Xue Yufeng

+65 92372711 | xyf.oco@gmail.com | yufeng-resume.web.app | github.com/XlightNtrEnx | linkedin.com/in/xue-yufeng/

Education

Singapore University of Technology and Design 8 Somapah Rd, Singapore 487372
Bachelor of Computer Science First Class Honours *Sep. 2023 – Present*
4.86/5.00 GPA (Excl. Humanities Modules)

Experience

Intern Sep 2025 – Dec 2025
Panasonic R&D Center *202 Bedok South Ave 1 #02-11, Singapore 469332*

- Had fun with my colleagues and learnt a lot about signal processing, data analysis, and machine learning. Main project was on the FMCW Radar which I picked up on and greatly improved from the previous intern according to my supervisor.

Tutor Mar 2023 – Jan 2024
Envision Learning *35 Circuit Road #01-442, Singapore 370035*

- Developed soft skills through collaboration with colleagues and meaningful interactions with students.

Projects

Panasonic FMCW Radar | *Matplotlib, NumPy, Pandas, Python, PyQtGraph, SQLite* Sep 2025 – Dec 2025

- Helped to refactor the project to use cleaner code and architecture and impressed my supervisor to comment that I was quite fast with the project yet clean. This allowed our research efficiency to be high as we could explore different ideas quickly by adding new features fast.
- Learned various signal processing algorithms (Fast Fourier Transform, Butterworth filter, Taubin's circle fit, MUSIC Algo for Direction of Arrival estimation) to apply them and train an AI with very low Mean Absolute Error (MAE).
- Stored different types of data using SQLite and folders to let my supervisor access them easily as well as provided visualization (including realtime) for them using Tensorboard, QtGraph, and Matplotlib

Ascenda Booking Platform | *NestJS, PostgreSQL, Railway, TypeScript* May 2025 – Aug 2025

- Reduced fuzzy search time from minimum 200ms to maximum 30ms of 69875 records which improved UX for autocomplete search.
- Improved team efficiency by setting up a simple CI/CD pipeline that ensures all test pass on a staging database before automatically deploying main branch to Railway production.
- Secured our API by storing all session data in an internal Redis server and the session data will only be transmitted if secure protocol is used and authentication details are valid.
- Helped solidify team's vision by playing an Architect role which involved designing the deployment diagram and ERD.
- Ensured Quality Assurance (QA) by implementing mock dependencies, writing test cases, and designing a reporter that summarizes test results.

SimpleDB | *JUnit* May 2025 – Aug 2025

- In this school project, I implemented a Two-phase locking mechanism to allow our database to operate on a concurrent level that came with deadlock detection and avoidance through maintenance of a Wait For Graph.

CSEShell | *C* May 2025 – Jun 2025

- I with my teammates implemented a low level shell in C that could handle customization, disk operations, and process management.

ParcelEye AI Tracker | *Android SDK, AWS, Docker, ExoPlayer, FFmpeg, Flask, Gradle, Java, MongoDB, Nginx, Python, SAM2.1, Spring Boot* Jan 2025 – Apr 2025

- Proposed the project, which was accepted by my team. It consists of an Android app livestreaming to an AI server and playing the AI's output to track parcels, addressing the real-world problem of parcel theft at our school.
- Developed a Flask API to control and feed a livestream into the SAM2.1 AI using Nginx and FFmpeg.
- Created a Spring Boot API integrated with MongoDB for authentication using JWT.

- Implemented ExoPlayer in the Android app to consume HLS streams from the AI and integrated the authentication flow with the API and AI.
- Hosted the AI and API on AWS using EC2 and configured the VPC to secure our servers.
- Containerized the AI server, reducing image size from 32GB to 5GB using multi-stage builds.

FPGA Game | *Alchitry AU, Alchitry Labs, Lucid* Jan 2025 – Apr 2025

- Our FPGA controlled six 7-segment displays. I ensured there was no excessive current flow by writing code that rapidly cycles through all displays, creating the illusion that they are all lit simultaneously.
- The FPGA needed to control the digits on the displays at 6V but could only output 3.3V signals, so I researched and used a PNP BJT transistor to step up the voltage.

Cat Dog Classifier | *Google Colab, Pillow, Pytorch* Dec 2024 – Dec 2024

- Applied concepts (residual connections, hierarchical layers, data normalization, and standardization) from proven models (ResNet, VGG, etc.) to build a model achieving 95% accuracy in training and validation for binary classification of cats and dogs.
- Used GradCAM to highlight the parts of images the AI uses to distinguish cats from dogs.

Resume Website | *CRACO, FFmpeg, Firebase, GitHub Actions, React, Styled-Components, Jotai, Vite* Oct 2024 – Present

- Made a script that utilizes FFmpeg to loop through each image and convert to .webp if the final size is smaller. This reduced the total size of important images from 53.5mB to 41.5mB which would improve UX as less time is wasted on loading images.
- Applied 3D, animation, and transition techniques in CSS to implement the landing page that has an easter egg.
- Used Firebase BaaS to authenticate users but later dropped it in favor of a better UX for recruiters.
- Utilized GitHub Actions to implement Continuous Integration/Continuous Deployment (CI/CD) with Firebase hosting.
- Applied SOLID principles for the first time in a significant project, leveraging states, contexts, and providers to that end.
- Migrated from CRACO to Vite for improved RAM usage during development.

Technical Skills

Languages: C, CSS, HTML, Java, JavaScript, Lucid, Python, SQL (MySQL, Postgres), TypeScript
Frameworks: Flask, JUnit, NestJS, Node.js, Qt, React, Spring Boot
Developer Tools: Android Studio, AWS, CRACO, Docker, Firebase, Git, GitHub Actions, Google Colab, Railway, Supabase, VS Code, Vite
Libraries (Java): Dotenv, ExoPlayer, Firebase Admin, Jakarta Mail, Java Net, Java IO, JUnit, MongoDB Driver, Netty, OkHttp, Retrofit, Spring Boot
Libraries (Python): Flask, Matplotlib, NumPy, OpenCV, Pandas, Pillow, PyQtGraph, PyTorch, Requests, SQLite, Tensorboard
Libraries (Javascript/TypeScript): Jotai, NestJS, React, Styled-Components, TypeORM