

Coursework of
COMP1678 User-centered Web Engineering
Module Leader: Dr Solomon H. Ebenuwa

Name: Abdullah Al Sohel

ID: 001287370

Programme: MSc Computing and Information Systems

University of Greenwich

LOGIN DETAILS:

Customer 1:
username: sohel
Password: asdasd

Customer 2:
username: samir
Password: asdasd

Admin:
username: admin
Password: admin

localhost: <http://127.0.0.1:8000/>

Database name: GreenwichLandsMarket

Note: Please don't forget to migrate all the migration table schemas before run this project as it has been developed on laravel entirely ~~100~~ 😊

Table of Contents

Table of Contents.....	2
Table of Figures.....	3
Introduction.....	5
Part 1: Analysis Design and Front-End Implementation.....	6
Analysis Design.....	6
Using UCD Requirement Elicitation Tools.....	6
List of Requirements Using UCD.....	11
Front-End Implementation.....	12
Appropriate Usage of Modern UCD.....	12
Developing and Implementing the Complete Front-End Website.....	15
Part 2: System and Back-End Implementation.....	28
Back-End Functionalities.....	28
CRUD Functionalities:.....	28
Database Design:.....	29
Front-End Integration:.....	30
Security and User Authentication.....	30
Evaluation and Testing.....	35
Functionalities and Project Requirements Alignment:.....	35
Back-End Testing Evidence:.....	46
Conclusion.....	49
References.....	50

Table of Figures

- [Figure 1: Persona 1](#)
- [Figure 2: Persona 2](#)
- [Figure 3: Card Sorting](#)
- [Figure 4: Desktop view](#)
- [Figure 5: Laptop View](#)
- [Figure 6: Navigation buttons](#)
- [Figure 7: View from admin](#)
- [Figure 8: Admin Dashboard](#)
- [Figure 9: CRUD buttons in admin dashboard](#)
- [Figure 10: detailed view](#)
- [Figure 11: Order List](#)
- [Figure 12: Cart](#)
- [Figure 13: Buttons with interaction](#)
- [Figure 14: flash message for adding item to cart](#)
- [Figure 15: item increase, decrease and remove](#)
- [Figure 16: Empty cart after removing item](#)
- [Figure 17: Interactive buttons in dashboard](#)
- [Figure 18: Search functionality](#)
- [Figure 19: Download report](#)
- [Figure 20: Header](#)
- [Figure 21: Header in admin dashboard](#)
- [Figure 22: Header in individual picture page](#)
- [Figure 23: ERD diagram](#)
- [Figure 24: code for password encryption](#)
- [Figure 25: code for unique username](#)
- [Figure 26: session for login and logout](#)
- [Figure 27: CSRF token](#)
- [Figure 28: login page](#)
- [Figure 29: login page 2](#)
- [Figure 30: login successful](#)
- [Figure 31: login functionality](#)
- [Figure 32: registration page](#)
- [Figure 33: registration page 2](#)
- [Figure 34: registration successful](#)
- [Figure 35: registration functionality](#)
- [Figure 36: Create Picture Item button](#)
- [Figure 37: Create Picture Item form](#)
- [Figure 38: Create Picture successfully](#)
- [Figure 39: Latest picture showing in gallery](#)
- [Figure 40: Latest item in database](#)
- [Figure 41: Cart functionality 1](#)
- [Figure 44: Test cases](#)

[Figure 45: Test cases in code 1](#)

[Figure 46: Test cases code 2](#)

Introduction

The advent of the digital era has significantly reshaped the landscape of commerce, particularly in the realm of art and cultural heritage. This project, titled "Development of an E-commerce Website for Greenwich Landmark Pictures," is an ambitious endeavor aimed at harnessing these technological advancements. The primary objective is to create a dynamic online platform dedicated to selling high-quality images of the Old Royal Naval College's most iconic landmarks. This e-commerce website will not only serve as a digital marketplace but also as a virtual exhibition space, allowing users from various demographics, including those with physical, health, and social challenges, to effortlessly explore and purchase these visual representations of cultural heritage.

The website's core functionality revolves around a user-centric design, ensuring accessibility and ease of navigation for a diverse range of users, including senior citizens. The platform will incorporate essential e-commerce features such as user registration, login capabilities, and an intuitive picture purchasing process. Key functionalities include a home page displaying a curated selection of landmark images, user account management, and a robust content management system (CMS). Additionally, the site will offer advanced search capabilities, allowing users to effortlessly find images based on specific criteria. The focus on a user-centered design (UCD) approach will ensure the platform is intuitive and accommodating to all user categories, particularly emphasizing the inclusivity of individuals with varying physical and social needs.

The technical framework of this project is built on a stack of modern web technologies and tools. The front-end development utilizes Bootstrap 4 for responsive design, ensuring the website's adaptability across different devices and screen sizes. JavaScript enriches the user interface with interactive elements, enhancing the overall user experience. The back-end is powered by PHP, with Laravel serving as the primary framework to streamline development and ensure robust application architecture. Eloquent ORM aids in efficient database interactions, while MySQL, facilitated by MySQL Workbench, forms the backbone of data storage and management. Version control and collaborative development are managed through Git and GitHub, ensuring a streamlined and efficient workflow. This amalgamation of technologies and tools represents a comprehensive approach, combining aesthetics, functionality, and user accessibility to create a seamless online experience for purchasing landmark images.

Part 1: Analysis Design and Front-End Implementation

Analysis Design

The process of analysis design is pivotal in aligning the website's development with user-centric design (UCD) principles (Mao et al., 2005). This involves a thorough understanding of user needs and behaviors, which is achieved by employing various UCD requirement elicitation tools. By focusing on the specific context of the Greenwich Landmark Pictures e-commerce website, this stage ensures that the resulting platform is not only functional but also intuitive and user-friendly (Insfran & Fernandez, 2008).

Using UCD Requirement Elicitation Tools

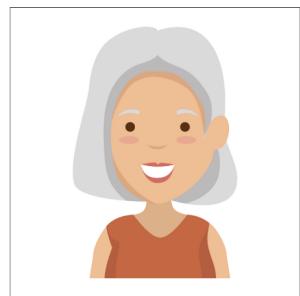
For the effective elicitation of requirements, a combination of UCD tools has been utilized:

1. Personas
2. Questionnaires
3. Card Sorting

Personas:

Developed to represent the diverse user base (Suryanto et al., 2020), including demographics such as age, work, and character. In the Persona section of the report, the user profiles for Margaret Thompson and Alex Rivera provide insight into the design considerations for the e-commerce website.

Margaret Thompson, The History Enthusiast



"A picture is a portal to the past, an invitation to uncover stories untold."

Age: **72**
Work: **Retired History Teacher**
Family: **Married, with grown children and grandchildren**
Location: **Canterbury, Kent**
Character: **The Lifelong Learner**

Goals

- To build a collection of landmark pictures that reflect her passion for history.
- To share her finds with her grandchildren and friends in her local history club.

Frustrations

- Difficulty reading small text and navigating complex websites.
- Overwhelming number of choices without adequate search filters.

Bio

Margaret is a retired history teacher who has a deep interest in historical landmarks and architecture. She enjoys exploring and learning about historical sites through photographs and stories.

Motivation



Personality

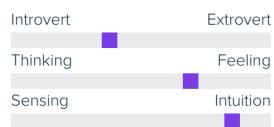
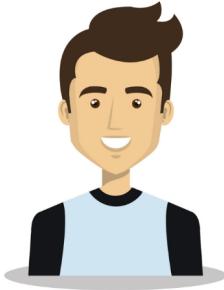


Figure 1: Persona 1

Margaret Thompson is a 72-year-old retired history teacher with a passion for historical landmarks. Her goals include expanding her collection of landmark images and sharing her passion with friends and family. She faces challenges with small text and complex navigation due to mild vision impairment. The website design for Margaret includes larger text options and a high-contrast mode to improve readability and ease of use.

Alex Rivera, The Young Professional



"Design is intelligence made visible, and each image is a brushstroke in my canvas of creativity."

Age: **29**
Work: **Graphic Designer**
Family: **Single, no kids**
Location: **Shoreditch, London**
Character: **The Creative Innovator**

Goals

- To find unique, high-resolution images for his creative projects.
- To efficiently purchase and download images without a time-consuming process.

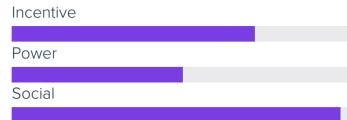
Frustrations

- Challenges in managing multiple images in his cart, which complicates the process of purchasing various types of images for different projects.
- Difficulty with websites that don't offer smooth pagination, making it a tedious task to browse through large collections of images.

Bio

Alex is a proactive graphic designer known for his fast-paced workflow and his need for a diverse range of high-quality visual content. He heavily relies on the convenience and efficiency of digital resources to meet the demanding deadlines of his creative profession.

Motivation



Personality

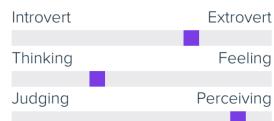


Figure 2: Persona 2

Alex Rivera, a 29-year-old graphic designer, seeks efficiency and diversity in the selection of visual content for his work. His frustrations are centered around inefficient cart management and poor pagination in extensive image libraries. His bio highlights a need for a fast-paced, convenient digital resource. Motivated by the desire to produce unique designs and engage with an online artistic community, the website design for Alex incorporates a seamless cart experience and smooth pagination to navigate through images effortlessly.

Both personas guide the website's design to be user-friendly, accessible, and efficient, reflecting the diverse needs of its user base.

Questionnaires:

Distributed to gather direct feedback from potential users, focusing on usability and accessibility preferences (Karanikolas et al., 2012).

Who	Type of User	Questions	Responses
John	Customer	Which devices are used by you while shopping online?	Mobile and laptop sometimes.
		Do you need easy access to view and manage your cart?	Yes, I'd like a clear and accessible way to view and update my cart before checkout.
Sam	Visitor	Do you need any specific requirement?	As a visitor, I should be able to browse products without logging in and be asked to login when trying to add products to cart so that it will stay there whenever I login back.
		What functionalities do you expect from the search feature?	I expect the search the image with name or keywords or description and filter the image by price.
		How do you find the process of registration on a website?	It should be quick, maybe with an option to register via social media accounts.

Card Sorting

Utilized to determine the optimal structure and categorization of the website's content, reflecting advanced database design practices (Laravel Documentation, 2023).

