ERIC TANG

SOFTWARE ENGINEER | FULL STACK WEB DEVELOPER

erictang1201@gmail.com | +61 42 3853 101 | Sydney, Australia | linkedin.com | ericdtang.com | github.com

SKILLS:

Programming Languages: JavaScript (ES6), PHP, TypeScript, Python, SQL
Frontend: React, HTML, CSS/SCSS, Redux, Next.js, Tailwind CSS, Apollo Client
Backend: Laravel, Node.js, MySQL, GraphQL, REST APIs, SilverStripe CMS
Tools & Practices: Git, AWS, Docker, Agile, Jest, PHPUnit, Test-Driven Development (TDD)

EXPERIENCE:

MathsOnline (Sydney, Australia)

Software Engineer   *Apr 2024 – Sept 2024*

- Engineered a custom CRM integration SDK, using Domain-Driven Design (DDD) principles to decouple core business logic from API dependencies, this enabled seamless refactoring of legacy code, and improved application maintainability by providing reusable abstractions over CRM features.
- Built an entire testing library extending Laravel's testing framework, providing pre-configured mocks and assertions for core modules while maintaining extensibility for custom testing scenarios, allowing developers to write unit and integration tests with ease while establishing testing best practices.
- Optimised MySQL database performance by implementing DB replication, caching, and ORM optimisations in Laravel, reducing database load by 30% and lowering the platform's cloud computing costs.
- Created interactive UIs using jQuery, AJAX and HTML templates for efficient JSON API interactions and server-side rendering, achieving optimised page loads and a smooth, SEO-friendly user experience.

Writer's Toolbox (Auckland, New Zealand)

Software Engineer   *Jan 2022 – Apr 2024*

- Engineered high-performance, scalable GraphQL APIs and PHP backend services, handling 500+ concurrent client-side requests while maintaining sub-200ms response times.
- Designed and developed a library of 15+ reusable, accessible UI components with React, TypeScript, and Sass, promoting consistent designs and flexible composition patterns, which accelerated new feature development by 40%.
- Led R&D and built POCs for large-scale system upgrades, including Golang and Kafka-based microservices, and data warehousing pipelines using Singer ETL and Apache Spark, enabling scalable infrastructure and enhanced data analytics.
- Improved UI responsiveness for a React data visualisation tool by 90%, leveraged techniques such as memoisation and window capture to optimise render cycles, reducing load times by 10x and significantly enhancing user experience.
- Built an import wizard that streamlined bulk data imports, implemented client-side validation rules, background job queues for asynchronous processes, and caching strategies in Apollo Client to optimise data operations and UI updates.
- Developed a flexible payment management system that integrated marketing workflows into the backend CMS, this ensured updates to pricing models and marketing discounts can be instantly calculated and applied in the purchasing UI.

EDUCATION:

University of Auckland (Auckland, New Zealand)

B.S. Computer Science and Psychology   *2019 – 2021*

- Certificate of Outstanding Achievement (A+ grade) in COMPSCI 210, COMPSCI 230, COMPSCI 345, COMPSCI 373, and COMPSCI 399.

PERSONAL PROJECTS:

Tracker.gg - Live Game Stats Tracker

A website that helps League of Legends players track their matches and visualise their in game performance statistics.

- Tech stack: Next.js, React, TypeScript, Tailwind CSS, Vercel.
- Integrated Riot Games' developer APIs to fetch, transform, and display live game data with optimised, cached game assets.
- Leveraged React Context to efficiently manage application state and ensure seamless communication between components.
- Utilised Zod for robust API schema validation, enforcing data integrity during API consumption.

AdminFlow - Company Dashboard

A full stack web application for managing companies and employees data.

- Tech stack: Laravel, Vue.js, Inertia.js, PHP, TypeScript, Tailwind CSS.
- Followed Test-Driven Development (TDD) practices to ensure robust, maintainable features and decoupled tests.
- Applied Laravel best practices, such as the use of Eloquent ORM and factory pattern to generate meaningful test data for rapid development iterations.
- Set up a CI pipeline with GitHub Actions that runs automated tests and linting to enforce code quality.