*Please **rate the new Quick notes tab** by taking a short survey.*

AI Development Residency - Working Session

Oct 14, 2026

Morgan Denner a.nicole.wolf@gmail.com daudagafar83@gmail.com gelonb@gmail.com Joe Reynolds Katerina Berezina l_gomer@yahoo.com Marie-France Han mousavimehr.sarah@gmail.com orlacastillo@gmail.com p.panteleon@gmail.com Preeti Anubolu sara.roohan@gmail.com sasfizsofia@gmail.com sbeeguer@gmail.com shaydlgarza@gmail.com Shiyao Liu valeriametla@gmail.com Dominic Bradley ogalichuks7@gmail.com Xiaolu Bai Cameron Guest mlouischarles@gmail.com willsrobjr124@gmail.com jasmineelsal@gmail.com mgutin10@gmail.com Tanya Sharma vtephang@gmail.com barbsduch@gmail.com ksope2002@gmail.com lilyhalitov@gmail.com tech.kee23@gmail.com lskwok@mailbox.org

Class debrief covered GitHub mechanics and AI skill verification with Socratic learning strategies.

Foundational concepts and enterprise AI development

- Participants reviewed core concepts including web architecture, APIs, and MCP functionality.
- Learners contrasted informal vibe coding with structured enterprise AI development methodologies.
- An upcoming product management curriculum will launch during the next week of training.

AI agent skills verification and quality assurance

- Morgan explained that measurable verification receipts prevent unverified AI hallucinations and ensure skill compliance.
- Automated gate checks on GitHub run continuous integration scripts to catch coding errors early.
- Testing uncovered that automated checks produced false positives and false negatives 60% of the time.

Skill routing and local storage architecture

- AI agents route tasks by matching requests against skill description triggers stored locally.
- Skills require local computer storage within specific user directories rather than cloud environments.
- System hooks enforce always-on behavior and select appropriate skills for specific execution tasks.

Socratic learning and Claude integration

- Morgan outlined a socratic educational philosophy emphasizing self-directed technical instruction through Claude.
- Learners must build habits of evaluating AI responses and requesting plain language explanations.

GitHub repositories and open source licensing

- A repository functions as an application code bucket complete with full version history.
- The MIT license permits free copying, modification, and commercial distribution as open source software.
- Local work trees store cloned repository copies on user computers for development work.

GitHub workflow and terminal authentication

- Morgan demonstrated renaming files, creating pull requests, and merging changes into the main repository.
- Morgan walked through terminal authentication using gh auth login with HTTPS and browser login.
- Morgan highlighted cleaning up temporary local folders post-merge as standard development hygiene.

Customized Claude learning instructions

- Lando shared custom Claude master instructions designed to accommodate ADHD and simplify complex processes.
- Lando recommended having Claude suggest continuous updates to personal learning instructions after each session.

Model selection and credit management

- Morgan prioritized Claude 4.8 for coding tasks to optimize credit consumption and extend feature capacity.
- Sara noted that benchmark rankings place Gemini and Meta models at the top for programming tasks.

- Strict development standards enable enterprise-ready applications regardless of the underlying AI model version.

Working session schedule

- Morgan scheduled a 3-hour working session on Saturday for breakout groups and coding exercises.

Next steps

- [ ] [Morgan Denner] Write explanations: Draft written explanations for dense technical information to facilitate participant comprehension.
- [ ] [The group] Study concepts: Utilize Claude as a personal tutor to research and learn GitHub and software development concepts discussed in class.
- [ ] [The group] Study open source: Research open source licenses and their impact on software distribution during independent study time.
- [ ] [Lando C] Share Claude Instructions: Post the personalized Claude master instructions in the chat for all class members to use.
- [ ] [Morgan Denner] Provide Meeting Transcript: Distribute a written transcript of the current session to all class participants.
- [ ] [The group] Prepare Saturday Session: Determine the topics or specific work to present during the upcoming Saturday working session.

Want to see more? View the full notes
Tip: You can always access your full notes from the left sidebar.