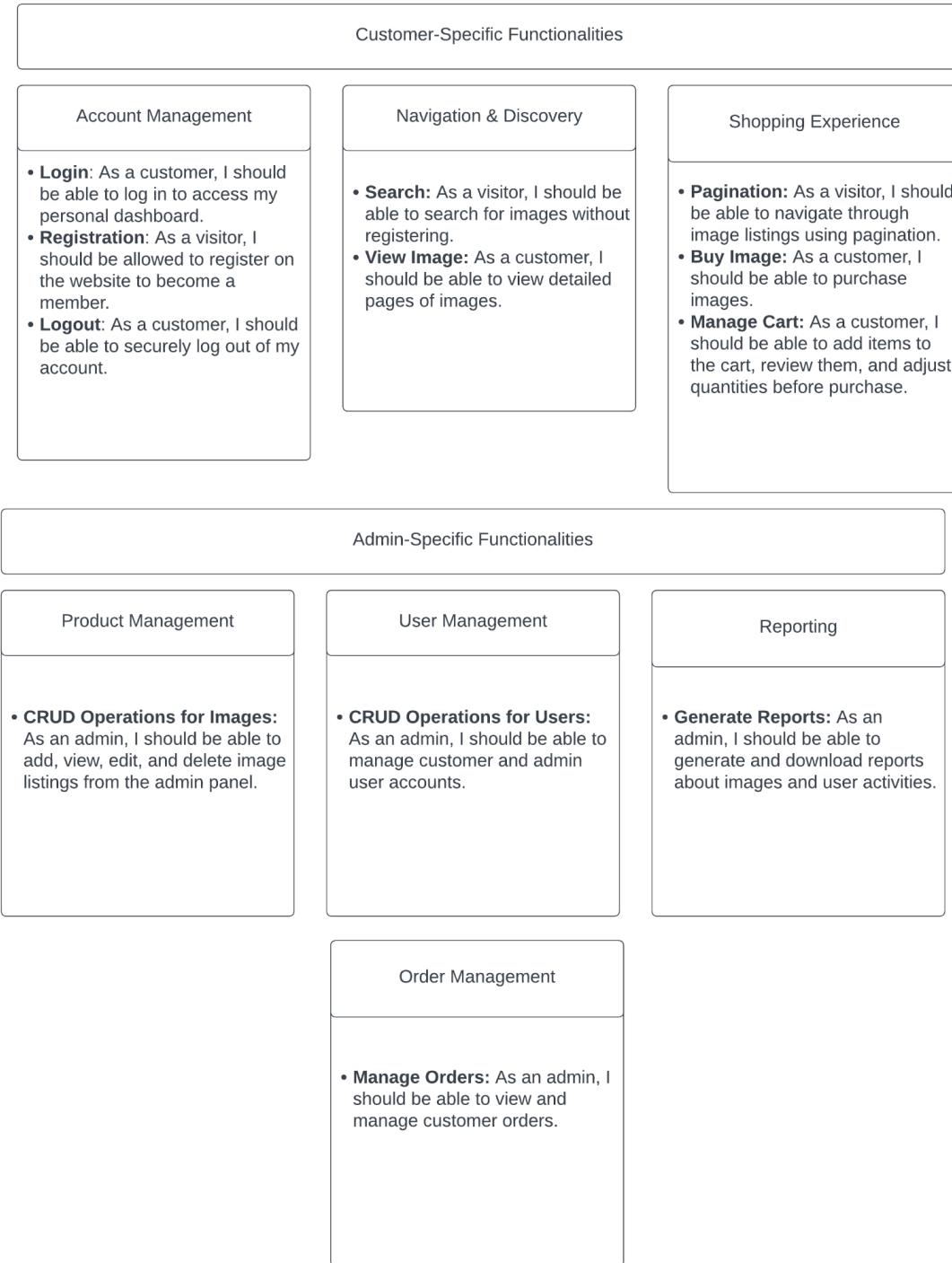


Figure 3: Card Sorting

List of Requirements Using UCD

Functional Requirements:

Customer-Specific:

- Login/Logout: Secure authentication process for users to access and leave their accounts.
- Registration: Simple account creation for new users.
- Search: Keyword and filter-based search functionality for images.
- View Image: Detailed view of images upon selection.
- Pagination: Incremental display of image listings.
- Buy Image: Seamless image selection and purchase process.
- Manage Cart: Ability to edit and review items in the shopping cart.

Admin-Specific:

- CRUD Operations: Manage (create, read, update, delete) images and user accounts.
- Reporting: Generation and retrieval of administrative reports.

Non-Functional Requirements:

- Responsiveness: Consistent user experience across all device types.
- Performance: Fast load times and responsive interactions.
- Security: Robust protection for user data and transactions.

These requirements aim to deliver a user-friendly and secure e-commerce platform for both customers and administrators, emphasizing performance and accessibility.

Front-End Implementation

The front-end implementation phase transitions the insights from the analysis design into a responsive and visually appealing UI that caters to both desktop and mobile users (Kim, 2013).

Appropriate Usage of Modern UCD

When addressing the user experience across different devices, it is essential to consider how the interface adapts to various screen sizes while maintaining functionality and aesthetic appeal.

For the responsive UI, two distinct interfaces have been designed:

Desktop Interface:

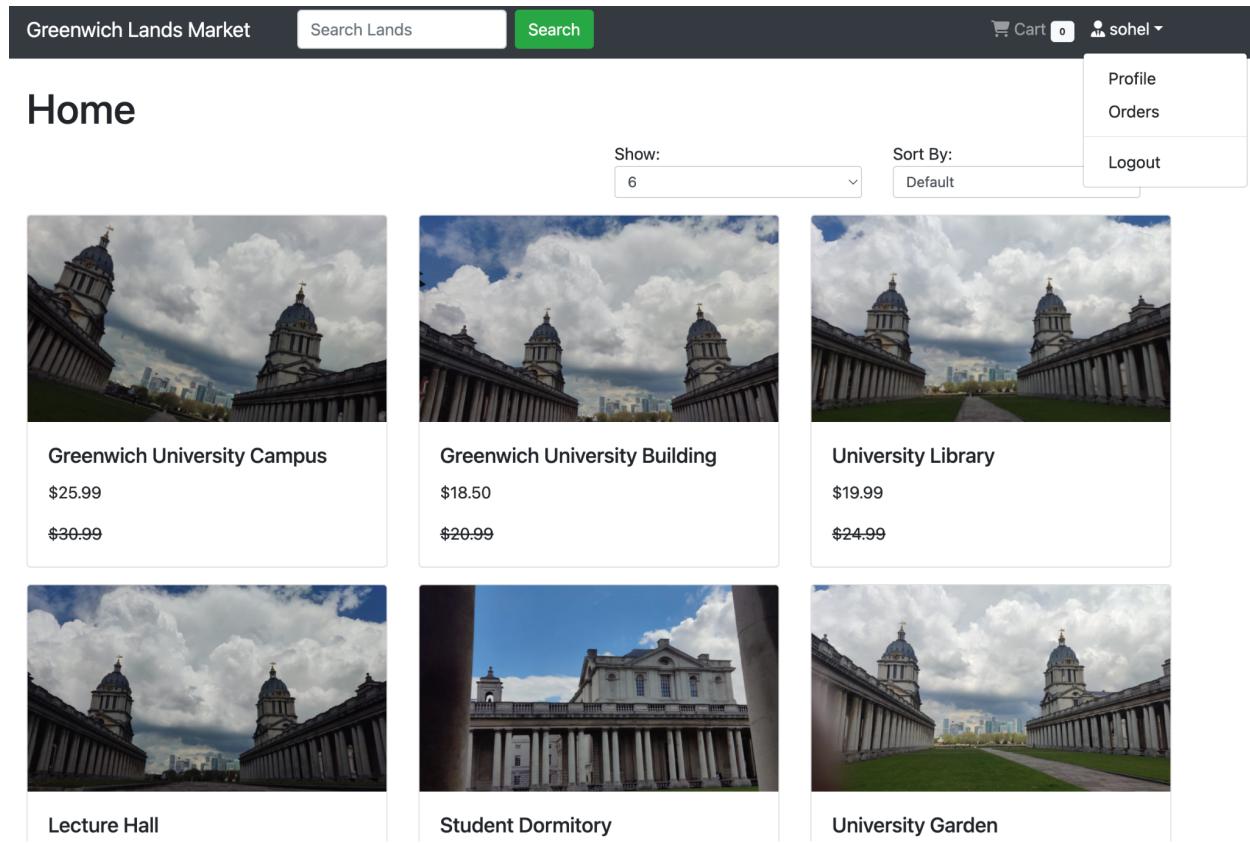


Figure 4: Desktop view

The desktop view features a login button and dropdown menu, with a focus on a three-card row display for products (Insfran & Fernandez, 2008). Post-login, customers can view their cart and access a dropdown menu from their username, offering profile management and logout options. For admins, there's an added link to the admin dashboard. The product display organizes images into three-card rows for easy browsing, and filter components like 'Show' and 'Sort' dropdowns are conveniently located at the top right.

Laptop Interface:

The screenshot shows the Greenwich Lands Market website's laptop view. At the top, there is a dark header bar with the text "Greenwich Lands Market" on the left, a search bar containing "Search Lands" and a green "Search" button in the center, and a hamburger menu icon on the right. Below the header, a "Cart" icon with a value of "0" and a user profile icon labeled "sohel" are visible. The main content area is titled "Home". It features two filter dropdowns: "Show: 6" and "Sort By: Default". Below these are three items, each with a thumbnail image of a building, the item name, and a price. The first item is "Greenwich University Campus", the second is "Greenwich University Building", and the third is "University Library" with a price of "\$19.99".

Show:	Sort By:
6	Default

Greenwich University Campus	Greenwich University Building	University Library \$19.99

Figure 5: Laptop View

This Interface demonstrates the adapted layout for a more compact screen, an important aspect of responsive web design (Kim, 2013). The laptop layout streamlines the interface, housing the cart and user menu within a hamburger icon to conserve space. The 'Show' and 'Sort' dropdowns are resized and repositioned to fit the more compact screen while remaining accessible.

