Récy&Co

Sorting is fun !

Recy&Co (RecyandCo.fr), sorting is fun! is an educational and entertaining web project, designed to raise awareness among children and families about selective sorting. It takes the form of an interactive website including local sorting instructions, a “Where to throw?” search engine, and a drag & drop game with a mascot, Recy the raccoon.
Developed with an eco-design approach (lightweight design, responsive, offline access), this project aims to make sorting simpler, more fun, and more accessible.
Intended to be shared with the inhabitants of the Communauté de Communes d’Évron, this project aspires to become a local, free, and evolving educational tool.

Holberton
SCHOOL

Roche Samira

Formation Holberton School Laval
RNCP5 : Web and Mobile Web Developer

Table des matières

A. Initial Objectives

At the start of the Récy&Co project, the main objectives defined were as follows:

Create an educational web application to raise awareness among children and families about waste sorting, making this learning experience fun and accessible.

The MVP (Minimum Viable Product) was to include:

- An interactive website built with HTML, CSS, JavaScript, and Python (Flask)
- A MySQL database to manage users (registration, login, scores, badges)
- Six main pages: home page, local sorting guidelines, "Where to throw?" search engine, drag & drop sorting game, user space, and About page
- Integration of Récy the raccoon mascot to make the site appealing
- Adherence to eco-design principles (lightweight site, responsive, optimized)

The project was initially designed for residents of the Communauté de Communes d'Évron, with the ambition of becoming a local, free, and scalable educational tool.

B. What Was Actually Accomplished

The Récy&Co MVP was developed and delivered on schedule, with all essential features operational. The final application consists of six fully functional and interconnected pages.

1. Technical Architecture

The application is based on a clear and maintainable three-tier architecture:

- **Frontend**: HTML5, CSS3, and vanilla JavaScript to ensure lightness and performance
- **Backend**: Python with Flask framework to handle business logic and HTTP routes
- **Database**: MySQL with SQLAlchemy ORM, comprising 6 relational tables (users, scores, badges, Shop, user_inventory, and user_badges)

User authentication is secured via JWT (JSON Web Tokens) with storage in HTTP-only cookies, and passwords are hashed with bcrypt.

2. Developed Features

Home Page The home page introduces the Récy&Co project and its mascot, Récy the raccoon. It offers clear navigation to all site features and includes calls-to-action to encourage users to register and start playing.

Bins Page This page displays sorting guidelines organized by bin type (yellow for recyclable packaging, green for glass, gray for household waste, etc.). When users click on a bin, an associated list of waste items appears with visual icons and practical tips for each type of waste. This organization facilitates understanding and memorization of sorting rules.

Sorting Guide Page The smart search engine allows users to type a keyword (for example, "bottle") and instantly get all corresponding results (plastic bottle, glass bottle, etc.), with the appropriate bin and associated sorting guidelines for each. This feature addresses a concrete daily need.

Sorting Game Page The interactive drag & drop game constitutes the educational core of the application. Users must drag waste items into the correct bins. Each correct answer earns 1 point, while an incorrect answer doesn't earn points but allows learning from the mistake. The scoring system encourages practice and progressive learning.

User Space After registration and login, each user accesses their personal profile displaying their information (email, username, total score, registration date) as well as detailed statistics: number of games played, best score achieved, and total number of correctly sorted waste items. This space reinforces engagement and motivation.

About Page This page presents the project context, my personal journey and training at Holberton School Laval, as well as the eco-design principles applied. It also integrates legal notices and regulatory obligations necessary for public website deployment (GDPR, privacy policy, terms of use).

3. Evolution of Geographic Scope

Although the project was initially designed for the Communauté de Communes d'Évron, analysis of the sorting guidelines JSON file revealed that the information was general enough to be applicable nationwide. The project was therefore expanded to serve families throughout France, thus increasing its potential impact and educational value.

4. Deferred Features

In accordance with Agile methodology and the MoSCoW prioritization defined during planning, certain features were deliberately excluded from the MVP to focus on essential functionalities:

- Bonus shop and virtual currency system
- Advanced customization features (avatars, elaborate themes)

- Integration with external national databases
- Multilingual translation

These elements constitute future development paths once the MVP is validated and deployed.

C. Metrics and Performance

The Récy&Co project stands out for exceptional technical performance and strong commitment to eco-design.

