

DAVID CHOY

Telephone: +44 7544 812 110

Email: davidchoy.dev@gmail.com

Location: London, UK

LinkedIn: www.linkedin.com/in/pkdchoy

GitHub: <https://github.com/pkdc>

Profile

Full-stack software engineer with end-to-end delivery experience across React/TypeScript, Django/DRF, and PostgreSQL. Builds and ships production web applications end-to-end, from spec and design through to Docker-based deployment. Combines engineering with UX/UI design capability in Figma. Effective at translating ambiguous domain requirements into structured, maintainable systems.

Technical Skills

- **Programming Languages:**
 - JavaScript, TypeScript, Python, Golang
- **Frontend Engineering:**
 - React, TypeScript, Vite, HTML5, CSS3 (Flexbox, Grid), Tailwind CSS, React Router, Apollo Client (GraphQL - client-side), responsive design, CSS animations, JavaScript animations (requestAnimationFrame), Jest, Vitest
- **Backend & Security:**
 - Python (Django, Django REST Framework, FastAPI), Go (channels, goroutines), RESTful APIs, WebSockets, PostgreSQL, SQLite, OIDC/OAuth2 (mozilla-django-oidc, Auth0), bcrypt, HTTPS, rate limiting
- **UI / UX & Product:**
 - User Research, User Journeys, wireframing, prototyping, Figma
- **Databases & Data Modelling:**
 - PostgreSQL (Django ORM), SQLite, DBML schema design
- **Data Science & Machine Learning:**
 - NLP with Transformer-based models (BERT), embeddings, scikit-learn, XGBoost, Keras/TensorFlow, Pandas, NumPy, Matplotlib, Seaborn
- **DevOps & Infrastructure:**
 - Docker, Nginx, Heroku, Render, Netlify, GitHub Pages, GCP
- **System & Networking:**
 - Linux CLI, SSH, environment variables, TCP/IP, HTTP, nmap
- **Tools & Methodologies:**
 - Git, GitHub, Agile/Scrum, TDD knowledge, GitHub Actions, Trello, Google Maps & Geocoding APIs
 - Design-to-Code Workflow: Figma to React (TypeScript) using Figma MCP in Claude Code for component scaffolding and layout translation
- **CMS:**
 - WordPress (commercial experience: configuration, maintenance, troubleshooting)

Experience

Software Engineer Trainee | 01Founders | Sept 2021 – Oct 2023

- Delivered multiple full-stack, team-based projects using React, Golang, and WebSockets, covering frontend and backend.

- Designed and implemented real-time chat systems supporting private and group messaging with Gorilla WebSocket.
- Built frontend architecture including authentication flows, routing, shared state management, and core social features.
- Developed backend services in Go, leveraging goroutines and channels for concurrency and responsiveness.
- Designed and implemented SQLite database schemas, authentication systems, and RESTful APIs.
- Implemented security-focused features including HTTPS (Let's Encrypt), bcrypt password hashing, UUID-based cookie sessions, and rate limiting in Go using channels to prevent system overload.
- Debugged and optimised frontend and backend performance using Chrome DevTools.
- Actively contributed to code reviews, documentation, GitHub Issues tracking, and collaborative problem-solving.

Web Developer | Business Clan | June 2019 - Oct 2020

- Wrote custom CSS code in WordPress for clients' website.
- Troubleshoot issues and solved technical problems of clients' website promptly.
- Successfully trained colleagues in the IT team on CSS and received a special commendation for the planning and delivery of the session.
- Effectively communicated with the support team of relevant 3rd party tools/plugins, and followed their instructions to resolve issues.

PhD Candidate | Institute of Cancer Research | Feb 2015 - Jan 2016

- Worked in the Radiotherapy Physics Modelling team on one of the MR linac related projects.
- Implemented an algorithm to accelerate IMRT dose calculation by pre-calculating the D_{ij} matrix.
- Designed, implemented, and tested the gamma evaluation algorithm for checking the correctness of the accelerated IMRT dose calculation algorithm, by comparing the calculated dose values to the reference values.

Software Engineer | Xceptor | Summer 2012

- Planned test process.
- Created manual test cases.
- Automated tests for Xceptor Data Hub (Coded UI tests).
- Worked as part of the support team to handle phone enquiries on Xceptor Data Hub.

Selected Projects

Unknot – Conflict Resolution Research Platform

React TypeScript | Vite | Django DRF | PostgreSQL | Docker | GitHub Actions | mozilla-django-oidc | Auth0

Deployed - <https://unknot-me.com>

- A guided conflict resolution research platform. Users engage with a four-step flow: read a conflict scenario, respond with their emotional reaction and proposed action, share a personal conflict story, then compare anonymised stories from other participants.
- Features session-based anonymous identity with optional OIDC authentication via Auth0, allowing users to persist and return to their sessions.
- Progressive disclosure UI and load-balanced content distribution to ensure even data coverage.

