Cyril De Graeve

React / Node.js Developer
XP: 3 years

contact@degraeve.dev / +33 6 76 04 54 31

Synthesis

As a freelance developer, I offer a range of web development services that cover all stages of the process of creating a site, SaaS or application. I am particularly proficient in React, Next.js, Node.js and PostgreSQL. With my practice and knowledge of the incredible Javascript ecosystem, I am able to solve complex problems by creating functional solutions and modern aesthetic interfaces tailored to your needs.

Training & Certification

2021 Fullstack JavaScript Developer Training (O'Clock)
2022 Professional Title of Web and Mobile Web Developer (DWMM)

Technical & functional skills

Programming language: Javascript, TypeScript, SQL

Web: HTML5, CSS3, React, Node.js, Express.js, Next.js, RESTful APIs, Tailwind, Framer-motion, Responsive Design, SEO, Testing, Web servers (Nginx, Apache)

Databases: PostgreSQL, Supabase, Prisma, Drizzle, Sequelize

Cloud Computing: Docker

OS: Linux, Windows WSL

Source Control: Git, GitHub

Methodologies: Agile, Scrum

Professional experiences

SERIOUS PUBLISHING

2024 (2 months)

Freelance

> https://www.serious-publishing.fr

Fullstack JavaScript & DevOps Developer

CONTEXT:

For the Parisian publishing house Serious Publishing, I was responsible for the complete overhaul of their website, moving from a PHP/MySQL stack to a modern architecture based on JavaScript and PostgreSQL. The objective was to improve performance, security, user experience and add new features to increase the visibility of the publisher, highlight its activities and increase conversions (online book purchases).

TECHNICAL ENVIRONMENT:

React, Next.js, Node.js, Express, Tailwind, Shadcn, Framer-motion, TypeScript, PosgreSQL, NextAuth, JWT...

Front-end:

> Technologies used: React, Next.js, TypeScript, Tailwind CSS, Framer Motion.
> User interface design and development:

- Implemented Next.js for server-side rendering (SSR) and performance optimization.
- Using TypeScript for better maintainability and error reduction.
- Integration of Tailwind CSS for a responsive and modular design.
- Added animations with Framer Motion to improve user experience.

> SEO optimization and GDPR compliance:

- Implementation of SEO best practices: meta tags, dynamic sitemap, optimized URLs.
- Compliance with the GDPR.

> Creation of a personalized Back Office adapted to the client's content management needs.

Back-end:

> Technologies used: Node.js, Express.js, PostgreSQL, NextAuth.js, JWT.
> RESTful API development:

- Creation of a REST API with Express.js to manage CRUD operations (for the Back-office).
- Structuring the code according to the MVC model for better organization.

> Authentication and security management:

- Implemented NextAuth.js for user authentication (secure sessions).
- Use of JWT for managing access and refresh tokens.

> Database:

- Migrating to PostgreSQL to take advantage of its advanced features.
- Database schema design, query optimization and use of indexes to improve performance.

> Back office security.

E-commerce features:

> Complete online store:

- Development of the user shopping cart system with data persistence.
- Dynamic calculation of shipping costs based on weight, destination and current postal rates.
- Product management with the possibility of adding/updating/deleting via the Back office.

> Integration of secure payments:

- Integration of the PayPal API to accept payments via PayPal or credit card.

Responsive Design:

> Mobile-first design to ensure an optimal experience on smartphones.
> Using and optimizing Tailwind for adaptability across all screens.
> Testing on different browsers and devices to ensure cross-platform compatibility.

Deployment and DevOps:

> Server configuration:

- Setting up a VPS server under Linux (Ubuntu).
- Installation and configuration of Nginx as a reverse proxy to serve the Next.js application and manage SSL certificates.
- Using PM2 for deploying and managing Node.js processes.

> Safety and performance:

- Configuring firewalls and security rules (UFW).
- Installation of SSL/TLS certificates via Let's Encrypt to secure communications.
- Optimization of Nginx settings to improve response times and manage caching.