Development Metrics

- 6 fully developed and tested functional pages
- Over 140 commits on GitHub, demonstrating iterative and documented development
- 6 relational tables in the MySQL database with well-defined relationships
- Over 20 automated tests with Pytest to ensure backend reliability
- 100% responsive design on desktop, tablet, and mobile, tested on all screen formats

Technical Performance (Lighthouse)

Lighthouse tests demonstrate the application's technical excellence:

- **Performance**: 91+/100
- **Best Practices**: 91+/100
- **SEO**: 91+/100
- **Accessibility**: High scores thanks to semantic HTML and appropriate color contrasts

Environmental Impact (EcoIndex)

The project's eco-responsible commitment translates into remarkable results:

- **Average EcoIndex score**: ~91/100 across all pages
- **Grade**: A (excellent) on all tested pages
- **Ranking**: Top 5% of the most environmentally-friendly websites worldwide

These performances place Récy&Co among the most eco-designed sites.

Eco-design Optimizations Implemented

- Use of vanilla JavaScript instead of heavy frameworks

- Image optimization in WebP format
- Minification of CSS and JavaScript files
- Reduction of HTTP requests
- Lightweight architecture and optimized code
- Average page size kept under 100 KB (DOM size between 58 and 235 elements depending on pages)

Development Duration

The complete MVP was developed in 4 to 5 weeks, following Agile methodology with weekly sprints, demonstrating excellent time and priority management.

D. Lessons Learned

1. What Worked Well

Several technical and methodological choices proved particularly relevant and contributed to the project's success.

Three-Tier Architecture and Technology Choices

Adopting a three-tier architecture (Frontend, Backend, Database) proved to be a major structural choice. This clear separation of responsibilities greatly facilitated code maintenance and strengthened application security. With each layer having its well-defined role, modifications and bug fixes could be performed in a targeted manner without impacting other components.

The choice of vanilla JavaScript for the frontend, although less common in the framework era, proved judicious. This choice directly contributed to the excellent eco-design scores obtained (grade A on EcoIndex). The absence of a heavy framework kept the application lightweight, fast, and environmentally friendly, while demonstrating solid mastery of pure JavaScript.

For the backend, Flask proved to be the ideal choice. The Python-Flask combination with SQLAlchemy ORM enabled elegant database management by transforming raw SQL into Python objects. This object-oriented approach made the code more readable, maintainable, and facilitated CRUD operations (Create, Read, Update, Delete).

Agile Methodology and Work Organization

Implementing weekly sprints was decisive in meeting deadlines. This organization established a sustained and regular work rhythm, with clear objectives to achieve each week and end-of-sprint deliverables that structured project progress.

MoSCoW prioritization (Must have, Should have, Could have, Won't have) was a valuable tool for staying focused on essentials. It helped clearly understand what the MVP really was and avoid scattering on secondary features. This method ensured critical functionalities were developed first, allowing delivery of a functional product within the allocated time.

Using Trello as a project management tool reinforced this organization. The Kanban board provided clear visualization of task progress and kept focus on each sprint's objectives, avoiding dispersion.

Successful Eco-Design

The excellent EcoIndex scores obtained (grade A, top 5% worldwide) result from a series of coherent technical decisions: absence of heavy frontend framework, well-refactored code for better maintainability, minimal API calls in JavaScript files (only essentials), semantic HTML structure with a single `<h1>` tag per page, use of a single font without multiple downloads, and application of the DRY principle (Don't Repeat Yourself).

This experience demonstrated that it's not difficult to reconcile performance and eco-design. Even with frameworks, it's possible to adopt an eco-responsible approach, but choosing lightweight technologies greatly facilitates achieving these goals.

Effective Solo Development Management

Working alone on the entire project (frontend, backend, database, design) could have been a major challenge. However, several factors maintained high productivity: discipline provided by Trello and weekly sprints, consistency in work (regularity rather than sporadic intensity), and especially an intelligent sequential development strategy.

By first developing the backend and database to a stable version, then moving on to frontend and API calls, this methodical approach avoided many back-and-forth changes and cascading modifications. This logical progression built solid foundations before adding the user interface.

Versioning and Documentation

The 140+ commits on GitHub aren't just a quantitative indicator. Each commit represents a save point allowing rollback in case of problems and precise tracking of project evolution. This rigorous versioning practice provided valuable security during development.

