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, Apollo Client
Backend: Laravel, Node.js, MySQL, GraphQL, REST APIs, SilverStripe CMS
DevOps & Testing: Git, AWS, Docker, Agile, Unit/Integration Testing, Test-Driven Development (TDD)

EXPERIENCE:

Self-Employed (Sydney, Australia)

Freelance Web Developer *Jan 2025 – Present*

- Integrated Stripe to handle online payments for a small SaaS product, built RESTful APIs and webhooks in Next.js to communicate with Stripe servers and trigger application logic accordingly, ensuring secure transactions and reducing manual invoicing efforts by 80%.
- Developed a custom dashboard using Laravel and Vue.js to manage org, employee, and inventory data for a client's logistics business, used Eloquent best practices such as eager loading and factory pattern to engineer efficient, well-tested database operations, optimising business operating efficiency through clear inventory and delivery management.
- Built responsive, mobile-first landing page and application UI for an open-source professional networking platform featured on DEV.to, collaborating with other contributors such as UX/UI designers on Figma to ensure pixel-perfect designs.

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.

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.

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:

rct-kit - A lightweight, React-like library for frontend development built with TypeScript and Vite.

- Recreated core features of React from scratch, such as JSX syntax support, functional components, and stateful hooks.
- Built a diffing/reconciliation algorithm, and used a fibre tree data structure to efficiently compare different rendered UI states, mimicking React's actual implementation of the virtual DOM and the algorithm used to update the UI.
- Emulated React's rendering lifecycle and component update flow, and designed a simple state system for functional components which allowed the use of reactive variables through calling a custom useState hook.