

Laien Tang

Agent Full-stack Engineer / AI-native Full-stack Engineer

Phone: 19115620274 Email: tangartificial@126.com GitHub: github.com/LaneTang Portfolio: atomweb-mauve.vercel.app/zh

Profile

B.Eng. candidate in Automation with project experience in AI Agent prototypes, agent workflows, frontend content systems, and embedded real-time control. I use AI coding tools for requirement clarification, system design, task decomposition, implementation, acceptance review, and iteration, focusing on turning LLM capabilities into runnable, traceable, and evaluable product systems.

Education

Southwest University | B.Eng. in Automation Sep 2022 - Jun 2026
GPA: 3.78 / 4.0

Technical Skills

Agent Engineering: Multi-Agent Workflow, RAG, Tool Use, structured context, Prompt Engineering, Checker / Evaluator, JSON Schema, runtime logs and traces
Full-stack Development: Python, JavaScript / TypeScript, Node.js, Astro, React Islands, Tailwind CSS, HTML / SVG, Playwright, Vercel
Engineering Tools: Git, GitHub, Linux, npm, Zod, content schema validation, AI Coding Workflow, Codex / Claude Code / Cursor-style workflows
Robotics / Embedded Systems: C / C++, STM32, FreeRTOS, BLDC, FOC, PID, UART / CAN / I2C / SPI, DMA, basic ROS / MoveIt

Projects

GrowthLens Agent | Psychological Assessment Report Generation Agent Personal Project
Python / DeepSeek API / Multi-Agent / RAG / Checker / Playwright / JSON Schema

- Designed and implemented an assessment report generation agent that uses assessment JSON as the only factual source to generate parent-facing, traceable, non-diagnostic visual PDF reports.
- Split report generation into Retrieval, Planning, Metric Writing, Generation, Checking, and Patch stages, using structured JSON as intermediate artifacts between stages.
- Built an expert knowledge layer by distilling strong sample reports, expert guidance, review comments, and metric explanations into skills, rubrics, metric reference cards, and retrieval materials.
- Implemented ContractValidator + CheckerAgent quality gates to detect missing metrics, schema mismatch, level conflicts, diagnostic wording, and factual hallucination; preserved generation evidence via retrieval_trace, final_check, and run_summary.
- Built a local HTML / SVG / Playwright renderer to deterministically render final_report.json into HTML, PDF, PNG, and page-check outputs, decoupling text generation from PDF layout.

juhuo-ai-event-promo | AI Community Social Media Promotion Skill Personal Project
Skill / Fact Table / Quality Gate / Multi-platform Copy / HTML

- With one event plan and one instruction, generate a complete social media promotion package for the Juhuo AI community.** Designed a reusable Skill that treats the event plan as the SSOT and generates WeChat, Xiaohongshu, QQ copy, HTML layout, and visual-material prompts.
- Used Fact Table, Need Confirmation, Quality Gate, and final consistency pass to prevent the agent from fabricating time, venue, speakers, registration methods, QR codes, or event promises.
- Modularized the workflow into fact extraction, copy generation, HTML layout, visual prompt, and file manifest, forming a portable AI-native instruction pack.
- Outputs: mp.weixin.qq.com/s/9IAt3Ce2TSNTTQxYxJf6fg | mp.weixin.qq.com/s/5wo_6UtSM5-vh6ZXkvy3xw

AtomWeb | Personal System Frontend and Content Publishing System Personal Project
Astro / React Islands / Tailwind CSS v4 / Zod / SSG / Vercel

- Website: atomweb-mauve.vercel.app
- Built an Astro-based personal system frontend that models projects, ideas, long-form writing, and real-world actions as content domains such as works, ideas, thoughts, and impact, rather than as a static portfolio.
- Implemented a search -> timeline -> detail browsing loop, /works?inspect= inline inspect, timeline URL state synchronization, and a unified metadata/search foundation for works, ideas, and thoughts.
- Designed an external content source architecture based on PTA (Public Token Access), migrating content assets into the private atomweb-content repository while keeping schema, validation, routing, and rendering logic in the codebase; content schema validation, a published branch, and Vercel Deploy Hook form a lightweight publishing pipeline that decouples content maintenance from site engineering.
- Used AGENTS.md, release files, and npm run check:release to constrain agent-assisted development with explicit boundaries, version records, and acceptance criteria.

AerialRoboArm | DACMAS-based AI-native UAV Robotic Arm Engineering Project Project Lead
DACMAS / AI Coding Workflow / C / STM32F103 / FreeRTOS / BLDC / FOC / PID / ELRS / AS5600

- Applied my DACMAS methodology by documenting requirements, system architecture, module boundaries, context materials, action plans, and acceptance criteria, enabling AI coding tools to participate in real embedded engineering.
- Designed a BSP-Driver-Module-Task-App five-layer software architecture for an onboard robotic arm, using DataHub to decouple sensor acquisition, RC input, state machines, and motion-control tasks.
- Reworked SimpleFOC core logic in pure C with fixed-point arithmetic on a resource-constrained STM32F103; used LUT-based trigonometry to improve control-computation efficiency on a no-FPU MCU.
- Integrated BLDC motor, AS5600 magnetic encoder, ELRS remote control, and servo gripper drivers; optimized communication with DMA, I2C + semaphore, and UART + RingBuffer.
- Used real hardware integration as the acceptance loop, iterating on control frequency, communication stability, state-machine correctness, actuator response, and resource usage.

AI Coding and Engineering Methodology

- Proposed and practiced my personal DACMAS methodology in the second half of last year, making requirements, context, modules, action plans, and acceptance criteria explicit to address consistency and hallucination issues in AI Coding.
- Current development workflow: conduct research and write-fragments to organize ideas and build my own "brain context"; use grill-me / grill-with-doc / zoom-out / brainstorming to extract key context; then generate specs and plans under the Superpowers architecture.
- During execution, run long coding tasks for the main implementation, perform HITL result review, and then revise, version-control, and iterate based on the review results.
- Use README, AGENTS.md, release files, schema, checkers, traces, and execution artifacts to constrain AI collaboration and make Agent / AI Coding workflows closer to deliverable software engineering.

Community and Practice

Juhuo AI Community | Co-Founder

AI Community / Event Design / AI Tool Practice

- Co-founded and operated a campus AI research and practice community, organizing Claude Code salons, AI writing sessions, and AI Practice Month to promote AI coding, agent workflows, and AI-native toolchains, and to explore how AI can connect with real needs.
- Toolized community promotion, content production, and event review workflows, turning repeated operational work into reusable agent workflow projects.
- Organized, planned, and hosted multiple community salon events; in May 2026, helped plan the Juhuo AI Practice Month, with more than 300 offline participants.

Awards

- **National Second Prize**, Higher Education Press Cup National Undergraduate Mathematical Contest in Modeling, 2024
Responsible for bench-dragon kinematic modeling, spiral motion equation derivation, binary-search-based collision detection algorithm design, and paper writing.
- **Merit Award**, CAMCM International Mathematical Contest in Modeling, 2023
Responsible for computer-vision semantic segmentation implementation and paper writing.

Additional

- English: IELTS 7.0; able to read technical documentation, papers, and development materials, and communicate in English.
- Interests: AI Agent, Agent Infra, AI Coding Workflow, LLM Application, Embodied AI, AI-native Product Engineering.