Esther Cheng

esther.cheng@duke.edu | (267) 885-7068 | linkedin.com/in/estcheng | github.com/esther-cheng

Education

Duke University, Trinity College of Arts and Sciences, Durham, NC — Cumulative GPA: 3.99 — *May 2026*
Bachelor of Science in Computer Science & Psychology
National Merit Scholar and Dean's List with Distinction (Top 10%)

Relevant Courses: Data Structures & Algorithms; Computer Architecture; Discrete Math for Computer Science; Human Skills for Software Engineering; Quantitative and Qualitative Methods in User-Centered Research; Fundamentals of Decision Science; Technical and Social Analysis of Information and the Internet

Work Experience & Organizational Leadership

Duke University Innovation Co-Lab – Durham, NC — *August 2024 – Present*
Software Developer

- Performed full-life cycle software development on a large-scale project using AGILE methodology
- Built Tablet Interface webapp leveraging MongoDB and Next.js to track class attendance and course satisfaction for supervisors and attendees; implemented an admin interface to visualize data for class supervisors and administration
- Assisted students with various coding projects, helping with quality assurance, code review, and peer tutoring

Duke University Computer Science Department – Durham, NC — *January 2024 – Present*
ECE/CS 250 (Computer Architecture) – Undergraduate Teaching Assistant (UTA)

- Conducted weekly office hours and answered EDStem discussion questions to field students' concerns and help with homework assignments and projects using C, MIPS Assembly, and Logisim
- Led a weekly recitation section to review and clarify course content for around 20 students every semester

Duke University Housing and Residence Life – Durham, NC — *August 2023 – May 2024*
Craven Quad Council Communications Chair

- Managed a $40,000 budget to plan dorm-wide sporting and multi-cultural events for over 500 undergraduate residents

Technical Skills & Projects

Skills: Python, Java, C, MIPS Assembly, JavaScript/TypeScript (React), Next.js, Figma, MongoDB, Redis, Docker, Git, HTML/CSS

AI Course Advisor — **Python (Flask), TypeScript (React), MongoDB, Redis, Docker, Figma, CSS, HTML**
Duke University Code+ Program — *July 2024*

- Implemented **Duke Single Sign-On (SSO) Authentication** through Duke OAuth API with backend-for-frontend workflow using authentication tokens and HTTP cookies to increase user endpoint security and reduce system attack risk
- Integrated **OpenAI's API** (GPT-3.5) with Duke course catalog through retrieval-augmented generation (RAG) by creating vector embeddings of user queries and searching **MongoDB** database for semantic matches to provide a smoother class search function for students/advisors
- Improved chatbot responses with model finetuning and prompt engineering to increase reliability and accuracy by 15%
- Created new user interface using **Figma** to mockup, **Typescript/React** for functionality, and iterative user testing to design an accessible and user-friendly chatbot with features like Quick View of class information and a user survey
- Worked with Duke University Registrar, Academic Advising Center, and Student Information Services to develop search functionality for university resources to improve awareness of available on-campus support and increase usability

Flowcate — **Python, CSS, HTML, JavaScript**
HackDuke Award-Winning Project: Locating Menstrual Product Dispensers on Campus — *September 2023*

- Integrated **Django (Python)** with Arduino in order to link dispenser boxes and their locations back to a web server
- Designed interface that updates in real-time as products are dispensed with **GoogleMaps API**, **JavaScript**, and **Python**

Personal Website — **CSS, HTML, JavaScript**

- Designed a personal website utilizing **HTML**, **CSS**, and **JS** to showcase personal projects and extracurricular endeavors

Cache Simulator — **C**
Write-Through and Write-No-Allocate Cache Simulator — *December 2023*

- Utilized a dynamically-allocated list of block structures to simulate the functionality of a write-through, write-no-allocate cache with LRU replacement in **C**
- Used bit manipulation to implement math used to find tag, index, and block offset of memory addresses

Other Hobbies & Interests

Fluent in English & Mandarin; Crocheting, Weightlifting, Rock Climbing, Dance, Canine Behavior