MODULE: CS2TP Team Project

GROUP: Group 38

TUTORS: XXXXXXXX, XXXXXXXX

TITLE: Final Evaluation Report on Crep Culture

XXXXXXXX (xxxxxx404), XXXXXXXX (xxxxxx311), XXXXXXXX (xxxxxx908), George Hotten (xxxxxx352), XXXXXXXX (xxxxxx780), XXXXXXXX (xxxxxx855), XXXXXXXX (xxxxxx189), XXXXXXXX (xxxxxx465)

24th March 2025

WORD COUNT: 3185 words

Final Evaluation Report

Table of Contents:

Section 1 - Project Information:

Project Title: Crep Culture

Project Logo:

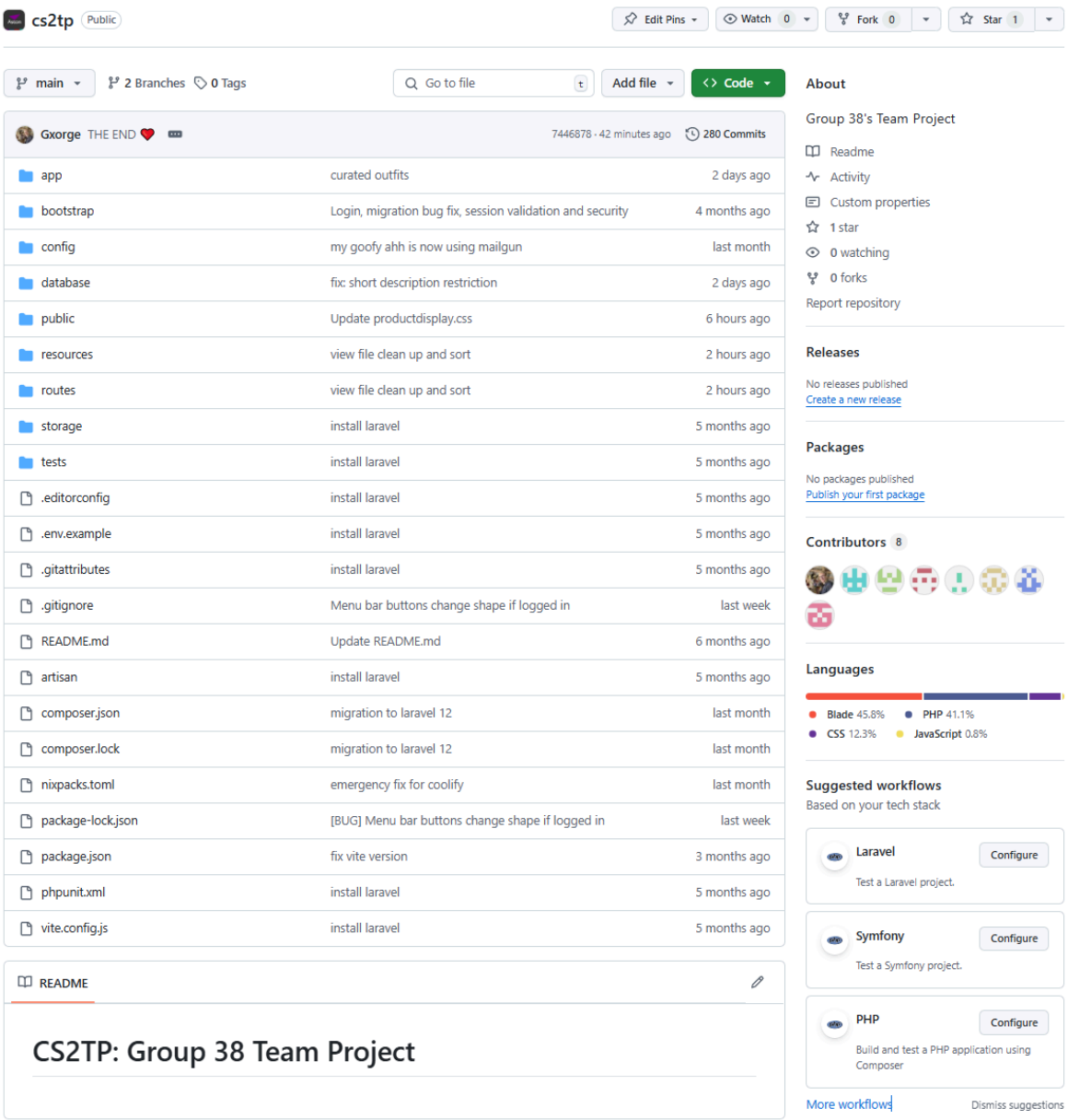
Project Summary:

Crep Culture is an online retailer that focuses on selling highly desired sneakers and trainers from some of the most well-known brands in the sneaker world such as Nike, Adidas, Jordan, and Yeezy. Targeted towards collectors, sneakerheads, and the average consumer, Crep Culture offers authenticated and exclusive trainers, giving customers access to highly sought-after styles and limited edition drops. With a strong commitment to quality, style, and dependability, Crep Culture aspires to be the go-to online hub for both sneakerheads and casual shoppers, transforming the sneaker market with a seamless online shopping experience and exceptional customer support. The website uses HTML and CSS for the frontend and PHP and Laravel for the backend while using MySQL for the database.