You should review Gemini's notes to make sure they're accurate. Get tips and learn how Gemini takes notes

Oct 14, 2026

AI Development Residency - Working Session

Invited Morgan Denner a.nicole.wolf@gmail.com daudagafar83@gmail.com gelonb@gmail.com Joe Reynolds Katerina Berezina l_gomer@yahoo.com Marie-France Han mousavimehr.sarah@gmail.com orlacastillo@gmail.com p.panteleon@gmail.com Preeti Anubolu sara.roohan@gmail.com sasfizsofia@gmail.com sbeeguer@gmail.com shaydlgarza@gmail.com Shiyao Liu valeriametla@gmail.com Dominic Bradley ogalichuks7@gmail.com Xiaolu Bai Cameron Guest mlouischarles@gmail.com willsrobjr124@gmail.com jasmineelsal@gmail.com mgutin10@gmail.com Tanya Sharma vtephang@gmail.com barbsduch@gmail.com ksope2002@gmail.com lilyhalitov@gmail.com tech.kee23@gmail.com lskwok@mailbox.org

Attachments AI Development Residency - Working Session

Meeting records Recording

Summary

Class debrief covered GitHub mechanics and AI skill verification with Socratic learning strategies.

Initial Class Learnings
Students reviewed learnings on web mechanics and GitHub workflows. Discussion covered AI agent error rates and skill storage configurations.

GitHub Repository Setup
Demonstration covered repository creation and terminal authentication. Participants committed to using Claude for active self-teaching.

AI Model Management
Discussion compared Claude model versions and credit efficiency. Upcoming working sessions were announced for Saturday.

