



An-Najah National University  
Faculty of Engineering & Information Technology  
Computer Department

Presented in partial fulfilment of the requirements for Bachelor degree in Computer  
Engineering



**HARD WHERE**

Mohammad Omar | Osama Khanfa  
11923761 | 11924807  
Supervised by: Dr. Muhannad Al-Jabi  
January 2024

<b>Introduction:</b>	<b>3</b>
<b>Abstract:</b>	<b>4</b>
<b>Problem and theoretical background:</b>	<b>5</b>
<b>Methodology:</b>	<b>6</b>
<b>Results And Discussion:</b>	<b>8</b>
Frontend:	8
• Nuxt.js:	8
• Ionic with Vue:	9
Backend:	10
• DotNet with Clean Architecture:	10
• ORM:Entity FrameWork Core:	10
• Image Handling:	11
• API Documentation:	11
Database:	12
Real-Time Capabilities:	12
Web Application Features:	13
• Authentication	13
• Authorization	13
User-Friendly Interface:	14
Product Details	15
Wishlist:	16
Cart Screen:	17
Checkout Page:	18
Checkout or Place an order:	19
Order Summary:	19
Payment page for an order:	20
Payment:	20
User Notification System:	22
• Notifications:	22
• Mailing System:	22
Messaging and support system:	23
Messages List:	23
User Profile Management:	24
User Update Profile:	25
Admin Dashboard:	25
Messaging :	27
My shop:	28

Edit item:	30
Mobile application:	32
User Interface and Experience	32
Design Principles	32
Navigation Overview	32
Screenshots and Explanations of Each Key Screen:	
Login page:	
The login page of the "HardWhere.ps" app presents a clean and user-friendly interface designed for straightforward user access. Here's a detailed description:	33
Home/Shop screen:	35
Search Functionality Screen	36
Product Detail Screen	38
Cart Screen	39
Support Chat	40
Account Settings	41
Technologies	43
Ionic Framework	43
Vue.js	43
TypeScript	44
Development Process with Ionic, Vue, and TypeScript	44
Features and Functionalities	45
Product exploring and Search Capability	45
User Account Management	45
Shopping Cart and Checkout Process	45
Real-time Communication with Support	45

## **Introduction:**

Hey there! Ever wonder what happens to old electronic stuff when we're done using them? That's where HardWhere.ps comes in! It's like a special place for computer engineering students to buy and sell used electronic parts. Imagine giving a second life to those gadgets you don't need anymore! This project is all about making tech life more sustainable and fun.

## **Abstract:**

HardWhere.ps is like a cool marketplace designed currently for students! Imagine a place where you can buy and sell used electronic parts easily. It's made with a powerful mix of technologies like Nuxt.js, Ionic with Vue, and a smart database called Supabase with PostgreSQL. The platform is super safe with encrypted passwords and special codes (JWT) for user logins. It's user-friendly too, with easy-to-use interfaces, a chat system, and a dashboard for managing stuff. This project is not just about buying and selling—it's about making a positive impact by reusing electronic parts and reducing waste.

## **Problem and theoretical background:**

HardWhere.ps addresses the community's need for a reliable platform to buy and sell used electronic parts. With a user-friendly interface, secure transactions, and real-time communication channels, the platform fosters a positive user experience. The inclusion of user-added products encourages community engagement, and the robust admin dashboard ensures efficient management of orders, communication, and product catalog integrity. Technologies like Nuxt.js, Ionic with Vue, DotNet with Clean Architecture, and effective image handling contribute to the platform's scalability, efficiency, and maintainability.

HardWhere.ps arises as a pioneering solution tailored for computer engineering students. This mobile platform serves as a unique marketplace, fostering the resale and reuse of used electronic parts. Specifically designed for computer engineering enthusiasts and students, HardWhere.ps aims to extend the life cycle of components, providing a sustainable avenue for trading and repurposing hardware. As computer engineering students navigate their academic journeys, this innovative platform empowers them to contribute to a circular economy, minimizing electronic waste while unlocking the full potential of used hardware projects components.

# Methodology:

## Technology Stack Selection:

- Select frameworks and tools that align with the goals of the platform, including Nuxt.js for the front end, Ionic with Vue for the mobile app, DotNet with Clean Architecture for the backend, and Supabase with PostgreSQL for the real-time database.

## Security Implementation:

- Develop a robust security framework, incorporating encryption for stored passwords in the database.
- Implement JWT for user authentication, ensuring secure token generation and validation during user interactions.

## Database Design:

- Design a relational database structure using PostgreSQL to effectively represent electronic components, user data, and transactional information.
- Utilize the Supabase platform for real-time capabilities, fostering collaboration and ensuring instant updates for users.

## Frontend Development:

- Implement the user interface using Nuxt.js for the web application and Ionic with Vue for the mobile app.
- Prioritize a user-friendly design with intuitive navigation, product cards, and search functionalities for an optimal browsing experience.

## Backend Development:

- Develop the backend using DotNet with Clean Architecture, ensuring modular and maintainable code.
- Implement APIs with versioning to accommodate future updates and enhancements.

Authentication and Authorization:

- Integrate JWT for secure token generation during user authentication.
- Establish role-based authorization policies to control user access to different features and resources.

### Admin Dashboard Implementation:

- Develop a comprehensive admin dashboard allowing efficient order management, real-time communication with users, and product editing capabilities.

### Messaging System Integration:

- Integrate a messaging system to enable user support and inquiries. Implement real-time notifications for users and admins, enhancing communication and engagement.

# Results And Discussion:

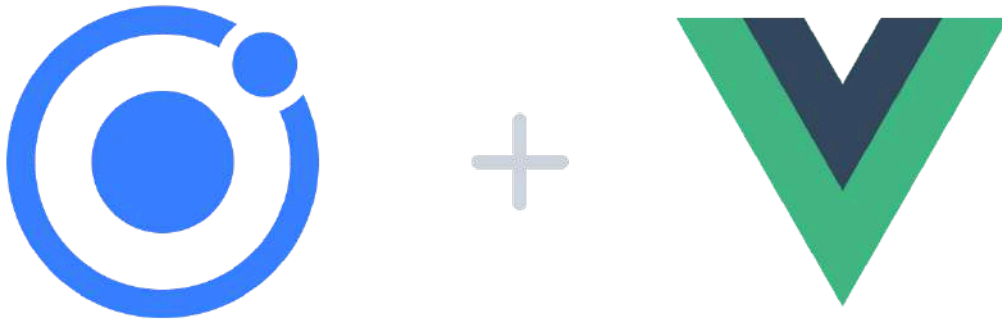
Technologies Used:

## Frontend:

- Nuxt.js:
  - Universal Web Applications: Nuxt.js enables the creation of universal web applications with server-side rendering (SSR), ensuring better performance and search engine optimization (SEO).
  - Automatic Code Splitting: The framework supports automatic code splitting, leading to optimized loading times and improved user experience.
  - Modular Development: Nuxt.js promotes modular development through its plugin and module system, allowing for easy integration of additional functionalities.

