

# Adryian (Adrian) Aheyeyu

(214)-336-17-87 | [LinkedIn](#) | [GitHub](#) | [adrian.de.coder@gmail.com](mailto:adrian.de.coder@gmail.com)

## EDUCATION

### Bachelor of Science in Software Engineering

University of Texas at Dallas

Area of Focus: Full-Stack Development

Graduating: May 2026

Richardson, TX

GPA: 3.4

## SKILLS

**Programming Languages:** Java, C, C#, C++, MIPS Assembly, JavaScript

**Web Development:** HTML, CSS, React, Firebase, Express.js

**Tools:** Node.js, GitHub, Git, Raspberry Pi, MARS, Microsoft Office Suite

**Operating Systems:** Windows, Linux, Unix, macOS, Raspbian

**Soft skills:** Leadership, Communication, Teamwork, Critical Thinking, Attention to Detail, Adaptability

**Other:** Markdown, Software Debugging and Testing, Computational Complexity Analysis, Agile Methodology

## PROJECT EXPERIENCE

### Password Manager (JavaScript):

*September 2024 - Present*

- Created a functional full-stack application, utilizing tools like **React**, **Firebase**, **Express** and **Crypto**.
- Back-end: Utilizes Firebase Database System to ensure scalability and AES-256-CTR to provide security while handling data.
- Front-end: Created warm and welcoming UI with simple features that make the design more modern and appealing to use.

### My Remote Dictionary Server Implementation (Java):

*August 2024 - Present*

- The program utilizes in-memory data storage to improve the speed of client-server communication.
- Added master-replica replication to allow for additional increase in performance.

### Multi-Threaded File Hashing Algorithm(C):

*March 2024 – May 2024*

- An algorithm to hash any file, which can process up to 500 real-life pages of text.
- Utilized a variety of sys-calls and POSIX thread API to process large files in less than a second.
- Cut down the average execution time by creating a binary tree of hash values that add up to the total hash.

### Client-Server TCP Database (C):

*February 2023 – March 2024*

- Predecessor of my JEDIS implementation.
- Utilized a variety of sys-calls and built-in TCP modules in C to develop client-server communication.
- Built and customized a variety of common requests to simulate a database environment.
- Introduced multi-threaded programming to optimize the number of simultaneous connections the server side can handle.

### File encryptor (MIPS Assembly):

*October 2023 – November 2023*

- Use a key to encrypt or decrypt a file of any extension.
- Built through utilization of sys-calls, stack and macros.
- Optimized to become unpickable through hashing.

### Payment Interface (C++):

*September 2021 – October 2021*

- Added capability to process files through defined linked lists to minimize memory limitations.
- Ability to search through, analyze and find mistakes within provided input.

## RELEVANT COURSEWORK

Data Structures and Algorithm Analysis

Mathematical Foundations of SE

Operating Systems

Systems Programming

Computer Architecture

Computer Networks

Computer Science I/II

Discrete Math for Computing

Software Engineering

## OTHER

Languages: Russian (Fluent), English (Fluent)

Hobbies: Embedded Programming, Server Management, Puzzle Solving