Mobile Interface:

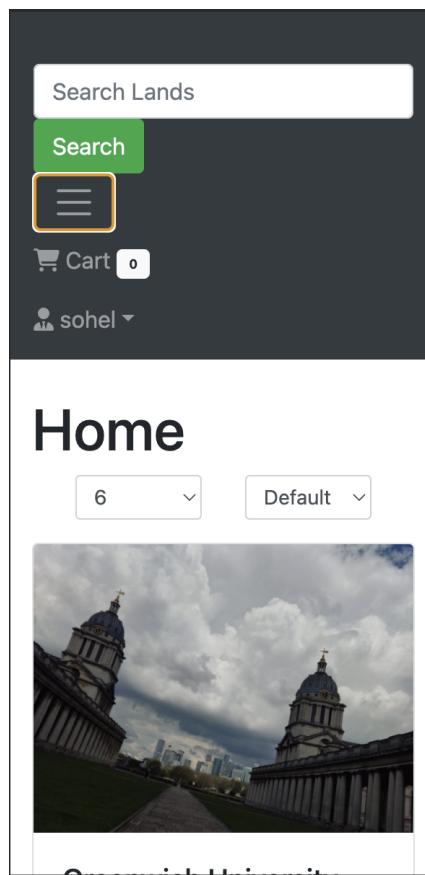


Figure 5: Mobile View

On mobile devices, the interface adopts a navigation style common in Android apps, a key consideration for discoverability in mobile contexts (Kim, 2013). The hamburger icon shifts to the left side, emulating a typical Android app navigation menu. Image cards are presented in a single column for clear viewing on narrow screens. Filter dropdowns for 'Show' and 'Sort' are centered on the screen, ensuring they are easily reachable and functional for on-the-go users.

Developing and Implementing the Complete Front-End Website

The navigation buttons, interactive elements, and layout design components have been developed with attention to detail, ensuring that the website is user-friendly and navigable (Insfran & Fernandez, 2008). The development has been executed with the end-user experience at the forefront. The design process involved incorporating all necessary components that contribute to a frictionless and intuitive navigation experience.

Navigation Buttons:

The navigation system is designed with clear and concise labeling, ensuring that users can effortlessly move through the website. The buttons are placed in intuitive locations, following established web conventions to minimize user confusion and learning curve. This strategic placement facilitates quick access to various sections such as the home page, product galleries, user accounts, and informational pages.

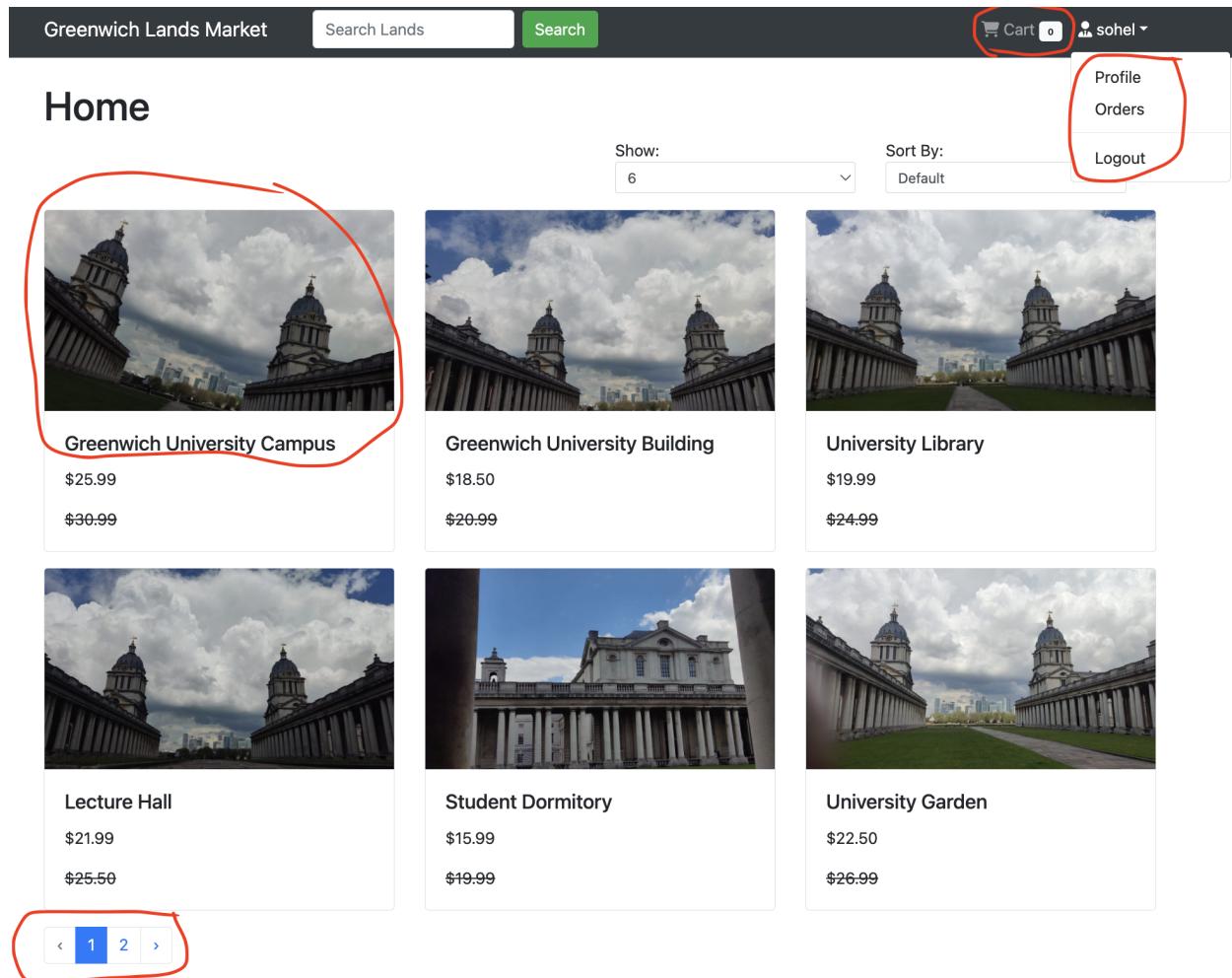


Figure 6: Navigation buttons

Navigation Buttons (Figure 6): Shows how users interact with the website, highlighting the importance of clear labeling and strategic placement (Suryanto et al., 2020). The Profile button takes users to their profile. Orders button displays completed orders in the table. Logout lets users logout, and the paginated button shows the next 6 images. Finally, the huge red mark lets users click on the image to see complete details.

The screenshot shows the admin dashboard of the Greenwich Lands Market. At the top, there is a navigation bar with 'Greenwich Lands Market' on the left, a search bar with 'Search Lands' and a 'Search' button in the middle, and 'Admin Dashboard' and a user profile icon on the right. A green success message box displays the text 'You have been logged in successfully.' Below the header, the word 'Home' is centered. On the right side, there are two filter dropdowns: 'Show:' set to 6 and 'Sort By:' set to Default. Below these filters, there are three image cards. Each card contains a thumbnail image of a building, its name, and a price. The first card is labeled 'Greenwich University Campus' with a price of '\$25.00'. The second card is labeled 'Greenwich University Building' with a price of '\$40.00'. The third card is labeled 'University Library' with a price of '\$10.00'.

Image	Name	Price
	Greenwich University Campus	\$25.00
	Greenwich University Building	\$40.00
	University Library	\$10.00

Figure 7: View from admin

In the admin view, “Admin Dashboard” becomes “Cart”. Check out the admin dashboard.

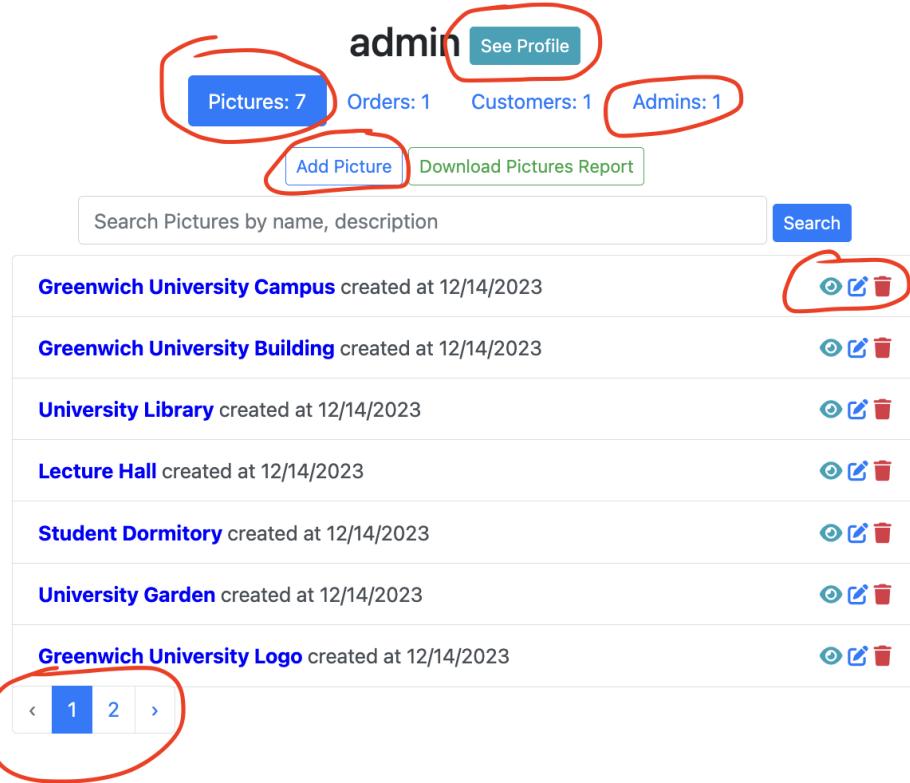


Figure 8: Admin Dashboard

To make it easy for admins to move between sections on the Admin dashboard page, the menu buttons were carefully planned out. The screenshot shows Pictures, Orders, Customers, and Admins with buttons. When the admin clicks orders, a table of client orders appears. By clicking "See Profile", Admin may view his profile here. Finally, each row of the table has three distinctive buttons. The eye symbol displays the item, the edit icon lets the admin edit it, and the trash icon removes it from the database and refreshes the page.