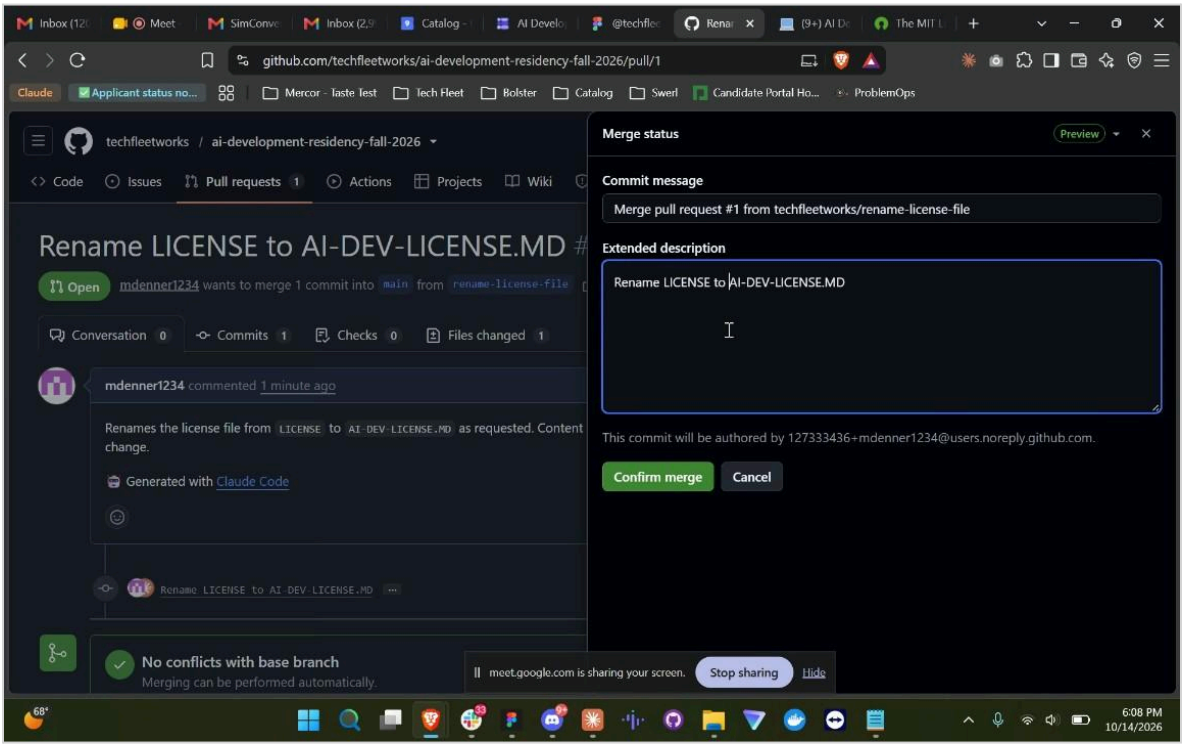
Next steps

- ☐ [Morgan Denner] Write explanations: Draft written explanations for dense technical information to facilitate participant comprehension.
- ☐ [The group] Study concepts: Utilize Claude as a personal tutor to research and learn GitHub and software development concepts discussed in class.
- ☐ [The group] Study open source: Research open source licenses and their impact on software distribution during independent study time.
- ☐ [Lando C] Share Claude Instructions: Post the personalized Claude master instructions in the chat for all class members to use.
- ☐ [Morgan Denner] Provide Meeting Transcript: Distribute a written transcript of the current session to all class participants.
- ☐ [The group] Prepare Saturday Session: Determine the topics or specific work to present during the upcoming Saturday working session.

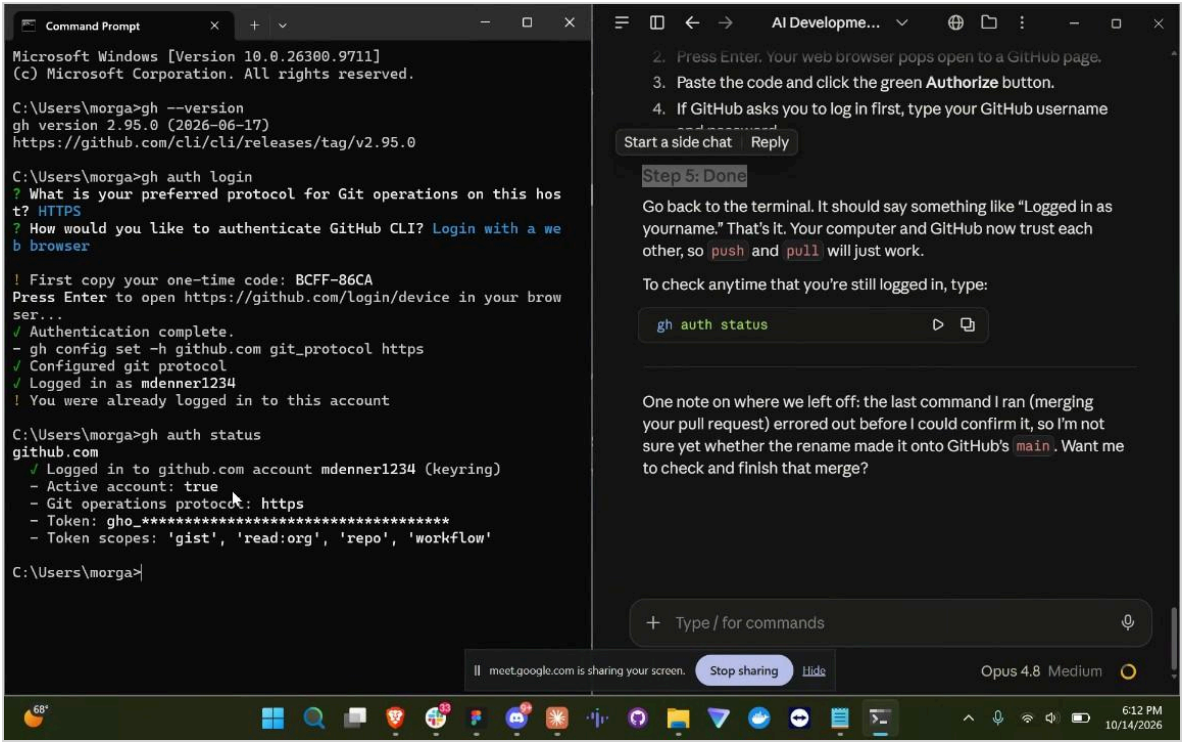
Details

Student Reflections on Initial Class Learnings: Sara Roohan shared using Claude to understand GitHub workflows. Shay Delagarza noted having zero coding background and learning internet, cloud, and GitHub fundamentals from scratch. Vichea Tephang discussed gaining a better understanding of web mechanics, two-tier and three-tier architectures, and how Claude works under the hood, planning to review big jams and projects for reinforcement. Tanya Sharma shared experience with open code and APIs, noting the benefit of learning basic internet concepts and instructing Claude to use plain language. Amanda Wolf found the first lecture a helpful refresher due to a background in computer information systems. Maria Gutin discussed learning web

mechanics, enterprise AI development versus wipe coding, and preparing for upcoming product management topics, while Morgan Denner noted that all students will take an entire product management course the following week.