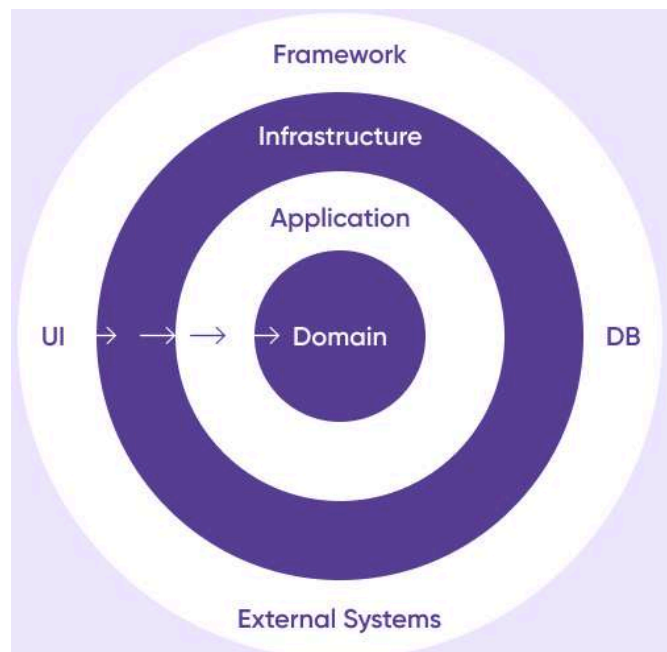
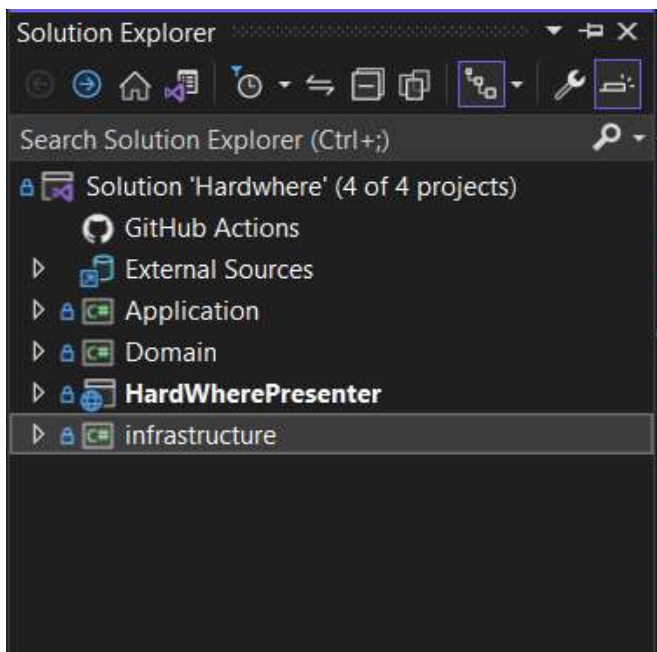


- Ionic with Vue:
  - Cross-Platform Development: Ionic, coupled with Vue, provides a cross-platform development environment, allowing for the creation of mobile applications compatible with various operating systems.
  - Pre-Built UI Components: Ionic comes with a library of pre-built UI components, ensuring a consistent and visually appealing user interface across different platforms.



## Backend:

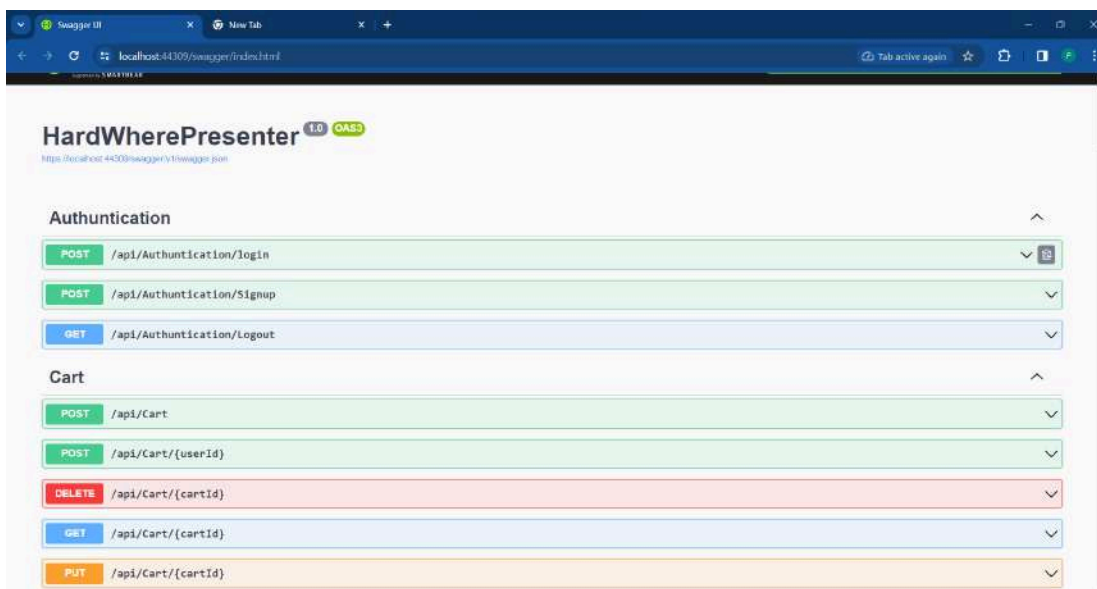
- DotNet with Clean Architecture:
  - Robust Backend: Using DotNet ensures a robust and scalable backend, capable of handling the complexities of an e-commerce platform efficiently.
- ORM:Entity FrameWork Core:
  - In the development of HardWhere.ps, the project benefits significantly from the integration of Entity Framework Core (EF Core).
  - Employing a code-first approach, EF Core allows for the intuitive definition of data models directly in C# code, fostering a clear and maintainable representation of the database schema within the application.
  - The seamless integration of LINQ with EF Core further streamlines the querying process, providing developers with a familiar and expressive syntax.



