Section 1

I will build a professional portfolio website to showcase the projects I'm most proud of. I chose this option because I feel that having a strong digital portfolio is as important as, if not more important than, my resume to future employers interested in my work. Displaying complex or novel projects proudly demonstrates my experience with, understanding of, and passion for the technologies I use, beyond the surface-level knowledge gained through my studies. While GitHub lets me show off my code, this portfolio helps employers gain a deeper understanding of my personality, experience level, and thought process, which helps me stand out from the hundreds of other faceless resumes.

Section 2

Welcome Statement: (1-2 sentences)

Welcome to my portfolio! Feel free to take a look around; there's a comment card at the bottom if you have any questions!

Purpose and Goals: (100-150 wc)

The purpose of this website is to serve as a digital portfolio, allowing me to showcase my proudest accomplishments, toughest challenges, and favorite learning experiences.

If you're an employer, I hope to show you who I am and why you should hire me, beyond bullet points on a resume. If you're a fellow technology lover, I hope to inspire you to follow your goals (that's cheesy). If we got into an argument on the internet, you googled me and found this: 1) I hope I was right; otherwise, this is awkward, and 2) I hope this shows you I know what I'm talking about.

Quote:

"Our greatest glory is not in never falling, but in rising every time we fall." - Confucius

Background Image:

I want to either embed my GitHub page or include a picture of me with my family.

Section 3

About me:

Hello! My name is Jackson. I'm 22 years old and currently a senior in Information Technology at Mizzou. I have experience building and deploying Python and C# applications on both desktop and web applications; you can check out my GitHub to see what I've been working on recently.

I currently work for Socket Fiber, where I wrote and maintain CalendarBuddy [link to projects page]. I also provide advanced networking, provisioning, and other technical support to field technicians as needed.

In my free time, I enjoy all things technology: writing software, playing video games, watching TV, building my homelab, and kitting out my "smart" home. I live with my fiancé, Emily, and our two children (cats), Oliver and Mae, who, quite frankly, are over my smart home shenanigans.

Education:
Classical Conversations - High School Diploma - 2021
Southwestern Illinois College - Dual Enrollment and Transfer Credits - 2020-2022
University of Missouri - Columbia - Bachelor's of Information Technology, Web and mobile dev cert, cybersecurity cert, information system cert, media and design cert - 2023-2026

Headshot:

Resume Video:

- Script:
    - Hello! My name is Jackson… I enjoy building solutions to make people's lives easier, typically my own, but hey, like Bill Gates said, "I will always choose a lazy person for a difficult job because a lazy person will find an easy way to do it". I hope to work on technologies that are going to define tomorrow, whether that's monitoring for SMRs or guidance systems for ICBMs. I love a good challenge, and I'm always up to learning something new to solve a problem. I'd love to connect if that's something you're interested in!
- Link: https://youtu.be/jdPK4rYgCBg

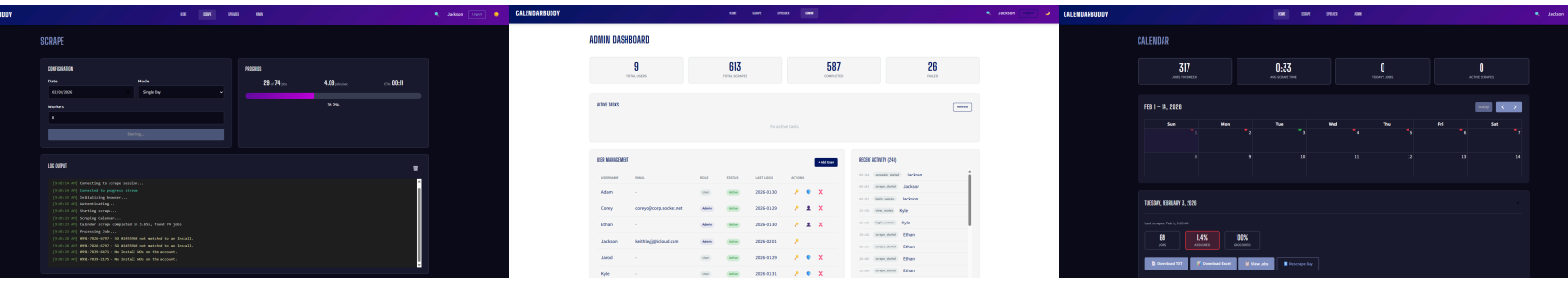
Section 4

Projects:

1. CalendarBuddy
    a. This is a Python-based automation platform I designed, developed, and deployed to save a collective 80 hours a week in manual data collection and interpretation. This project has 4 core parts: the CalendarBuddy, a FastAPI, Redis, Celery, and a PostgreSQL-based web interface; and three async, Playwright-based applications: the Scraper, Spreader, and Assigner, which are imported as modules in the CalendarBuddy's Docker environment. Each of these programs handles a different task and was independently developed before I decided to combine them into a single "centralized" program. The Scraper accesses my company's installation calendar, scrapes all relevant jobs, uses a worker pool to

process up to 16 jobs concurrently, and stores the information in the PostgreSQL database. The Spreader is a VRPTW (Vehicle Routing Problem with Time Windows) solver that helps users build optimized routes for technicians, allowing for fine-grained logic customization using a primarily imported configuration. Finally, the Assigner automatically processes returned assignments, making sure the job is assigned to the correct Technician and contracting company.