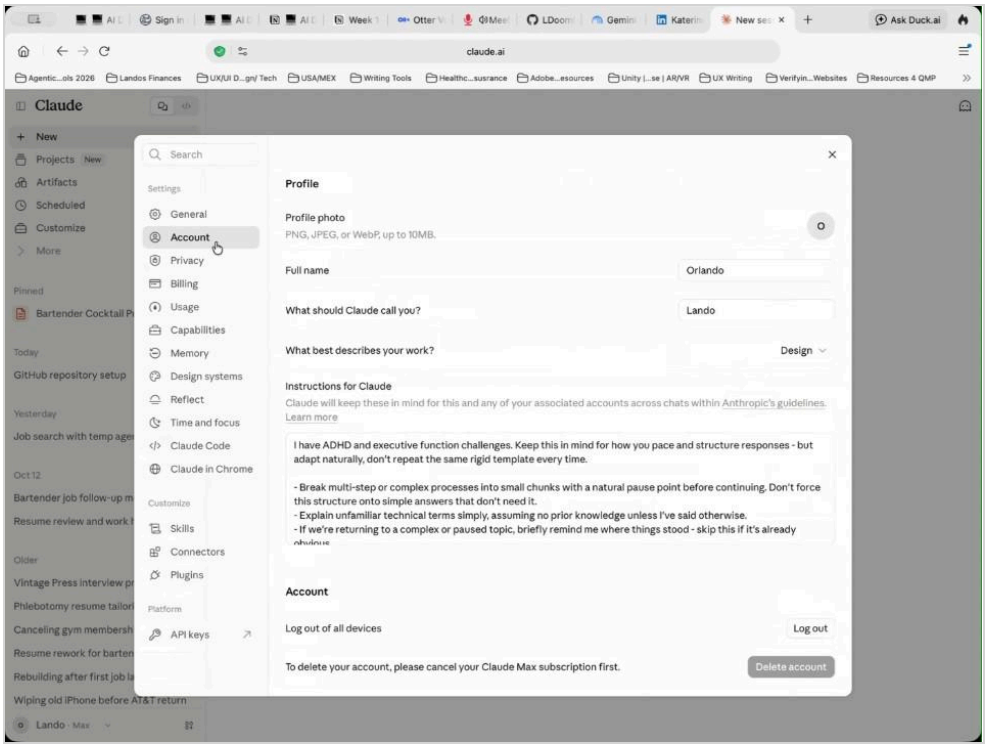


Introduction to GitHub and AI Skills: Morgan Denner introduced the evening's core focus on GitHub and AI skills, noting that secure shell keys are useful for remoting into servers. Morgan emphasized that current concepts will become foundational during weeks three through eight and advised that industry approaches to AI are still pioneering rather than master-expert driven.