Writing docstrings and comprehensive technical documentation proved beneficial in two ways: it facilitated code understanding during development and ensures another developer could take over the project quickly understanding technical choices and implemented logic.

2. Challenges Encountered and How They Were Overcome

Like any development project, Récy&Co presented its share of technical, methodological, and personal challenges. Each obstacle was an opportunity to learn and develop problem-solving strategies.

Major Technical Challenges

JWT authentication represented the first significant technical blocker. Initially, JWT tokens were stored in localStorage on the client side. However, after deepening security considerations, it became necessary to migrate to HTTP-only cookie storage, more secure against XSS (Cross-Site Scripting) attacks. This modification had a cascading impact: all facade routes (HTTP layer) had to be adapted to handle cookie sending and receiving, and the frontend JavaScript code required significant adjustments to retrieve tokens from cookies rather than localStorage.

This issue was resolved through methodical documentation research and using artificial intelligence as a learning tool. This experience strengthened understanding of security challenges in modern web applications.

Developing the drag & drop game constituted another major technical challenge. Understanding JavaScript events dragstart, dragover, drop and their coordination required significant investment. It wasn't simply about copying existing code, but truly understanding the underlying workings of the browser's Drag and Drop API to adapt it to the sorting game's specific needs.

Learning occurred through a combination of YouTube tutorials showing concrete drag & drop implementations, and exchanges with AI to clarify concepts and solve specific problems. This multi-source approach proved effective in mastering a complex feature.

Architecture Problems and Refactoring

A circular import problem occurred in the backend facade routes structure. This type of error, common in medium-sized Python projects, happens when two modules import each other, creating an infinite loop. Resolution required complete refactoring of HTTP routes organization, rethinking import structure and extracting certain functionalities into separate modules.

This experience provided better understanding of the importance of modular design and dependency management in a Python/Flask project.

Doubts and Questioning About JavaScript

JavaScript was the least mastered technology at project start. This relative weakness generated significant doubts: was it the right choice to use vanilla JavaScript rather than a more guided framework? Wasn't the learning curve too steep?

These questions were particularly present during drag & drop development, where DOM manipulation and event management required fine language understanding. In hindsight, this choice proved judicious: it forced real skill development in JavaScript, without the sometimes misleading abstractions of frameworks, and achieved eco-design objectives.

Time Management and Flexibility

Drag & drop game development took 5 full days instead of the initially planned 2-3 days. This delay pushed back the entire schedule by a week, jeopardizing deadline compliance.

The solution resided in applying Agile methodology itself: rather than sacrificing quality or abandoning essential features, an additional week was integrated into the schedule, bringing MVP development to 5 weeks instead of 4. This flexibility, inherent to Agile approach, delivered a complete, quality product without compromising initial objectives.

This experience taught the importance of regular schedule re-evaluation and accepting that some complex features require more time than expected.

Autonomous JavaScript Learning

JavaScript being the least familiar technology, its learning occurred "on the job" throughout the project. This learning-by-doing approach, though sometimes uncomfortable, proved very effective. The combination of YouTube video tutorials (to see concrete implementations), official documentation (to understand concepts), and AI assistance (to solve specific problems) created a complete learning environment.

Drag & drop development, though difficult, was the catalyst for real skill advancement in JavaScript, transforming an initial weakness into acquired competence.

Personal Challenges: Solo Work and Motivation

Working alone on the entire project presented significant psychological challenges, particularly in moments of doubt. The absence of a team to validate technical choices, share difficulties, or simply reassure during difficult moments was sometimes burdensome. Personal questioning ("Am I capable of completing this project?", "Are my technical choices relevant?") was all the more difficult to manage in the absence of immediate feedback.

Maintaining motivation was possible thanks to support from C26 cohort colleagues at Holberton School. Exchanges, mutual encouragement, and sense of belonging to a group were decisive. Remembering that other students were succeeding in their projects maintained confidence in one's own abilities and perseverance through difficult times.

This experience highlighted the importance of professional networks and mutual aid, even in individual projects, and strengthened stress management and self-motivation skills.

3. Technical Skills Developed

The Récy&Co project was an opportunity to acquire and consolidate a wide range of technical skills covering the entire web development stack.

