

Kord Boniadi

 [LinkedIn](#) |  (949) 485-0398 |  [kboniadi.github.io](#) |  kordboniadi@gmail.com |  [Github](#)

EDUCATION

B.S. in Software Engineering (in-major GPA: 3.98)

University of California Irvine, Irvine, CA

SKILLS

Languages: (proficient): C++, Java, Python, Go, C#, JavaScript, HTML/CSS (familiar): TypeScript, SQL

Tools/Frameworks: (proficient): Linux, Kubernetes, Docker, AWS(EC2, S3, Lambda, DynamoDB), Flask, React.js, Blazor, React Native, gRPC, Protocol buffers, MongoDB, PostgreSQL (familiar): Redis, RabbitMQ, Angular, Ionic

EXPERIENCE

Software Engineer | Kible Inc.

Jun. 2019 – Present

- Improved our cloud optimized browser's latency by 2ms by refactoring H.264 decoding algorithm in Emscripten.
- Utilized gRPC and Protobuf to improve old and develop new endpoints for the core API written in Go.
- Improved service deployment and cleanup by a measurable amount through optimizations using Kubernetes.
- Implemented UI components for the main website's buy, and user authentication pages.
- Collaborated with the QA team to identify bugs and bottlenecks that were affecting performance.
- Leveraged knowledge in Git, Go, C++, gRPC, Protobuf, Docker, Kubernetes, [Ultralight](#), JavaScript, MongoDB

Software Engineer (volunteer) | Community for Global Innovation

Sep. 2022 – Present

- Created an iOS app using React Native to connect college students with pro bono lawyers to provide free legal services.
- Designed and developed a server-less API and Web App backend using Flask hosted using AWS Lambda.
- Implemented caching using Redis to improve latency and performance of frequently pinged API endpoints.
- Stored and managed data using DynamoDB to maximize query speeds and to keep with a server-less design.
- Leveraged knowledge in Python, JavaScript, AWS(S3, Lambda, DynamoDB), Docker, Redis, Flask, React Native

Software Engineer Intern | Apple Inc.

Jun. 2022 – Sept. 2022

- Spearheaded the development of a next generation web app for R&D teams to improve data access.
- Integrated new and legacy API's to ensure a seamless migration from existing services.
- Improved page load times by nearly 30% by optimizing asset caching mechanisms and pre-rendering content.
- Collaborated with both software and hardware engineers across multiple teams.
- Leveraged knowledge in Git, C#, JavaScript, HTML/CSS, REST API, Blazor, RabbitMQ

Software Engineer (Contract) | Agil Solutions

Jun. 2021 – Jun. 2022

- Developed a web admin portal for a client using C# and the Blazor framework with an ASP.NET backend.
- Utilized the CircleCI pipeline to facilitate the integration of multiple changes to the core codebase as well as the automation for deployment of version updates.
- Maintained and contributed new features and components to multiple React.js web portal projects simultaneously.
- Collaborated with team which involved daily stand-ups and weekly deliverables. Relied on agile methodology.
- Leveraged knowledge in Git, C#, JavaScript, HTML/CSS, REST API, Blazor, React.js, ASP.NET, CircleCI

PROJECTS

Search Engine

- Developed a Python based Web search engine from the ground up that is capable of handling tens of thousands of Web pages, under harsh operational constraints and having a query response time under 300ms.
- Achieved constraints by implementing and designing the engine to use an inverted index, vector space model, tf-idf factor, cosine similarity score, MapReduce paradigm and distributed query evaluation.
- Utilized: Python, Flask, MapReduce paradigm

Multi-Threaded Distributed System Based Online Game

- Developed a Java based multiplayer game with a scalable infrastructure with a team of four developers.
- Designed the backend using the microservice architecture to assimilate independently operating modules to ensure minimal downtime and optimal scalability.
- Implemented a custom messaging queue to facilitate communication in real-time between the different services.
- Developed and implemented a server architecture that included a robust network structure and thread pool management to ensure the server backend could properly and efficiently manage a high amount of traffic.
- Created and hosted a PostgreSQL database which persisted user data, game statistics, and global leaderboards.
- Utilized: Java, Microservices, JavaFX, PostgreSQL