Add Picture Download Pictures Report														
<input type="text" value="Search Pictures by name, description"/>														
<input type="button" value="Search"/>														
<input type="button" value="View"/>														
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Greenwich University Campus created at 12/14/2023</td> <td style="text-align: right; padding: 5px;"> </td> </tr> <tr> <td style="padding: 5px;">Greenwich University Building created at 12/14/2023</td> <td style="text-align: right; padding: 5px;"> </td> </tr> <tr> <td style="padding: 5px;">University Library created at 12/14/2023</td> <td style="text-align: right; padding: 5px;"> </td> </tr> <tr> <td style="padding: 5px;">Lecture Hall created at 12/14/2023</td> <td style="text-align: right; padding: 5px;"> </td> </tr> <tr> <td style="padding: 5px;">Student Dormitory created at 12/14/2023</td> <td style="text-align: right; padding: 5px;"> </td> </tr> <tr> <td style="padding: 5px;">University Garden created at 12/14/2023</td> <td style="text-align: right; padding: 5px;"> </td> </tr> <tr> <td style="padding: 5px;">Greenwich University Logo created at 12/14/2023</td> <td style="text-align: right; padding: 5px;"> </td> </tr> </table>	Greenwich University Campus created at 12/14/2023		Greenwich University Building created at 12/14/2023		University Library created at 12/14/2023		Lecture Hall created at 12/14/2023		Student Dormitory created at 12/14/2023		University Garden created at 12/14/2023		Greenwich University Logo created at 12/14/2023	
Greenwich University Campus created at 12/14/2023														
Greenwich University Building created at 12/14/2023														
University Library created at 12/14/2023														
Lecture Hall created at 12/14/2023														
Student Dormitory created at 12/14/2023														
University Garden created at 12/14/2023														
Greenwich University Logo created at 12/14/2023														
< 1 2 >														

Figure 9: CRUD buttons in admin dashboard

Admin Dashboard (Figure 8 and Figure 9): Illustrates the admin's perspective, showcasing the CRUD functionality which is central to managing the platform (Laravel Documentation, 2023). After clicking on eye icon, it will let admin to go to page of detailed picture

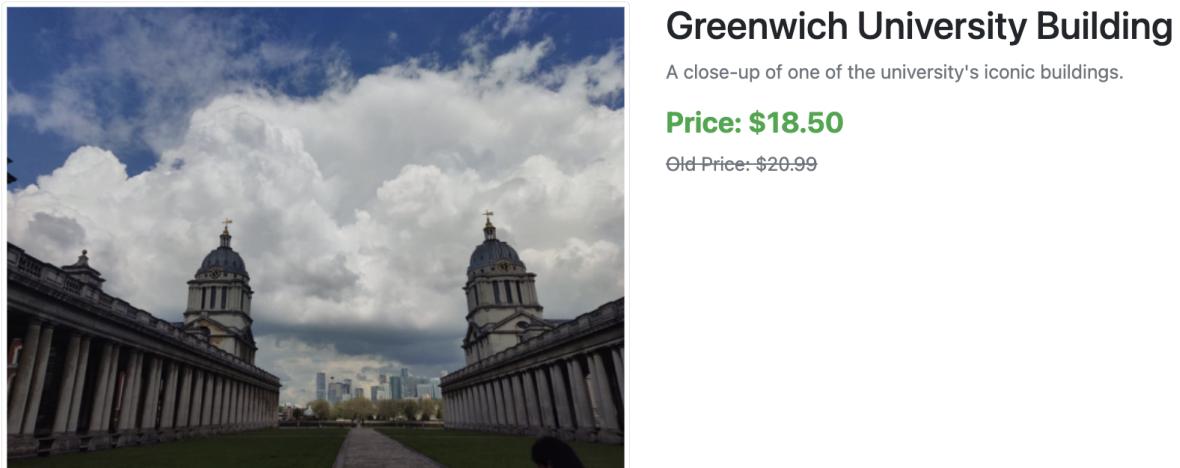


Figure 10: detailed view

As it can be seen, the whole page with the picture's details is showing.

The screenshot shows the admin dashboard with the following elements:

- User info: admin [See Profile]
- Dashboard stats: Pictures: 7, Orders: 1 (highlighted in blue), Customers: 1, Admins: 1
- Buttons: Download Orders Report
- Search bar: Search Orders by customer name
- Search button: Search
- Order details for "sohel's Order":
 - Total: \$167.92
 - Ordered on 14 Dec 2023
 - Order Items:
 - Greenwich University Campus - Qty: 3
 - University Sports Field - Qty: 5

Figure 10: Order table

Clicking the Orders tab opens the customer's order table. Each row has a customer name and a link to their information for the admin. Blue-colored order items are also navigable.

Order List

# Order ID	Date	Total Amount	Products
1	14 Dec 2023	\$167.92	Greenwich University Campus (x3) University Sports Field (x5)

Figure 11: Order List

Clicking the orders button on the home page displays customers' orders. Each row contains ordered products with links to their pages.

Your Cart

#	Product	Quantity	Price	Remove
1	 University Clock Tower	<button>-</button> 3 <button>+</button>	\$70.50	<button>Remove</button>
Total			\$70.50	
<button>Continue Shopping</button>			<button>Confirm Order</button>	

Figure 12: Cart

Customers can choose to either proceed to the home page by clicking the continue shopping button or they can confirm the order, which will then take them to the success page. Both options are available to them here.

Interactive Elements:

To create a dynamic and engaging user interface, JavaScript and CSS have been employed. These technologies have allowed for interactive elements such as dropdown menus, form validation to function smoothly. JavaScript has been used to handle user events and triggers, while CSS provides the visual transitions and animations that make the interaction feel more natural and responsive.

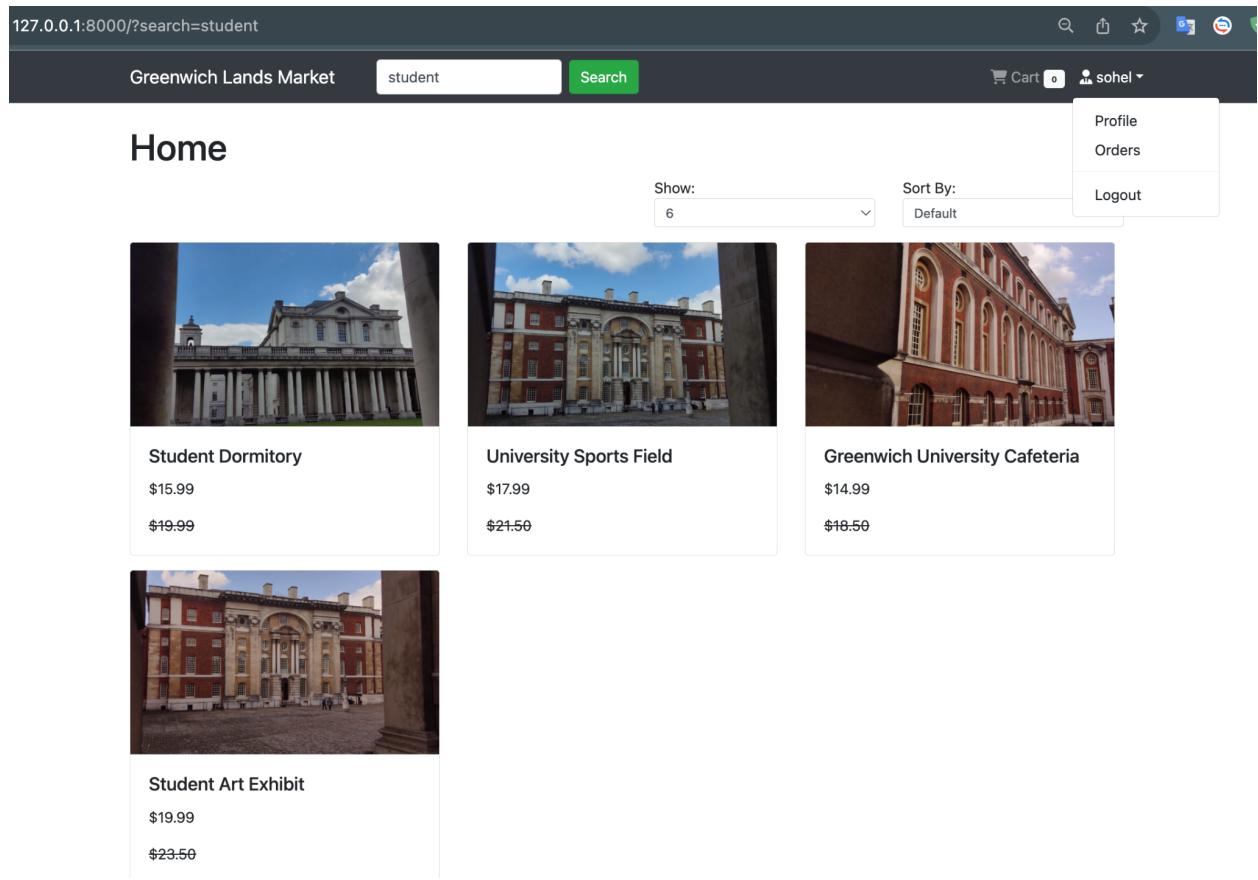


Figure 13: Buttons with interaction

If the user types what they want and clicks search, it will query the database and display the result on the home page. By using JavaScript, clicking on a user's username displays a menu with three buttons. Show and sort by dropdown input and all options are also available.

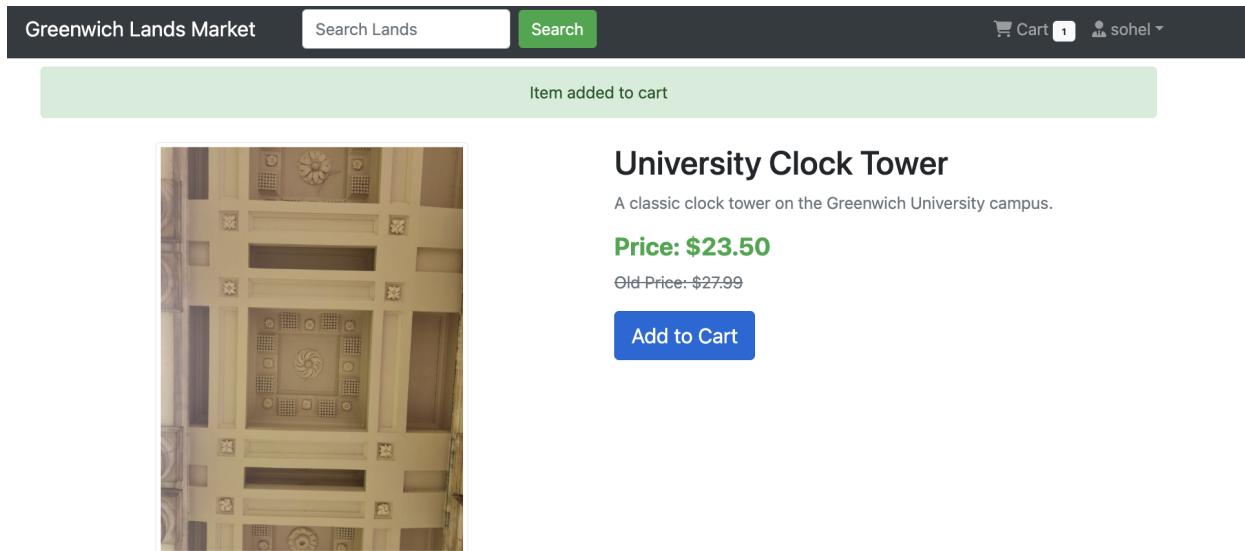


Figure 14: flash message for adding item to cart

After clicking on add to cart, the cart button on top right is changed from 0 to 1. Also Flash messages pop up with success messages. Lets see whats inside cart page.

