

Project Name: artStock Documentation

# Introduction

## I.1. Overview

artStock is a website used as a platform where artists and art enthusiasts can connect with each other, share ideas, receive feedbacks, collaborate on projects, and build a community around their shared interest in art. Overall, the goal would be to create a thriving online ecosystem for artists and art lovers to come together and support each other.

## II.2. Technical Stack

1. Framework: Laravel 10
2. Database: MySQL
3. Front-end: HTML, CSS, JavaScript, Bootstrap 5
4. Back-end: PHP 8.2.1
5. Server: Apache
6. Version Control: Git, GitHub
7. Development Environment: IntelliJ, Laragon

# Requirements Analysis

## II.1. Non-functional requirements

1. Performance: The website should load quickly, respond quickly to user input.
2. The website should be secure against common web-based attacks such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
3. Usability: The website should be easy to use and navigate, with clear and consistent design elements and user interface components.
4. Maintainability: The website should be designed and built in such a way that it is easy to maintain and update over time, with clean and well-organized code and documentation.

## II.2. Functional requirements

### III.2.1. User group

* Super admin:
* Sign up.
* Sign in.
* Manage user accounts.
* Manage data and content.
* Purchase management.
* Perform system maintenance and updates.
* Monitor system performance.
* Add to cart.
* Update cart.
* Make purchases
* View users’ feedback.
* Admin:
* Sign up.
* Sign in.
* Manage user accounts.
* Manage data and content.
* Purchase management.
* Perform system maintenance and updates.
* Monitor system performance.
* Add to cart.
* Update cart.
* View users’ feedback.
* Make purchases
* User:
* Sign in.
* Create and edit content.
* View, comment, interact with content and users.
* Search.
* Add to cart.
* Update cart.
* Manage account information.
* Make purchases.
* Guest:
* Sign up.
* Sign in.
* View content.
* Search.

### III.2.2. Functional Analysis

▶ Sign up

|  |  |
| --- | --- |
| **Privileged users** | Super Admin, Admin, Guest |
| **Description** | Create an account |
| **Activation** | Click on “Sign up” button on header. |
| **Input** | * First name * Last name * Username * Date of birth * Email * Gender * Phone number (optional) * Location (optional) * Avatar (optional) * Cover (optional) * Payment method (optional) |
| **Processing flow** | 1. Redirect to sign-up page. 2. Retrieve information from form. Perform validation. 3. If validation fails, show error messages. 4. If validation succeeds, redirect to sign-in page. |
| **Output** | Log recorded.  Redirect to sign-in page and display success notification. |
| **Note** | Only Super Admin can assign roles on sign-up process. |

▶ Sign in

|  |  |
| --- | --- |
| **Privileged users** | Super Admin, Admin, User, Guest |
| **Description** | Sign in |
| **Activation** | Click on “Sign in” button on header. |
| **Input** | * Username or email * Password |
| **Processing flow** | 1. Redirect to sign-in page. 2. Retrieve information from form. Perform validation. 3. If validation fails, show error messages. 4. If validation succeeds: 5. Create a login session, save the username, and account ID to the session. 6. Redirect to user’s page. |
| **Output** | Log recorded.  Redirect to user’s page. |
| **Note** | Super Admin and Admin must log in by another log-in page. |

▶ Manage user accounts

|  |  |
| --- | --- |
| **Privileged users** | Super Admin |
| **Description** | Delete and ban users. Modify user’s role. |
| **Activation** | Click on “Manage” button on user’s record. |
| **Input** |  |
| **Processing flow** | 1. Redirect to user management page. 2. On delete or ban user, click on respective button. 3. Finish re-confirmation. |
| **Output** | Log recorded.  Redirect to super admin’s page. Show success notification. |
| **Note** | Implementing IP ban on banning user. |

▶ Manage data and content

|  |  |
| --- | --- |
| **Privileged users** | Super Admin, Admin |
| **Description** | Delete content |
| **Activation** | Click on “Delete” button on content’s record. |
| **Input** |  |
| **Processing flow** | 1. Redirect to content’s page 2. Click on “Delete” button 3. Finish re-confirmation |
| **Output** | Log recorded. Redirect to the management page of deleted content’s user. |
| **Note** |  |