- Image Handling:
  - Static Path for Images in DotNet:
    - Performance Enhancement: Serving images through a static path in DotNet improves performance by directly delivering static assets without the need for dynamic generation.
    - Efficient Resource Handling: This approach ensures that images are efficiently loaded, contributing to a smoother user experience.

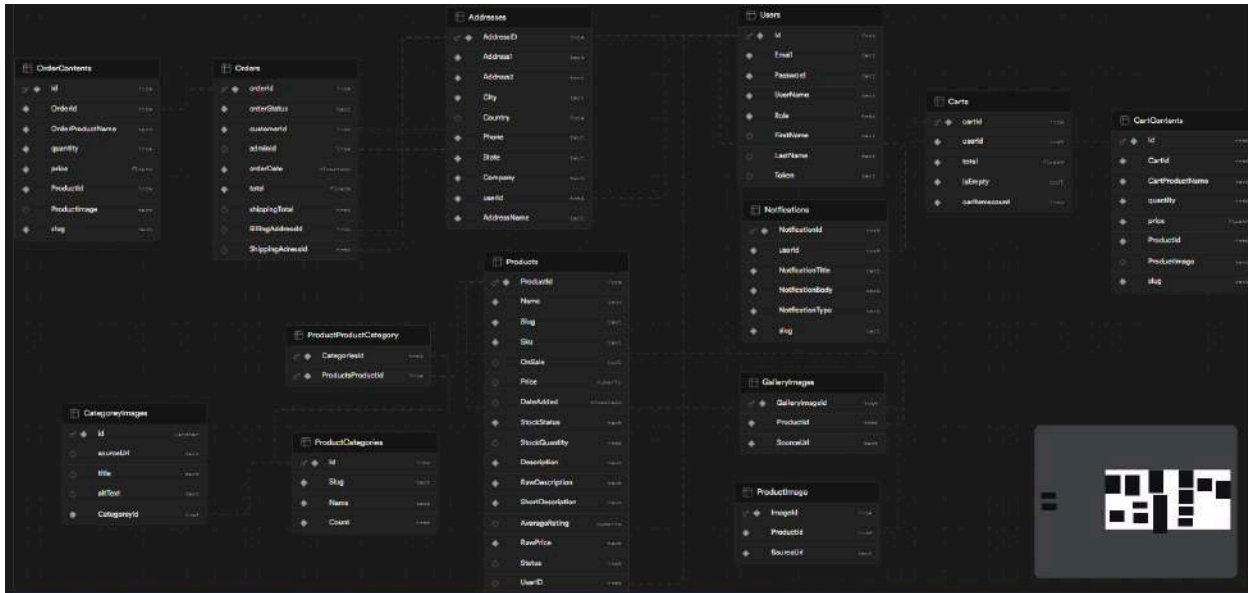
```
app.UseStaticFiles(  
    new StaticFileOptions()  
    {  
        FileProvider = new PhysicalFileProvider(  
            Path.Combine(Directory.GetCurrentDirectory(), @"Uploads")  
        ),  
        RequestPath = new PathString("/images")  
    }  
);
```

- API Documentation:
    - REST APIs Are introduced in the system, using different CRUD operations and methods of HTTP LIKE (PUT,POST,Patch ...)
- Swagger :



## Database:

In the foundation of HardWhere.ps lies a robust and reliable relational database, leveraging the capabilities of Supabase built on PostgreSQL. This choice underscores the commitment to data integrity, scalability, and efficient management within the platform. PostgreSQL.



## Real-Time Capabilities:

Supabase introduces real-time functionality, allowing instant updates and notifications when changes occur in the database.

Real-time features enhance collaboration by ensuring that changes made by collaborators are immediately reflected to all users.



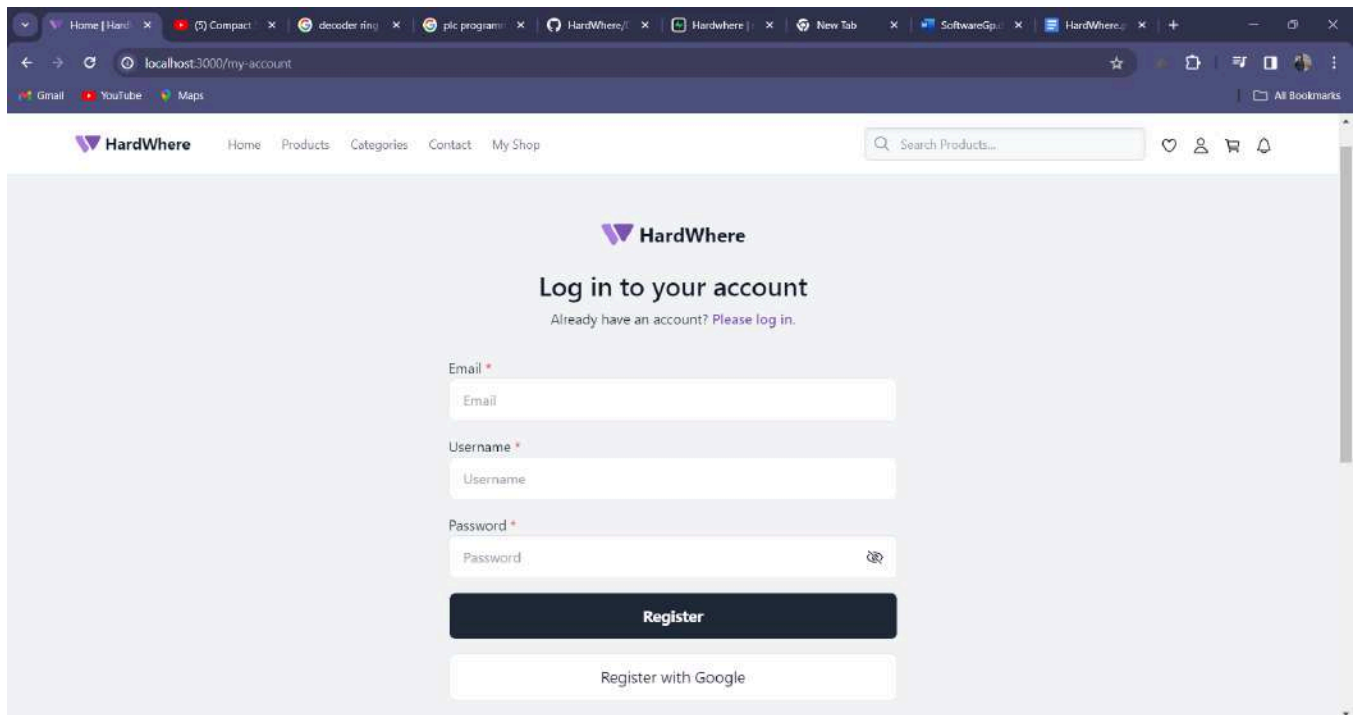
## Web Application Features:

- Authentication

HardWhere.ps prioritizes user security through a dual-layered approach. Utilizing JSON Web Tokens (JWT), the platform generates secure tokens during user authentication, ensuring verified access. User passwords are stored encrypted in the database, enhancing data protection

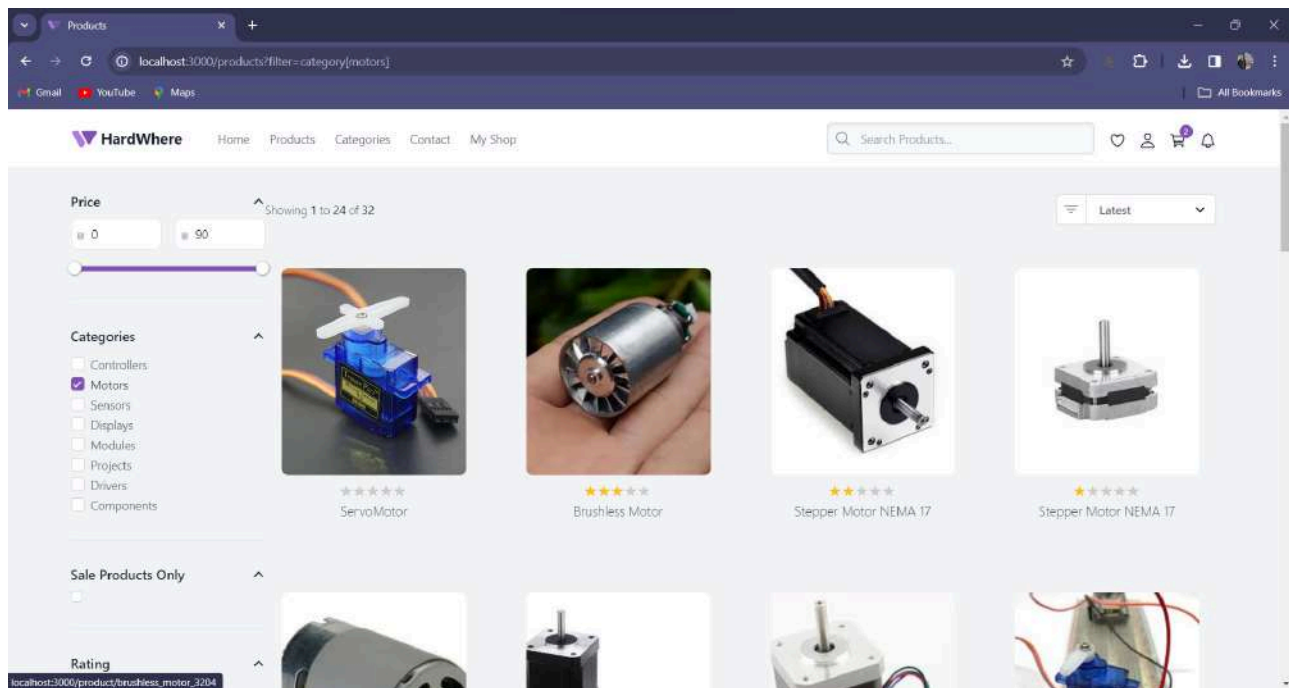
- Authorization

Authorization is meticulously managed based on roles and policies, enabling fine-grained control over user access to resources. This comprehensive authentication and authorization strategy forms the backbone of a secure and user-friendly experience within the platform.

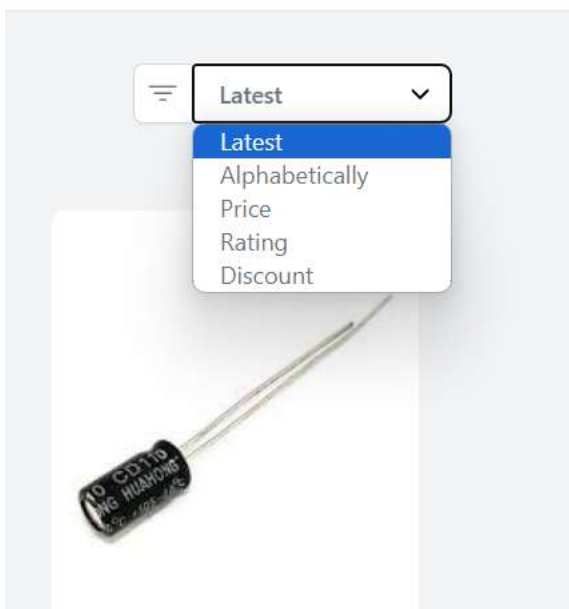


## User-Friendly Interface:

- The user interface is designed to be intuitive, with visually appealing product cards for easy navigation and engagement. It's a responsive design also



