

➔ See this CV at <https://www.ycmjason.com/cv>

Jason Yu

Principal Frontend Engineer • Tech Speaker • Open Source Contributor

✉ jason@ycmjason.com • 📍 London, UK

Passionate software engineer with 8+ years of experience building scalable frontend architectures, leading TypeScript migrations, and empowering developer communities through open source and public speaking. Creator of 12k+ download mobile apps and innovative dev tools.

[Personal Website](#)

[GitHub](#)

[LinkedIn](#)

[X](#)

[dev.to](#)

[YouTube](#)

👤 Work Experience

Boltzbit (Principal Frontend Engineer)

2025 - Present

- Architected a **scalable frontend monorepo** to facilitate seamless integration with external teams.
- Spearheaded the **migration to TypeScript**, improving code reliability and developer experience.
- Led the frontend architecture for a **no-code, AI-empowered app builder**:
 - Developed a **widget registry** enabling users to compose apps via modular components.
 - Created a **Widget SDK** to support third-party extensions and ecosystem growth.

🔍 Fishball Ltd (Founder)

2025 - Present

Building multiple apps, SaaS, and open-source projects:

📱 Mobile Apps

- 🕒 **WearDa Metronome**
 - 🖥️ Jetpack Compose, Kotlin
 - A WearOS vibrating metronome with **>12k downloads** and **4.5★ rating**.
- 🇬🇧 **Ace It: Life in the UK Test**
 - 🖥️ React Native, TypeScript
 - A study app for the UK Citizenship Test.
- ♠️ **Big2Cal**
 - 🖥️ React Native, TypeScript
 - Score-tracking app for the popular Hong Kong card game **Big2**.

🌐 Web Apps

- 🏠 **mjcaLapp**
 - 🖥️ React, TypeScript
 - Mahjong score-tracking app with analytics and stats.

📈 Bloomberg LP (Senior TypeScript Engineer)

2023 - 2025

- Led migration to TypeScript strict mode
 - Increased visibility of strict mode errors:
 - Developed `@ts-expect-strict-error`, a **TypeScript Language Service Plugin**, showing strict mode errors in IDEs (without affecting `tsc`).
 - Allowed suppressing errors with `@ts-expect-strict-error-next-line`.
 - Prevented new strict mode errors:
 - Built a CLI tool leveraging `@ts-expect-strict-error`.
 - Enforced checks via a new CI step, blocking PRs with unmarked strict errors.
 - Reduced existing strict mode errors:
 - Introduced Grafana dashboards to track progress and a leaderboard to gamify error elimination.
 - Eliminated **15% of 10k errors** in the first month.
 - Reduced technical debt by **1,500+ strict mode violations** through team collaboration.
- Defined best practices and patterns for scalable TypeScript code.

📄 Attest Technology Limited (Frontend Engineer)

2019 - 2023

- Developed an *A* algorithm-based SVG line drawing module* for signature capture.
- Improved error detection and component testing DX, reducing bug reports by **40%**.
- Advocated for the **"rule of least power"** as a coding principle.
- Led migration from **vanilla TypeScript** to **Vue.js**, modernizing the tech stack.

🏠 The Hut Group (Full Stack Engineer)

2017 - 2019

- Built a **vanilla JavaScript component framework**.
- Founded & led **"JS Club"** to promote modern JavaScript practices.
- Developed a CLI tool to improve developer workflows.

🎬 YouTube

📄 Technical Talks & Tutorials

- **TRY {} CATCH {} – FIXED! (with a surprisingly old trick)**
 - Analyzed limitations in standard JavaScript error handling and demonstrated an IIFE-based pattern to improve code integrity and type inference.
- **7 MODERN IIFE TRICKS IN 10 MINUTES**
 - Presented seven practical use cases for Immediately Invoked Function Expressions (IIFE) in modern development, including async operations, variable scoping, and encapsulation.
- **Your code: powerful, but not clean. Here's why**
 - Applied the "Rule of Least Power" to software engineering, advocating for "depowering" code conventions to enhance long-term readability and maintainability.
- **Introducing ts-migrating: Upgrade TSConfig Progressively**
 - Demonstrated `ts-migrating`, an open-source tool designed to help developers upgrade strict TypeScript compiler options progressively across large codebases.

🔧 Open Source & Side Projects

- [@fishballpkg/linkz](#)
 - A config-driven static site generator that powers my personal website.
 - Generates a **linktree-like** landing page with full markdown support.
 - Powers this very site.
- [ts-migrating](#)
 - A plugin enabling progressive upgrades to stricter `compilerOptions` (e.g., `strict`, `noUncheckedIndexedAccess`, `erasableSyntaxOnly`) by allowing problematic lines to fall back to legacy settings.
- [@fishballpkg/acme](#)
 - A zero-dependency, minimalist ACME client in TypeScript designed to simplify certificate generation.
 - Leveraged modern web standards (e.g., Web Crypto API) and implemented a custom **ASN.1 Encoder** using low-level `Uint8Array` manipulation. 🤖
- [one-page-pdf](#)
 - A CLI tool using **Puppeteer** to render webpages into single-page PDFs.
 - Used to generate the PDF version of this very CV.
- [diary.ycmjason.com](#)
 - AI-powered diary using LLM, inspired by Tom Riddle's diary.
 - [Find out more on GitHub](#)
- [Keyboard-Keyboard](#)
 - Web-based **musical keyboard** using **Web Audio API**.
 - [Find out more on GitHub](#)
- [Perfect Colour](#)
 - Vision quiz testing ability to recognize RGB values.
 - [Find out more on GitHub](#)

🎤 Public Speaking

- **Live Coding: Recreating Vue's Reactivity System!**
 - 📅 Oct 2023
 - 📍 [Vue.js Slovenia Meetup](#)
 - 📺 [Recording](#)
- **Vue 3 Reactivity Reimagined**
 - 📅 Nov 2019
 - 📍 [Vue.js London Meetup](#)
 - 📺 [Recording](#)
- **Live coding: The new Composition API**
 - 📅 Oct 2019
 - 📍 [Vue.js London Conference](#)
 - 📺 [Recording](#)
- **Vue without View - An Introduction to Renderless Components**
 - 📅 Jun 2019
 - 📍 [Vue.js London Meetup](#)
 - 📺 [Recording](#)
- **Building a Simple Virtual DOM from Scratch**
 - 📅 Nov 2018
 - 📍 [Manchester Web Meetup](#)
 - 📺 [Recording](#)

📄 Articles

📖 All articles available at dev.to/ycmjason

Here are some hand-picked highlights:

- **Typing Partially Applicable Functions in TypeScript**
 - 📅 Jun 2023
 - 🔗 [Read](#)
- **Detecting Barcodes from the Browser**
 - 📅 Nov 2021
 - 🔗 [Read](#)
- **Writing Cleaner Code with the Rule of Least Power**
 - 📅 Apr 2020
 - 🔗 [Read](#)
- **Building a Simple Virtual DOM from Scratch**
 - 📅 Dec 2018
 - 🔗 [Read](#)

🎓 Education

Imperial College London

2014 - 2017 • BEng Computing

- Courses: Haskell, Java, C, C++, Prolog, Python, AI, Operating Systems, Compilers, Networking.