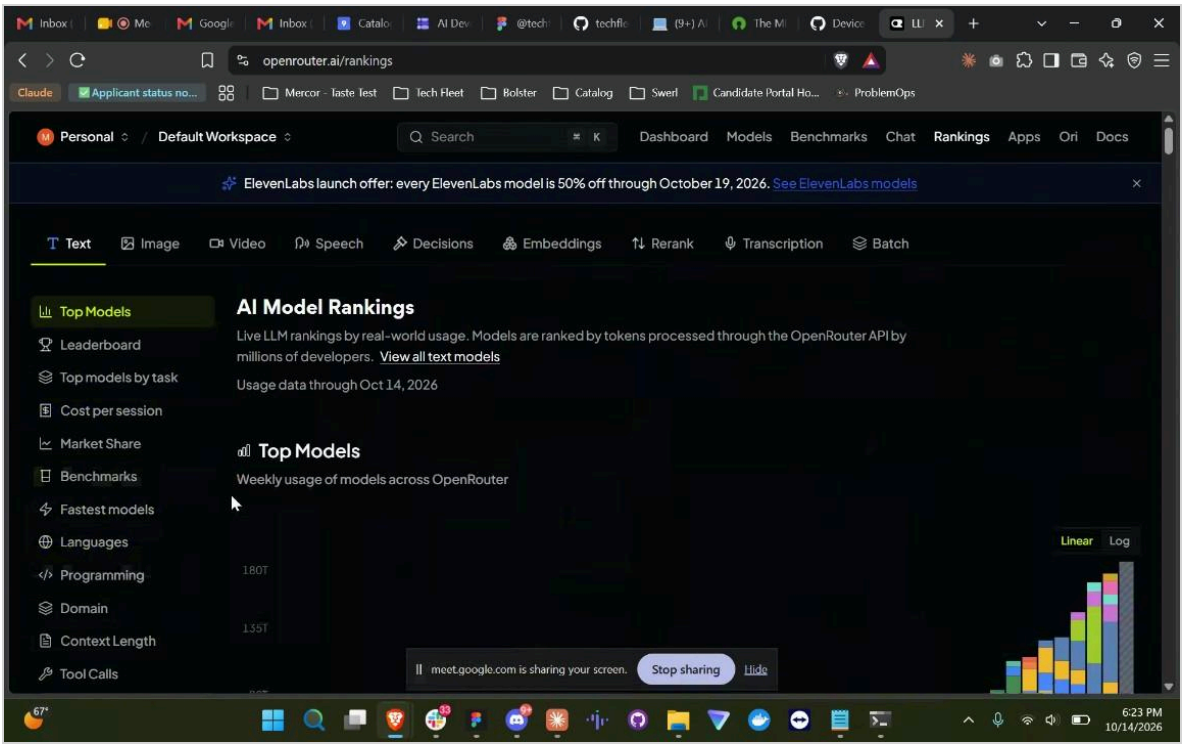


Challenges of AI Memory and Skills Verification: Morgan Denner discussed the problem of large language models lacking persistent memory and failing to consistently apply skill instruction files without hallucinating. Morgan described experimenting with a custom skill system that automatically reads skills, matches them to directives, and generates measurable verification receipts to prove that the agent actually executed the skill correctly.

GitHub Code Changes and Continuous Integration Checks: Morgan Denner explained local code repositories, pulling code, and pushing changes for platforms like TechFleet, mentioning over 400 to 500 code changes made. To prevent unknown bugs and regressions, Morgan detailed the use of continuous integration checks and gate checks on GitHub, which run scripts to test and verify code quality before pushing updates live.



Evaluating Agent Responses and False Positives: Dominic Bradley shared a personal practice of grouping related topics in desktop sessions and asking Claude to teach them through structured prompts. Morgan Denner and Dominic Bradley discussed the reliability of agent skill execution, with Morgan warning that agents frequently produce false positives and false negatives—reaching a 60% error rate in recent tests where 700 out of 1,400 tests failed—where they claim correctness despite errors, necessitating strict requests for empirical evidence and proof from agents.



Local Versus Cloud Skill Storage and Configuration: Lando C asked about the storage location of skills, and Morgan Denner clarified that skills for local Claude desktop must be stored in the local user's folder rather than synced with the cloud. Lando C noted success by pasting skills directly into project folders, while Morgan reiterated the importance of demanding evidence of skill utilization from agents.

Exploring Hooks and Custom Enforcement Systems: Paul Panteleon asked whether a combination of skills and hooks could ensure tasks are properly executed, and Morgan Denner confirmed that hooks and always-on settings can enforce instructions. Morgan described designing a custom system that uses machine-backed receipts to prove which skills were selected and applied.

Skill Routing, System Prompts, and Beginner Accessibility: Amanda Wolf discussed trying to set up routing behavior and system prompts in local models, noting a lack of similar direct visibility in Claude. Morgan Denner tested Claude on routing behavior and explained that routing depends on skill description triggers. Katerina Berezina expressed feeling overwhelmed by technical screen switches and

unfamiliarity with GitHub files, asking for foundational explanations of what GitHub does and how files work together.

Claude as a Primary Teaching Tool and Socratic Learning Dynamic: In response to student concerns about technical complexity and screen navigation, Morgan Denner instructed students to use Claude as their primary teacher to explore unknown concepts in immersive detail. Morgan emphasized a Socratic teaching dynamic where students drive exploration and practice rather than relying on Morgan for all answers, while Shiao Liu questioned whether to focus on foundational learning before working sessions. Morgan advised students to absorb what they can during sessions, review recordings, and use them as triggers for independent study.