CODEBERRY *2024 (5 days)*

Freelance

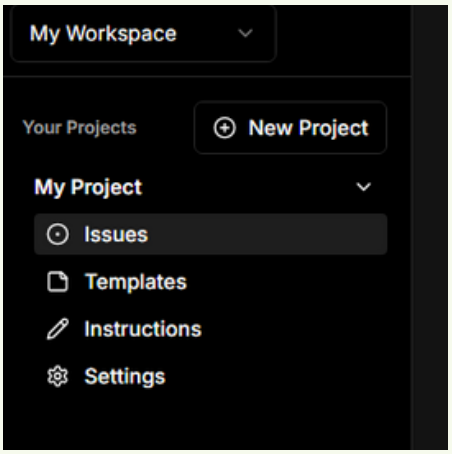
Fullstack JavaScript Developer

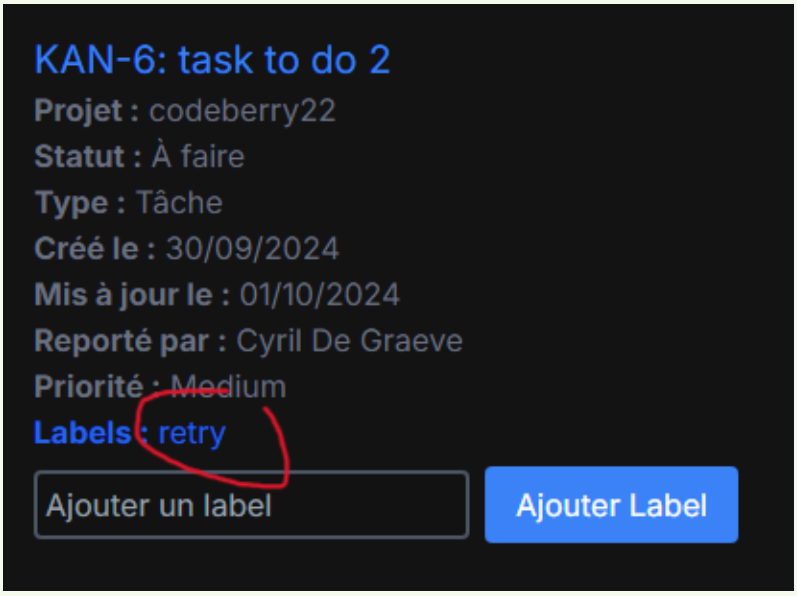
CONTEXT:

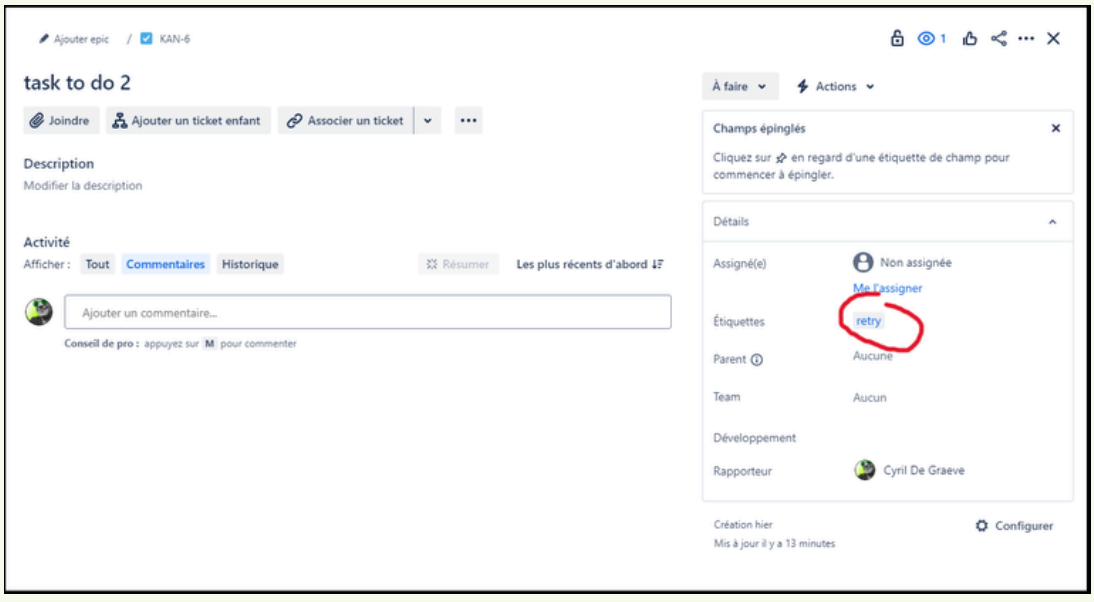
For a startup in the financing phase, I participated in the development of the MVP (Minimum Viable Product) of a SaaS intended to automatically generate Pull Requests from Jira and/or Linear tickets. My main role was to integrate Jira into the application to retrieve certain tickets and automate their processing, which facilitates the workflow of the development teams and improves the efficiency of the code management process.

