Brandon Emilio Dauria
b.dauria24@gmail.com • 339-208-0892 • www.linkedin.com/in/bdauria/ • Cranston, RI

SKILLS

Software Languages: Java, JavaScript, HTML, CSS, SQL, Python, Typescript, GO, PostgreSQL, C++
Frameworks: Flask, React.js, Bootstrap, Angular, Drupal, Vue.js, Node.js, Tailwind.css
Tools: Git, Docker, AWS, Cloudflare
Languages: English, Mandarin

EXPERIENCE

Nterprisers | Providence, RI April 2025 – Dec 2025
Junior Backend Engineer – Full Time

- Developed and maintained Python-based backend services supporting an early-stage web platform for local manufacturing and small businesses, contributing to 100% of core backend functionality used during the initial go-to-market launch.
- Designed and implemented 10–20 RESTful API endpoints using Python to support core platform features and front-end data consumption, enabling ~30% faster feature iteration by streamlining request handling, response serialization, and business logic as requirements evolved.
- Built and optimized PostgreSQL database schemas across 10+ relational tables, applying data modeling best practices, indexing strategies, and query tuning to reduce average query response times by ~20–30% under increasing data volume.
- Collaborated with 3–5 cross-functional stakeholders across product, marketing, and data teams to translate evolving business requirements into scalable backend solutions, reducing implementation rework by ~25% and improving development alignment.
- Wrote unit-level tests and conducted manual API testing through request/response validation and edge-case coverage to verify backend correctness, reducing regression during rapid development cycles, and improving overall code reliability.

GigChomp | Boston, MA June 2023 – Sept 2023
Full Stack Developer – Internship

- Engineered a full-stack web application using Flask and React, implementing user authentication, profile management, and job posting workflows, enhancing daily active users by ~50% through streamlined UX improvements, optimized backend services, and feature prioritization based on user feedback.
- Optimized backend data flow, session management, and API interactions in Flask by introducing caching mechanisms, query refinements, and request batching, resulting in ~30% faster data retrieval, reduced server load, and improved responsiveness under typical and peak traffic conditions.
- Led a UI/UX redesign initiative using Figma prototypes, iterative user feedback, and component refactoring in React, improving navigation flows, responsiveness, and user engagement metrics, contributing to a 50% increase in active users within six months.
- Designed and optimized RESTful API endpoints and backend request workflows, including structured request validation, response shaping, and error handling, reducing API latency by ~25% and improving frontend–backend data consistency for all core application features.
- Implemented server-side input validation and authorization checks across Flask routes to enforce access control, prevent malformed or unauthorized requests, and reduce backend error rates during periods of high traffic.

Biblio.io | Singapore | https://github.com/bdauria1/Project-Alexandria Sept 2022 – Dec 2022
Frontend Developer- Internship

- Integrated seven custom React components—including search filters, navigation elements, and reusable UI modules—using modern component patterns, significantly improving site functionality, modularity, and usability by ~20%.
- Improved frontend architecture and performance by implementing lazy loading, code splitting, memoization, and efficient asset management, reducing initial page load time by ~15%, navigation latency by ~20%, and improving responsiveness under frequent state updates.
- Collaborated with cross-functional product and backend teams to align the React frontend with REST APIs, implementing robust data handling, fallback states, and error handling to ensure ~25% fewer frontend–backend sync issues and a more seamless user experience.
- Refactored frontend state management, component lifecycles, and rendering logic to minimize unnecessary re-renders, optimize performance under high data load, and improve overall UI responsiveness and stability for dynamic page content.
- Enhanced frontend reliability and fault tolerance by adding defensive rendering patterns and React error boundaries, reducing UI crashes caused by unexpected API responses, and improving user trust in the application.

PROJECTS

Nutrition Application | https://github.com/krisch0u/cs411 Nov 2023

- Built a React-based nutrition tracking platform integrating Yelp and Nutritionix APIs, implementing structured data fetching, transformation, and aggregation to improve meal tracking accuracy
- Leveraged React Hooks (useState, useEffect) for controlled state management of location-based searches and meal data, reducing unnecessary API re-fetches and improving request efficiency.
- Optimized Axios API calls using batching, caching, and request deduplication strategies, improving average response times by ~15%, and increasing perceived application performance for end users.
- Designed reusable UI components, API utility layers, and data handling abstractions, separating concerns between frontend presentation and backend integration.
- Implemented robust client-side error handling, edge-case management, and fallback UI states, ensuring application stability when external APIs failed, returned incomplete data, or were throttled during peak usage.

Logistical Web Application | https://github.com/bdauria1/cs460 Sept 2024

- Developed and deployed a dynamic Flask-based web application on AWS using a MySQL backend, album management, and photo-sharing features.
- Integrated the Python backend with HTML, CSS, and Bootstrap to deliver responsive, user-friendly interfaces, improving overall usability.
- Enhanced backend performance and scalability by optimizing SQL queries through indexing strategies and query restructuring, reducing page load times under typical usage conditions.
- Designed and implemented relational MySQL database schemas supporting user accounts, media assets, and entity relationships, ensuring data integrity, normalization, and scalable access patterns.
- Configured cloud deployment and environment management on AWS, including application setup, environment variables, and testing workflows, to support debugging, staging, and production readiness.

EDUCATION

Boston University | B.S. in Computer Science - Dean's List May 2024