

Kiệt Nguyễn

Senior Frontend Developer

AI-NATIVE DEVELOPER

I build the web apps that keep e-commerce orders moving.

kiet080797@gmail.com
linkedin.com/in/kiet-nguyen-dev
github.com/kiet0807
kietnguyen.io.vn/en
Ho Chi Minh City, Vietnam

01 Summary

Senior frontend developer with 6+ years of experience building production web applications in React and TypeScript, across e-commerce, UK wealth-management software and fulfillment platforms. Primary frontend developer of Onflow's order management system: restructured the codebase, redesigned the UI/UX, migrated it to strict TypeScript and helped cut the release cycle by 50%; built Fulfillment Connect and the shared packages from scratch. Works AI-first with guardrails: about 80% of the code is drafted by AI agents, and every change is specified, reviewed and tested before it ships.

6+	50%	1.6k	800+
years building production frontends	shorter release cycle on the Onflow frontends	files migrated to strict TypeScript, zero JavaScript left	test files guarding three production codebases

02 Experience

2025 – Present

Senior Frontend Developer · Onflow Fulfillment

E-commerce fulfillment service built on its own technology platform: OMS, WMS, TMS and Fulfillment Connect.

- Took over the Onflow OMS frontend as primary developer (3,700+ commits): restructured it into seven feature-based modules, redesigned the UI/UX and migrated 1,600+ files to strict TypeScript.
- Moved the OMS build from Create React App to Vite and upgraded its core dependencies for faster builds and deployments; authored most of the test suite and the architecture lint rules.
- Built the Fulfillment Connect console from an empty repository to production in seven weeks (nine feature modules today), then created Onflow Shared Packages: the UI, API runtime and tooling both apps share.
- Delivered inbound, outbound, courier and reporting screens for the Operations System (OPS) and built two internal Google Apps Script tools: a warehouse inbound booking form and a Taiga-to-Sheets sync.
- Established an AI-assisted workflow: authored the agent guides (AGENTS.md, CLAUDE.md) in three repositories and built 7 reusable skills, 3 subagents and session hooks, matching the model to each task to save tokens; about 80% of the code is drafted by AI agents and 100% is reviewed before merge.
- Impact: release cycle 50% shorter, deployments over 50% faster, production hotfixes nearly eliminated.

`React · TypeScript · Vite · TanStack Query · Vitest · Sentry`

2024

Full-stack Developer · Freelance

Company website for Diamond Owl Solution, an IT services firm.

- Designed and built the website end to end with Next.js on a headless Strapi CMS.
- Modeled the content, exposed it through a GraphQL API and integrated email and third-party services.
- Containerized the stack with Docker for portable deployment on the client's infrastructure.
- Worked in a three-person team, turning requirements from the team lead and the client into features.

`Next.js · TypeScript · Strapi · GraphQL · Docker`

2022 – 2024

Frontend Developer · TMA Solutions × GBST

GBST builds wealth-management software for investment platforms and pension providers. Contributed to two of its products: Composer and Catalyst.

- Developed React features for Composer, the wealth administration platform behind some of the UK's largest investment platforms.
- Delivered new React features for Catalyst, a self-service digital portal for investors and advisers, alongside its legacy Ember.js application.
- Worked in an Agile Scrum team, communicating directly with the overseas client to refine requirements; covered new features with Jest and Testing Library tests.
- Recognized twice: for delivering a release ahead of schedule, and for high performance and support of colleagues.

`React · Ember.js · Jest · Formik · Jenkins`

2021

Frontend Developer · PAVN

Online marketplace for buying and selling digital services.

- Built the storefront with Next.js and TypeScript, using server-side rendering for performance and SEO.
- Managed server state with React Query and built the interface with Material UI.
- Worked in a Scrum team, reporting progress daily to the product owner.

`Next.js · TypeScript · React Query · Material UI`

2020

Frontend Developer Intern · Just Enough

Bakery e-commerce app with online payment.