TECHNICAL ENVIRONMENT:

React, Next.js, TypeScript, Tailwind, Shadcn, Framer-motion, Supabase, Drizzle, Server Actions, Clerk, Vercel...







Installation and environmental management:

> Setting up the development environment with Next.js to benefit of the latest features.
> Configuring continuous deployment on Vercel for fast and reliable hosting.
> Managing user authentication with Clerk to secure access to the application.

Jira integration in the application:

> Database:

- Using Supabase for data management.
- Created the 'jira_issues' table to store Jira issue information.
- Performed schema migrations with Drizzle ORM to facilitate changes and maintain database consistency.

> Back-end:

- Development of an API with Next.js API Routes to interact with the Jira API.
- Retrieving Jira tickets via the Jira REST API, with authentication and permissions management.
- Creation of functions to automate the generation of Pull Requests based on retrieved tickets.

> Front-end:

- Display of Jira tickets in the application in real time.
- Using Next.js Server Actions to manage server-side actions directly from React components.
- Implementation of an intuitive user interface with Tailwind CSS and Shadcn/UI.

Adding a 'label' field to the Jira application and tickets:

> Database:

- Added a 'label' field to the jira_issues table to associate labels with issues.
- Management of corresponding migrations in Supabase.

> Back-end:

- Created an API route to add and update Jira ticket labels from the application.
- Development of functions to synchronize labels between the application and Jira.

> Front-end:

- Updated the UI to allow users to manage labels.
- Using Framer-Motion to add smooth animations when adding or editing labels.

With a solid foundation for the MVP, I was able to help quickly deliver a working product **in just 5 days**, allowing the startup to present a demo to potential investors.

SAJIN PHOTO

2024 (2 months)

Freelance

Fullstack JavaScript & DevOps Developer

CONTEXT:

For a professional photographer wishing to offer his clients a real-time photo sharing experience during events, I designed and developed a web and mobile application for managing and sharing event photos. The goal was to enable the photographer to instantly share photos taken at events with attendees, improving customer engagement and overall satisfaction.

TECHNICAL ENVIRONMENT:

React, Node.js, Express, Next.js, TypeScript, Bcrypt, Chakra UI...

Event management:

> Features:

- Creation and management of personalized events with metadata (date, location, description).
- Hierarchical organization of photo albums by event for simplified navigation.

> Implementation:

- Designed the data model for events and albums in the database.
- Used Next.js for dynamic routing and server-side rendering of event pages.

Upload and secure storage:

> Features:

- Intuitive interface for mass uploading of high resolution photos by the photographer.
- Secure image storage with access control to ensure confidentiality.

> Implementation:

- Integration with a cloud storage service (AWS S3 type) for optimal scalability.
- Implementation of an authentication and authorization system with JWT and Bcrypt.
- Management of asynchronous uploads with real-time feedback to the user.

Intuitive visualization:

> Features:

- Clean photo gallery with sorting, filtering and search functionalities.
- Grid view with support for zoom and full screen mode.

> Implementation:

- Using Chakra UI to build a responsive, accessible and responsive interface.
- Optimization of image loading with lazy loading and compression techniques.

Envoi de photos personnalisé :

> Features:

- Ability for users to select photos and share them via email.
- Personalization of messages with dynamic templates.

> Implementation:

- Integration of a transactional email sending service (SendGrid).
- Created secure APIs to manage sending requests and track statuses.
- Queue management for sending emails to ensure reliability.

Simplified administration:

> Features:

- Dashboard for the photographer to manage events, photos and users.
- Real-time statistics on user engagement (views, shares, downloads).

> Implementation:

- Development of a back office with React and Chakra UI, secured by authentication.
- Management of roles and permissions to differentiate access between administrator and users.
- Using RESTful API with Express.js for CRUD operations.

Deployment and DevOps:

> Server configuration:

- Setting up a VPS server under Linux (Ubuntu) to host the application.
- Installation and configuration of Nginx as reverse proxy and web server.
- Using PM2 for Node.js process management and continuous deployment.

> Safety and performance:

- Installation of SSL/TLS certificates via Let's Encrypt to secure communications.
- Configuring firewall (UFW) and security rules to protect the server.
- Optimized Nginx settings to improve performance and manage caching.