#	Product	Quantity	Price	Remove
1	University Clock Tower	- 3 +	\$70.50	Remove

Total \$70.50

[Continue Shopping](#) [Confirm Order](#)

Figure 15: item increase, decrease and remove

Here users can increase and decrease their item quantity by clicking these buttons and flash messages will show up as well.

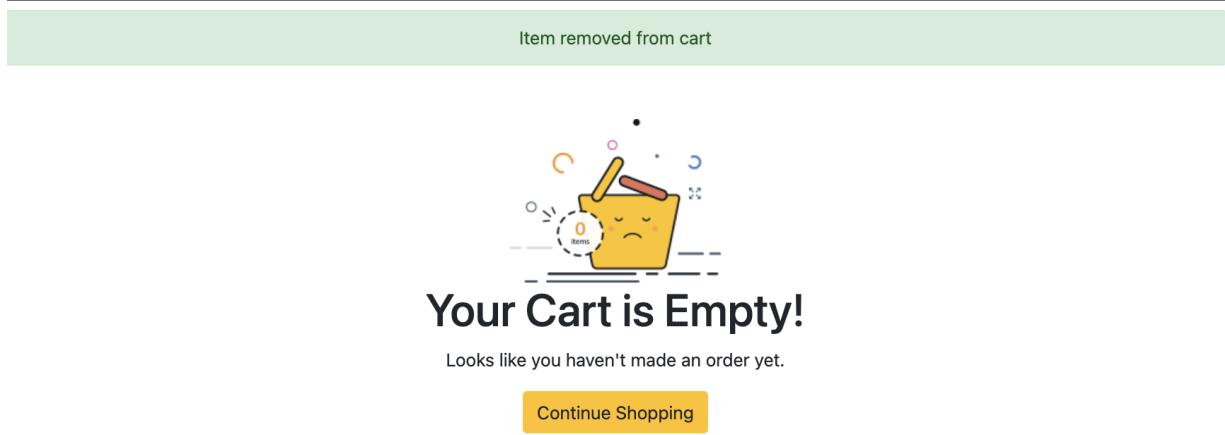


Figure 16: Empty cart after removing item

Also clicking on the remove button will remove the item from the cart with a flash message as well.

Customer	Total	Ordered On	Items
sohel	\$167.92	14 Dec 2023	Greenwich University Campus - Qty: 3 University Sports Field - Qty: 5
samir	\$29.98	14 Dec 2023	Greenwich University Cafeteria - Qty: 2
samir	\$69.50	14 Dec 2023	University Garden - Qty: 1 University Clock Tower - Qty: 2

Figure 17: Interactive buttons in dashboard

Also admin can interact with different buttons to visit different tables of different items like orders, customers and admins.

127.0.0.1:8000/admin?tab=orders&order_search=samir

The screenshot shows the Greenwich Lands Market admin dashboard. At the top, there's a header with the title "Greenwich Lands Market", a search bar labeled "Search Lands", a "Search" button, and a link "Admin Dash". Below the header, the user "admin" is logged in, with a "See Profile" button. There are summary statistics: Pictures: 7, Orders: 2, Customers: 2, Admins: 1, and a "Download Orders Report" button. A search bar contains the text "samir", and a "Search" button is next to it. Two order results are listed under "samir's Order".
Order 1 (Total: \$29.98): Ordered on 14 Dec 2023. Item: Greenwich University Cafeteria - Qty: 2.
Order 2 (Total: \$69.50): Ordered on 14 Dec 2023. Items: University Garden - Qty: 1, University Clock Tower - Qty: 2.

Figure 18: Search functionality

Admin can search pictures, orders, customers and admins as well.

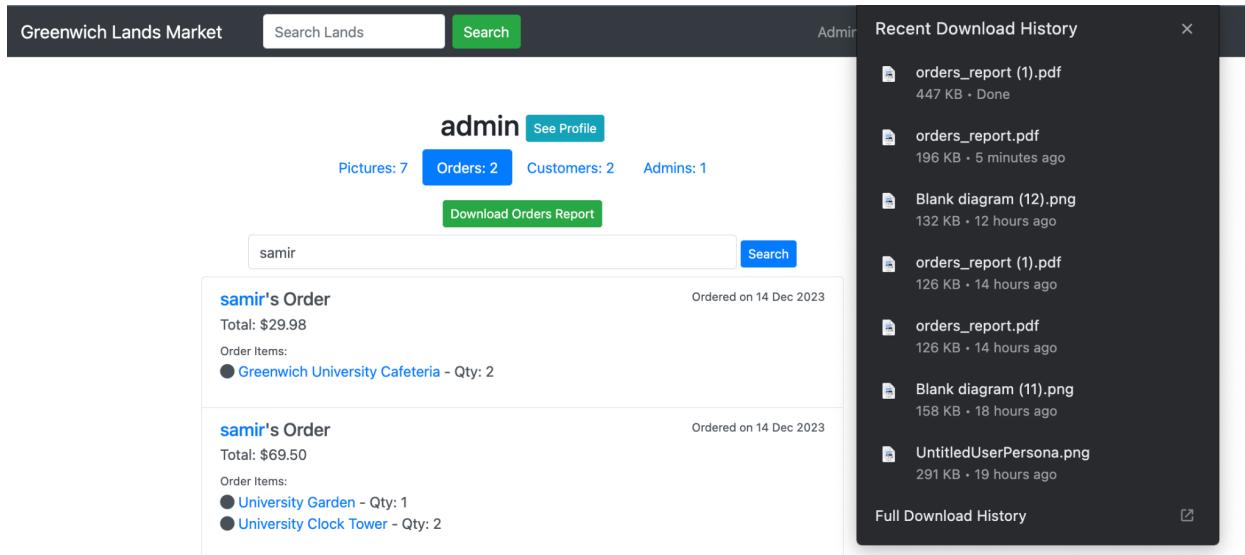


Figure 19: Download report

Lastly, clicking on the Download Report button will download a report of the pdf file. It can be a report of Pictures, Orders or users. Interactive Elements (Figure 13 to Figure 19): JavaScript and CSS provide dynamic content display and user interactions, enhancing the overall user experience (Karanikolas et al., 2012).

Layout and Design Components:

Attention to detail has been applied in crafting the layout and design components, with headers and footers that contain key information and navigational links. The search bar is prominently positioned to offer easy access for users to quickly find the images they desire. Media content, like images, is displayed in a way that is both aesthetically pleasing and functional, with consideration for loading times and quality.

The screenshot shows the Greenwich Lands Market homepage. At the top, there is a dark header bar with the text "Greenwich Lands Market" on the left, a search input field with placeholder "Search Lands" and a green "Search" button in the middle, and "Admin Dashboard" and "admin" on the right. A red oval highlights the entire header area. Below the header, the word "Home" is displayed in a large, bold, black font.

Below "Home", there are two sets of filters: "Show:" with a dropdown menu set to "6" and "Sort By:" with a dropdown menu set to "Default".

The main content area displays six cards, each featuring a small image of a building or landscape, the name of the location, and two price options:

- Greenwich University Campus**
\$25.99
\$30.99
- Greenwich University Building**
\$18.50
\$20.99
- University Library**
\$19.99
\$24.99
- Lecture Hall**
\$21.99
\$25.50
- Student Dormitory**
\$15.99
\$19.99
- University Garden**
\$22.50
\$26.99

At the bottom left, there is a navigation bar with arrows and the number "2", indicating there are two more pages of results.

Figure 20: Header

Navbar with logo button or home page link, search input with search button, admin dashboard or cart depending on user type, and user button are always there.

The screenshot shows the Admin Dashboard interface. At the top, there is a header bar with the title "Greenwich Lands Market", a search bar labeled "Search Lands", a green "Search" button, and a user profile section for "admin". Below the header, the main content area is titled "admin" with a "See Profile" link. It displays statistics: Pictures: 7, Orders: 3, Customers: 2, Admins: 1. There are buttons for "Add Picture" and "Download Pictures Report". A search bar allows searching by name or description. The main list contains seven items, each with a thumbnail, title, creation date, and three small icons (edit, delete, etc.). Below the list is a navigation bar with pages 1, 2, and >.

Picture Title	Created At	Action Icons
Greenwich University Campus	12/14/2023	
Greenwich University Building	12/14/2023	
University Library	12/14/2023	
Lecture Hall	12/14/2023	
Student Dormitory	12/14/2023	
University Garden	12/14/2023	
Greenwich University Logo	12/14/2023	

Figure 21: Header in admin dashboard

Also navbar with its components are present here in the admin dashboard.

The screenshot shows an individual picture view page. At the top, the header bar is identical to Figure 21. The main content features a large image of the Greenwich University Building under a cloudy sky. To the right of the image, the title "Greenwich University Building" is displayed, followed by a description: "A close-up of one of the university's iconic buildings." Below this, the price "Price: \$18.50" is shown in green, with the original price "Old Price: \$20.99" in smaller text below it.

Figure 22: Header in individual picture page

The navbar is presented here on the picture view page as well.

These screenshots serve as evidence of the thoughtful design and technical implementation that have gone into developing a comprehensive and fully functional front-end for the website. The aim has been to create a user-friendly environment that makes browsing, selecting, and purchasing images an enjoyable and hassle-free experience.

Part 2: System and Back-End Implementation

The back-end of the e-commerce platform is designed to provide robust and secure functionality for managing the website's operations. The underlying database architecture is structured to support a variety of operations that administrators and users will conduct.

Back-End Functionalities

The robustness of the back-end functionalities aligns with industry standards for web applications, ensuring that administrators can efficiently manage user profiles, product listings, and transactions (Laravel Documentation, 2023).

CRUD Functionalities:

- **Users:** Administrators can create new user profiles, read user information, update user details, and delete users if necessary. Users can manage their profiles, change passwords, and maintain session tokens for persistent logins.
- **Products:** Admins have the ability to create new product listings, read details on existing products, update product information, and remove products from the listing.
- **Carts:** The cart functionality allows users to add products, modify the quantity, and mark them as purchased. Each cart is tied to a user account for personalized shopping experiences.
- **Orders:** After shopping, users can place orders which are then managed by admins. Each order is linked to the user's account and contains information about the payment and order IDs.
- **Profiles:** User profiles are managed with the capability to add personal information such as names, birth dates, contact details, and addresses.

Database Design:

Here is a ERD (Entity-Relationship Diagram) diagram for the tables that have been used for the project.