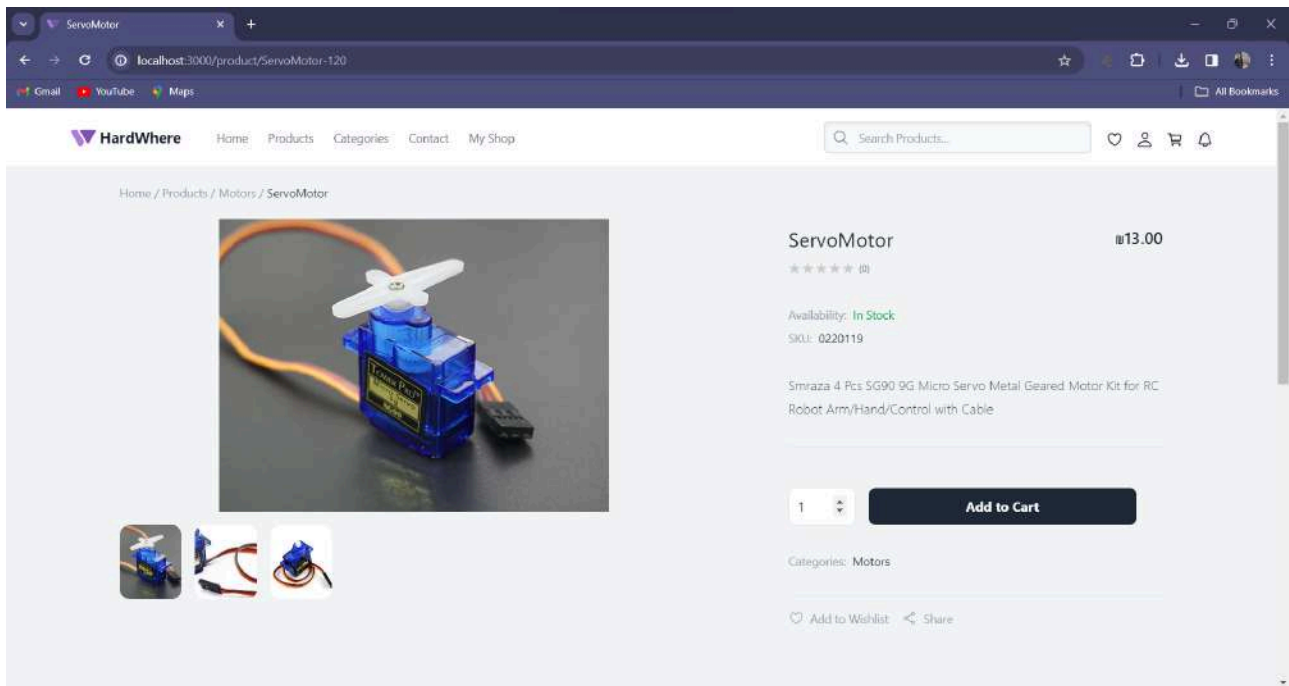
- Enhanced Search and Filter: Users can easily find products through advanced search functionality and various filters, including category, rating, and price



Here, You can order the contents based on their price, data added, name Des or ASC.

## Product Details

To show more details about the product you can click on any product to get something like this:



**Routing** introduced to that product to show exactly this product belongs to which category.

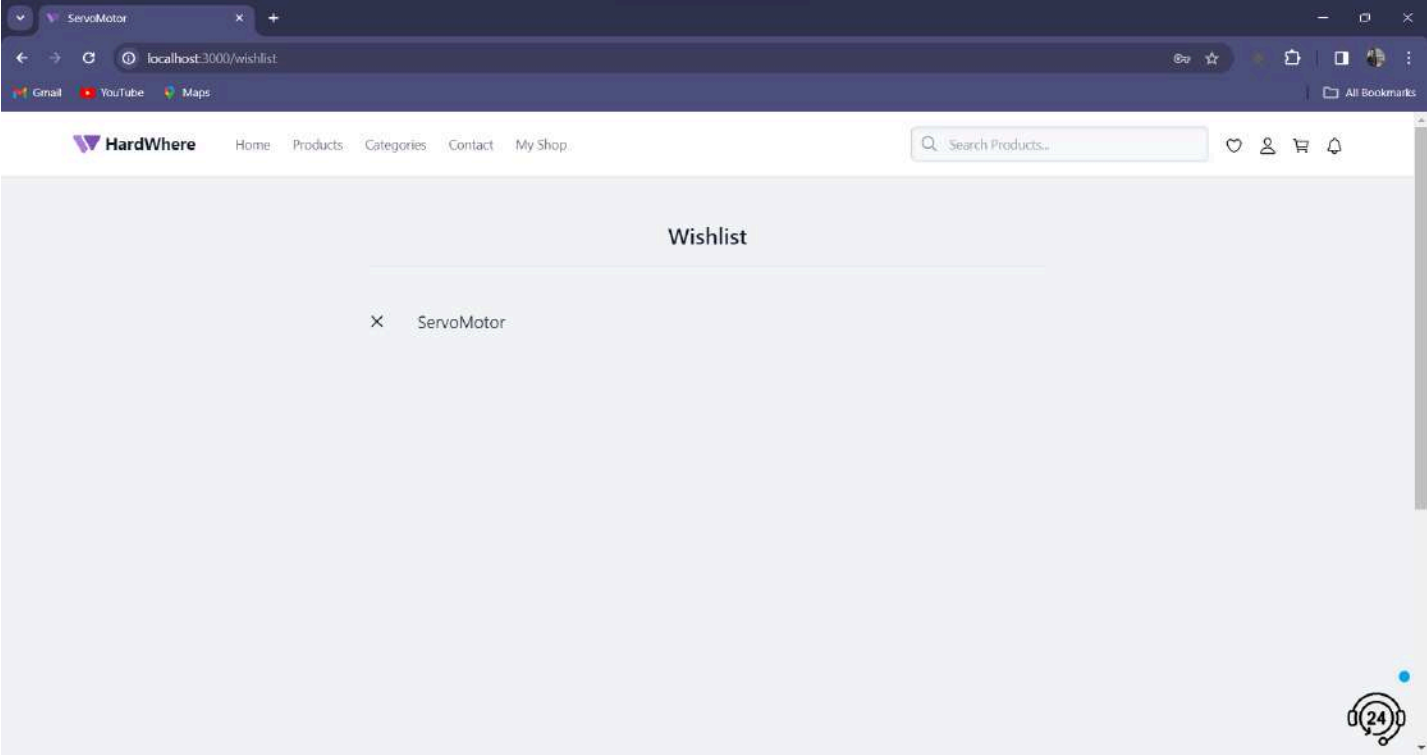
**Add To Wishlist:** you can add this product to the wishlist as a viewer.

