JAMIL MOHAMMAD

Computer Science Undergraduate | American University of Beirut

Beirut, Lebanon • +961 71 106 455 • jamilm2006@gmail.com • LinkedIn • GitHub • jamilm.dev

PROFESSIONAL SUMMARY

Third-year Computer Science undergraduate at the American University of Beirut (GPA 3.64/4.00, Dean's Honor List). Comfortable across the stack: Java for object-oriented design and data structures work, TypeScript/React frontends, Node.js and Cloudflare Workers backends, C/C++ for systems and network programming, and Python for numerical work. Recent projects span full-stack web apps, OS kernel modifications, a networked multiplayer game over TCP/IP, and a from-scratch N-body physics simulator. Experienced in collaborative Agile team environments and motivated by software that solves concrete, real-world problems.

EDUCATION

American University of Beirut (AUB)

August 2024 – Present

Bachelor of Science in Computer Science | Minor in Mathematics | Minor in Philosophy GPA: 3.64 / 4.00

- Completed Coursework: Data Structures (A+), Linear Algebra (A+), Software Engineering (A+), Systems Programming (A), AI Ethics (A), Computer Organization & Design (A−), Operating Systems (B+), Advanced Algorithms & Data Structures (B), Random Variables & Probability and Statistics (C+)
- In Progress: Applied Cryptography
- Upcoming: Machine Learning, Extended Reality (XR), Numerical Computing, Computer & Information Security
- Dean's Honor List – Spring 2024–25 (Semester GPA: 4.22 / 4.30)
- Member of the AUB Honor Society – awarded for exceptional academic standing

PROJECTS

UniFlow – AI-Powered University Course Planner

Software Engineering – AUB, 2026

React • TypeScript • Node.js • Supabase • Clerk • Vercel • Render • Groq (Llama 3.3 70B) • Hugging Face • Playwright • GitHub • Live Demo

- Collaborated in a 4-member Agile team using Jira and Scrum with a rotating Scrum Master role; designed and fully deployed a web app helping AUB students plan course schedules and track degree progress
- Personally responsible for the full frontend (React + TypeScript), backend API (Node.js), and production deployment on Vercel and Render
- Single-handedly built a Playwright-based course fetcher that scrapes and parses live course data directly from AUBsis (AUB's student information system), keeping the platform up to date each semester
- Built a fully anonymous course and professor review system with a Groq-powered (Llama 3.3 70B) chat agent that summarizes aggregated review data, and automated content moderation via Hugging Face inference
- Implemented a suite of student tools including schedule planning, degree progress tracking, and AI-driven course recommendations and academic guidance
- Added full mobile support, ensuring a responsive and accessible experience across all screen sizes
- Implemented secure authentication and user management using Clerk, with Supabase (PostgreSQL) as the database

N-Body Gravitational Simulator

Independent Project, 2026

C++20 • RK4 Numerical Integration • CMake / Make • JSON Configs • Python / Matplotlib • FFmpeg • Cross-platform (Linux / Windows) • GitHub

- Designed and built a config-driven N-body gravitational simulator in C++20, modeling Newtonian dynamics with $O(N^2)$ pairwise forces and Plummer softening across arbitrary scenarios loaded from JSON (solar system, three-body figure-8, binary star + planet)
- Implemented a fixed-step fourth-order Runge-Kutta integrator from scratch, conserving total mechanical energy to $\approx 10^{-14}$ relative drift over 25,300 steps of the Chenciner–Montgomery three-body figure-8 orbit

- Wrote a Python visualization package (matplotlib) that generates trajectory plots, velocity-field quiver overlays, energy-drift diagnostics, and FFmpeg-rendered MP4 animations with fading trails
- Built cross-platform via both Makefile and CMake; validated numerically identical results on Ubuntu 24.04 and Windows 11 (MSYS2 / MinGW-w64)

xv6 Operating System Extensions

Operating Systems – AUB, 2026

C • xv6 (x86 and RISC-V) • QEMU • Ubuntu (VirtualBox) • GDB • Systems Programming • GitHub