- Implemented the product browsing, cart and checkout flows with Next.js and Theme UI.
- Integrated a Strapi and MongoDB backend, with order confirmation emails sent through Nodemailer.
- Delivered features under the guidance of a mentor and the team lead.

`Next.js · TypeScript · Theme UI · Strapi · MongoDB`

03 Projects

COMPANY PRODUCT 2026

Onflow OMS

Role: Primary frontend developer

Multichannel order management: orders from marketplaces, websites and social channels on one screen, inventory in sync, and every order handed to the fulfillment warehouse. Frontend restructured into seven feature-based modules, with a redesigned UI/UX, a typed data layer and lint-enforced import rules.

−50% release cycle · **>50%** faster deploys · **2.1k** TypeScript files, no JavaScript

`React · TypeScript · TanStack Query · React Hook Form · Vitest`

COMPANY PRODUCT 2026

Onflow Fulfillment Connect

Role: Built from scratch

Fulfillment Connect lets a business keep the OMS or ERP it already runs and plug Onflow fulfillment into it. This is its console: create and track shipments, handle returns, run inbound orders, watch inventory and pull reports. Released behind feature flags.

9 feature modules · **7** weeks to production · **197** test files

`React · TypeScript · Zustand · TanStack Query · Unleash`

INTERNAL LIBRARY 2026

Onflow Shared Packages

Role: Author and maintainer

The frontend library shared across the company: components, hooks and runtime published as three private npm packages. UI, core (API client, session, analytics, feature flags) and config (ESLint, Prettier, TypeScript presets, git hooks).

3 published packages · **2** apps built on them · **173** test files

`React · TypeScript · Sass · ESLint · GitHub Packages`

04 How I work with AI

AI-assisted development, with guardrails

≈80% of my code is drafted by AI · **100%** of diffs reviewed by me · **3** repositories with agent guides

01 Spec first
I write down what to build and how it will be checked before the agent writes a line.

02 Rules and skills
Each repository carries an agent guide, lint rules and reusable skills, so the agent works the way the team does.

03 The agent drafts
The agent writes the code and its tests in small steps. Subagents search, review and verify, each on a model sized for the task, which saves tokens.

04 I review, CI decides
I read every diff and own the result. Nothing merges until types, lint and tests pass.

- ✓ Strict TypeScript, no JavaScript left
- ✓ Architecture rules checked by lint in CI
- ✓ Tests run on every pull request: 800+ files across three codebases
- ✓ Agent guides and skills in the repository
- ✓ A human reads every diff before it merges

05 Skills

CORE	React 6 yrs, TypeScript 6 yrs, JavaScript 6 yrs, HTML 6 yrs, CSS / Sass 6 yrs
FRAMEWORKS & STATE	Next.js 3 yrs, TanStack Query 3 yrs, React Hook Form 2 yrs, React Router 3 yrs, Zustand, Ember.js
UI & STYLING	Bootstrap / reactstrap 3 yrs, Material UI, Recharts 2 yrs, react-i18next 2 yrs
TESTING & TOOLING	Vitest 2 yrs, Jest, React Testing Library 3 yrs, Vite 2 yrs, ESLint 3 yrs, GitHub Actions 2 yrs, Google Apps Script
BACKEND & DATA	Node.js (NestJS, Express), Strapi, REST 6 yrs, GraphQL, PostgreSQL / MySQL, MongoDB, Docker 2 yrs
AI & AGENTS	Claude Code, Agent Skills, Subagents, Hooks, AGENTS.md / CLAUDE.md, Model routing, Token optimization, MCP
CURRENTLY EXPLORING	Claude Certified Architect, Prompt caching, React 19, Tailwind CSS 4, React Three Fiber, Turborepo

06 Education and recognition

2015 – 2020 **B.Sc. in Computer Science**
University of Information Technology (UIT), VNU-HCM

2023 **Recognition for high performance and teamwork**
TMA Solutions

2022 **Recognition for delivering a release ahead of schedule**
GBST