DISC Classifier - Email Personality Classification System

Python | FastAPI | Pandas | Logistic Regression | NLP | BERT | XGBoost | RegEx

GitHub - <https://github.com/pkdc/disc-predictor-for-business>

- Developed a text classification tool to predict DISC personality styles from corporate emails using BERT embeddings and an XGBoost model.

- Built a custom-labeled dataset from the Enron email corpus using a hybrid strategy (manual, rule-based, and pseudo-labeling).
- Wrapped the trained model in a FastAPI endpoint for classification via API.

NotFbook – Real-Time Social Network

React | Golang | WebSocket | CSS Modules | SQLite

Deployed - <https://notfbook.netlify.app/>

BE Github - <https://github.com/pkdc/social-network-backend>

FE Github - <https://github.com/pkdc/social-network-frontend>

- Led the design and implementation of the real-time chat system, a core feature of the application, using Go and Gorilla WebSocket on the backend and React on the frontend.
- Engineered the complete frontend chat interface, including WebSocket context management, dynamic chat lists, and a full-featured chatbox supporting both private and group messaging.
- Developed the full stack for the chat notification system, enabling real-time alerts and user interactions (e.g., accept/decline requests).
- Owned the frontend development for key application areas, including the landing page, user authentication flows, and the core post/comment system.
- Architected the frontend state management using multiple React Contexts to handle complex logic for authentication, social graphs (following), and group systems.
- Optimized chat system performance and responsiveness by strategically implementing React hooks for state management and leveraging goroutines for concurrent backend processing.
- Enhanced user experience by designing and implementing fluid animations for the landing page, loading sequences, and chat interface interactions.

GoPulse - Secure Forum

Golang | Concurrency (Goroutines & Channels) | HTTP/TLS | SQLite | GCP

Deployed - <https://go-pulse-789.onrender.com>

GitHub - <https://github.com/pkdc/forum-security>

- Engineered a concurrent rate-limiting system from scratch using Go Channels and Goroutines, protecting the server from request floods without external dependencies.
- Implemented HTTPS with automatic certificate provisioning/renewal (Let's Encrypt ACME) in Go using autocert, serving HTTP-01 challenges on port 80 and terminating TLS on 443.
- Implemented environment-adaptive TLS infrastructure supporting both PaaS-managed and bare-metal SSL termination using native ACME (Let's Encrypt).

Tetris - Full-Stack Arcade Platform with Leaderboard

JavaScript (ES6+) | Go (Golang) | RESTful APIs | Jest

Deployed - <https://tetris-with-scoreboard.onrender.com>

GitHub - <https://github.com/pkdc/tetris-with-scoreboard>

- Implemented a full-stack Tetris application using a modular, class-based JavaScript architecture to manage game state, collision detection, timing, and rendering flow.
- Designed a decoupled Go backend service for score persistence, exposing a lightweight RESTful API consumed by the frontend to power a responsive leaderboard with real-time search and dynamic pagination.
- Validated core line-clearing and scoring logic using Jest with a DOM-simulated test environment, covering boundary conditions, consecutive and non-consecutive row clears, and scoring rules.

Miles & Miles - Design-to-Code Mobile Prototype

React TypeScript | Vite | Figma MCP | Claude Code

GitHub - <https://github.com/pkdc/miles-and-miles-mobile>

Deployed - <https://miles-and-miles-mobile-happy.netlify.app>

- Designed a multi-screen mobile flow in Figma based on a brief for a luxury car rental company adding a new booking feature, producing user journeys and high-fidelity prototypes.
- Implemented the design as a working React TypeScript prototype using Claude Code with the Figma MCP Server to convert Figma components into code.
- Built reusable UI components, scroll interactions, form validation, date/time modal pickers, and React Router navigation with disabled state management.

Thymynd – Decision-Making for Public Good

HTML5 | CSS3 | JavaScript

Deployed - <https://pkdc.github.io/thymynd>

GitHub - <https://github.com/pkdc/thymynd>

- A mobile-responsive website for a decision-making platform exploring personalised scenario-based tools for education, social services, and justice sectors.
- Built with vanilla HTML5, CSS3, and JavaScript.

Education

Software Engineering Programme, 01Founders, 2023

Master of Science - Computer Science, University of Oxford, 2014

Bachelor of Science - Physics, Imperial College London, 2013

Professional Certificates

UX and AI Skills Bootcamp, The School of UX, December 2025

Data Science Bootcamp, The Developer Academy, March 2025

Microsoft Certified: Azure Fundamentals, Microsoft, March 2021

The Bits and Bytes of Computer Networking, Google, July 2020

Using Python to Interact with the Operating System, Google, June 2020