- Replaced the default round-robin scheduler with a Multi-Level Feedback Queue (3 priority queues, exponentially-doubling 1/2/4-tick quanta, periodic priority boost to prevent starvation), modifying the PCB, context-switch path, and timer trap handler
- Added kernel-thread support via a clone() syscall sharing the parent address space, then built a user-space thread library with thread_create / thread_join and a ticket lock for FIFO mutual exclusion
- Extended the shell with wildcard expansion, command history (arrow-key recall), and a username/password login gate; added tree, diff, find (recursive pattern match), and an in-file keyword search tool

Networked Connect 4 in C

Systems Programming – AUB, 2025

C • Alpine Linux • POSIX Sockets (TCP/IP) • OpenMP • GDB • Valgrind • GitHub

- Built as part of a 3-member team; developed a full-featured console Connect 4 game in C supporting local 2-player, bot vs. player, and networked client-server modes
- Implemented TCP client-server networking using POSIX sockets (socket / bind / listen / accept / connect), letting two players run the game across separate machines on a LAN
- Built a Minimax bot opponent with alpha-beta pruning across three difficulty levels (easy / medium / hard), parallelizing the search's scoring and root-move evaluation with OpenMP; debugged with GDB and profiled with Valgrind for memory correctness
- Applied explicit memory management, Makefiles, and secure coding practices covered in CMPS 241; deployed and tested on Alpine Linux

The Moral Agent – Comparative Ethics AI Web App

AI Ethics – AUB, 2026

TypeScript • React • TanStack Start • Tailwind CSS • Cloudflare Workers • Google Gemini API • Groq API (Llama 3.3 70B fallback) • Server-Sent Events • Prompt Engineering • GitHub • Live Demo

- Designed, built, and deployed a full-stack TypeScript web app where a user-submitted ethical dilemma is sent in parallel to two AI agents — one reasoning from Kantian deontology, the other from Millian utilitarianism — with both responses streamed side-by-side in real time using Server-Sent Events
- Engineered multi-page system prompts grounded in course readings (O'Neill on Kant, Mill's Utilitarianism) that lock each agent into its framework's methodology and vocabulary, isolating moral framework as the only variable across two instances of the same language model
- Built a Cloudflare Workers serverless backend that fans out parallel requests to the Google Gemini API with transparent fallback to Groq's Llama 3.3 70B on quota exhaustion or upstream failure, transforms upstream SSE streams, and surfaces canonical philosophical terminology to the reader via inline highlighting in the column's accent color
- Deployed publicly on Cloudflare Workers with API keys stored as encrypted secrets, configured CI/CD via GitHub integration, and documented the work in an accompanying written report and presentation deck

Bionic Reader – Firefox Reading Extension

Independent Project, 2026

JavaScript • WebExtensions API (Manifest V2) • pdf.js • Content Scripts • Unicode Text Processing • GitHub • Get it on AMO

- Built and published a Firefox extension — live on Mozilla's Add-ons store (AMO) — that emboldens the leading letters of each word to guide the eye and speed up reading, with an on/off toggle on any page (toolbar button or keyboard shortcut) and three adjustable fixation strengths
- Bundled a custom pdf.js-based reader that applies the bionic effect to PDFs in place — working around pdf.js rendering text onto a canvas with no DOM to style — while preserving the document's original layout, columns, and line breaks
- Wrote a Unicode-aware word splitter that bolds the correct leading fraction of each word while keeping contractions intact and leaving numbers unbolded; architected as a Manifest V2 WebExtension running entirely client-side, with no user data leaving the browser

Probability Distributions Visualizer *Random Variables & Statistics – AUB, 2025*

Python • NumPy • Matplotlib • Statistical Computing

- Built a personal visualization tool to sample from and graph key probability distributions (normal, binomial, Poisson, exponential) as a supplement to coursework
- Used NumPy for random variable simulation and Matplotlib to plot PDFs, CDFs, and histograms, reinforcing theoretical concepts through visual exploration

Data Structures & Algorithms Library *Data Structures & Advanced Algorithms – AUB, 2025*