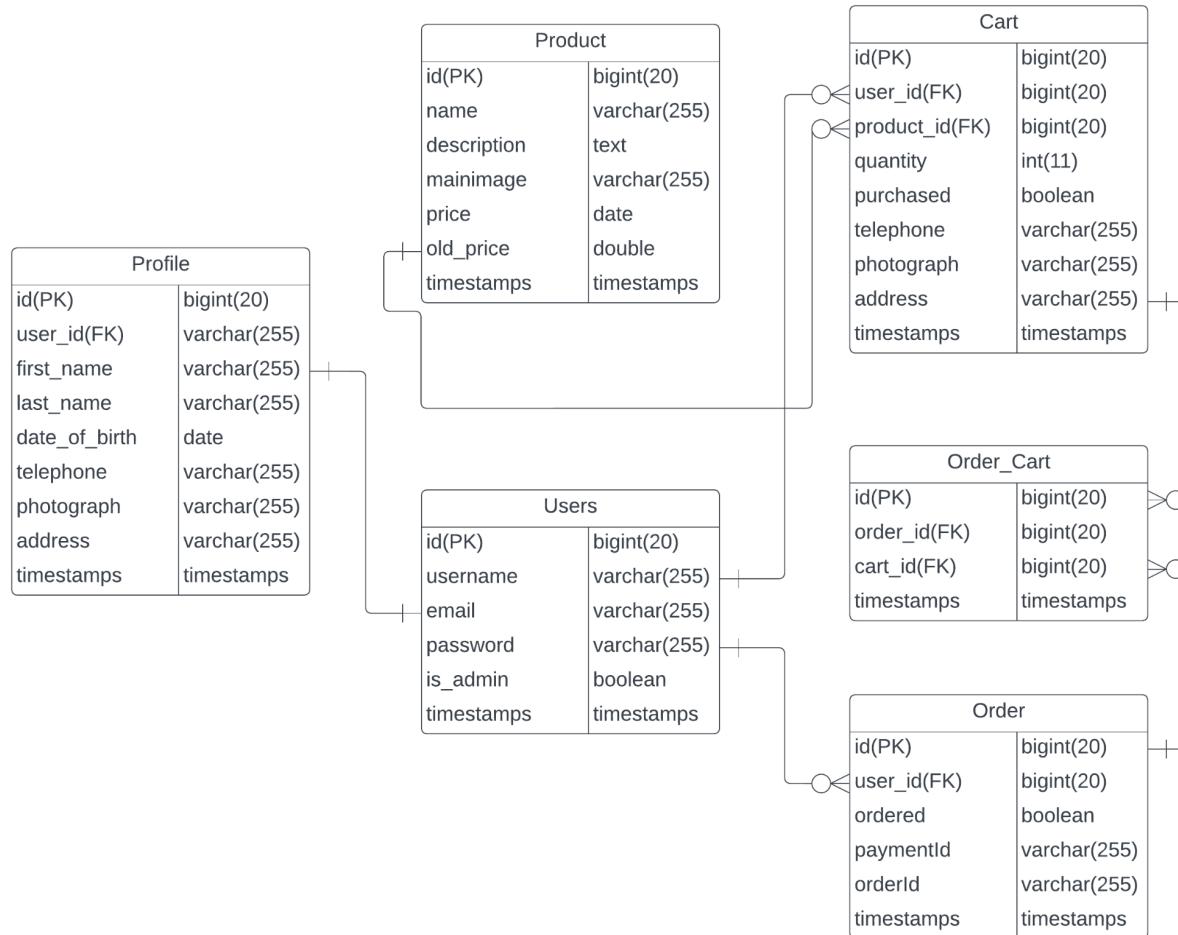


Figure 23: ERD diagram

Figure 23 illustrates the ERD, reflecting the relationships between users and carts, and the connection of orders to users, emphasizing the relational aspect of the database design (Chen, 1976).

- one user has one profile, a profile belongs to one user.
- one user can have zero or many carts, a cart belongs to one user.
- one user can have zero or many orders, an order belongs to one user.
- one product can be in zero or many carts, a cart is related to one product.
- an order can have zero or many carts through Order_Cart, a cart can be in zero or many orders through Order_Cart.

Front-End Integration:

The database is seamlessly integrated with the front-end through a series of route endpoints and server-side logic:

- When a customer logs in, the Users table is queried to authenticate credentials.
- Product views are generated by fetching data from the Products table, displaying images, descriptions, and prices.
- The shopping cart experience is driven by the Carts table, where items are added or removed by the user.
- Upon checkout, an Order is created, with details being stored and managed in the corresponding table.
- User profiles can be viewed and edited through the Profiles table, providing a personalized experience.

Security and User Authentication

Password Encryption:

Password encryption is implemented using Laravel's built-in hashing functions to enhance security measures (Taylor Otwell, 2023). Passwords are stored using robust hashing algorithms. When a user creates or updates a password, it is encrypted before being saved in the database. This means actual passwords are never stored, enhancing security against unauthorized access.

```
public function register(Request $request)
{
    $request->validate([
        'username' => ['required', 'min:3', 'max:255', Rule::unique
            ('users', 'username')],
        'email' => ['max:255'],
        'password' => ['required', 'min:6', 'max:255', 'confirmed'],
    ]);

    $user = User::create([
        'username' => $request->username,
        'email' => $request->email,
        'password' => Hash::make($request->password),
    ]);

    // Create an empty profile for the user
    $user->profile()->create([
        'user_id' => $user->id,
    ]);

    Auth::login($user);

    session()->flash('success', 'You have been registered and login
successfully.');
}
```

Figure 24: code for password encryption

Here password from password field is being stored with Hash algorithm in the database instead of plain text.

Username Validation:

Username uniqueness is validated through Laravel's validation rules to prevent duplication and ensure integrity (Laravel Documentation, 2023). If a username is already taken, the user is prompted to select a different one, thus maintaining unique identities within the platform.

```
public function register(Request $request)
{
    $request->validate([
        'username' => ['required', 'min:3', 'max:255', Rule::unique
            ('users', 'username')],
        'email' => ['max:255'],
        'password' => ['required', 'min:6', 'max:255', 'confirmed'],
    ]);

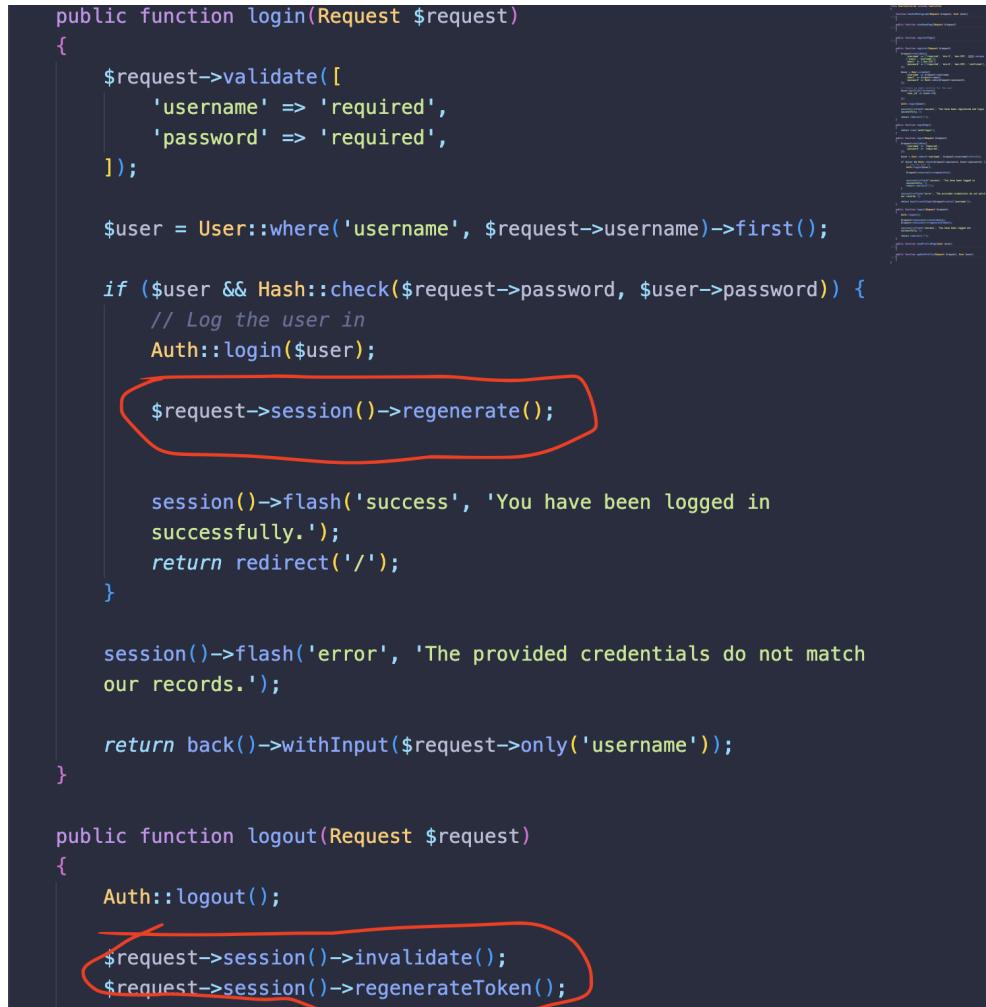
    $user = User::create([
        'username' => $request->username,
        'email' => $request->email,
        'password' => Hash::make($request->password),
    ]);
}
```

Figure 25: code for unique username

Here the username from the text field is being checked with the Rule validation module of laravel in order to prevent duplicate username.

Session Management and Logout Mechanisms:

The session management strategy employs tokens, a standard practice for maintaining stateful interactions over the stateless HTTP protocol (Fielding et al., 1999). Sessions are managed through tokens that are created upon login and invalidated on logout. This token is checked with each request to verify a user's authenticity. Logout mechanisms are designed to immediately invalidate these tokens, effectively ending the session and protecting against session hijacking.



```
public function login(Request $request)
{
    $request->validate([
        'username' => 'required',
        'password' => 'required',
    ]);

    $user = User::where('username', $request->username)->first();

    if ($user && Hash::check($request->password, $user->password)) {
        // Log the user in
        Auth::login($user);

        $request->session()->regenerate();
    }

    session()->flash('success', 'You have been logged in successfully.');
    return redirect('/');
}

session()->flash('error', 'The provided credentials do not match our records.');

return back()->withInput($request->only('username'));
}

public function logout(Request $request)
{
    Auth::logout();

    $request->session()->invalidate();
    $request->session()->regenerateToken();
}
```

Figure 26: session for login and logout

Red marks denote the session regeneration and session invalidate and generate another token for next session for login and logout functionalities.

