

Zackary Santana

Miami, FL | zasant014@fiu.edu | <https://www.linkedin.com/in/zackary-santana> | <https://zackaryjamesantana.com> | <https://attach.fly.dev>

Education

Florida International University

Expected Graduation: April 2023*

GPA: 3.6

- Bachelor of Science in Computer Science
- Advance UNIX Programming, Algorithm Techniques, Artificial Intelligence, Data Mining, Data Structures, Database Administration, Database Management, Computational Geometry, Human-Computer Interaction, Mobile App Dev, Programming Languages, Operating Systems, Software Engineering

Work Experience

Software Engineer Intern

June 2022 – August 2022

MongoDB (<https://github.com/evergreen-ci/evergreen>)

- Direct experience working on distributed systems internal facing **CI/CD** testing tooling, handling 1-2 engineering requests every week from other engineers at MongoDB, dog-feeding and pushing those updates live to increase other team velocities.
 - This ranged from changing the **CLI** to accept new flags, represent new database models, or fix commands under certain conditions like overriding defaults, to creating/adjusting **REST** endpoints to better align with the docs.
- Concurrently worked on uncoupling archiving and restarting logic for tests to allow for specific grouping of tests to restart only failed- teams often used these groupings with **800+ tests** that would run on average for **2 hours**. Restarting them before would restart all tests, with my changes it would cut down the restarted tests from 800+ → 1-5, this reduced restart time/machine usage by up to **90%**.
- Used tools like **Splunk** to query and test about the data of restarted tests, creating new pipelines in **MongoDB** to support the database model changes, **Golang** as the primary language for the tooling, **TypeScript/React.js** for UI changes.

Software Engineer Intern

June 2021 – August 2021

MetLife

- Focused on improving UX (user experience) of a smoke test website used by every team, 200+, in my department. After analyzing the site, I noticed there were sequentially queries that would block each other from being executed causing the load time for the website to range from **10 seconds to 30 seconds on a cold start**. I implemented concurrent execution of the queries, optimized the error handling on the queries, and added pagination to the site to reduce the initial load time **down to 1 second**.
- Split the tasks of querying **MongoDB** to the backend (**Restify**) for the endpoints and pinging the **JVM's** (endpoints) to the frontend (**Angular.js**) which reduced the server load- improving overall server response time, and decreasing the time from requesting the site to viewing the site.

Skills

Languages: TypeScript, JavaScript, Golang, SQL, Python, Java, C, Dart, Bash, HTML, CSS

Web Technology: React.js, Express.js, Prisma, Node.js/NPM, Flask, Angular.js, SCSS, Next.js, Svelte 3, Vite, Webpack, Tailwind

General Technology: Git, Docker, UNIX/Linux/WSL, MongoDB, PostgreSQL, SQLite, Firebase, Prometheus, Grafana

Additional Technology: Deno, Preact.js, Twins, Remix.js, Fresh, Cypress, Jest

Productivity: JIRA, Figma, Notion, Microsoft Teams, Trello, Slite, Slack

Projects

CodeConstants (<https://codeconstants.com>)

October 2022 - Present

Web Application

- Large-scale application hosted via various custom docker images, it includes the main application which is a **React.js** and **Express.js** server that fetches from 3 different microservices.
 - **Golang** microservice: Provides an API that receives data from the frontend, authenticating requests with the TypeScript microservice, and “promoting” certain results to the Python microservice.
 - **Python** microservice: Responsible for a stable API that shows posts/info that are personally verified by an admin of the site, or automatically scrapped using a cron job and stored.
 - **TypeScript** microservice: Handles general utility cases for all microservices like authentication and emailing verification.
- Targets serving a fast website that has other working examples like a visualizer for sorting, random number generator with persistence, and a simple reaction timer game.

UPE - FIU (<https://upefiu.io>)

January 2022 - June 2022

Web Application

- Medium-scale application that is mostly monolithic that serves news, updates, and event information relating to UPE @ FIU. The daily active users is around 30 with a session span of around 3 pages which is held in a **Prometheus** server and visualized through a **Grafana** dashboard. Every week with an event, there are around 80-100 daily active users.
- **Prisma** is used as an ORM to provide a type-safe way to interact with the database, all pages are SSR and hydrated when the client receives them.