-port contention, and added USB discovery and offline clock synchronization while preserving upstream attribution.

OpenClaw + Freqtrade

Research workflow and runtime control platform

Python, Freqtrade, Vue 3, Docker, XGBoost
github.com/ETO-ze/freqtrade-local

- Connected local market-data caching, dynamic universe screening, model training, backtesting, and approved configuration synchronization to a cloud Freqtrade runtime.
- Integrated Random Forest, HistGradientBoosting, and XGBoost for factor experiments; screened candidates using liquidity, market capitalization, volatility, funding rates, and BTC/ETH market conditions.
- Implemented approval checks using backtest returns, profit factor, drawdown, trade count, and concentration; built a read-only Vue dashboard and local GUI for inspection, process control, and synchronization.

DDAE Optimized

Engineering refactor of a seismic denoising workflow

TensorFlow, NumPy, SciPy, CLI, YAML
github.com/ETO-ze/DDAE_optimized

- Refactored an existing notebook-based DDAE implementation into CLI scripts, YAML configuration, and reusable data-processing workflows for supervised and self-supervised/unsupervised training.
- Added epsilon handling, clipping, and an optional MSE anchor to correlation-based loss; implemented trace-wise window extraction and overlap-add reconstruction to standardize training and reconstruction.

Alpha Sentinel Local

Local monitoring deployment and service integration

Python, FastAPI, SQLite, Docker Compose, Streamlit
github.com/ETO-ze/alpha_sentinel_local

- Integrated a local monitoring deployment with periodic token-list retrieval, change detection, alert records, health/status endpoints, and data-source controls.
- Organized services with Docker Compose and SQLite persistence; integrated trading-pair metrics and candlestick views, keeping the complete UI and private configuration separate from the partial public source.