

# Saturn Amarbat

773-802-9165 • samarb2@uic.edu • linkedin.com/in/saturn-amarbat-tech • github.com/saturn-amarbat

---

## EDUCATION

University of Illinois Chicago

December 2026

Bachelor of Engineering in Computer Science

Coursework: Machine Learning, Data Structures, Systems Programming, Software Design, Programming Practicum,

## SKILLS & TECHNICAL TOOLS

Languages: Python, Java, C, C++, x86-64 Assembly, JavaScript

Technologies: Git, GDB, Valgrind, PyCharm, VS Code, Linux/Ubuntu, Excel, Access

## EXPERIENCE

Office Assistant | Institute of Health Research and Policy

Chicago, IL | Nov 2023 - Present

- Systematized administrative workflows by devising standardized Excel templates and technical job aids, decreasing training ramp-up time for new hires by 50%
- Analyzed departmental expenditures using internal databases (EDDIE, Chrome River), generating detailed reports that improved purchase tracking accuracy and budget compliance.
- Orchestrated operational logistics for a research facility supporting 200–500 active staff, managing complex building maintenance logs and inventory budgets to ensure continuous research uptime.

## PROJECTS

### OpenStreetMap Navigation System | C++, Graph Algorithms, JSON, STL

- Architected a high-performance navigation backend processing 7,300+ vertices and 22,000+ edges, utilizing the JSON library to parse and construct graph topology from real-world OpenStreetMap JSON data.
- Implemented Dijkstra's Algorithm with priority queues and adjacency lists to execute sub-second path-finding queries on weighted graphs, optimizing for both memory efficiency and routing speed.

### Inverted Index Search Engine | C++, Data Structures, Query Processing

- Engineered a text search engine using an Inverted Index with Hash Maps and Sets to map tokens to documents, achieving O(1) lookup speeds for complex Boolean queries (AND, OR, NOT).
- Optimized query retrieval logic using set operations (union, intersection) and a custom text processing pipeline to sanitize and tokenize large lyric datasets for accurate results.

### Binary Analysis & Reverse Engineering | x86-64 Assembly, GDB, C, System Security

- Reverse-engineered a binary executable by analyzing x86-64 assembly dumps and stack frames to deduce control flow logic.
- Executed a precise buffer overflow attack to manipulate the runtime stack and bypass security checks, demonstrating memory safety mastery.