Backend Skills: Flask and Server Architecture

Working with Flask mastered several essential backend development aspects. Blueprints were used to organize the application into logical modules, facilitating code maintenance and scalability. Session and cookie management was deepened, particularly for implementing persistent authentication.

SQLAlchemy ORM represented a major advancement in understanding data persistence. Mastering relationships between tables (one-to-many, many-to-many) enabled designing a coherent and performant data model. ORM queries were deepened, enabling effective data manipulation without writing raw SQL, while understanding performance implications of each query.

Web application security constituted a major learning axis. Implementing authentication with JWT (JSON Web Tokens) provided understanding of stateless session management mechanisms. Using HTTP-only cookies for token storage strengthened understanding of XSS attack vectors and security best practices. Password hashing with bcrypt completed this security training, ensuring users' sensitive data is never stored in plain text.

Frontend Skills: JavaScript, HTML, and CSS

JavaScript was the technology requiring the most learning. DOM (Document Object Model) manipulation became a mastered skill, enabling creation of dynamic and reactive interfaces. Event management (click, submit, input, and especially drag & drop specific events) was deepened.

The native JavaScript drag & drop system represented a particular technical challenge. Mastering dragstart, dragover, drop, dragenter, and dragleave events, as well as managing the dataTransfer object, created a smooth and intuitive user experience for the sorting game.

An important discovery was that Jinja2, initially perceived as a frontend tool, is actually a backend template engine that generates HTML server-side before sending it to the client. This understanding clarified the separation between server-side rendering and client-side interactions.

In HTML, learning semantic HTML (appropriate use of `<header>`, `<nav>`, `<main>`, `<section>`, `<article>` tags) improved accessibility and SEO. Responsive design was mastered, ensuring optimal user experience on all devices (desktop, tablet, mobile). Optimization (minification, compression, choosing lightweight image formats like WebP) was integrated from conception.

Database Skills: MySQL

Working with MySQL consolidated understanding of relational database design. Relationships between tables (foreign keys, referential integrity constraints) were mastered, enabling correct modeling of complex relationships like those between users, scores, games, and badges.

SQL queries (though abstracted by SQLAlchemy) were deepened, particularly for understanding joins, aggregations, and performance optimization. This dual SQL/ORM competence ensures flexibility in data persistence approach.

DevOps Skills and Tooling

Git usage exceeded basic versioning. Understanding why to version (traceability, ability to roll back, collaboration) and how to version well (clear commit messages, atomic commits) transformed Git into a true project management tool. Using branches was learned, enabling work on isolated features before merging into the main branch.

Automated testing with Pytest represented a significant advancement in development rigor. Writing over 20 unit tests validated proper backend functionality and quickly identified regressions during modifications. This "test before continuing" approach strengthened code quality and reliability.

Postman was used to test HTTP routes before even developing the frontend, validating that endpoints return expected data with correct status codes. This practice considerably accelerated debugging by isolating backend problems from frontend problems.

Cross-Cutting Skills: Debugging and Optimization

Debugging became a skill in itself. Chrome DevTools was used intensively to inspect DOM, analyze network requests, consult cookies and local storage, and identify JavaScript errors. Strategic use of console.log() to trace code execution resolved many logical bugs.

Performance optimization was approached systematically through Lighthouse, which provides precise scores on performance, accessibility, best practices, and SEO. Each Lighthouse recommendation was analyzed and, when relevant, applied.

Using GreenIT-Analysis introduced a new dimension: eco-design. This tool measured the site's environmental impact (number of requests, page sizes, DOM size, estimated energy consumption) and enabled optimization decisions guided by these metrics. Obtaining A grades on all pages validates this approach's effectiveness.

E. Retrospective and Perspectives

What I Would Do Differently

In hindsight, if I were to restart this project, I wouldn't fundamentally change my approach or technical choices, which proved relevant and consistent with eco-design and performance objectives. The three-tier architecture, choice of Flask and vanilla JavaScript, as well as Agile methodology all proved themselves.

However, I would pay particular attention to drag & drop game planning. This feature, more complex than expected, required 5 days instead of the initially estimated 2-3. Better anticipation of its technical complexity, perhaps with a quick prototype at project start, would have allowed earlier schedule adjustment and avoided the one-week delay on the entire project.