HYPER-FREE

2023 – Today

Freelance

> https://hyper-free.cyrildegraeve.dev

Fullstack JavaScript Developer & Copywriter

CONTEXT:

As a Fullstack JavaScript and Copywriter developer, I created HYPER-FREE, a platform presenting my packaged service offers in web development and copywriting. My goal is to combine the power of code and the art of words to help freelancers, artists, coaches, photographers, startups, SMEs... boost their online presence and increase their reputation. This initiative allows me to offer comprehensive solutions that precisely meet the needs of my clients, while highlighting my unique expertise derived from 20 years of experience in journalism.

TECHNICAL ENVIRONMENT:

React, Next.js, TypeScript, Panda CSS, Ark UI, Framer-Motion, i18next...

Development of the web platform:

> User interface design:

- Creation of a modern and responsive design, ensuring an optimal user experience on all devices.
- Using Panda CSS (and Ark UI) for flexible layout and quick style customization.
- Implemented reusable components with TypeScript to improve code maintainability.
- Creation of a contact form allowing the visitor to specify their request as best as possible.
- Internationalization of content with the i18next library to manage dynamic translations (French/English).

> Performance optimization:

- Implemented server-side rendering (SSR) with Next.js to improve page loading time.
- Use of optimization techniques such as code splitting and lazy loading.

Copywriting and marketing content:

> Creation of impactful content:

- Writing engaging texts to present my web development and copywriting services.
- Highlighting my experience as a journalist to establish a connection with the target audience.

> Niche Marketing Strategy:

- Development of messages tailored to the specific needs of potential customers.
- Optimizing content for SEO to improve online visibility.

Personalized service offers:

> Development of four service packages (creation from scratch and/or complete redesign).

> Detailed presentation of the offers:

- Creation of dedicated components for each formula detailing the advantages and specificities.
- Integration of calls to action (CTA) to encourage contact.

SEO optimization and performance:

> Improved online visibility:

- Use of SEO best practices: meta tags, optimized URLs, XML sitemap.
- Implementation of structured content with schema.org to improve SEO.

> Analysis and monitoring:

- Google Analytics and Google Search Console integration to track site traffic and performance.
- Continuous data-driven adjustments to improve conversion rate.

By offering **a rare combination of web development and copywriting skills**,
I meet the needs of clients looking for a complete solution for their online presence.

ARTIFICIAL LIFE COACH

2023 (2 months)