Section 2 - Project Artefacts:

- Link to deployed website: https://cs2tp.hotten.cloud/
- Link to GitHub repository: https://github.com/georgeataston/cs2tp
- Link to Trello board: removed for privacy

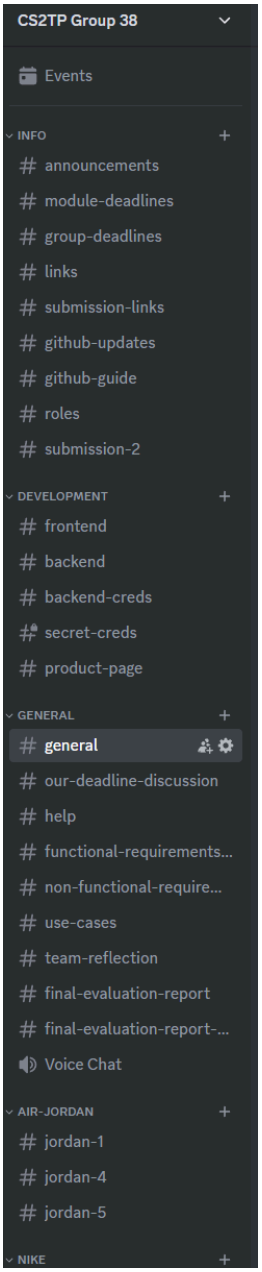
GitHub Repository:



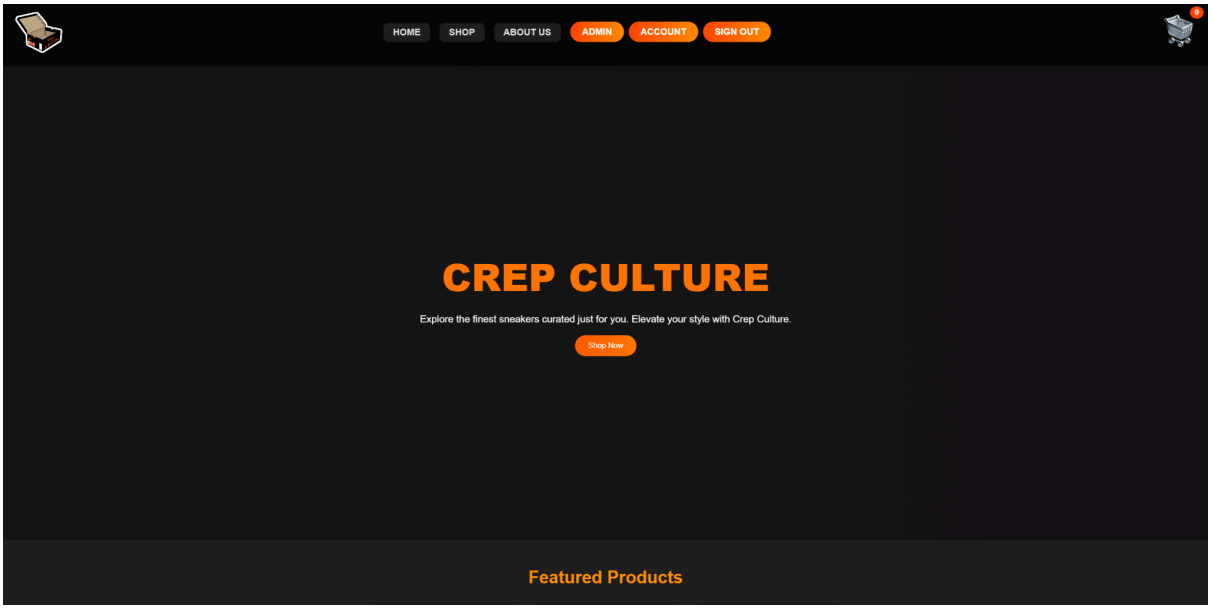
Trello Board:

Removed for privacy

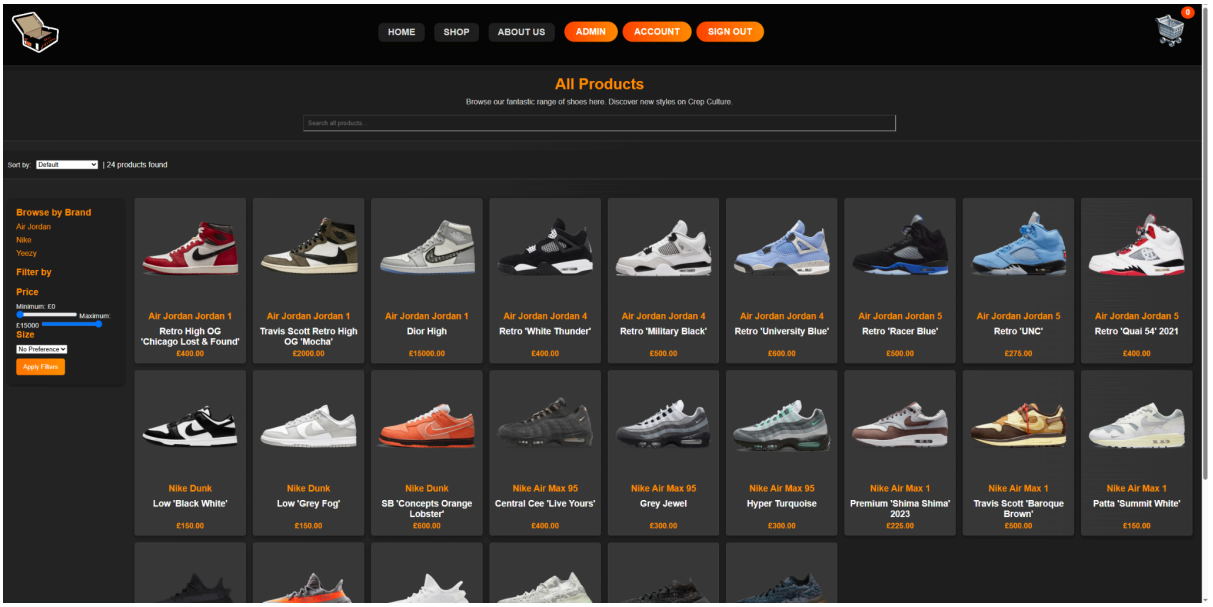
Discord:



