

Bering Tafa

[LinkedIn](#) | [+4915226058411](tel:+4915226058411) | beringtafa.com | beringtafa5@gmail.com | [GitHub](#)

Skills

- C++ | C | Rust | Kotlin | Java | Python | SQL | JavaScript | TypeScript | React | Wasm | Git | Tailwind | AWS | Firebase
- English (C1) , Albanian (Native) , German (A2)

Experience

Software Engineer TeamViewer *Bremen, Germany* **11/2024 - Present**

- Resolved critical production issues (mostly involving complex synchronization challenges) that were preventing successful client acceptance testing, enabling the acquisition of new contracts **and preserving existing multi-million-euro revenue streams**.
- Designed and implemented a **Kotlin Multiplatform Bluetooth framework**, enabling multiple internal teams to deliver bluetooth functionality in platform-specific application variants (Android/iOS).
- Lead and developed a **Kotlin Multiplatform Database Management framework** using SQLite3, improving maintainability and standardizing data handling across platforms and TeamViewer Frontline teams.
- **Proactively** refactored legacy components to improve performance and long-term maintainability beyond assigned responsibilities.
- **Technical Tools used** : Kotlin Multiplatform, Java, Swift, C, Git

Software Developer IBM Safer Payments (IRIS) *Koblenz, Germany* **10/2023 - 03/2024**

- Maintained and enhanced core components of the IBM Safer Payments fraud detection platform used by banking clients.
- Contributed to the annual release cycle by implementing a memory management enhancement that **improved transaction data handling and storage control for banking customers**.
- Implemented efficient memory chunk mapping and data reshuffling mechanisms to optimize runtime performance and resource utilization.
- **Technical Tools used** : C++, Python, JavaScript (React), Git

Software Engineer Intern Apple *München, Germany* **06/2022 - 09/2022**

- Developed a user-space virtualization solution on macOS to simulate Motion co-processor behavior. This consisted mostly of rewriting the iOS threading implementation using POSIX threads to ensure compatibility with the macOS environment.
- Built a testing environment that supported validation of existing drivers and experimentation with new sensor algorithms, **reducing dependency on physical hardware during development**.
- Collaborated with senior engineers to integrate this simulation framework into existing driver development workflows.
- **Technical Tools used** : C, C++, Shell Script (Python and Bash), Git

Teaching Assistant Jacobs University Bremen *Bremen, Germany* **09/2021 - 05/2023**

- Courses: **Programming in C and C++ | Algorithms and Data Structures | Software Engineering**
- Delivered online and in-person tutorials for **200+** students, strengthening understanding of core computer science concepts.
- Developed structured and visually intuitive learning materials, **increasing tutorial attendance** and student engagement.
- Graded assignments and provided detailed, constructive feedback with personalized academic guidance to the students.
- Prepared data-driven performance analyses with the professor to evaluate student progress and course effectiveness.

Education

Bachelor of Science Jacobs University Bremen *Bremen, Germany* **09/2020 - 05/2023**

- Major in **Computer Science** and Minor in **Mathematics**.
- Part of **Google Developer Student Club** : Taught Algorithms and Data Structures and guided students to Google coding competitions.

Projects

Real-Time Collaborative Editing Extension

[Go to Project](#)

- Developed an extension for the [Lapce](#) code editor (Rust-based), enabling multiple users to edit the same file simultaneously.
- Explored Conflict-free Replicated Data Types (CRDTs) to ensure consistency across users in a distributed environment.
- Investigated serverless collaboration strategies and distributed systems techniques to reduce latency and improve reliability.
- **Technical Tools used** : Rust, JavaScript, WebSockets

Neural Network Framework from Scratch

[Go to Project](#)

- Designed and implemented a **neural network framework in C without external machine learning libraries**.
- Built core components from scratch, including matrix operations, forward and backpropagation, weight initialization, and explicit memory management, based on research literature on neural network mathematics.
- Validated the implementation by using this framework to create neural networks in test applications (e.g., handwritten digit recognition).
- **Technical Tools used** : C