Freelance

> https://artificial-life-coach.vercel.app

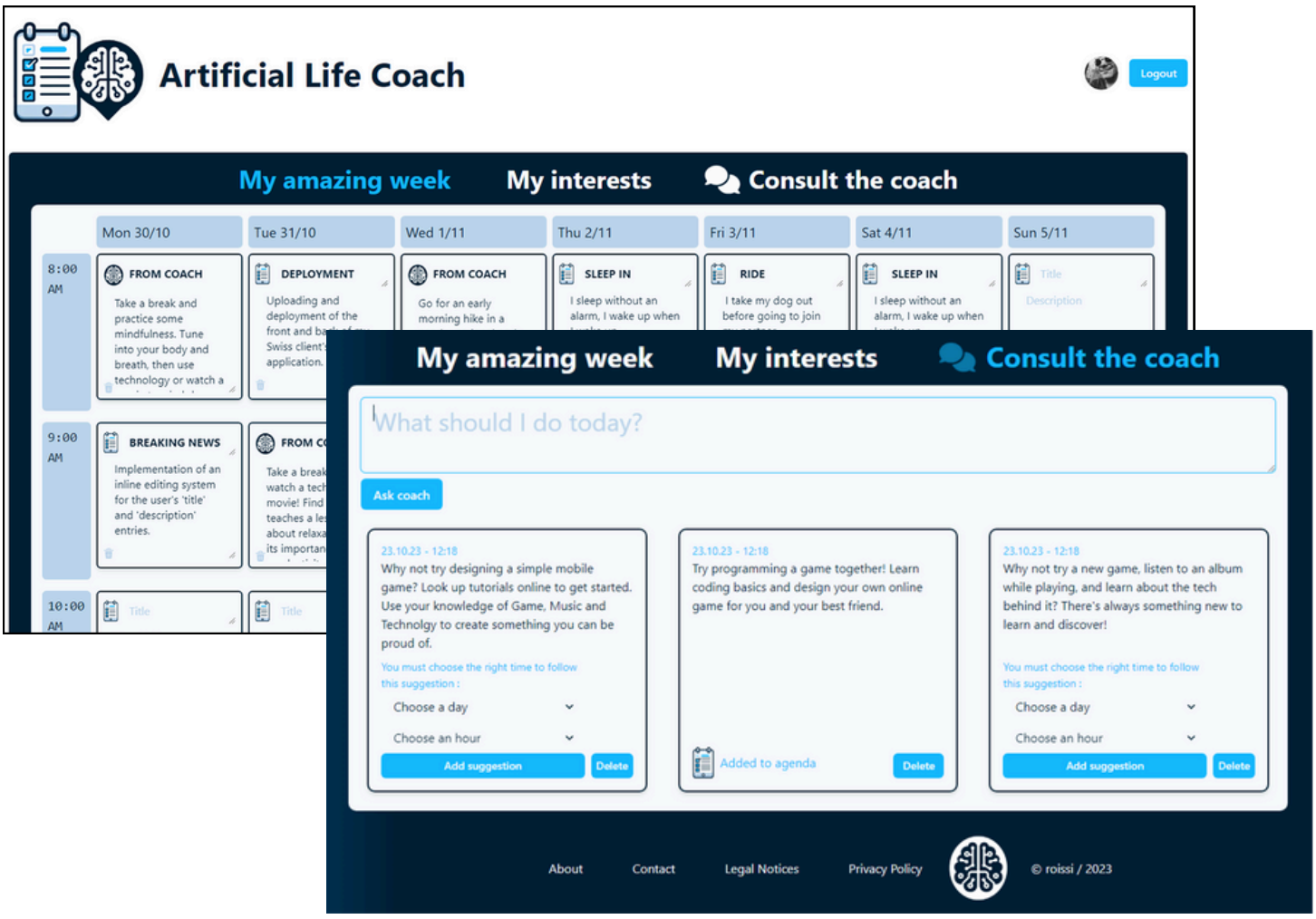
Fullstack JavaScript Developer

CONTEXT:

I designed from A to Z (Front-end and Back-end) "Artificial Life Coach", a web and mobile application offering users an artificial life coach to manage their weekly or daily interactive agenda. The goal was to create a personalized experience that allows users to plan their activities based on their current interests and needs, while benefiting from tailored advice.

TECHNICAL ENVIRONMENT:

React, Node.js, Express, Next.js, PostgreSQL, Sequelize, Bcrypt, JWT, Chakra UI, Framer-Motion...



Main features:

> Personalization of interests and needs:

- Users can enter their interests (one or more) and set their current needs for a specific duration (in days).
- This information is transmitted to the virtual coach, who generates personalized suggestions to enrich their daily lives.

> Interaction with the virtual coach:

- Users can ask the coach questions, such as:
  - “I'm not in good shape, give me an idea of a suitable activity for the afternoon.”
  - “I have an important meeting tomorrow morning, how can I prepare to be at my best?”
- The coach provides tailored answers and advice in real time thanks to OpenAI API integration.

> Interactive calendar management:

- Possibility of adding the coach's suggestions directly to the personal calendar, at the desired date and time.
- Users can also enter their own schedules and activities, similar to Google Calendar.

AI integration with OpenAI API:

> Using the OpenAI GPT-3.5-turbo-1106 API to generate intelligent, contextual responses of the virtual coach.
> Optimization of prompts to obtain relevant responses while respecting API limits.

Front-end development:

> Design of an intuitive and responsive user interface, suitable for desktop computers and smartphones.
> Use of CSS Modules or Styled Components for effective style management.
> Managing application state with React's Context API.

Back-end development:

> Creation of a RESTful API to manage authentication, user data, interactions with the calendar and requests to the OpenAI API.
> Implementation of a secure authentication system with JWT for tokens and Bcrypt for password hashing.
> Database management with PostgreSQL and Sequelize as ORM.
> CORS management:

- Configuration of CORS (Cross-Origin Resource Sharing) to enable secure communication between the Front-end and the Back-end.
- Implementation of precise rules to only authorize trusted origins, thus strengthening the security of the application.

Safety and performance:

> Validation and sanitation of user inputs to prevent SQL injections and XSS attacks.
> Cache implementation for frequent responses to reduce response times and costs related to the OpenAI API.
> Optimizing database queries and using indexes to improve performance.

CLAIRIOS

2024 (2 months)

Freelance

> https://clairios.com

Développeur Front-end + DevOps

CONTEXT:

For Clairios, an IT services company specializing in outsourcing and consulting, I designed and developed a dynamic and modern website. The main objective was to highlight the services offered, improve the company's online visibility and offer an optimal user experience to visitors, in particular SMEs and VSEs looking for reliable IT solutions.