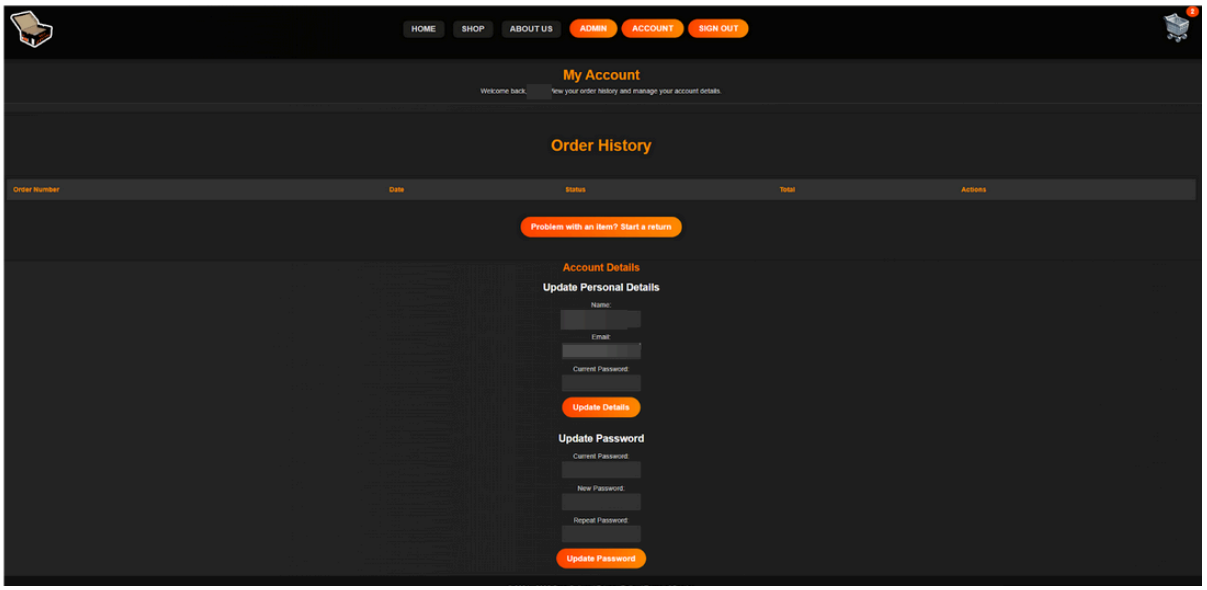
Home Page:



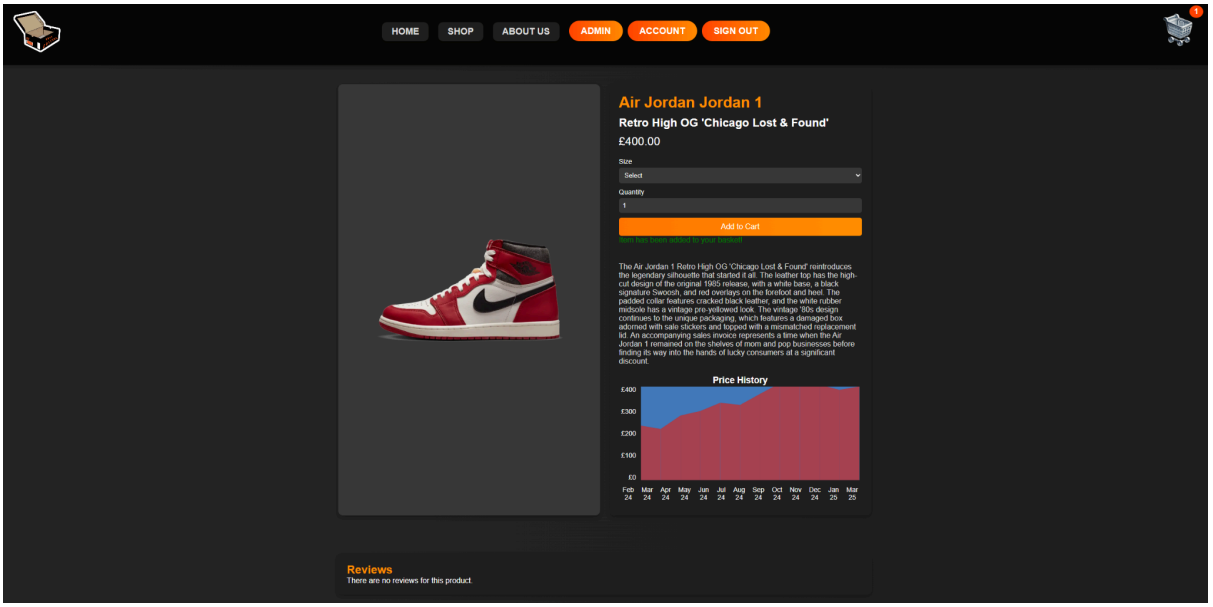
Shop Page:



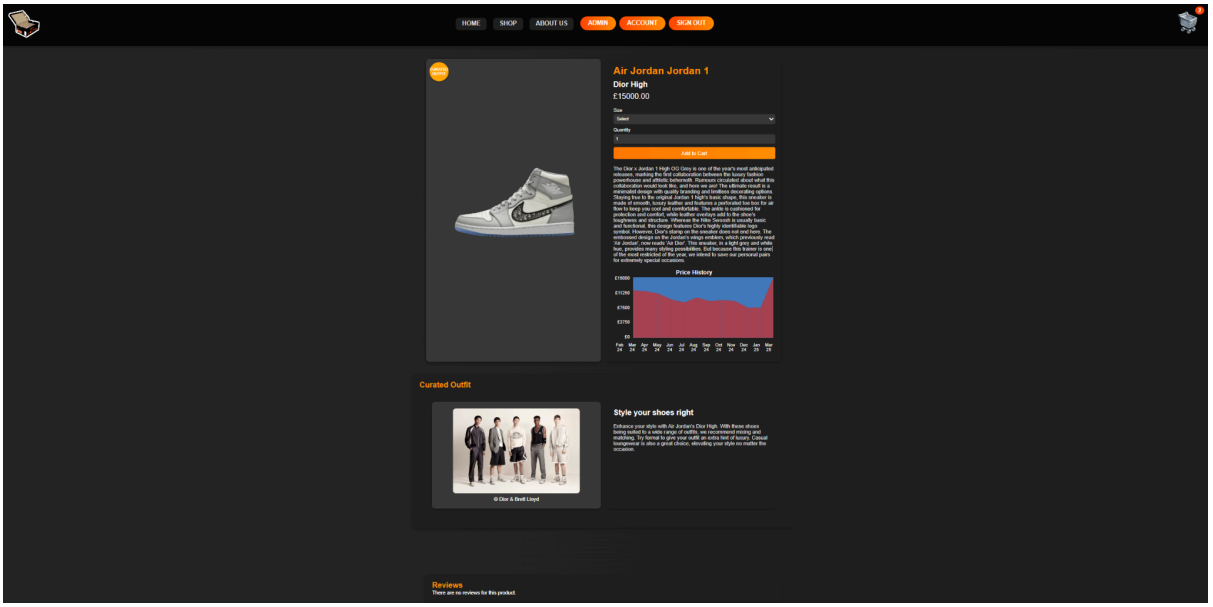
Account Page:



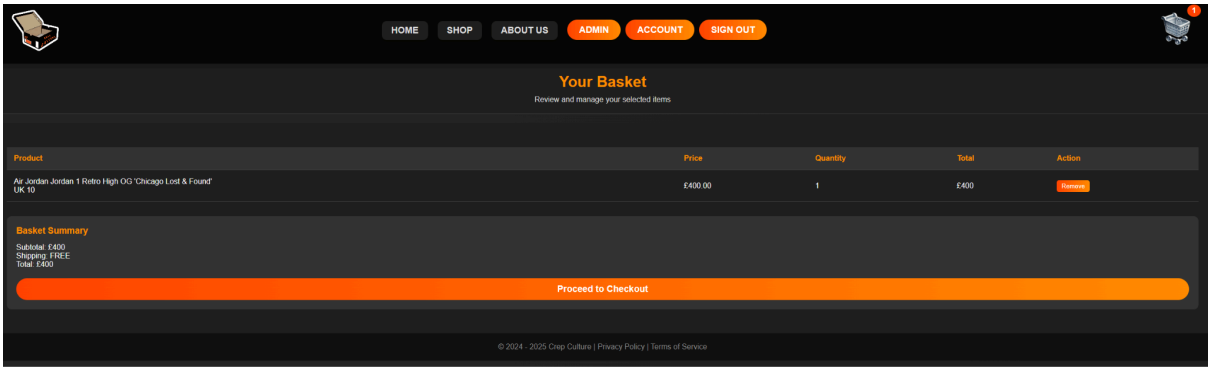
Product Listing with Price Graph Unique Feature:



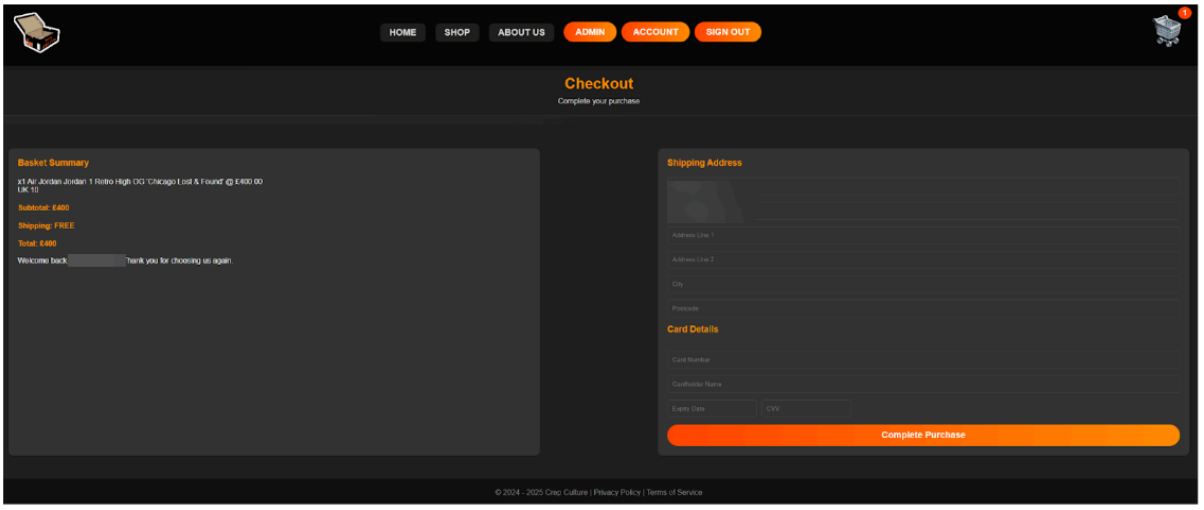
Product Listing with Curated Outfit Unique Feature:



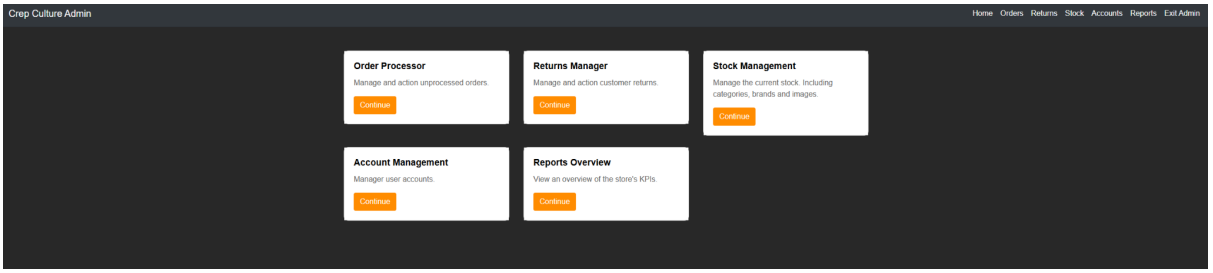
Basket Page:



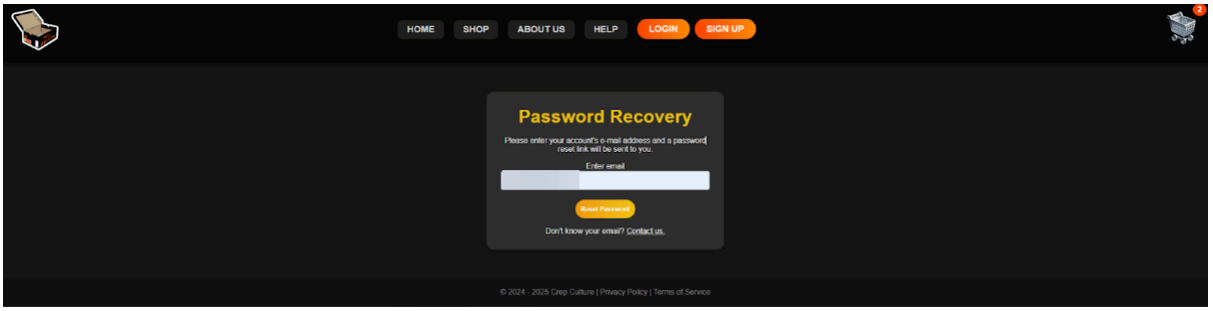
Checkout Page:

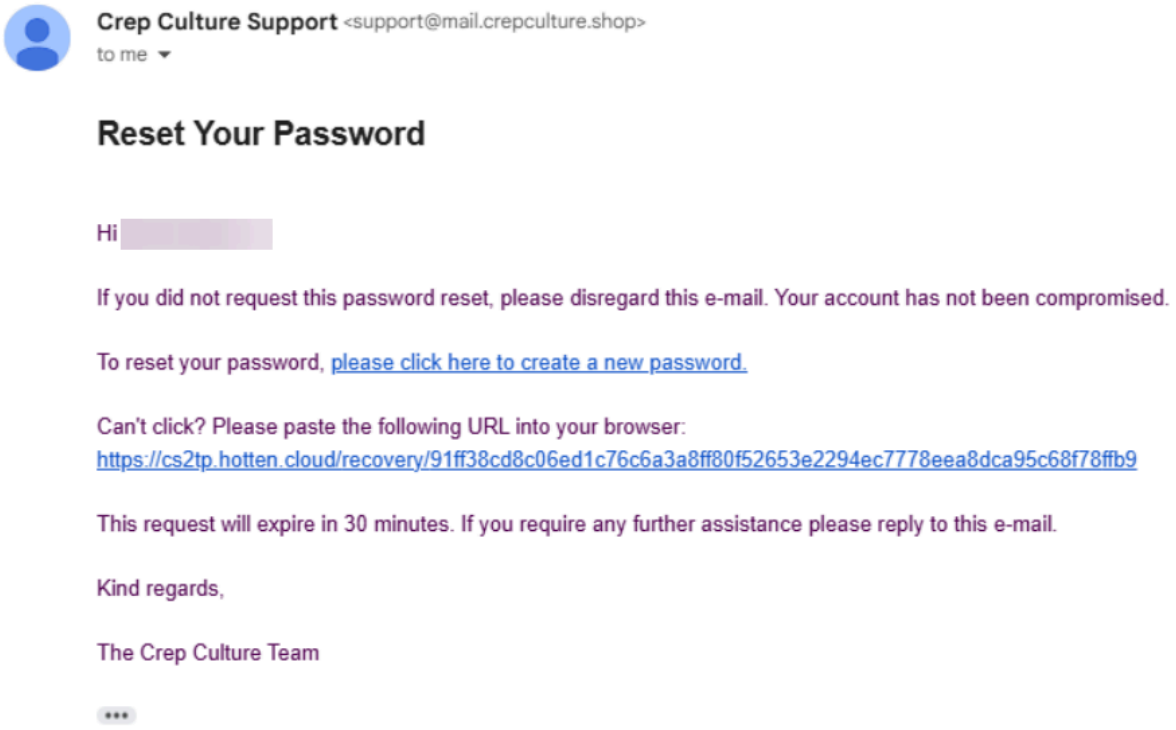


Admin Page:



Password Recovery:





Crep Culture Support <support@mail.crepculture.shop>
to me

Reset Your Password

Hi

If you did not request this password reset, please disregard this e-mail. Your account has not been compromised.

To reset your password, please click here to create a new password.

Can't click? Please paste the following URL into your browser:
https://cs2tp.hotten.cloud/recovery/91ff38cd8c06ed1c76c6a3a8ff80f52653e2294ec7778eea8dca95c68f78ffb9

This request will expire in 30 minutes. If you require any further assistance please reply to this e-mail.

Kind regards,

The Crep Culture Team

Section 3 - Software Quality Assurance:

Maintainability: Maintainability of a piece of software is the ease of modification or improvement without starting from scratch. For a website like Crep Culture, this means being able to quickly add new features, fix errors, or update items without

compromising the rest of the site. We gave simplicity, structure, and documentation top priority in order to simplify management of our website. For instance, we broke up our code into smaller pieces that each served a specialised purpose instead of creating one large block to manage everything. This makes it simple for anyone to grasp what the code is doing and to edit or replace a particular part without compromising the whole site. We also choose Laravel as it offers us a clear framework to apply. This let us divide up various tasks on the website - such as page display, order processing, user account management - into distinct parts of the code. For instance, we made sure all of the code for out-of-stock warnings was kept apart from the programming handling the user interface when we developed the stock inventory logic. Later, when we wanted to modify the way the out-of- stock warnings were shown, we saved time and avoided mistakes by not changing the remainder of the code. We also followed changes to our code using GitHub. This let us go back and fix mistakes anytime anything stopped working. It also let team members work simultaneously on various parts of the site without interfering with one another.

Usability: Usability refers to how easy and convenient it is for users to learn and understand to use software. It's also beneficial for measuring how well the software meets the needs and expectations of users. For Crep Culture, usability is a critical quality attribute as it directly impacts user satisfaction and the overall success of the platform. Usability is important as users expect a seamless and enjoyable shopping experience across all devices which is why we have integrated mobile support. Crep Culture's mobile support allows the platform to be effectively used on smartphones and features a responsive design that adapts to the different screen sizes ensuring that users can easily navigate and browse products with minimal issues. Another component of usability that Crep Culture focuses on is navigation and accessibility; the use of a well-optimised search function with filters and sorting options. This was added to increase ease of use for the end user as they will not struggle with finding specific products.

Efficiency: Efficiency refers to how efficient a website is at performing a given task without wasting time or using system resources. For us at Crep Culture, pages are expected to load quickly, the user can browse smoothly, and the site will remain reliable even under immense traffic load. This was done in order to provide clean and lightweight code that does not burden the system. For example, on the website where we showed products and processed searches, the website would only load the information that is required - nothing more. It helps to mitigate the load on the server and increases response speed for the user. We also worked on minimising needless calls to the database and to use efficient queries while the website communicates with the database. In other words, instead of searching the entire stock database when a user added an item on his basket, we shortened the process only to the particular product. This helps the site to run more smoothly by reducing processing time. With this built in mind, we were constantly testing performance with load simulations so that it would perform as well as possible for as many users as

possible. We used these to find out and fix slowdowns before they came to be real problems. We focused on how memory, processing capability, server space were being utilised to ensure Crep Culture is quick, dependable, and ready to scale as the site expands.

Portability: Portability is the ability of a software to run on different platforms or environments without requiring significant modifications. For Crep Culture, portability is a key attribute as our website will not be just limited to one certain type of device, operating system, or browser. We want our website to function seamlessly on different operating systems such as Windows, Android, Linux, macOS, etc. Our tests included running the website on multiple devices and not just limiting it to a certain one device. The same is also true with browsers to check compatibility, it was run on Google Chrome, Mozilla Firefox, Safari, etc. Issues were faced with browser compatibility where we addressed menu inconsistencies of buttons appearing larger on different browsers. To compliment the portability of Crep Culture we included responsive designs to ensure layout functionality on different screen sizes. Using CSS queries we provided an optimal viewing experience from desktops to mobile support. Font sizes were reduced and padding was adjusted to ensure a consistent frame that was dynamic to different screens.

Section 4 - Software Testing and Validation:

When testing our website, we decided to use integration, system and usability testing. To ensure we had the most amount of exposure to things going wrong, we decided on manual testing as opposed to using unit testing. This meant that we were able to validate the true customer experience and explore the different actions end users might take, as opposed to scripted events.