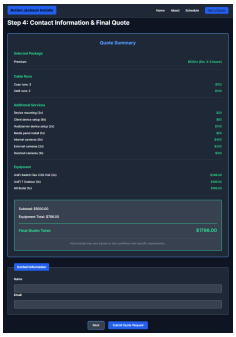
b. I am very proud of this project; in my opinion, this was and still is my only truly non-frivolous program. My team and I use it multiple times a day, and it has almost completely automated my job, which I got in no small part due to this suite of programs. This project has taught me everything I know about CI/CD, from deployment best practices and automated linters to finally realizing why everything is a web app these days.

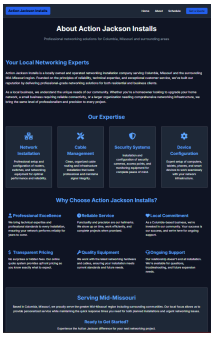


I'm working on adding the assigner to the website right now, ss will reflect later

2. ActionJacksonInstalls.com
    a. This is an EJS-based website for a potential business I've been pondering over the last year or so. I started this project around the time AI coding agents were really coming into the spotlight, so I decided an interesting spin on it would be to try to write as few lines of code as possible, acting more as a Software Architect or Senior Developer and letting the agents write the code and fix any bugs while I supervised. I used ChatGPT to develop a PRD document and an MVP, which I fed into the Claude Code CLI to write the code. In the CLAUDE.md, I specified that the linter, type checking, and Claude's built-in security review be performed before creating any commits. I then tied this to a commit webhook, which forces Claude to show that these were run before it can create the commit. I then set up a Codex integration with GitHub, so Codex performs an automated security and bug scan on every commit, providing detailed logs of any issues it finds. I only performed manual checks on the code during pull requests; the goal was to see how well the AI agents could manage on their own, given a strong set of documentation and structure. There's still a lot I want to add to this site, but for now, it's taken a backseat to other projects.
    b. I learned a lot from this project (EJS, Claude webhooks, MongoDB, S3-compatible storage, AI + CI/CD integration, domain acquisition, and website hosting), but overall, I'm not super pleased with the website itself. I ran into a lot of common issues with AI agents: bland/boring UIs; messy codebases; outright

lies in the website's documentation; tests that didn't actually test anything; broken authentication; and more. I would say my biggest takeaway from this project is just how far AI is from being able to "replace" software engineers.





3. Browse Guard
    a. This is my Mizzou IT Capstone. My team of 4, named MAD J, is currently developing a web-based browser extension that utilizes custom heuristics and machine learning to identify and rate potentially unsafe emails, websites, downloads, and links. Also included is a web-based portal that allows users to view their historical safety rating and the ratings for any user accounts they may have set up. Users will also be able to check existing passwords for security and generate new ones if needed. We're also toying with the idea of adding parental controls, but that is a stretch goal at this point.
    b. This project has taught me a lot about developing software as a team. While I was already familiar with Git commands and had a decent understanding of repository etiquette, working on a team made me much more conscientious about commit messages, branch usage, and pull requests. This is also the only extension I've developed, so I've learned all of that too.

Section 5

Personal Reflection

I'm only just beginning my career in software development, but my problem-solving skills have been developing for much longer, shaped by years of hands-on troubleshooting and self-directed learning. I'm at my best when faced with difficult challenges, especially ones without a clear or predefined solution. When something breaks, doesn't scale, or simply wastes time, my instinct is to understand how it works and then figure out how to make it better, regardless of whether that means writing code, reworking a system, or learning something entirely new.

My academic experience at Mizzou has provided an excellent foundation to grow these skills through structured labs, assignments, and projects. Coursework in software development, networking, cybersecurity, and systems design helped formalize knowledge I had previously picked up informally and exposed me to new ways of thinking about technology. Working within defined requirements and deadlines forced me to balance functionality, maintainability, and security; skills that translate directly to real-world development and team environments.

Outside the classroom, my personal and professional projects have played the biggest role in shaping my identity as a developer. Building tools like CalendarBuddy has given me a wealth of experience, from understanding how automation reshapes workflows to seeing how expectations change when a program transitions from a "project" to an enterprise solution. I've learned that writing code is often the easiest part of building software; documentation, team collaboration, deployment, and long-term planning are just as important, if not more so, once other people rely on your work.

Looking toward the future, my career goals place me working on the bleeding edge of technology. I'm drawn to complex systems, challenging problems, and environments where learning never really stops. While I know I still have a lot to learn, I'm confident in my ability to adapt quickly, take ownership of problems, and contribute meaningful solutions wherever I go.

This section was generated with the assistance of ChatGPT, mainly because I ran out of time to finish it.