Rolv Apneseth

rolv.apneseth@gmail.com rolvapneseth.com Rolv-Apneseth

Experience

Junior Front-End Developer, Kinesense Ltd. – Remote, Ireland Nov 2022 – Apr 2025

- Primary developer for the front-end of a web-based video investigation tool built using React, adapting the company's flagship desktop application to serve law enforcement and security clients globally in a more accessible format.
- Collaborated closely with the senior developer in charge of the project on architectural decisions, ensuring seamless integration between front-end and back-end systems. In addition, conducted thorough code reviews for back-end changes, both to identify potential bugs and to remain up-to-date on relevant functionality.
- Built maintainable custom components (e.g. interactive video timeline with events, video overlays for motion detection, progress indicators for video import and analysis) using modern React and CSS features to minimise complexity while still enhancing user experience.
- Leveraged Redux for state management and RTKQuery for efficient API handling, ensuring scalability and robust debugging capabilities.
- Developed comprehensive unit tests with Jest and maintain Azure DevOps CI/CD pipelines, ensuring high code quality and reliability.

Junior Front-End Developer, Bottletop Media – Remote, Ireland Sept 2021 – Nov 2022

- Optimised and maintained websites for over 1000 existing clients using modern HTML, CSS, and JavaScript.
- Implemented custom styling and interactive features for clients, leading to increased customer satisfaction.
- Using modern CSS and a touch of VB.NET, created demo websites with unique, custom layouts for big potential clients, greatly increasing their interest in the company's product.
- Developed a custom Selenium (browser automation) script in Python to automate the creation of page structures on the company's proprietary CMS software for new clients' websites, speeding up the initial setup stage by over 200%.

Skills

Languages: Rust | TypeScript | JavaScript | Lua | Python | Bash | SQL | Go

Technologies: React | Redux | HTML | CSS | SCSS | Tailwind CSS | Docker | GitHub Actions

Interests: Open-Source Software | Linux | Self-Hosting | Computer Building

Soft Skills: Self-Driven | Passionate | Fast Learner | Curious | Receptive to Feedback

Natural Languages: English | Portuguese | Norwegian

Personal Projects

frankfurte-rs Crate | Docs | Code

Rust, Clap, Tokio, Reqwest, CI/CD

- Rust library and CLI to interface with Frankfurter, a currency exchange rate API, to enable efficient retrieval and processing of exchange rate data via safe and correct bindings.
- Implemented custom, restrictive types to validate dates and currency values, ensuring only valid data is sent to API.
- Created comprehensive unit and integration tests, including property based tests for custom types to ensure they can handle any inputs. Tests are only run against locally hosted versions of the API setup via a custom Docker compose file, for efficient testing without impacting the public API.

World Wonders API Demo | Code

Rust, Axum, Docker, Open API, CI/CD

- An API written in Rust using Axum, which provides information about famous architectural wonders from around the world, created to explore back-end technologies as well as my interest in history.
- Developed a Docker image for seamless self-hosting, accessible via Docker Hub, and provided detailed documentation through a web page along with an OpenAPI specification.
- Automated CI/CD processes using GitHub Actions, including running unit and integration test, rebuilding the Docker image, and deploying it to the VPS, leading to an improved developer experience with a one-step deployment process.

ps-typer PyPi | Code

Python, PyQt5, PyQtGraph, NLTK, SQLite

- A GUI application written in Python using PyQt5, made for practising typing with a modern, minimalistic UI, made using the PyQt5 library for Python. I used this application to go from 40 WPM to 85 WPM and continue to use it to this day.
- Leveraged various corpora of text from the NLTK library to generate near endless content of full, proper sentences for a user to practice typing with. This was one of my main goals in creating this project.
- Implemented a user statistics system with the use of a local SQLite3 database, so that users can track the progression of their typing speeds. The results are visualised using PyQtGraph for easy consumption.

Open Source Contributions

Yazi ***(Rust, Tokio)*****:** A modern terminal file manager

languagetool-rust ***(Rust, Tokio, Clap)*****:** Rust bindings to connect with a LanguageTool server API

rustywind ***(Rust)*****:** Formatting tool for sorting Tailwind CSS classes

wpaperd ***(Rust, Linux, Clap)*****:** Minimal wallpaper daemon for Wayland on Linux

libmacchina ***(Rust, Linux)*****:** A Rust library providing access to all sorts of system information

macchina ***(Rust, Clap)*****:** CLI tool for displaying system information

tldr ***(Documentation, Linux)*****:** A collection of help pages for command-line tools