Java • OOP • Generics • Algorithm Analysis • Graph Algorithms

- Implemented all major data structures from scratch — linked lists, stacks, queues, deques, BSTs, AVL trees, red-black trees, heaps, hash tables, and graphs — and applied each to real-world problem scenarios (e.g., LRU cache with hash map + doubly linked list, priority scheduling with heaps, social graph traversal with BFS/DFS, Dijkstra's shortest path)
- Applied advanced algorithm design techniques including divide-and-conquer, dynamic programming (LCS, rod cutting, matrix chain), greedy algorithms (Huffman codes, activity selection), and amortized analysis
- Leveraged OOP principles — generics, interfaces, inheritance, polymorphism, and encapsulation — to design a modular, reusable library mirroring industry-grade API design patterns
- Benchmarked time and space complexity empirically to validate theoretical Big-O guarantees and identify performance bottlenecks

TECHNICAL SKILLS

Programming Languages	**Strong:** C, C++, Java, JavaScript (React, Node.js), TypeScript **Intermediate:** Python, HTML/CSS, MIPS Assembly **Beginner:** SQL, MATLAB
Tools & Platforms	Git, GitHub, Linux, Windows, Jira, VS Code, GDB, Valgrind, Make, CMake, MSYS2 / MinGW-w64, FFmpeg, Vite, Vercel, Render, Cloudflare Workers, Clerk, Playwright, Docker (basic)
Databases	PostgreSQL, Supabase, SQL, Relational Schema Design, SQL vs NoSQL trade-offs
Core Concepts	Data Structures & Algorithms, OOP & Design Patterns, Advanced Algorithm Design (Divide & Conquer, Dynamic Programming, Greedy), Graph Algorithms (Dijkstra, Bellman-Ford, Floyd-Warshall, MST), Systems Programming, OS Internals (Processes, Threads, Scheduling, Synchronization, Virtual Memory), Computer Architecture (Pipelining, Cache Design, ALU, Floating Point), TCP/IP Networking, POSIX, Agile & Scrum, SDLC, Microservices, DevOps, Cloud Deployment, Linear Algebra
AI APIs & Tooling	Groq API, Gemini API, Hugging Face (moderation), Prompt Engineering, AI Agent Design, Ethical AI, NumPy, Matplotlib; upcoming: Machine Learning, Extended Reality (XR)
Mathematics	Linear Algebra, Discrete Mathematics, Probability & Statistics, Mathematical Reasoning; NumPy & Matplotlib for sampling and graphing random variables and probability distributions (Minor in Mathematics)

SOFT SKILLS

Communication • Team Collaboration • Time Management • Attention to Detail • Problem Solving • Adaptability • Performing Under Pressure

CERTIFICATIONS & TEST SCORES

TOEFL iBT – 105 / 120 (C1 Advanced English) *May 2025*

SAT – 1340 / 1600 *October 2023*

Lebanese Baccalaureate – 14.615 / 20 *2024*

ACTIVITIES & AFFILIATIONS

- Association for Computing Machinery (ACM), AUB *2025 – Present*
- Computer Science Student Society, AUB *2024 – Present*
- AUB Honor Society *2025 – Present*
- Active Volunteer, UNICEF Club, AUB *2025 – Present*

  On-site during the Lebanon war: prepared and distributed food for displaced families under time-sensitive, high-pressure conditions. Co-organized campus fundraisers raising $900+ (Mother's Day campaign for mothers with cancer, Christmas gift drive for children at AUBMC) and contributed to an off-campus recycling initiative with Ahla Fawda at their EcoHub in Hamra.
- Philosophy Student Society, AUB *2025 – Present*
- Selected Volunteer, 42nd AUB Outdoors *2026*
- Photography Club, AUB *2025 – Present*
- Music Club, AUB *2024 – 2025*
- Performed electric guitar at Kerr Got Talent (AUB dorm talent show) *January 2026*
- ULYP Change Program *August 2023 – August 2024*
- HARMUN Model United Nations *March 2023*

LANGUAGES

English: Proficient (C1 – TOEFL 105) **Arabic**: Native **French**: Basic **Italian**: Basic

Lebanese Driving License (Arabic) – valid until 2035; International Driving Permit available upon request.