Joshua Arrevillaga

arrejosh@bu.edu | 781-521-9240 | linkedin.com/in/joshua-arrevillaga | https://github.com/2004joshua

Education

Boston University College of Engineering Boston, MA
B.S. in Computer Engineering Expected May 2026
Minor in Electrical Engineering

Relevant Coursework*: Cloud Computing, Intro to Operating Systems, Software Design, Smart and Connected Systems, Applied Algorithms for Engineers, Computer Organization, Intro to Electronics, Logic Design*

Experience

Red hat Expected May 2025
Incoming Software Engineering Intern

- Develop and maintain code in Python and Golang for RHEL, OpenShift, and other Red Hat products.
- Add features, debug, and collaborate with cross-functional teams, including Quality Engineering and Product Management.

HeadStarter.AI May 2024 – Sept 2024
Software Engineering Fellow, Remote

- Completed five software engineering projects using HTML, CSS, Python, React, Next.js, and OpenAI API, building diverse applications as part of Track B
- Led a team of four fellows to take on a startup's backlog, developing an AI-driven customer support tool that automated ticket creation, reducing manual input by 40%
- Implemented OpenAI API to streamline customer interactions, enabling the system to generate accurate summaries from user input and automatically log them, improving operational efficiency by 35%

Projects

IBM Autopilot Dashboard Sept 2024 – Present

- Created React UI for administrators to launch tests and monitor Kubernetes/OpenShift clusters through an embedded terminal
- Consolidated existing health checks (GPU PCIe Link Bandwidth, Memory Evaluation, Network Reachability, etc.) into UI dashboard for cluster health monitoring
- Integrated node filtering options, allowing administrators to sort and view nodes based on health status, hardware type, and node ID for targeted monitoring and diagnostics
- Implemented OpenShift login for secure administrative access, utilized networked programming, REST APIs, and Kubernetes/OpenShift

Virtual Disk File System Feb 2024 – Mar 2024

- Developed virtual disk file system in C with 8,192 blocks, each 4KB, enabling operations such as open, read, write, and delete through a custom library
- Designed an inode-based structure utilizing direct, and single and double indirects for metadata management supporting file sizes up to 40 MiB
- Utilized block management using bitmap and root directory to manage up to 64 files, with support for 32 simultaneous open file descriptors

User Level Pthread Library Feb 2024 – Mar 2024

- Built custom user-level threading library for Linux, implementing functions pthread_create, pthread_exit, and pthread_join, allowing for thread creation, execution, and termination without reliance on libpthread
- Developed preemptive thread scheduler using setjmp and longjmp for context switching, with periodic timer interrupts (SIGALRM) to manage fair, round-robin scheduling of up to 128 threads
- Designed a thread control block (TCB) structure to manage thread states, stacks, and context, ensuring efficient execution and isolation between threads within a single process

Barbie's World Platformer Nov 2023 – Dec 2023

- Developed multi-level Barbie-themed platform game using Java and LibGDX in Android Studio, featuring dynamic gameplay with enemies, power-ups, and interactive environments
- Engineered core game mechanics, including sprite collision detection, jumping physics, and environmental interactions, improving gameplay fluidity and responsiveness
- Developed custom level progression mechanics, ensuring increase in difficulty and gameplay variety through strategic placement of enemies, power-ups, and obstacles

Skills

Programming Languages: C | C++ | C# | Python | MATLAB | Verilog | HTML | CSS | JS | Go | MIPS
Frameworks/Tools: React | Next.js | Arduino | ESP-IDF | Linux | Firebase | Git | Kubernetes | OpenShift | REST API's
Languages: Spanish (Fluent), English (Native)