Protection of Authentication Credentials:

CSRF protection, a critical security feature, following best practices for web security (OWASP Foundation, 2023). Credentials are protected during transit with SSL/TLS encryption, ensuring data sent between the user's browser and the server is encrypted. The system also implements measures to protect against common vulnerabilities like SQL injection and Cross-Site Request Forgery (CSRF), which can expose user credentials.

```
to create account</p>
</div>

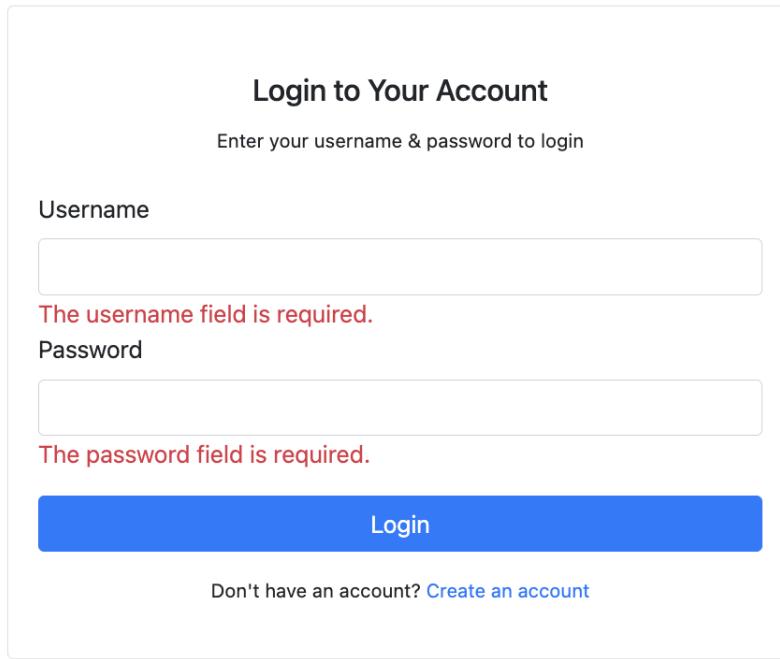
<!-- Registration Form -->
<form class="row g-3" method="POST" action="/register">
    @csrf
    <!-- Username -->
    <div class="col-12">
        <label for="yourUsername" class="form-label">Username</label>
        <input type="text" value="{{ old('username') }}" name="username" class="form-control" id="yourUsername" required>
        @error('username')
            <span class="text-danger">{{ $message }}</span>
        @enderror
    </div>
```

Figure 27: CSRF token

Csrf token is being used here for the post request to prevent site attack from the hacker.

Evaluation and Testing

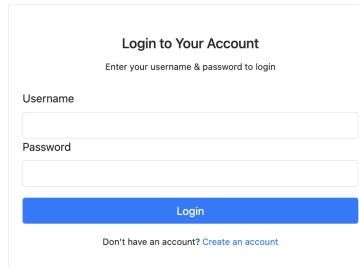
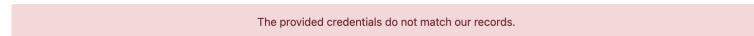
Functionalities and Project Requirements Alignment:



The image shows a login form titled "Login to Your Account". It includes a placeholder text "Enter your username & password to login". There are two input fields: "Username" and "Password". Below each field is a red error message: "The username field is required." and "The password field is required." respectively. A blue "Login" button is centered below the fields. At the bottom, there is a link "Don't have an account? [Create an account](#)".

Figure 28: login page

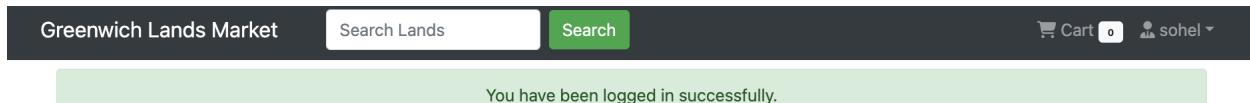
If a user tries to login without entering credentials, it will show error messages.



This image shows the same login form as Figure 28, but with a different error message. The "Username" and "Password" fields are empty. A red flash message at the top of the page reads "The provided credentials do not match our records." The rest of the form, including the "Login" button and account creation link, remains the same.

Figure 29: login page 2

If the user provide incorrect credentials, it will show flash message with error message.



Home

Show:

6

Sort By:

Default



Figure 30: login successful

After login successfully, it will redirect to the home page with successful flash message.

```
public function login(Request $request)
{
    $request->validate([
        'username' => 'required',
        'password' => 'required',
    ]);

    $user = User::where('username', $request->username)->first();

    if ($user && Hash::check($request->password, $user->password)) {
        // Log the user in
        Auth::login($user);

        $request->session()->regenerate();

        session()->flash('success', 'You have been logged in
successfully.');
        return redirect('/');
    }

    session()->flash('error', 'The provided credentials do not match
our records.');

    return back()->withInput($request->only('username'));
}
```

Figure 31: login functionality

Backend logic implementation for login users.

Create an Account

Enter your personal details to create account

Username

Email

Password

The password field confirmation does not match.

Confirm Password

[Create Account](#)

Already have an account? [Log in](#)

Figure 32: registration page

Registration process will not happen until both password fields match. And it will show an error message under the password field.

Create an Account

Enter your personal details to create account

Username

The username has already been taken.

Email

Password

Confirm Password

Create Account

Already have an account? [Log in](#)

Figure 33: registration page 2

Also if the username already exists in the database, it will let the user know about it.

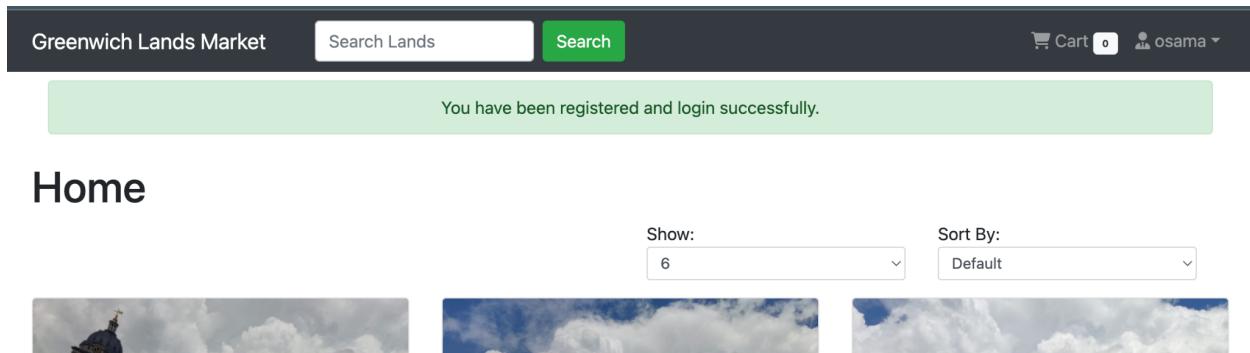


Figure 34: registration successful

After registration, it will login the user and redirect to the home page with a flash message.

```
public function register(Request $request)
{
    $request->validate([
        'username' => ['required', 'min:3', 'max:255', Rule::unique
            ('users', 'username')],
        'email' => ['max:255'],
        'password' => ['required', 'min:6', 'max:255', 'confirmed'],
    ]);

    $user = User::create([
        'username' => $request->username,
        'email' => $request->email,
        'password' => Hash::make($request->password),
    ]);

    // Create an empty profile for the user
    $user->profile()->create([
        'user_id' => $user->id,
    ]);

    Auth::login($user);

    session()->flash('success', 'You have been registered and login
successfully.');

    return redirect('/');
}
```

Figure 35: registration functionality
Code for the registration process with validation rules.

The screenshot shows the Greenwich Lands Market Admin Dashboard at the URL 127.0.0.1:8000/admin?tab=pictures&picture_search=university. The top navigation bar includes 'Greenwich Lands Market', a search bar, and 'Admin Dashboard'. The main area is titled 'admin' with a 'See Profile' button. It displays a summary: Pictures: 7, Orders: 1, Customers: 1, Admins: 1. Below this are buttons for 'Add Picture' and 'Download Pictures Report'. A search bar allows searching by name or description, with a 'Search' button. The main content is a list of seven university pictures, each with a delete icon. The pictures are:

- Greenwich University Campus created at 12/14/2023
- Greenwich University Building created at 12/14/2023
- University Library created at 12/14/2023
- Student Dormitory created at 12/14/2023
- University Garden created at 12/14/2023
- Greenwich University Logo created at 12/14/2023
- University Sports Field created at 12/14/2023

Pagination at the bottom shows pages 1, 2, and 3.

Figure 36: Create Picture Item button

“Add picture” button for letting the admin to use the forms for creating picture that will be shown to the gallery of home page

Create New Picture

Picture Name

Description

Main Image
 WhatsApp Image 2023-12-14 at 3.47.19 pm (3).jpeg

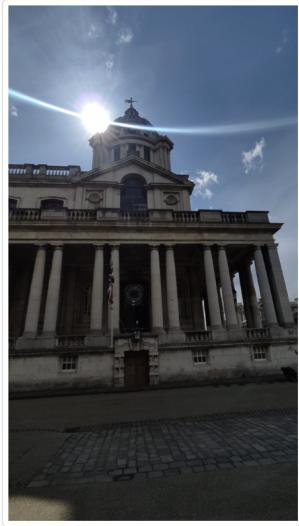
Price

Old Price

Figure 37: Create Picture Item form

Picture item form for adding pictures to the database and showing it to the homepage gallery.

Picture created successfully.



Greenwich Dome

When sun arises, dome rises as well

Price: \$31.50

Old Price: \$35.00

Figure 38: Create Picture successfully

After adding the image, it will redirect to that particular image's route with a successful flash message.

127.0.0.1:8000/?page=2

 Greenwich University Logo \$9.99 \$12.50	 Classroom Interior \$20.50 \$22.99	 University Sports Field \$17.99 \$21.50
 Greenwich University Cafeteria \$14.99 \$18.50	 Student Art Exhibit \$19.99 \$23.50	 Greenwich Dome \$31.50 \$35.00

< 1 2 >

Figure 39: Latest picture showing in gallery

Latest item that has been inserted to the database is shown here alongside with other pictures.

