

# Chih-Kai Wang

Software Engineering Intern Candidate | Python, TypeScript, AI Research Tools

[Email](#) | [GitHub](#) | [Portfolio](#) | Taipei, Taiwan

---

## PROFILE

I build developer tools and interactive research applications with Python and TypeScript. I am seeking software engineering and AI application internships, with an interest in reliable testing and mathematical software.

---

## EDUCATION

**National Taipei University of Education** | Expected 2028

B.S. student, Mathematics Education Division, Department of Mathematics and Information Education.

---

## SELECTED PROJECTS

**[Finite Witness](#)** | JavaScript, WebMCP, Web Workers

- Built an interactive workbench for exhaustive counterexample search over labeled simple graphs with 3-6 vertices, keeping enumeration off the main UI thread.
- Exposed the same application operations through eight WebMCP tools. Certificates retain the claim, graph metrics, and exact searched prefix for inspection.

**[HonestCI](#)** | TypeScript, Node.js, GitHub Actions

- Built a CLI and GitHub Action that checks JUnit evidence for missing reports, stale results, zero tests, and unexpected test-count drops against a trusted baseline.
- Published version 1.0.4 on npm and documented reproducible failure scenarios, including a runner that exits successfully without executing tests.

**[RigorGraph](#)** | Python, JSON Schema, Offline Reports

- Built version-controlled claim-evidence records, deterministic audits, and self-contained HTML reports, distributed through a Python CLI and GitHub Action.
- Added checks for changed evidence bytes, stale reviews, and invalid status transitions. Published package 1.0.1 on PyPI as a public beta.

---

## OPEN-SOURCE CONTRIBUTION

**[Codex Dream Skin: merged Windows verification fix](#)** | July 2026

- Narrowed native-window fallback handling so unrelated protocol errors remained failures, and repaired helper loading in standalone verification.

---

## RESEARCH SOFTWARE

**[ProofWeave Core](#)**: experimental structured claim certification with Lean. **[SAIR Proof Press](#)**: Lean-checked solvers with frozen artifacts and released-input evaluation summaries.

---

## TECHNICAL SKILLS

**Languages**: Python, TypeScript, JavaScript | **Tools**: Git, GitHub Actions, JSON Schema, MLX, LaTeX

**Formal methods**: Lean 4 / Mathlib (developing through research projects)

*Last updated: September 2026*