# Furnom Samuel Dam

furnomd1@umbc.edu|LinkedIn|Silver Spring, MD|GitHub

## EDUCATION

**University of Maryland, Baltimore County,** Baltimore, MD.
Graduation Date: May 2026.
*Bachelor of Science in Computer Science.*
**Relevant Coursework:** Data Structures, Algorithms, Computer Architecture, Operating Systems, Software Engineering, Artificial Intelligence.

## WORK EXPERIENCE

**Beta Computers Limited** Sep 2018
*Information Technology Intern*

- Observed and learned C# programming fundamentals under professional guidance.
- Developed early technical curiosity and foundational problem-solving skills.

## TECHNICAL SKILLS

**Languages:** Python, C/C++, JavaScript, Go, SQL, C#, TypeScript.
**Frameworks:** Flask, React, Tailwind, Gin, FastAPI, Node, Express, GraphQL, NextJS.
**Databases:** MongoDB, Redis, MySQL.
**Tools:** Git, Docker, Linux, AWS, Jira, Jest, Unittest, Postman, Swagger, Railway, GCP, Vercel, GitHub Actions.

## PROJECTS

- **GEPO** Jan 2026
  - Built a GitHub analytics platform using Next.js, Node.js/Express, GraphQL, MongoDB, and Redis; automated Docker deployments to Cloud Run via GitHub Actions and frontend to Vercel.
  - Instrumented Redis caching, achieving near 100% hit rate, reducing GitHub API calls, and improving response times.
  - Developed 68 Jest unit tests for middleware, services, and APIs, ensuring production-ready coverage.
- **Accessibility Map** Dec 2025
  - Collaborated in a 4-person engineering team to build a campus-wide accessibility navigation system for students and visitors with mobility needs.
  - Built backend authentication, threshold logic, and MySQL/MariaDB schema for 300+ accessibility nodes (doors, elevators, paths), enabling accurate routing across 17+ campus buildings.
  - Integrated FastAPI backend with the JS frontend and transitioned all services to Railway production APIs; resolved CORS, routing, and building-validation bugs to stabilize full-campus navigation.
  - Improved reliability and performance in production, processing 614+ API calls with <200 ms latency, <120 MB memory, and 0% error rate after final fixes.
- **Studivio** July 2025
  - Developed a full-stack study productivity and note-taking app using Flask, React, and MongoDB, implementing JWT authentication for secure user access.
  - Integrated OpenAI GPT, Whisper, and AssemblyAI APIs to process PDFs (≤ 60 pages / 25 MB) and audio files (≤ 1 hour / 50 MB) with near-instant summarization, enabling students to break down complex concepts into actionable points.
  - Instrumented backend performance with Prometheus across local test traffic (13 requests): ~36 ms GETs, ~15 ms POSTs, ~18 s PDF summaries; Python GC reclaimed ~14.4k objects.