Our workflow involved all code being submitted as a pull request on GitHub before it is merged into the development or production branches. When a PR is created, an automatic system builds the website and publishes to a URL (for example, dev-pr52-cs2tp.hotten.cloud). This allowed us to fully explore and test new features and how they interact with older features. Our main focus would be around areas of the website where code and logic was edited. This ensures that bugs are quickly discovered and fixed, before they become a bigger problem rooted deeper into the codebase.

This workflow meant that we were constantly testing the integration of code being produced, with Git assisting with any conflicts. Once these conflicts were resolved, the server rebuilds the website and testing can be restarted.

For our system testing, our checkout and basket system was the most heavily tested section. This was due to the large amount of data requested from the backend, what

data needed to be stored to retain a user's basket and how data is validated before and after checkout. To effectively test this, we designed the following process:

- Find an item on the shop page that has out of stock items
- By editing the HTML, enable the size's button and attempt to add to your basket
- The server should refuse. Add an in-stock size to your basket.
- Log in and out of an account to ensure basket retention.
- Proceed to checkout, ensuring the correct prices and item information is displayed.
- Test the validation of the form's input fields, such as card numbers, expiry and CVVs.
- Submit the order.

This test system helped us discover numerous bugs in the system, as the stock system was a feature that was revised alot during development. This also ensured when data logic was changed, we could validate that the core functions of our site were still functional.

To test usability, we asked students from different courses to browse the website. We asked them to envision they were recommended our site and are currently after a new pair of shoes. We then let the students browse the website and attempt to buy an item, manage their account and start returns. This allowed us to refine our user experience by changing the interface to ensure that some functions were more intuitive to find and use. The biggest problem students had was how to start a return. Based on this feedback, we created a dedicated help page with FAQs and added a button directly next to a user's orders to start a return.

Section 5 - UI Design Considerations:

The UI design for Crep Culture was thoughtfully designed and put together to ensure that users are fully engaged and find it easy to use. The main objective we sought out was to develop an aesthetically pleasing and user-friendly platform that would both appeal to trainer enthusiasts and more casual buyers. The main design feature we considered was to have a clean and modern interface that reflects the style and energy of sneaker culture along with having visually appealing buttons and animations such as when hovering over a shoe to give a more elevated feel for the users. We intended for the homepage to showcase the latest sneaker information to gain the user's attention and highlight more exclusive products. The menu bar is clear and visible, at the top of the page and is very easy to use with clear categories such as “Home”, “Shop”, and “Contact”, making it easy for users to find what they are looking for. We also ensured the website was fully responsive and could adapt to different screen sizes for various devices such as smartphones, computers, tablets, etc. For devices like phones, we have a mobile-friendly navigation menu so smaller screens don't affect the user's shopping experience while also ensuring links and

buttons are large enough for easy navigation by touch.

The shop page has filters through which users can select their preferred option by brand, price, size, and availability. This allows users to easily select the exact sneaker they are looking for ideally. The search bar is in the center of the shop page where the user can directly search for a specific product or brand. Every product page has the information necessary on the shoe and reviews so customers can acknowledge if it's worth purchasing. This page will also have a price history graph which will show the customer how the price for certain products has been for the past year. The shopping cart is accessible on all of the pages by clicking the cart icon in the top right corner allowing users to easily view and manage selected products before proceeding to the checkout.

Section 6 - Security Considerations:

To ensure security against malicious input from users, we make use of Laravel's 'Eloquent'. Eloquent provides automatic escaping to prevent SQL injections. We also further utilized Laravel's input validation before passing any data to the database, ensuring it is always the correct type, in the correct format and escaped.

To ensure the security against our user's data, our database is isolated from the internet. It is only accessible on the local server via the website application. This drastically reduces the risk of a database breach. To further protect data, all passwords are hashed with salts, using 'bcrypt'.

Crep Culture uses server managed sessions for storing information about a logged in user, such as their user ID and if they're an administrator. These values cannot be changed on the user's end, preventing any fraud. The same system is used for the basket function, preventing users from tampering with their basket items and their prices.

Section 7 - Project Management:

We decided to use a flexible and cooperative approach to project management throughout the whole project. We were able to fulfill deadlines, maintain organisation, and adjust to unforeseen circumstances thanks to this approach. We didn't follow a strict methodology but instead followed a practical, task-orientated approach that worked the best for our team dynamic.

We first set up a Trello board to break the project into manageable tasks, assign responsibilities, and track progress. Each team member had their own cards which contained the features they needed to work on and we used columns like "Assigned Tasks", "In-Progress Tasks" and "Completed Tasks" to stay on top of what needed attention. This helped us keep the workload balanced and gave everyone visibility on what others were working on.

GitHub played an important role in our collaboration. We used it to manage our entire code, track contributions, and handle version control. When someone finished a task, such as creating a new page, adding a feature, or updating a database, they would commit their changes and push them to the repository. This reduced conflict and made it easier to review and resolve issues collaboratively. Branching also allowed us to experiment without affecting the main version of the project.

We mainly communicated via Discord, where we set up separate channels for announcements, deadline tracking, discussions, etc. This made discussions much easier and helped us overcome obstacles. We also had a few voice call meetings to plan major features or review our progress as a group.

