Neil Gomes

202-257-9768 | neilgomes273@gmail.com | linkedin.com/in/neilgomes | github.com/neilgomes327

Education

University Of Maryland College Park, MD
Bachelor of Science in Computer Science *Aug. 2023 – December 2026*

Experience

Barista November 2020 – Present
Starbucks Coffee Company *Silver Spring, MD*

- Operated in a fast-paced, high-volume environment while consistently meeting strict time and accuracy requirements
- Managed multiple concurrent tasks under pressure, prioritizing customer orders during peak hours
- Communicated clearly with team members to coordinate order fulfillment and resolve issues quickly

Projects

Budgeting & Finance Web App | *JavaScript, HTML/CSS, LocalStorage* | GitHub Fall 2025

- Collaborated in a team of three to design and implement a client-side budgeting application with persistent state using browser `localStorage`
- Implemented expense tracking with validation, categorization, sorting by date, and dynamic balance updates
- Developed monthly spending summaries aggregating expenses by category and comparing income versus expenses
- Contributed to goal tracking and alert logic to notify users when balances fall below configurable thresholds

Blockchain Implementation | *Python, Cryptographic Hashing* Fall 2025

- Implemented a simplified blockchain data structure consisting of linked blocks secured with cryptographic hash pointers
- Designed block validation logic to ensure integrity by verifying hashes and previous-block references
- Incorporated proof-of-work style hashing to simulate computational difficulty during block creation
- Validated blockchain correctness by detecting tampering and rejecting invalid chains

SmallC Mini-Compiler | *OCaml, Recursive Descent, Re (regex)* Spring 2025

- Built a lexer using OCaml `Re` regular expressions to tokenize a SmallC (C-like) language, including keywords, operators, literals, and identifiers
- Implemented a recursive-descent parser that follows a provided grammar and produces an AST with correct operator precedence (e.g., `||, &&, ==, <, +, *, ^`)
- Developed an evaluator/interpreter to execute SmallC statements and expressions with an environment-based variable store and type checking for `int` and `bool`
- Added control-flow support (`if/else`, `while`, `for`, `printf`) and raised meaningful runtime errors (type errors, undeclared variables, divide-by-zero)

Mock Memory Allocator | *The C programming Language, ValGrind, GNU Debugger* Fall 2024

- Implemented a mock `malloc` function to manage heap memory using a doubly linked list representation
- Designed algorithms to search for appropriately sized free memory blocks and return pointers to allocated regions
- Handled memory splitting and reuse to reduce fragmentation within a simulated heap
- Debugged memory errors and leaks using Valgrind and GNU Debugger

Technical Skills

Languages: Java, Python, C, SQL (Postgres), JavaScript,HTML/CSS, OCaml
Frameworks: React, Node.js
Developer Tools: Git,Github, Docker, VS Code, Command Line tools
Libraries: pandas, NumPy, Matplotlib