TECHNICAL ENVIRONMENT:

React, Next.js, TypeScript, Tailwind, Framer-motion...

User interface design:

> Modern and responsive design:

- Creation of a clear, intuitive and accessible user interface, highlighting services such as outsourcing, security, support and cloud solutions.
- Using Tailwind CSS for responsive design and quick style customization.
- Framer-Motion integration to add smooth animations, improving engagement and user experience.

> Accessibility:

- Compliance with WCAG standards to ensure optimal accessibility for users with disabilities.
- Accessibility tests carried out to verify compliance.

Front-end development:

> Performance optimization:

- Implemented server-side rendering (SSR) with Next.js to improve loading times and SEO.
- Use of code splitting (multiple components) and lazy loading to optimize resource loading.

> Optimized contact form:

- Creation of a user-friendly contact form with client and server side validation.
- Implemented an API to process submissions and send data securely.
- Added confirmation notifications to improve user experience.

Optimization for SEO:

> Improved online visibility:

- Incorporation of custom meta tags for each page (title, description, keywords).
- Use of structured data (schema.org) to improve the interpretation of content by search engines.
- Automatic generation of an XML sitemap to facilitate indexing.
- Optimizing URLs and content to target relevant industry keywords.
- Optimizing images with Next.js Image Component for better performance.

Deployment and DevOps:

> Server configuration:

- Setting up a VPS server under Linux (Ubuntu).
- Installation and configuration of Apache as a web server to serve the application.
- Configuring Node.js to run Next.js application in production mode.
- Using PM2 for managing Node.js processes, ensuring stability and automatic recovery in case of failure.

> Safety and performance:

- Installation of SSL/TLS certificates via Let's Encrypt to secure HTTPS communications.
- Configuring the firewall (UFW) and setting up security rules to protect the server.
- Optimized Apache settings to improve performance, including enabling Gzip compression and configuring caching.

> Deployment automation:

- Implementation of deployment scripts to facilitate continuous updates.
- Integration with Git for version control and continuous deployment.

TÉLÉGRAPHE

2023 (1 month)

