

Albert "AJ" Snow

(479) 774 - 1828 | ajsnow2012@gmail.com | github.com/AlbertSnows | savyb.fly.dev

Technical Experience

Languages: Clojure, JavaScript, PHP, Java, C#, Rust, Python, C++

Tools/Frameworks/Libraries: Node.js, Express, MySQL, MongoDB, Docker, GCP, React, Kubernetes, Spring Boot, JUnit, H2

Skills: Software Architecture, Database Design, API Design, Project Modeling, Algorithm Analysis, CI/CD, MVC, Functional-Programming, Microservices, ETL, OOP, TDD

Work Experience

Backend Engineer

2022 - 2023

Peerspace

- Operated in several Clojure and JS (Node.js + Express) micro-services which utilized MongoDB storage and GCP hosting
- Enhanced features and facilitated communication between services via two primary communication protocols
 - * Direct calls were handled through restful APIs
 - * Async calls leveraged Google's Pub/Sub architecture; typically to persist data in time-insensitive case
- Interfaced regularly with Kubernetes and Docker systems to deploy software images for testing and software releases
- Collaborated with teammates to create design documents for new and legacy features
- Enhanced and standardized development environments by various initiatives
 - * Updated or added outdated documentation across various services
 - * Added dev configs to streamline and improve feedback/REPL loops
 - * Added testing frameworks (e.g. Jest, clojure.test) and core tests to untested repos
 - * Standardized deployment methodologies for versioning
- Optimized, refined, and standardized MongoDB schemas to accommodate new feature development
- Debugged complex issues involving multiple over-the-wire processes across the system (e.g. when inconsistent data fired faulty events)

Software Engineer II

2019 - 2022

Paycom

- Perform full stack (JS/PHP) operations, including creating a backend API and writing user-facing scripts
 - Utilized MVC style framework, OOP style design, and software was regulated (ISO 9001:2015, SOX, PII)
 - Lead initiative to design a new core feature on our team that added ability to customize *all* fields on company position; widely adopted by clients after release
 - Expand various module features by adding data customization, new workflows, and new api routes
 - Push performance improvements such as code optimization, query aggregation, and ORM optimization (e.g. 100x speeds)
 - Create DB designs based on feature requests and employ various data modeling and software design techniques
 - Redesign the code structure in various parts of my code domain (e.g. refactoring code from 1k lines to 200)
-

Education

Kansas State University (2019)

Bachelor of Science in Computer Science (3.8 GPA)

Projects

Personal Website (CLJS, Fulcro)	<i>A personal website using ClojureScript and the Fulcro framework for personal use</i>
Spring Boot State Machine (Java)	<i>Exploratory project to grok the Spring Boot framework. Uses H2 and REST API.</i>
Utility Functions	<i>Repo for experimental code concepts meant to be transferable between languages</i>
Order State Machine (C++)	<i>Example CLI state machine to experiment with core ETL concepts in C++</i>
Hotshot (CLJS)	<i>Basic API to solve a fun math case; utilized Calva for Interactive Programming.</i>
Game Jam Unity Project (C#)	<i>Small game built by me in a weekend featuring lightbulb-themed puzzle platforming</i>
Thermal Modeling (Python)	<i>College concurrency project; helped Graduated student parallelize modeling code</i>

Open source contributions made to Fulcro's Template here: github.com/fulcrologic/fulcro-template. I added the ability to compile and deploy a prod ready uberjar for a project when it's ready to be released.