**Availability:** here you can see if the product has any items in stock or it's finished.

**Quantity:** you can add more than an item from that product if there is any in stock.

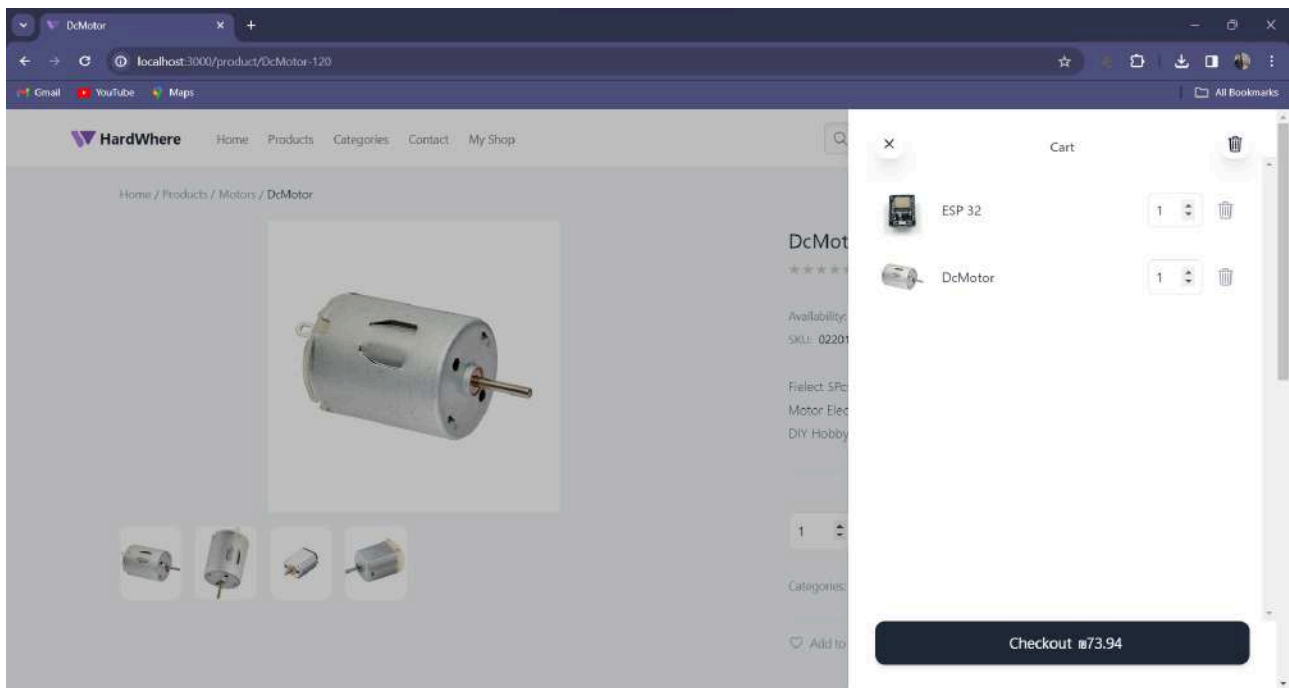
**Gallery images:** you can see the product images as a gallery.

# Wishlist:



Here you can view your items in the wishlist.

## Cart Screen:



- Here the cart is opened from the side when you click on it from the app header.
- You can directly update any item quantity from the cart side view, and delete items from it

The good point is the cart is shared between your website and mobile, so if you added items from the website and headed to your mobile app you will see those items also in your mobile app cart and that is called a realtime database.

## What about the viewer user Cart ?

No problem the viewer can store his data in the local storage and once he login to the system his cart will be stored to be shared among his other devices linked with that account

## Checkout Page:

The screenshot shows a web browser window with the URL localhost:3000/checkout. The page header includes the 'HardWhere' logo and navigation links for Home, Products, Categories, Contact, and My Shop. A search bar is present with the placeholder text 'Search Products...'. The main content area is divided into two sections: 'Billing Details' and 'Order Summary'.

**Billing Details:** This section contains several input fields for user information:

- EMAIL: Sami@gmail.com
- FIRST NAME: Sami
- LAST NAME: Jamal
- ADDRESS 1: 10street
- ADDRESS 2: Rafidea
- CITY: Nablus
- COUNTY: Nablus
- COUNTRY: Palestinian Territory
- PHONE (OPTIONAL): +970594391742

**Order Summary:** This section displays the items in the cart and the total amount:

- Item: DcMotor (quantity 1)
- Coupon Code: [input field] [Apply button]
- Subtotal: ₪25.00
- Shipping: + 10
- Total: ₪35
- [Checkout button]
- [Pay button]

A 24-hour support icon is visible in the bottom right corner of the page.

**Billing Details:** The user puts his details for the shipping and billing.

**Order Summary:** here the user can see the summary of his order before placing his order.

**Checkout:** it's better to name it place an order, this will place an order and the pay will be on site.

**Pay:** this is the payment and order way to ask for items.

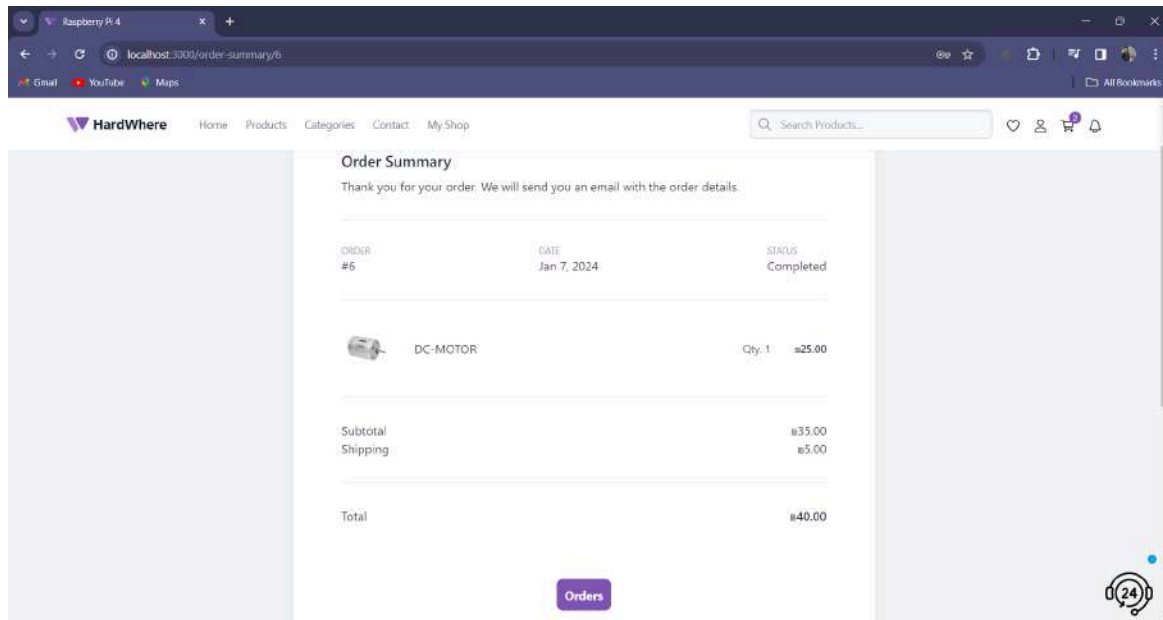
Payment is provided as a secured payment method using **stripe APIs** with .NET

If the user deletes the items the order summary data will be recalculated and no need to open that cart again for any edit you want to apply on items in your cart.

