

# OUSSAMA ANEZHA

Wuhan, China | +86 181 7127 5325 | oussama.anezha@gmail.com

oussamaanezha.com | GitHub: OUSSAMAanz | LinkedIn

Software engineering undergraduate building intelligent applications, algorithms and data/modeling systems, with an emphasis on traceable implementation, testing and reproducible experiments.

## EDUCATION & EXPERIENCE

Wuhan University | BEng Software Engineering, Sep 2024 - Jul 2028 (expected) | GPA 3.57/4.0

Capgemini | AI Engineer Intern, 2026-present: data pipelines, factors, AutoML, text signals and validation.

Amazon | Backend Development Intern, Jan-Mar 2025: Spring Boot APIs, authentication, MySQL, Redis and logging.

DEVOTECH | Software Development Intern, Mar-Sep 2022: Python, JavaScript, C++ and Java development/testing.

## PROJECTS | APPLICATIONS, ALGORITHMS & SYSTEMS

Architecture-to-code agent | Python / OpenAI / PlantUML

Parsed 11 architecture views; built bounded tool loops, path checks, timeouts, tests and JSONL audit trails with a generated game example.

Financial news intelligence | React / FastAPI / FinBERT

Built news deduplication, source and sentiment storage, forecast artifacts and fallback tests; forecasting models did not beat holdout baselines.

Spark & Sprout AI tutor | React Native / TypeScript / FastAPI

Built lesson-context prompts, exercises, animated feedback and chat; speech endpoints are implemented, with device/provider validation pending.

DSA knowledge graph | C++14 / STL / Graphviz

Implemented template containers, trees and AVL rotations, BFS/DFS/Dijkstra, sorting, search and concept-graph export.

Economic simulation & distributed learning | Python / PyTorch / FastAPI

Led UML modeling in a team project; repository includes rule agents, a PPO parameter server, rollout workers, experience buffering and checkpoints.

Unity game project | C# / Unity

Contributed to a team game with character, camera, interaction and audio scripts; inspected inventory pickup, UI refresh and item-description logic.

Apache Spark architecture recovery | Architecture / UML

Owned repository setup, layered/Master-Worker pattern analysis and technical context documentation for a course architecture-recovery study.

China-market engine & bias audit | Python / AKShare / vn.py

Built MA20/60 rules, next-open execution, costs and cash sizing; corrected timing/universe bias and reconciled cross-engine execution.

## PROJECTS | DATA, STRATEGIES & MODELING (CONTINUED)

CSI 300 factors & earnings events | Pandas / scikit-learn

Built price/volume factors, Rank IC/ICIR evaluation and rolling train/validation/test splits with first-publication timing.

Call-auction imbalance | Python / JoinQuant

Implemented five-level queue filters, imbalance ranking, sizing and exits; retained platform-data and simulated-short limitations.

Sentiment & execution simulation | Python / JoinQuant

Combined breadth, turnover and margin indicators; simulated early profit-taking and delayed loss exits, not measured human behavior.

Institutional holdings analysis | Python / Pandas / unittest

Converted disclosures into inferred position changes, crowding and forward returns; tested publication alignment and distinguished disclosures from trades.

Unified adaptive A-share strategy | Factors / Exposure / Trend

Combined factor selection, participation and trend filters; institutional signal is a price-volume proxy and test years are not globally untouched.

BTC multi-timeframe, grid & channel | Freqtrade / Python / Docker

Compared strategy families with funding/costs, look-ahead and recursive checks, cross-asset tests, intratrade risk and long/short attribution.

BTC ZigZag / ATR trailing | C++ / Python / Freqtrade

Screened candidates in C++, validated in Freqtrade; used completed bars and one-way stops, labeling development-period results in-sample.

Read-only order-flow bridge | Python / Bookmap

Detected short-window sweeps and liquidity exhaustion; exported normalized events without order placement or account access.

Gold signal & exit experiments | MQL5 / Python / Pandas

Compared ZigZag/RSI/EMA signals and trailing exits on historical M1 data with costs and documented rejected hypotheses.

Macro & earnings event engines | Tick data / MQL5 / Python

Tested macro OCO/confirmation and slippage; earnings EA modeled session exits and gap-risk sizing with data-quality caveats.

Bilingual interactive portfolio | HTML / CSS / JavaScript

Built role filters, project detail views, animated characters, contact options and a custom domain in English and Chinese.

## RESEARCH & EARLIER WORK

In progress: online CVaR scenario reweighting (mirror descent/dynamic regret); LLM software verification. Quantum compiler metamorphic testing is proposed. Earlier records: PPO/LSTM trading, US30 ONNX EA and 8GB CNN orchestration; artifacts need verification, with no production/profit claims. Adaptive planning remains a concept.

## SKILLS & LANGUAGES

Python, Java, C++, C#, JavaScript/TypeScript; Pandas, scikit-learn, PyTorch, FastAPI, Spring Boot, React Native, SQL, Redis, Git. Arabic (native), Chinese (HSK4), English (IELTS6.0), French (TCF B2), German (B1).

Team projects do not imply sole authorship. Related variants are grouped; research/backtests are not evidence of live profitability.