The platform is built using HTML for its page structure, CSS for a responsive and attractive design, and PHP for backend operations and server-side processing. Leveraging the Laravel framework accelerates development with built-in security features and easier database management through MySQL. This tech stack ensures a secure, scalable, and user-friendly e-commerce experience.

Section 8 - Teamwork Aspects:

Teamwork Ethics:

From the very beginning of the project, our team placed a strong emphasis on respect, communication, and shared responsibility. We made sure that every individual had a say in decisions and encouraged open discussion when there were problems. Although every member was expected to do their part in the work, we also made it a point to support one another if someone needed help or fell behind. We used Discord to keep in contact among us, whether it was answering questions, giving feedback, or working on a solution together. We did not blame one another if something went wrong but focused on finding solutions and what we could learn from the problem. Every individual was flexible, taking on extra workload as needed to keep the project on track. Our teamwork was built upon respect for one another, dependability, and a shared goal to create a successful and quality product.

Statements of Contribution:

XXXXXXXX (xxxxxx780) - Frontend + UI Design

I have:

- Created some of the design mockups.
- Created the logo for the website.
- Worked on the following for the frontend; shop page, product display, login page.
- Handled some of the CSS across the site as a whole along with my peers on the frontend team.
- Fixed bugs that caused issues with the CSS and logo and the navigation bar.

George Hotten (xxxxxx352) - Backend & Frontend

I have:

- Provided hosting, automatic previews and building of the website.
- Managed the group's GitHub, by ensuring we follow good git practices.
- Created and managed the group's Trello. Splitting the report specification into manageable chunks.
- Worked on most functionality within the admin pages, including stock, orders, reports and returns.
- Multiple frontend and backend bug fixes.
- Our unique feature, curated outfits.
- Ability for users to update their details, view past orders and return items.
- Review system.
- Designed the authentication system for users and administrators Includes a password reset system utilizing SMTP.
- All logic for the shop pages, including out of stock, filtering, searching, sorting, dynamic sizing and reviews.
- Revamped the stock system for dynamic sizing and stock levels. Includes a better basket system.
- Designed multiple databases, such as accounts, stocks, reviews, and more.
- Supported with the backend link with the frontend on multiple features, such as basket, checkout, shop page and product display.

XXXXXXXX (xxxxxx404) - Backend & Frontend

I have:

- Created the database schemas and order validation files on the backend to allow users to checkout multiple items within an order and get the correct information about a product with foreign keys such as product id and user id.
- Created 3 pages on the front end including the help page, policy page, and terms and service page.
- Helped my team with debugging several frontend issues to comply with project requirements while maintaining back-end logic.

XXXXXXXX (xxxxxx189) - Frontend

I have:

- Created the frontend for the user basket
- Created the frontend for the inventory management
- Created the low/out of stock banners on the shop page
- Helped to implement bug fixes where appropriate

XXXXXXXX (xxxxxx311) - Frontend + UI

I have:

- Created the homepage for the website, the about us page, the contact us page and the sign-up page.
- Designed the colour scheme for the website and provided site-wide HTML and CSS to be used for instance like for buttons, the hover animations, etc.
- Designed the basket logo along with an accompanying counter to display the items in the basket. All the pages are also now linked with each other to help the customer to navigate the website easily.
- Added to the product display page the price history graph which for each shoe will show its price history over the past year which is one of our unique features.

XXXXXXXX (xxxxxx855) - Backend

I have:

- Created the backend logic for order returns and stock management, ensuring accurate stock status.
- Created database schemas to support stock management and product structure, using foreign keys like product ID and user ID to link order data.
- Populated and structured the product page with content including brands, categories, and product listings to ensure accurate display and functionality.

XXXXXXXX (xxxxxx908) - Frontend

I have:

- Designed the frontend pages for the returns system, the user basket, and the checkout page.
- Created the front end for the user account page allowing users to view previous orders and update account details.

XXXXXXXX (xxxxxx465) - Frontend

I have:

- Contributed to the development of the website by designing the frontend for the admin order return page.
- Fixing bugs on the shoe page
- Created the front-end design for the admin pages.

Section 9 - Project Limitations and Recommendations for Future Work:

Limitations:

- The website may not be fully optimised for some old and low-end phones which could lead to potential performance issues.
- There might also be limitations in handling high traffic volumes as our platform grows bigger.
- Our site does not include AI-driven recommendations.

- Site does not include two-step verification to increase security when logging in.
- Payment card accepts expiry dates less than 2025 or allows random inputs.
- Does not accept valid shipping addresses allowing users to input randomly, which can lead to failed or delayed deliveries.

Future work:

- Introduce Augmented Reality (AR) which could allow users to use their phone cameras to try and see how the shoes would look.
- Create a community forum for enthusiasts where they can interact, share collections, and give more in-depth reviews.
- Introduce a loyalty program to give back to users for supporting us.
- Have limited edition events/sneakers.
- Incorporate a chatbox to help the page as the site grows and AI recommendations.

Conclusion:

In conclusion, a high-quality sneaker retailer website was created that met the functional and non-functional requirements set out in the project brief. The team demonstrated key project management and software engineering competencies, factoring in software quality assurances to ensure a complete and user-friendly full-stack software application. Challenges such as CSS styling inconsistencies were resolved through debugging and effective collaboration. While the current version is stable and complete, future developments could be optimised further for the target market of sneaker enthusiasts and also include the latest security technologies.