If the user has not had any orders before, his data won't be available and he needs to add it manually from here by filling the form, or from his settings in **MyAccount Tab**.

Any change happens here on billing details ,data can be updated automatically in the database to set for the last address used.

## Checkout or Place an order:



### Order Summary:

Here the user can see an order summary describing his order information.

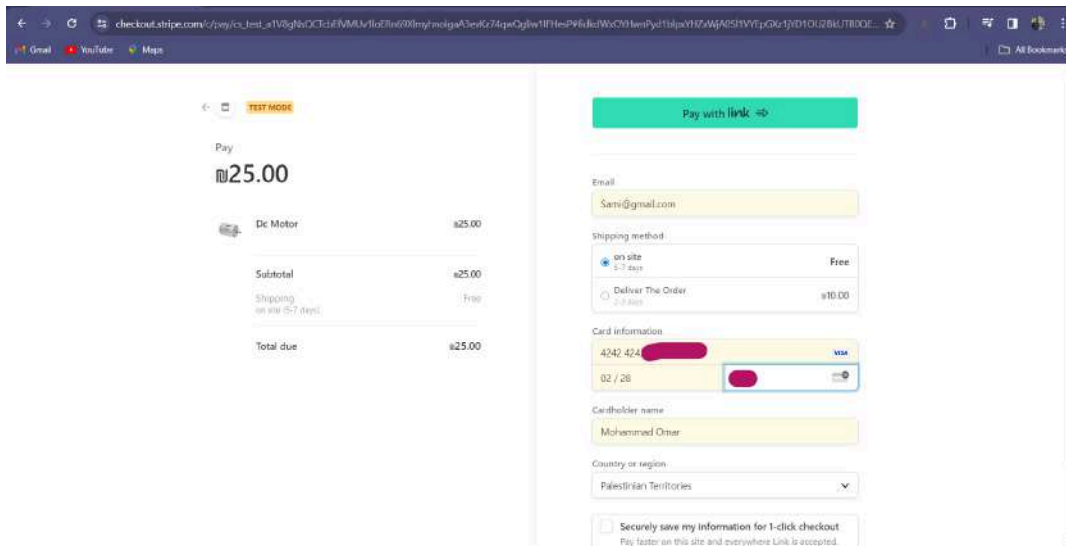
- Order Number that will be tracked to send updates on user order status.
- The data that shows when this order has been placed.
- The Status of the Order
  - The order has several states as (Pending, Processing, Shipped if it's ordered to a certain place and the final state is completed.
- The price and total of the order and each product in it.

The Orders button will take the user to the orders tab to see all of his orders.

Order	Date	Status	Total
6	January 7, 2024		35
7	January 7, 2024		50
8	January 7, 2024		75
9	January 7, 2024		35
11	January 8, 2024		73
12	January 14, 2024		23

By clicking on the Order id like (6), the user will be routed to the order summary page as shown above.

### Payment page for an order:



### Payment:

If the user hits the Pay option in the checkout page, this page will show up, a page from Stripe payment. **Stripe** is a payment services provider that lets merchants accept credit and debit cards or other payment methods.

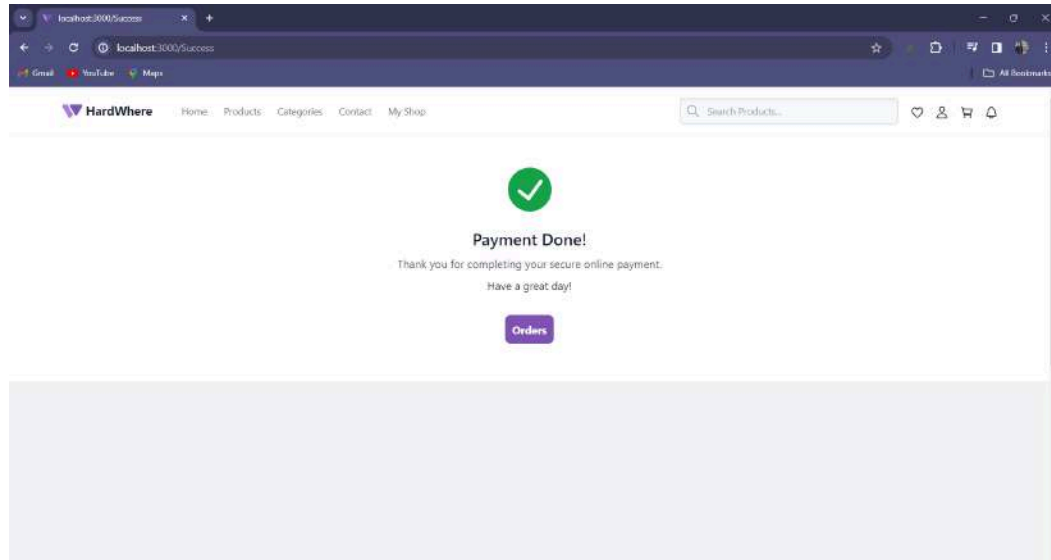
### Page overview:

- The total of the order will be changed based on the shipping method.
  - Shipping methods
    - On Site: that means no shipping, the user will take his order for the site of HardWhere.
    - Deliver the Order: a delivery operation will be made to the address he