Freelance

> https://neo-telegraphe.vercel.app

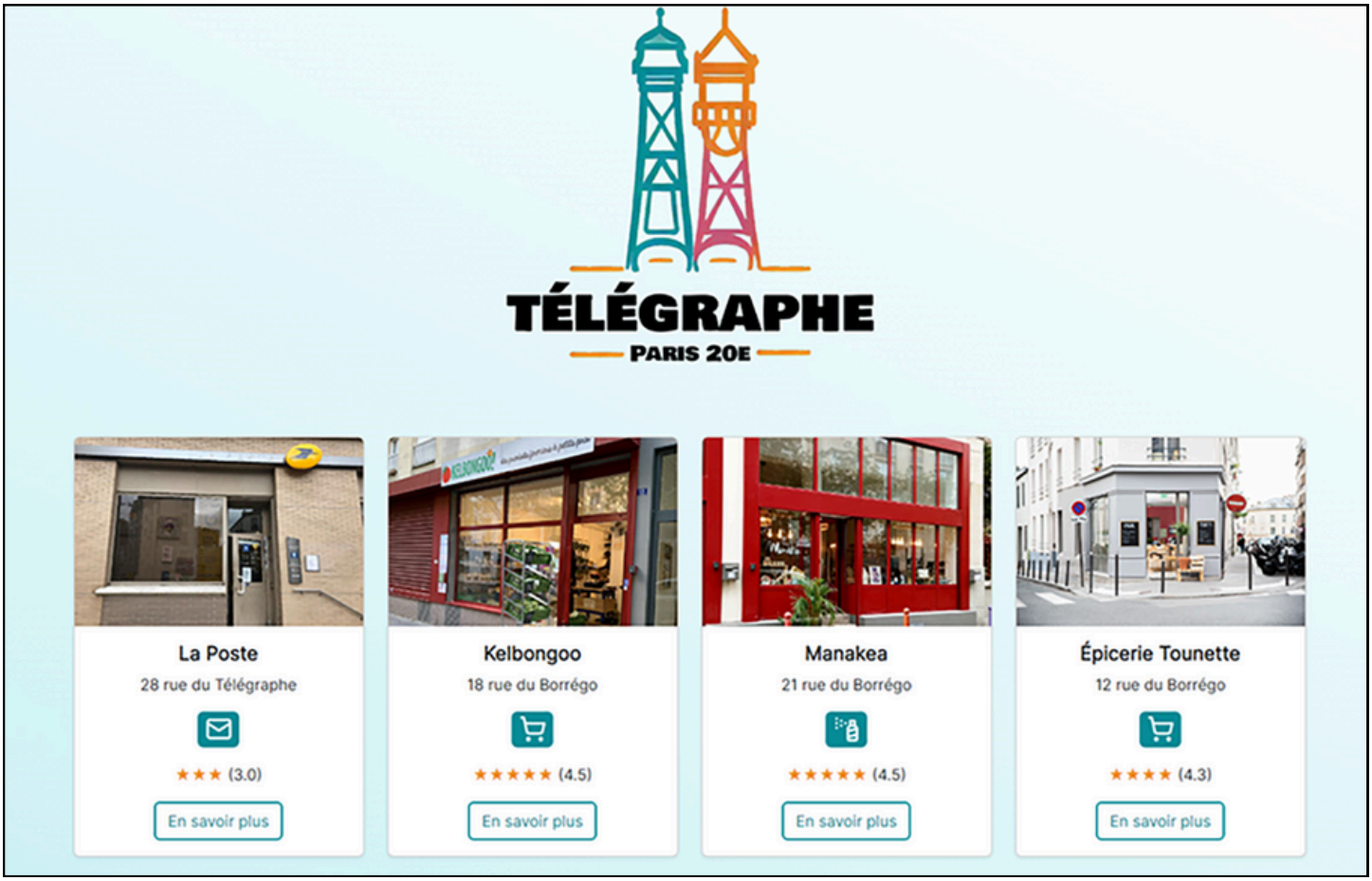
Fullstack JavaScript Developer

CONTEXT:

I created "Télégraphe", a web platform dedicated to discovering the shops in the Télégraphe district, located in the 20th arrondissement of Paris. The goal was to help residents and visitors easily find local shops and learn more about what they offer. This project aims to strengthen community ties, support local businesses and energize the neighborhood.

TECHNICAL ENVIRONMENT:

React, Node.js, Express, Next.js, PostgreSQL, Supabase, Bootstrap...



Main features:

> User authentication:

- Implementation of a secure registration and login system for community members.
- Management of user sessions with JWT to secure communications.

> Location of stores:

- Integration of the Google Maps API to display an interactive map of the neighborhood.
- Users can locate stores on the map with custom markers.

> Ratings and reviews:

- Consultation of notes and reviews left by other members on the stores.
- Star rating system and detailed comments to evaluate businesses.

> Adding personal opinions:

- Ability for users to write and save their own reviews of the stores visited.
- Moderation of reviews to ensure the quality and relevance of content.

> Responsive interface:

- Design of a fully responsive application, suitable for desktops, tablets and smartphones.
- Use of Bootstrap to facilitate responsive design and ensure a consistent user experience.

Front-end development:

> Design of an intuitive and friendly user interface, highlighting stores and reviews of the community.
> Using Next.js for Server Side Rendering (SSR) improving performance and SEO.
> Implementation of reusable components with React for better code maintainability.

Back-end development:

> Creation of a RESTful API with Express.js to manage CRUD operations of stores, reviews and users.
> Using Supabase (based on PostgreSQL) for database and authentication users.
> Managing relationships between tables (stores, users, reviews) to maintain integrity data.

Google Maps API integration:

> Using the Google Maps JavaScript API to display interactive maps.
> Customization of markers to represent different stores.
> Implementation of features such as zooming, filtering of stores by category, and the display of detailed information when clicking on the markers.

DON EFFICACE

28/02/2023 – 29/06/2023 (4 mois)

Fixed-term contract

Fullstack JavaScript Developer

CONTEXT:

As part of a 4-month fixed-term contract, I participated in the internationalization of the platform of the Giving What We Can association to create the French version of the new Don Efficace website. Don Efficace is a non-profit NGO whose mission is to finance actions aimed at reducing the suffering of all sentient beings (human and non-human) in the most effective way possible, directly or indirectly.
During this project, I worked closely with English and Swiss-German developers, which involved numerous pair programming sessions and enriching intercultural communication. I was also responsible for updating the marketing part of the site using Sanity, an open-source headless CMS written in JavaScript and designed for developers.

TECHNICAL ENVIRONMENT:

React, Node.js, Express, Next.js, TypeScript, Sanity, i18n, Jest...

