Guillaume Bournel

/users/11676080/guimauve   /guimauveb   Nantes, FR
in/in/guillaume-b-4a167b19b   guimauve.io
contact@guimauve.io

ABOUT ME

I am a passionate software developer who enjoys spending time in the various fields of programming. I am self-educated, and always keeping up to date and strengthening my skills play a big role in my everyday life.
The languages I have used the most include Rust, Typescript, Javascript, Python, C++, Bash script. I also tinker with Haskell from time to time. I have a strong interest in web development as well as OS development, embedded development, game development and compilers, even though most of my experience comes from web development. I am currently focusing on Rust and Typescript, as well as functional programming.

OBJECTIVE

A position developing software and services.

SKILLS

Languages
- Rust
- Typescript/Javascript
- Python
- C++
- C
- Bash script
- Haskell

Libraries and frameworks
- React/React Native (mostly using Typescript)
- Redux
- Django
- Actix
- Yew
- curses/ncurses
- curl/curlpp
- Flask
- jQuery

DevOps/Services
- Git
- Github
- AWS
- Integromat
- nginx
- Postgres
- MySQL
- Docker

SOFTWARE DEVELOPER EXPERIENCE

Full stack developer - Dashdoc   January 2021 - Present

Working on every end of the codebase (back-end - integrations - front-end - mobile application) playing various developer roles. From implementing quick but much needed features (such as small API or UI changes needed for a new customer) to larger projects involving all ends of the product (such as bigger features that have been requested by a few clients for some time).

Since I am the one to whom customer support goes first in case of trouble, I also play an important role in quick and critical bug resolution.

I write a lot of React using Typescript, whether it be by taking part in the refactoring of the UI by creating 'ui-kit' generic and reusable components using functional components, or simply by adding new features. I also work on back-end features, such as adding and updating API endpoints and creating in-house tools to help customer support and operations team.

I also work with Integromat to create new deployment scenarios, fix existing ones, add new modules and keep them up to date.

The languages and frameworks I have been using the most are Typescript/React and Python/Django.

Full stack developer - guimauve   August 2019 - Present

Creating all kinds of projects, consistently learning new technologies. Currently focusing on Rust and functional programming.

Blogger - guimauve.io   June 2020 - Present

I maintain an active blog discussing anything related to what I encounter during project development. I also write tutorials and guides.

Full stack web developer - institut-sylvie.fr   January 2020 - May 2020

Institut Sylvie - Beauty salon website

Fully responsive website

Here is my first 'real world' project. The shop owner gave me the chance to build my first website from scratch.

Back-end is handled by Flask. Front-end is built from HTML/CSS and plain JavaScript as I didn't feel the need for any library.
All persistent data are stored on an AWS MySQL database connected through MySQLdb Python library.
The website is hosted on a AWS EC2 instance.

This project taught me a lot about front-end and back-end web development as I had to do everything by myself, from designing the pages to tweaking nginx configuration files.

Features
- Fully responsive design
- The owner can post news/updates that are then displayed on the homepage
- Users can register and get notified about their next appointment, subscribe to the shop newsletter
- The owner can manage products and services displayed as well as create new categories of products or services
- The owner can also manage business hours in real-time
- Visitors can contact the shop through a simple form

Technologies

aws bootstrap css flask html js mysql
nginx python

OTHER EXPERIENCE

Front Desk - Kyriad Hotel - Nantes   December 2019 - July 2020

- Welcoming international guests in a warm and friendly manner.
- Answering questions and addressing complaints
- Knowing all essential aspects of the hotel operations.
- Registering guests and managing the rooming chart.
- Collecting payments by accepting cash, check, or charge payments from customers.

Auto Technician - Norauto - Saint-Herblain   June 2018 - September 2019

Performed maintenance and repairs on all makes and models, including:

- Oil changes
- Tire rotations and replacement
- Transmission flushes
- Front-end alignments
- Battery installations
- Headlight/taillight installations

Conducted necessary repairs and took vehicles for test drives to verify soundness.

PROJECTS (See more projects at guimauve.io/projects)

guimauve.io

I originally built my website using Flask and some vanilla Javascript but I couldn't easily publish content without having to manually deal with HTML and CSS.

I heard about Rust and all its great features and how everybody is loving it, so I decided to kill two birds with one stone: rebuild my blog so that I can easily upgrade it and publish content, and learn more about Rust!

I went with Actix, Diesel, Postgres and Yew.

Even though some of these technologies (especially Yew and more specifically its functional components) are still considered as "experimental" and not really meant to be used in production, I only encountered a few quirks that didn't really have any real negative impact on the project.

Sure this is not some huge app used by millions and with critical features, but I'm confident in saying that this stack could definitely be used for a more serious project.

Features
- Uses some cutting edge technologies, and it's really cool
- Fast, type and memory error-free thanks to Rust's strong typing system and borrow checker
- Uses a Redux-like store to access data from any component and speed-up rendering
- Super easy to upgrade UI with a React-like function component based system
- Articles can easily be created and edited using a basic Chapter > Content structure
- And a lot more cool things...

Technologies

aws psql trunk actix yew wasm
rust diesel

tinyDLM - C++ Download Manager

C++ download manager built around cURLpp and ncurses

I've never encountered a simple yet efficient download manager that would be free AND still maintained.

So I decided to build my own and figured out it would be a great way to learn more about C++, multithreading, sockets and desktop applications and I did learn a lot along the way!

I could have used multiple non-blocking sockets instead of threads and therefore only use one thread for the download manager part, but I really wanted to learn more about multithreading and its pitfalls.

The main window is actually made of two subwindows: the downloads list displayed as a curses menu and a 'downloads status' window sitting next to it and updated in its own thread.
Using different threads for windows was also a challenge since curses (actually ncurses) is not multi-thread safe, but using C++ mutexes I managed to make it work smoothly.

Features
- Works on macOS and Linux Debian based distributions
- Supports Direct Download Links
- Supports simultaneous transfers (tested up to 32),
- Transfers can be paused/resumed/killed,
- Inactive transfers can be cleared from the list,
- Displays current download speed and progress for each transfer,
- Displays current transfer status,
- Signals user input and URL/HTTP errors,
- Very low CPU/RAM consumption,
- Partially supports adding multiple links at the same time

Technologies

cmake cpp curl curlpp curses linux
macos multithreading

EDUCATION

Bachelor of Law - University of Poitiers

Public law/Tax law

Mathematics - Physics - Computer Science - University of Nantes

Mathematics/Computer science