Jay Mistry

jay@jellyware.co.uk | *www.jellyware.co.uk* | *www.github.com/jaym-01*

EDUCATION

Imperial College London - Electronic and Information Engineering:

- Obtained the **Dean's List** in my 1st and 2nd year for academic performance (top 10% of the year).
- Built a pipelined RISC-V 32-I (CPU) using System Verilog & C++.
- Built a C90 to RISC-V compiler with C++, Flex and Bison.

A-Levels, 2020-2022: Maths – **A***, Physics – **A***, Computer Science – **A*** (Coursework: Designed and built a normalised MySQL database, and a desktop app to perform CRUD operations (C#)), EPQ – **A**
AS-Level, 2021-2022: Further Maths (self-taught) – **A**

PROFESSIONAL EXPERIENCE

Full stack engineer at OncoFlow (Health tech AI startup) (3 months):

- Developed the frontend and backend from scratch for the product and company site independently, using Next.js (with **Typescript**), Tailwind CSS and **Python** FastAPI.
- Worked in a fast-paced environment, adding features and fixing bugs within a few hours or days.
- Implemented unit, integration and end-to-end tests using Jest, pytest and Playwright.
- Used GitHub: conducted code reviews and pushed code to production with pull requests.
- Created local testing environments to simulate production, using Docker and Bash scripts.
- Added optimisations such as caching to reduce the use of AI models, reducing cost and latency.

Volunteer at Zero Gravity - Helped a student from a disadvantaged background get into a top University, by informing and aiding decision-making on the application process, and providing feedback and support on their application. Both listening to them and coherently presenting ideas improved my communication skills.

SKILLS

Programming Languages/Frameworks: (Proficient): TypeScript, Tailwind CSS, Python; (Intermediate): React.js, Next.js, Jest, Playwright, C++, HTML/CSS.

Familiar with **Linux commands** and designing UIs with **Figma**.

PROJECTS

Implemented a Redis Server in C++ (www.github.com/jaym-01/RedisCpp):

- Used boost ASIO library to create TCP sockets that handle multiple clients using an event loop.
- Implemented a parser for the Redis serialisation protocol (RESP).

Contributed to an app that automates containerising and deploying Python backends (using Typescript & Python):

- Built the frontend for the app using Next.js, Tailwind CSS and shadcn/ui.
- Worked on the Python FastAPI backend, which involved adding a custom run command when containerising an app and fixing issues with pulling repos from GitHub.
- Learnt about creating docker files, containerising apps and deploying to AWS using App Runner.

Dashboard and embedded programming for Mini Power Grid (using Typescript & Python):

- Built a full-stack app with Next.js and Tailwind CSS that displays real-time data using MQTT.
- Deployed an MQTT broker to an AWS EC2 instance and shared the dashboard using ngrok.
- Programmed Raspberry Pi Picos to send and receive data from the dashboard using MQTT, with micro python.
- Designed and implemented an algorithm to synchronise with an HTTP server, with data changing periodically, that minimised latency and resource consumption.

App Automating repetitive tasks when video editing (using Typescript & Python):

- Used Python and Pydub to encode and decode the audio from audio and video files.
- Improved the algorithm for finding and removing silences, from a naive $O(n^2)$ to $O(n)$.
- Connected the Python script to an Electron app using child processes and built the frontend with React.js.