Internationalization of the website:

> Implementation of internationalization (i18n):

- Configuring Next.js to support multiple languages, particularly French and English.
- Using libraries such as react-i18next to handle dynamic translations (English/French).
- Managing JSON translation files to structure content in different languages.

> Content adaptation:

- Translation of existing content into French.
- Adjustment of formatting and style to adapt to the linguistic and cultural particularities of the French-speaking audience.

> Testing and validation:

- Implementation of unit tests with Jest to verify the correct functioning of multilingual components.
- Carrying out manual tests to ensure the consistency and quality of translations.

International collaboration:

> Effective communication:

- Participation in regular meetings with the English and Swiss-German teams.
- Use of English as the main working language to facilitate communication.

> Pair programming :

- Pair programming sessions to solve complex problems, share knowledge and harmonize development practices.
- Use of collaboration tools such as Visual Studio Live Share or video conferencing platforms with screen sharing.

Updating the marketing part with Sanity CMS:

> Sanity CMS integration:

- Configuration of Sanity as a headless CMS to manage the dynamic content of the site.
- Creation of personalized schemas for different types of marketing content (articles, pages, testimonials).

> Front-end development:

- Setting up dynamic pages in Next.js to display CMS content.

> Content Optimization:

- Collaborate with the marketing team to ensure content is optimized for the target audience.
- Implementation of SEO best practices to improve site visibility.

Technical improvements and optimization :

> Performance :

- Optimization of page loading time using Next.js static pre-rendering (SSG).
- Using Lazy Loading for non-critical images and components.

> Code quality:

- Adoption of TypeScript to improve code maintainability and robustness.
- Implementation of coding standards with ESLint and Prettier.

> Testing:

- Wrote unit and integration tests with Jest to ensure code reliability.
- Test coverage aimed at achieving a high level of confidence in production deployment.

J'ADOPTE UN HUMAIN

2021 (1 month)

END OF TRAINING PROJECT

Product Owner & Back-end Developer

CONTEXT:

At the end of training at O'Clock, I participated as Product Owner and Back-end Developer in the design of "I adopt a human", a responsive web application for a fictitious animal shelter. The main objective was to allow future adopters to choose the ideal companion according to their desires, values and needs. This project, carried out using the Agile method (Scrum) with a team of 4 other developers, aimed to modernize existing showcase sites, which are often not very interactive, and to provide shelters with a simplified and secure animal management system.

TECHNICAL ENVIRONMENT:

React, Node.js, Express, PostgreSQL, JWT, Bcrypt, Swagger, Jest...

Product Ownership:

> Agile project management:

- Implementation of the Scrum methodology to organize development over a period of one month.
- Animation of daily meetings (stand-up meetings), sprint planning, reviews and retrospectives.
- Prioritization of the product backlog based on project needs and available resources.

> Collection of needs and definition of specifications:

- Development of user stories with clear acceptance criteria.
- Close collaboration with the team to define key functionalities and ensure a common vision of the project.

Back-end development:

> RESTful API design and development:

- Used Node.js and Express.js to create a robust RESTful API.
- Structuring the code according to the MVC model for better organization and maintainability.
- API documentation with Swagger to facilitate Front-end integration and understanding by the team.

> Database management:

- Design of the relational schema with PostgreSQL, modeling the key entities: users, animals, profiles, etc.
- Implementation of relationships between tables to ensure data integrity.
- Optimization of SQL queries to improve application performance.

> Authentication and security:

- Implementation of authentication with JSON Web Tokens (JWT) to secure API endpoints.
- Hashing of user passwords with Bcrypt to guarantee the confidentiality of sensitive information.
- Management of roles and permissions for the three access levels:
  - User: animal consultation, profile creation, adoption request.
  - Staff: animal management (addition, modification, deletion), consultation of user profiles.
  - Administrator: user management, moderation, full access to all features.

> Development of the matching system:

- Creation of a matching algorithm between user and animal profiles.
- Taking into account multiple criteria such as the type of animal, behavior, specific needs, environment of the adopter.
- Results return in the form of an ordered list and a matching percentage, providing the user with the best potential matches.

Collaboration with the Front-end team:

> Coordination with Front-end developers working with React for API integration.
> Integration tests to ensure consistency between Front-end and Back-end.
> Resolution of technical issues and adjustments based on team feedback.

Testing and quality assurance:

> Writing unit tests and integration tests with Jest to verify correct operation Back-end modules.
> Implementation of continuous integration to automate tests and maintain code quality.