Understanding GitHub Repositories and Licensing: Morgan Denner demonstrated setting up a GitHub repository, defining it as an isolated folder or bucket of code with built-in version history for applications like TechFleet. Morgan explained the difference between public and private repositories and discussed software licenses, specifically detailing the Massachusetts Institute of Technology license which allows open-source usage, copying, modifying, merging, publishing, distributing, selling, and licensing of code as a public good.

Setting Up Repositories with Claude Code and Local File Paths: Gelon Brumfield asked about private repository configuration, and Morgan Denner advised setting projects as private for commercial work while recommending consulting Claude regarding license safety. Using Claude Code, Morgan demonstrated setting up a new repository, prompting Claude to explain repository mechanics in sixth-grade terms by comparing them to folders with complete version history and pull requests. Finally, addressing Joe Reynolds, Morgan showed the local file path on the computer's documents folder where the repository work tree is stored.

Code Changes and Repository Merging: Morgan Denner demonstrates how to make a basic code change by renaming a file from `license.md` to `a.dev.license`, pushing the change from their local computer to GitHub, and executing a merge into the main code repository. The underlying problem involves understanding how code

transitions from local updates to live versions, and Morgan Denner walks through the process of opening pull requests and cleaning up temporary folders. The final consensus is that code changes are successfully merged into the main bucket through this structured workflow.

GitHub Authentication and Terminal Setup: Morgan Denner explains the process of authenticating GitHub on a local computer for first-time users using terminal commands such as `gh auth login`, selecting the HTTPS protocol, and logging in through the browser. Joe Reynolds assists by pointing out that the two-factor authentication code comes from the terminal, noting their specific code as 631398. Morgan Denner then demonstrates verifying authentication status with `gh auth status` and performing clean-up practices for temporary files. The decision is made to ensure all participants know how to authenticate their environments properly.

Commitment to Tool-Based Learning: Morgan Denner addresses the problem of students feeling stuck when facing unfamiliar technical hurdles and requests a verbal commitment from participants—including Dominic Bradley, Stephanie Beeguer, Cameron Guest, Katerina Berezina, Tanya Sharma, Joe Reynolds, and Gelon Brumfield—to use Claude to teach them everything they do not know and to call Claude out when it makes mistakes. The participants agree to this pledge by repeating after Morgan Denner, establishing a consensus around active self-teaching.

Customizing Claude Master Instructions for Learning Styles: Lando C raises the problem of cognitive overload and executive function challenges, suggesting that setting master instructions in Claude account settings helps tailor outputs to specific learning styles, such as breaking complex processes into small steps. Morgan Denner expresses enthusiasm and requests that Lando C share their screen and paste the custom instructions into the chat so others can benefit from them. The consensus is that participants can share and adopt these tailored instructions to improve their interactions with the AI tool.

Comparison and Selection of Claude Model Versions: Morgan Denner discusses the problem of managing credit consumption versus model performance, explaining

their preference for Claude version 4.8 over newer versions like 5.5 due to better credit efficiency and strong coding benchmarks that allow for more feature work. Tanya Sharma inquires about model naming conventions, and Morgan Denner introduces Open Router as an application that permits using multiple models behind the scenes. Sara Roohan looks up current programming rankings on Open Router, listing top models such as Gemini 4 argon, Meta's Muse Wonder, stepfun, and Anthropic's Claude 5.5 variants, while Sara Roohan and Morgan Denner discuss past memory issues and credit limits on newer models. The discussion concludes with Morgan Denner sharing personal strategy regarding credit conservation while acknowledging that model preferences vary.

Active Learning Philosophy and Upcoming Session Schedule: Morgan Denner highlights the problem of passive learning through textbooks or presentations, arguing that the age of static learning is over in favor of using AI tools for exploration and experimentation. Morgan Denner outlines the schedule for upcoming working sessions, announcing a three-hour working session on Saturday to allow for deeper coding and group collaboration. The meeting concludes with Morgan Denner advising participants to work on homework and reach out via email or Discord.

You should review Gemini's notes to make sure they're accurate. Get tips and learn how Gemini takes notes

*How is the quality of **these specific notes?** Take a short survey to let us know your feedback, including how helpful the notes were for your needs.*