MySQL Workbench

Local instance 3306 - Warning - not supported

Schemas: GreenwichLandsMarket, GreenwichLandsMarketTest, Java_CRUD_APP, LondonMusics, MariataHomes, MariataHomesTest, myLaravelApp, sys

Tables: Products

Query:

```
1 •   SELECT * FROM GreenwichLandsMarket.products;
```

Result Grid:

	id	name	description	mainimage	price	old_price	created_at	updated_at
1	1	Greenwich University Campus	A scenic view of the Greenwich University camp...	Greenwich University Campus_1702575448.jpg	25.99	30.99	2023-12-14 17:37:28	2023-12-14 17:37:28
2	2	Greenwich University Building	A close-up of one of the university's iconic build...	Greenwich University Building_1702575929.jpg	18.5	20.99	2023-12-14 17:45:29	2023-12-14 17:45:29
3	3	University Library	The interior of the university library filled with bo...	University Library_1702576036.jpg	19.99	24.99	2023-12-14 17:47:16	2023-12-14 17:47:16
4	4	Lecture Hall	A spacious lecture hall with rows of seats and a...	Lecture Hall_1702576086.jpg	21.99	25.5	2023-12-14 17:48:06	2023-12-14 17:48:06
5	5	Student Dormitory	Exterior view of student dormitories at Greenwich...	Student Dormitory_1702576123.jpg	15.99	19.99	2023-12-14 17:48:43	2023-12-14 17:48:43
6	6	University Garden	A beautifully landscaped garden on the universit...	University Garden_1702576176.jpg	22.5	26.99	2023-12-14 17:49:36	2023-12-14 17:49:36
7	7	Greenwich University Logo	The official logo of Greenwich University.	Greenwich University Logo_1702576371.jpg	9.99	12.5	2023-12-14 17:52:52	2023-12-14 17:52:52
8	8	Classroom Interior	A well-lit classroom with desks and chairs.	Classroom Interior_1702576410.jpg	20.5	22.99	2023-12-14 17:53:31	2023-12-14 17:53:31
9	9	University Sports Field	An expansive sports field where students enjoy...	University Sports Field_1702576443.jpg	17.99	21.5	2023-12-14 17:54:03	2023-12-14 17:54:03
10	10	Greenwich University Cafeteria	The bustling cafeteria where students dine and...	Greenwich University Cafeteria_1702576483.jpg	14.99	18.5	2023-12-14 17:54:43	2023-12-14 17:54:43
11	11	Student Art Exhibit	An art exhibit showcasing the creative talents of...	Student Art Exhibit_1702576513.jpg	19.99	23.5	2023-12-14 17:55:13	2023-12-14 17:55:13
12	12	Greenwich Dome	When sun arises, dome rises as well.	Greenwich Dome_1702582746.jpg	31.5	35	2023-12-14 19:39:06	2023-12-14 19:39:06
13	13	University with Clouds	The Clouds are laughing ha ha ha	University with Clouds_1702585795.jpg	500	700	2023-12-14 20:29:56	2023-12-14 20:30:38
14	14	Campus view	Campus view from manhole view	Campus view_1702585885.jpg	123	130	2023-12-14 20:31:25	2023-12-14 20:31:25
	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

products 1

Query Completed

Figure 40: Latest item in database

Latest Item has been inserted to database successfully

Add picture to cart, change picture quantity by increasing or decreasing has been shown to [Interactive Elements section](#)

```
class OrderController extends Controller
{
    public function addToCart($productId)
    {
        $product = Product::findOrFail($productId);

        $cart = Cart::firstOrCreate([
            'product_id' => $product->id,
            'user_id' => Auth::id(),
            'purchased' => false
        ]);

        $order = Order::firstOrCreate([
            'user_id' => Auth::id(),
            'ordered' => false
        ]);

        if (!$order->orderItems()->where('cart_id', $cart->id)->exists())
        {
            $order->orderItems()->attach($cart->id);
        } else {
            $cart->quantity += 1;
            $cart->save();
        }

        return redirect()->route('pictures.show', $productId)
            ->with('success', 'Item added to cart');
    }
}
```

Figure 41: Cart functionality 1

For adding pictures to cart functionality.

```
public function viewCart()
{
    $carts = Cart::where('user_id', Auth::id())->where('purchased', false)->get();
    $order = Order::where('user_id', Auth::id())->where('ordered', false)->first();

    return view('cart/cart', [
        'carts' => $carts,
        'order' => $order,
        'isExists' => $carts->isNotEmpty() && $order !== null
    ]);
}

public function removeFromCart($productId)
{
    $order = Order::where('user_id', Auth::id())->where('ordered', false)->first();

    if ($order) {
        $cart = Cart::where('product_id', $productId)
            ->where('user_id', Auth::id())
            ->where('purchased', false)
            ->first();

        if ($cart) {
            $order->orderItems()->detach($cart->id);
            $cart->delete();

            return redirect()->route('cart.view')
                ->with('success', 'Item removed from cart');
        }
    }
}
```

Figure 42: Cart functionality 2

For viewing items in cart and removing items from cart functionality.

```
public function changeCartItemQuantity($productId, $changeType)
{
    $cart = Cart::where('product_id', $productId)
        ->where('user_id', Auth::id())
        ->where('purchased', false)
        ->firstOrFail();

    if ($changeType === 'increase') {
        $cart->quantity += 1;
    } elseif ($changeType === 'decrease' && $cart->quantity > 1) {
        $cart->quantity -= 1;
    }

    $cart->save();

    return redirect()->route('cart.view')
        ->with('success', 'Cart updated successfully');
}

public function confirmOrder(Request $request)
{ ... }
```

Figure 43: Cart functionality 3

Change item quantity by increasing or decreasing from cart implementation.

Back-End Testing Evidence:

```
sohel@Sohel-MacBook GreenwichLandsMarket % php artisan test

PASS Tests\Feature\AdminControllerTest
✓ admin can view admin dashboard
✓ admin can create picture
✓ admin can edit picture
✓ admin can delete picture
✓ admin can create customer
✓ admin can edit customer
✓ admin can delete customer
✓ admin can download pictures report
✓ admin can download customers report
✓ admin can download admins report

PASS Tests\Feature\OrderControllerTest
✓ user can add product to cart
✓ user can view cart
✓ user can remove product from cart
✓ user can change cart item quantity

PASS Tests\Feature\UserControllerTest
✓ user can view register page
✓ user can register
✓ user can view login page
✓ user can login
✓ user can logout

Tests: 19 passed (59 assertions)
Duration: 0.82s
```

Figure 44: Test cases

There are 19 tests for CRUD users, photographs, downloads, customers, admins, etc. Testing cart functions was successful. Most significantly, test cases have been implemented to ensure that users are able to log in, register, and log out of their accounts.

```
class UserControllerTest extends TestCase
{
    use RefreshDatabase; // Ensures the database is reset for each test

    /** @test */
    public function user_can_view_register_page()
    {
        $response = $this->get('/register');
        $response->assertStatus(200);
        $response->assertViewIs('auth.register');
    }

    /** @test */
    public function user_can_register()
    {
        $response = $this->post('/register', [
            'username' => 'newuser',
            'email' => 'newuser@example.com',
            'password' => 'password',
            'password_confirmation' => 'password',
        ]);

        $this->assertAuthenticated();
        $response->assertRedirect('/');
    }

    /** @test */
    public function user_can_view_login_page()
    {
        $response = $this->get('/login');
        $response->assertStatus(200);
        $response->assertViewIs('auth.login');
    }
}
```

Figure 45: Test cases in code 1

Implementation code for test scenarios that ensure users are able to correctly browse the register page and register.

```

/** @test */
public function user_can_login()
{
    $user = User::factory()->create([
        'username' => 'existinguser',
        'password' => Hash::make('password'),
    ]);

    $response = $this->post('/login', [
        'username' => 'existinguser',
        'password' => 'password',
    ]);

    $this->assertAuthenticatedAs($user);
    $response->assertRedirect('/');
}

/** @test */
public function user_can_logout()
{
    $this->actingAs(User::factory()->create());

    $response = $this->post('/logout');

    $this->assertGuest();
    $response->assertRedirect('/');
}

```

Figure 46: Test cases code 2

In order to guarantee that users are able to view the login page, log in from the page, and log out without any problems appearing.

Testing is crucial for verifying that back-end functionalities meet the project requirements. Figures 28 to 46 display various test cases and code implementations, showcasing rigorous quality assurance processes. These figures validate the effectiveness of the login, registration, and product management functionalities, ensuring they conform to the outlined requirements (Myers et al., 2011).

Conclusion

In conclusion, the e-commerce platform's development has been a comprehensive exercise in applying modern web development techniques and user-centered design principles. Challenges encountered have been instrumental in learning and adapting the project to meet security and usability standards (Nielsen, 1994). The project has successfully delivered a comprehensive e-commerce platform tailored to selling pictures of landmarks, achieving key milestones in user experience, functionality, and security.

Challenges faced during the project, such as ensuring data security and optimizing the user interface for various devices, provided valuable learning opportunities. Lessons learned from these challenges include the importance of thorough testing to identify potential security vulnerabilities and the benefits of a flexible design framework to accommodate different user needs.

Overall, the project stands as a testament to the successful application of user-centered design principles and modern web development practices. The experience has underscored the critical balance between aesthetic appeal and functionality in creating an engaging and secure online environment.

References

1. Chen, P.P. (1976). The Entity-Relationship Model—Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1(1), pp.9-36.
2. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. & Berners-Lee, T. (1999). Hypertext Transfer Protocol -- HTTP/1.1. RFC 2616.
3. Insfran, E. & Fernandez, A. (2008). A Systematic Review of Usability Evaluation in Web Development. In WISE 2008: Web Information Systems Engineering – WISE 2008 Workshops.
4. Karanikolas, N.N., Tritaki, A. & Karavasileiadis, O. (2012). Advanced Database Design and Modern Web Technologies Used in Nutrition Software Design. In 2012 16th Panhellenic Conference on Informatics, pp.1-5.
5. Kim, B. (2013). Responsive Web Design, Discoverability, and Mobile Challenge. ALA TechSource, August/September 2013.
6. Laravel Documentation. (2023). Database: Getting Started. Available at: <https://laravel.com/docs/8.x/database> [Accessed 4 December 2023].
7. Mao, J.-Y., Vredenburg, K., Smith, P.W. and Carey, T. (2005). The state of user-centered design practice. *ACM Transactions on Computer-Human Interaction*, [online] 12(1), pp.1–39. Available at: <https://doi.org/10.1145/1047671.1047677> [Accessed 2 December 2023].
8. Myers, G.J., Sandler, C. & Badgett, T. (2011). *The Art of Software Testing*. John Wiley & Sons.
9. Nielsen, J. (1994). *Usability Engineering*. Academic Press.
10. OWASP Foundation. (2023). Cross-Site Request Forgery (CSRF). Available at: <https://owasp.org/www-community/attacks/csrf> [Accessed 10 December 2023].
11. Suryanto, T.L.M., Faroqi, A., Ridwandono, D., Saftri, N.D., Ithriah, S.A. & Pratama, A. (2020). Exploration of User Experience Approach in Using the Digital Commerce Platform on Millennial User in Indonesia. In 2020 6th Information Technology International Seminar (ITIS), pp.1-5.
12. Taylor Otwell. (2023). Encryption. Laravel Documentation. Available at: <https://laravel.com/docs/8.x/encryption> [Accessed 11 December 2023].