

# Albert "AJ" Snow

(479) 774 - 1828 | [ajsnow2012@gmail.com](mailto:ajsnow2012@gmail.com) | [github.com/AlbertSnows](https://github.com/AlbertSnows) | [savyb.fly.dev](https://savyb.fly.dev)

---

## Technical Experience

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

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

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

---

## Work Experience

### Backend Engineer

2022 - 2023

*Peerspace*

- Operated in several Clojure and JS (Node.js + Express) microservices which utilized MongoDB storage and GCP hosting
- Led team initiatives to implement requested features for top clients
  - \* For example, calendar sync, resulting in 30% initial adoption and positive post-release feedback via CX team
- 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 and scope design documents for new and legacy features
- Enhanced and standardized development environments via various initiatives (e.g. documentation, tests, dev configs)
- 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

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

---

Open source contributions made to Fulcro's Template here: [github.com/fulcrologic/fulcro-template](https://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.