

Tikkitti



Capstone PROPOSAL

<https://github.com/KimTechHQ/Tikkitti>
KimTechHQ@pm.me



Project Proposal

Project: Tikkitti – A Personal Workflow Helper for IT Technicians

Prepared by: Kimberly Cano

Date: August 31, 2025

1. Executive Summary

Tikkitti is a personal workflow assistant designed to automate and streamline repetitive ticket documentation for IT technicians. Managing tickets is often time-consuming, requiring detailed logs of who was helped, where, when, and what actions were taken. Tikkitti reduces this overhead by automating data capture, assisting with ticket creation, and providing a dashboard for follow-ups.

The goal is to deliver a working prototype that demonstrates real-world efficiency improvements, making ticket documentation faster, more consistent, and easier to manage. By automating routine workflows, Tikkitti frees technicians to focus on solving problems instead of logging them.

2. Project Objectives

- Automate Ticket Documentation – Reduce manual input by capturing common ticket details (user, location, issue, resolution) through a quick-access form.
- Campus and Contact Auto-Detection – Use local network information and a profile table to infer campus location and likely user contact.
- Seamless Ticket Creation – Support Freshservice integration via API or automation tools (e.g., Playwright).
- Correction-Friendly Overlay – Provide an unobtrusive, transparent overlay to quickly adjust auto-filled information.

- Personal Dashboard – Display open tickets, follow-ups, VIP users, and unresolved issues in one view.
- Adaptive Learning – Improve accuracy over time by learning from user corrections and adjustments.

3. Project Scope

In-Scope:

- Development of a local desktop tool with hotkey-triggered quick capture form.
- Freshservice ticket integration through API or browser automation.
- Personal dashboard for monitoring open and follow-up tickets.
- Auto-detection of campus context and suggested contact information.
- Basic machine learning feedback loop for corrections.

Out-of-Scope:

- Enterprise-wide rollout or integration into all ticketing systems.
- Mobile app version.
- Large-scale multi-user deployment.
- Advanced analytics or reporting dashboards beyond personal use.

4. Deliverables

- Quick Capture Form & Hotkey – A functional, rapid-entry ticket form.
- Freshservice Integration – Working automation to create tickets directly in the system.
- Correction Overlay – A transparent UI overlay for reviewing and adjusting captured details.
- Personal Dashboard – Display of pending, follow-up, and VIP-related tickets.
- Documentation – User guide and technical write-up describing setup, functionality, and potential improvements.
- Capstone Presentation – Final project presentation and demonstration.

5. Project Timeline

Phase	Start Date	End Date	Duration
Proposal Submission	8/29/25	9/5/25	1 week
Project Planning	9/12/25	9/19/25	1 week
Development Phase 1: Quick Capture + API	9/19/25	10/3/25	2 weeks
Midterm Presentation	10/3/25	10/17/25	2 weeks
Development Phase 2: Dashboard + Overlay	10/24/25	11/7/25	2 weeks
Development Phase 3: Adaptive Learning	11/7/25	11/21/25	2 weeks
Testing & Refinement	11/21/25	12/5/25	2 weeks
Final Presentation	12/5/25	12/12/25	1 week

6. Risk Management

- Risk 1: API Access Limitations – Freshservice API may restrict certain actions.
- Mitigation: Use fallback browser automation (Playwright).
- Risk 2: Accuracy of Auto-Detection – Incorrectly guessed users or campuses could cause confusion.
- Mitigation: Provide correction overlay and allow quick edits.
- Risk 3: Time Management – Balancing development with coursework.
- Mitigation: Use phased milestones (aligned with midterm/final deliverables).
- Risk 4: Learning Curve for ML Component – Adaptive learning may take more time than expected.
- Mitigation: Keep ML basic (correction tracking) and defer advanced models to future work.

7. Conclusion

Tikkitti will streamline IT technicians' workflows by automating ticket documentation, providing intelligent suggestions, and centralizing follow-up tracking. This project not only addresses a real-world pain point but also demonstrates applied skills in automation, API integration, and usability-focused development.

By the end of the semester, the expected outcome is a functional prototype that saves time, reduces repetitive work, and highlights the potential for scaling into broader IT service management environments.

Acknowledgment of Tools Used

Sections of this proposal (grammar refinement, structure, and outline) were assisted by OpenAI's ChatGPT. All ideas, project design, and technical details were created and verified by the author.

Signature:

Kimberly Cano