In terms of development time allocation, I would devote more time to backend and less to frontend. Backend constitutes the application's core and its solidity conditions the entire quality. While frontend is important for user experience, investing earlier in exhaustive backend tests and robust architecture would have facilitated subsequent frontend integration.

Future Project Developments

Récy&Co, in its current MVP version, lays solid foundations for numerous developments. Several features identified during the planning phase were deliberately postponed and constitute a clear roadmap for next versions.

The leaderboard (player ranking) tops priorities. This feature would strengthen the game's competitive and motivating aspect, encouraging users to return regularly to improve their score and compare with other players.

The bonus shop with a virtual currency system would reward user engagement and add a collection dimension to the game. Users could unlock cosmetic elements or game advantages, creating a stronger engagement loop.

Complete implementation of the badge system would enrich gamification. Currently present in the database, this system still needs development of attribution logic and visual display of unlocked badges.

Production deployment is an essential step for Récy&Co to truly fulfill its educational mission. Going live would allow French families to actually access the application and benefit from its educational features.

Game improvement also constitutes a development axis: adding new difficulty levels, diversifying waste to sort, introducing additional game mechanics (time trial mode, daily challenges, etc.).

Finally, a backend and frontend code refactoring phase would enable even more rigorous application of the DRY principle (Don't Repeat Yourself). Although current code is functional and maintainable, certain portions could be optimized to eliminate redundancies and improve readability.

What This Project Brings Me Moving Forward

This project represents much more than a simple academic exercise. It constitutes a complete project design experience from A to Z: from initial idea to functional MVP delivery, through all phases of planning, development, testing, and documentation. This global vision of the software development cycle is precisely what's expected of an experienced developer.

For my apprenticeship search toward obtaining the RNCP 6 title (Application Designer Developer), Récy&Co constitutes a major asset. I now have a concrete, deployable, and technically solid project to present to recruiters. This project demonstrates not only technical skills (full-stack, architecture, security, testing), but also methodological skills (Agile, solo project management, documentation) and sensitivity to current issues like eco-design.

Beyond acquired technical skills, this project brought me something fundamental: confidence in my abilities. Moments of doubt were numerous, particularly during drag & drop development or facing circular import problems. But each overcome obstacle reinforced the certainty that I'm competent, that I can learn what I don't yet know, and that I can successfully complete complex projects autonomously.

This realization is perhaps the most valuable learning: I know I can do it. This confidence will be a solid foundation for approaching future challenges, whether in apprenticeship or in my future career as a developer.

F. Conclusion

The Récy&Co project, an educational web application dedicated to learning waste sorting, is part of a strong pedagogical and ecological approach. Initially designed for the Communauté de Communes d'Évron and extended to all of France, this project aimed to make recycling learning fun, accessible, and environmentally friendly. This ambition was fully realized.

The delivered MVP demonstrates complete mastery of full-stack web development: solid three-tier architecture with Flask and MySQL, responsive frontend in vanilla JavaScript, secure authentication system with JWT, interactive drag & drop game, and eco-designed design validated by exceptional EcoIndex scores (grade A, top 5% worldwide). The six functional pages, 140+ commits, 20+ automated tests, and 100% responsive design testify to rigorous and professional work.

Beyond technical results, this project was an intense human and formative experience. Working alone on the entire stack, managing moments of doubt, overcoming technical blockers, learning JavaScript in depth, mastering web application security: each challenge was a growth opportunity. Support from Holberton School's C26 cohort was precious in difficult moments, recalling the importance of community even in individual projects.

This project constitutes a success both academically and personally. It validates skills required for obtaining the RNCP 5 Web and Mobile Web Developer title, while laying solid foundations to pursue the RNCP 6 Application Designer Developer title through apprenticeship. Récy&Co is now a presentable project to recruiters, demonstrating complete technical skills, eco-design sensitivity, and ability to lead a project from start to finish.

But more importantly, this project reinforced an essential certainty: I am competent, I can learn, I can create, and I can succeed. This confidence in my abilities is the foundation on which to build my future developer career.

Récy&Co is not an end, but a beginning. Future developments (leaderboard, bonus shop, production deployment, game improvements) offer exciting prospects. And beyond this specific project, the acquired experience opens the way to even more ambitious projects.

Waste sorting may start with a simple bin, but with Récy the raccoon and a bit of code, it becomes a fun and educational adventure. Sorting is fun!