NEBIYOU ESSAYAS DESSALEGN

612-275-8573 | nebiyou.dessalegn05@gmail.com | Minneapolis, MN | Vermillion, SD
linkedin.com/in/nebiyou-essayas-dessalegn | nebanebish.com | github.com/Nebishhh

EDUCATION

University of South Dakota | **Bachelor's in Computer Science** *Vermillion, SD*

Honors Program · Dean's List · Expected May 2027 · GPA 3.80

- **Coursework:** Algorithms & Data Structures, Software Engineering, Artificial Intelligence, Information Retrieval, Operating Systems

SKILLS

Languages: Python, TypeScript, JavaScript, SQL, HTML/CSS
Applied AI: AWS Bedrock, Claude/OpenAI/Gemini APIs, MCP, embeddings, semantic search (pgvector)
Frameworks: React, Next.js, Express.js, FastAPI, Streamlit
ML/Data: scikit-learn, pandas, NumPy, feature engineering
Tools: Docker, Git, PostgreSQL, Supabase, Vercel

EXPERIENCE

University of South Dakota, Information Technology Services | **Service Desk Analyst** *Vermillion, SD | Aug 2026 – Present*

- Resolve hardware, software, and network issues for students, faculty, and staff across Windows, Microsoft 365, VPN, and Wi-Fi, documenting each case in TeamDynamix
- Restore secure access by troubleshooting Duo MFA failures, triaging suspected account compromises and escalating per security protocol, and creating guest accounts in Active Directory
- Review Knowledge Base articles for accuracy using KCS and flag outdated or unclear content
- Built KB Finder (below) for my own lookups, then shared it with coworkers for testing; it addressed a long-standing gap in the existing KB search, reached the general manager, and is now actively used at the desk

Better Mobile Security | **Project Intern, Software & AI Engineering** *New York, NY | Dec 2025 – Jan 2026*

- Reverse-engineered a commercial auto-insurance rating model spanning 90,000+ interlinked Excel formulas into a pure-Python engine with automated input/output detection mapped to JSON schemas
- Shipped the engine as a Dockerized REST API with Model Context Protocol (MCP) support, making the rating model callable by services and LLM agents; developed against Claude via AWS Bedrock
- Built a Streamlit dashboard enabling non-technical stakeholders to visualize and test the model without touching code

Freelance Web Developer *Remote | 2025 – Present*

- Build and deploy marketing websites for local small businesses, owning the work from outreach to handoff
- Built fast, mobile-friendly sites in React and Astro with booking flows and a reusable design system

PROJECTS

KB Finder | **Semantic Search for IT Support** *Deployed | kb-finder.nebanebish.com*

- Built a semantic search tool for USD's 3,109 IT knowledge base articles that matches plain-language problems ("my laptop screen broke") to the right technical article, where keyword search often missed
- Tested it against keyword search before making it the default, with test cases locked in advance so results couldn't be tuned
- Next.js, TypeScript, Python, PostgreSQL + pgvector, OpenAI embeddings, Supabase, Vercel; access code, daily search cap, and no query logging to control cost and privacy

PL Analytics | **Premier League Machine Learning** *Deployed | pl-analytics.nebanebish.com*

- Built three ML tools in one web app from 1.89M player appearances: a market-value predictor, a match-outcome predictor, and a player-style finder (Python, scikit-learn, FastAPI, React)
- Caught a hidden data leak where a feature used post-match information, inflating results, and rebuilt it from past matches only
- Chose a model that actually predicts draws over a higher-scoring one that mostly skipped them

CookOut AI | **Recipe Scaling & Event-Planning Web App** *Personal Project · In Progress*

- Built a web app that scales recipes to any guest count, checks which dishes fit each guest group's dietary needs, and merges everything into one shopping list
- Added AI recipe import from text, URLs, and photos via the Gemini API; a regex parser handles common input first, with AI only as a fallback, secured by SSRF protection and file-type checks
- Full-stack TypeScript monorepo (React, Express, Prisma, SQLite) with per-user auth, unit and end-to-end tests, and a Docker setup for VPS deployment

Finish the Lyrics | **Music Trivia Web App** *Personal Project · In Progress*

- Building a trivia game where players pick a song's real next line from a set of believable fake ones
- Python seeding pipeline pulls song data from the iTunes and Genius APIs; Claude Haiku generates the decoy lines

Smart Home Energy Optimization *CSC 444 (IoT) · 5-Person Team*

- Handled feature engineering and modeling for a model predicting per-minute home energy use (R^2 0.93 on a chronological split); found ML-based control did not beat the rule-based baseline