▶ Purchase management

|  |  |
| --- | --- |
| **Privileged users** | Super Admin, Admin |
| **Description** | Approve purchase, cancel purchase |
| **Activation** | Click on respective button on purchase’s record |
| **Input** |  |
| **Processing flow** | 1. Redirect to purchase’s page 2. Click on respective button 3. Finish re-confirmation |
| **Output** | Log recorded.  Redirect to the management page of deleted content’s user. |
| **Note** |  |

▶ Perform system maintenance and updates

|  |  |
| --- | --- |
| **Privileged users** | Super Admin, Admin |
| **Description** | Implement updates on UI, UX, back-end, etc. |
| **Activation** |  |
| **Input** |  |
| **Processing flow** |  |
| **Output** |  |
| **Note** |  |

▶ Monitor system performance

|  |  |
| --- | --- |
| **Privileged users** | Super Admin, Admin |
| **Description** | Keeping track of:   * Server usage * Response times * Error rates * Traffic patterns |
| **Activation** | Click on “Dashboard” button |
| **Input** |  |
| **Processing flow** | * Redirect to dashboard |
| **Output** |  |
| **Note** |  |

▶ View users’ feedback

|  |  |
| --- | --- |
| **Privileged users** | Super Admin, Admin |
| **Description** | View users’ feedback. Give feedback. |
| **Activation** | Click on “Response” button |
| **Input** |  |
| **Processing flow** | 1. Redirect to feedback’s page. 2. Click on “Feedback” button. Give feedback. |
| **Output** | Log recorded.  Redirect to feedback management page. |
| **Note** |  |

▶ Create and edit content

|  |  |
| --- | --- |
| **Privileged users** | User |
| **Description** | Create, edit, delete, hide content. |
| **Activation** | On creating: Click on “Create a post”  On editing: Click on gear symbol on top-left of content’s area. Choose desired action. |
| **Input** |  |
| **Processing flow** | 1. Redirect to a form containing the information of content (blank if creating). 2. Modify content. Perform validation. 3. Click “Submit”. |
| **Output** | Log recorded.  Redirect to user’s page. |
| **Note** |  |

▶ Add to cart

|  |  |
| --- | --- |
| **Privileged users** | Super Admin, Admin, User |
| **Description** | Add to cart with specific type, quantity. |
| **Activation** | Click on the product and choose “Add to cart”. |
| **Input** |  |
| **Processing flow** | * Redirect to product’s page. * Choose “Add to cart”. |
| **Output** |  |
| **Note** |  |

▶ Update cart

|  |  |
| --- | --- |
| **Privileged users** | Super Admin, Admin, User |
| **Description** | Modify product’s type and quantity. Delete a product in the cart. |
| **Activation** | Click on “Change” to modify purchase information.  Click on “+” or “-“ buttons to modify product’s quantity.  Click on “bin” button to delete a product from the cart. |
| **Input** |  |
| **Processing flow** | 1. Redirect to user’s cart. 2. Perform desired option. |
| **Output** |  |
| **Note** |  |

▶ Manage account information

|  |  |
| --- | --- |
| **Privileged users** | User |
| **Description** | Modify user’s information. |
| **Activation** | Click on the user’s avatar or hover on user’s avatar and choose Profile. |
| **Input** |  |
| **Processing flow** | 1. Redirect to user’s profile. |
| **Output** |  |
| **Note** |  |

▶ Make purchases

|  |  |
| --- | --- |
| **Privileged users** | Super Admin, Admin, User |
| **Description** | Make purchases. |
| **Activation** | Click on each product/ brand/ cart ‘s tick box, then choose “Purchase”. Website will redirect to purchase page. Choose payment method and confirm purchase. |
| **Input** |  |
| **Processing flow** | 1. Redirect to user’s cart. 2. Redirect to purchase’s page. 3. Make payment. |
| **Output** | Log recorded.  Redirect to user’s page. Show success notification. |
| **Note** |  |

# System Design

## III.1. Databases

### III.1.1. E-R Model