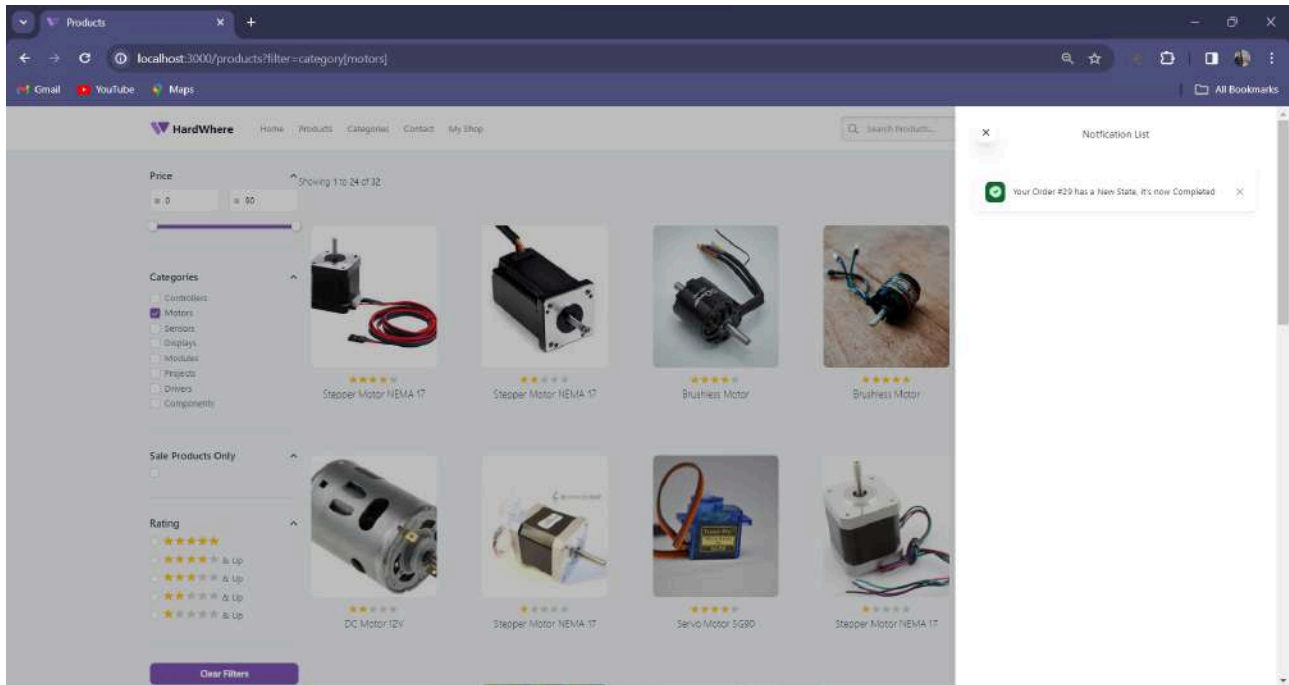
put in the shipping address form in the checkout page.

- After a successful payment operation. A new page will show up to tell the user that his payment is done. And an order will be placed also. This is a stripe dashboard view after payment is success

Amount	Payment method	Description	Customer	Date
₪25.00 ILS	Succeeded ✓	**** 4242	Sami@gmail.com	Jan 28, 11:20 PM



## User Notification System:

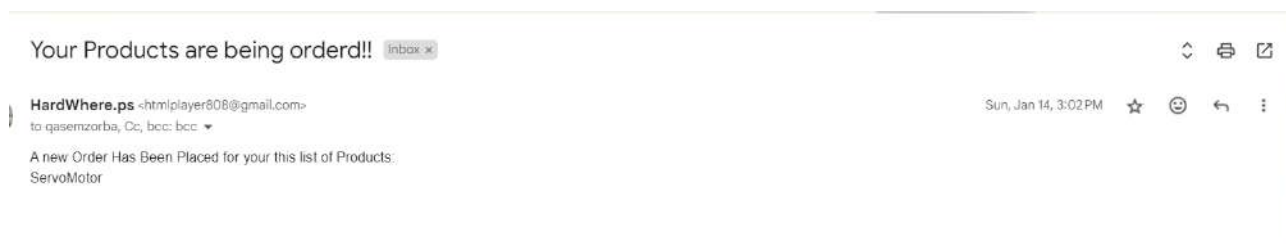


- **Notifications:**

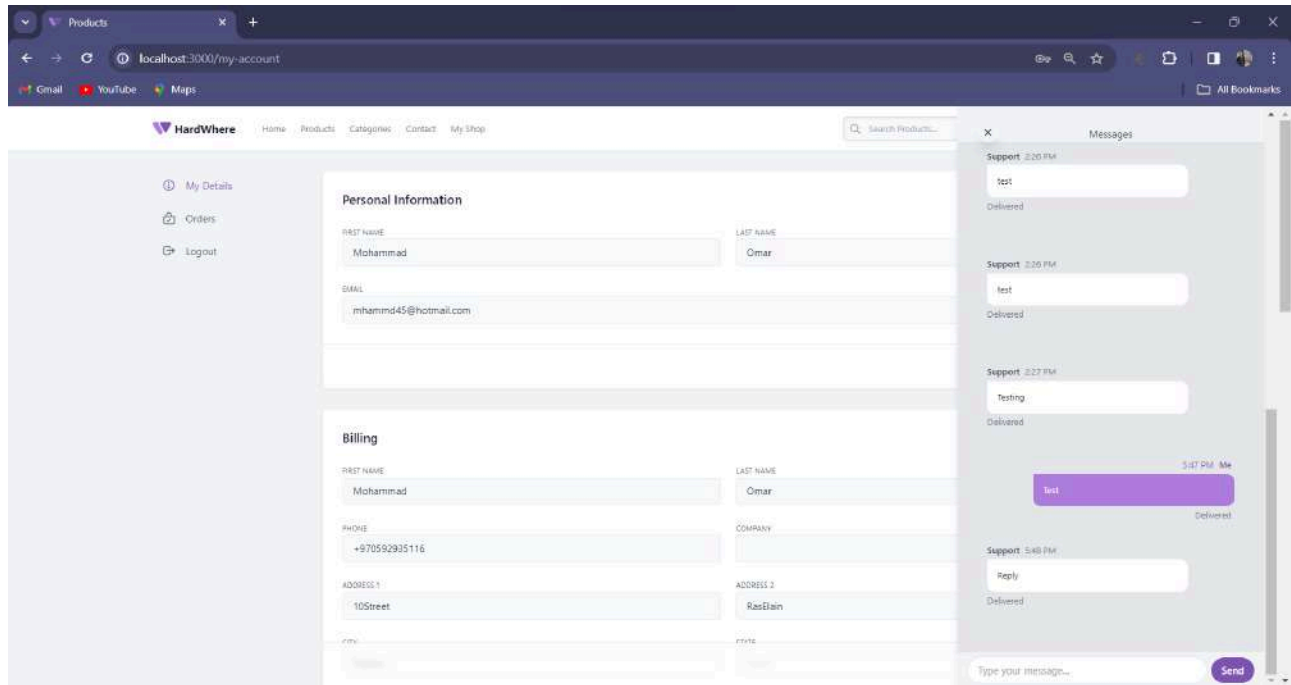
- It's a real time notifications service, built using SignalR Library in .NET.
- the user when placing an order, every update on his order will be shown here.
- Also a toast will be raised on the web to notify him without opening the notification list.
- Notifications for orders is divided into two types :
  - Notifications for the Owner
  - Notifications for the User who ordered that order.

- **Mailing System:**

- Not only notifications on the platform, also an email will be sent to both user types.
- We used a library called **MimeKit** with .NET to give best approach for building the email service.



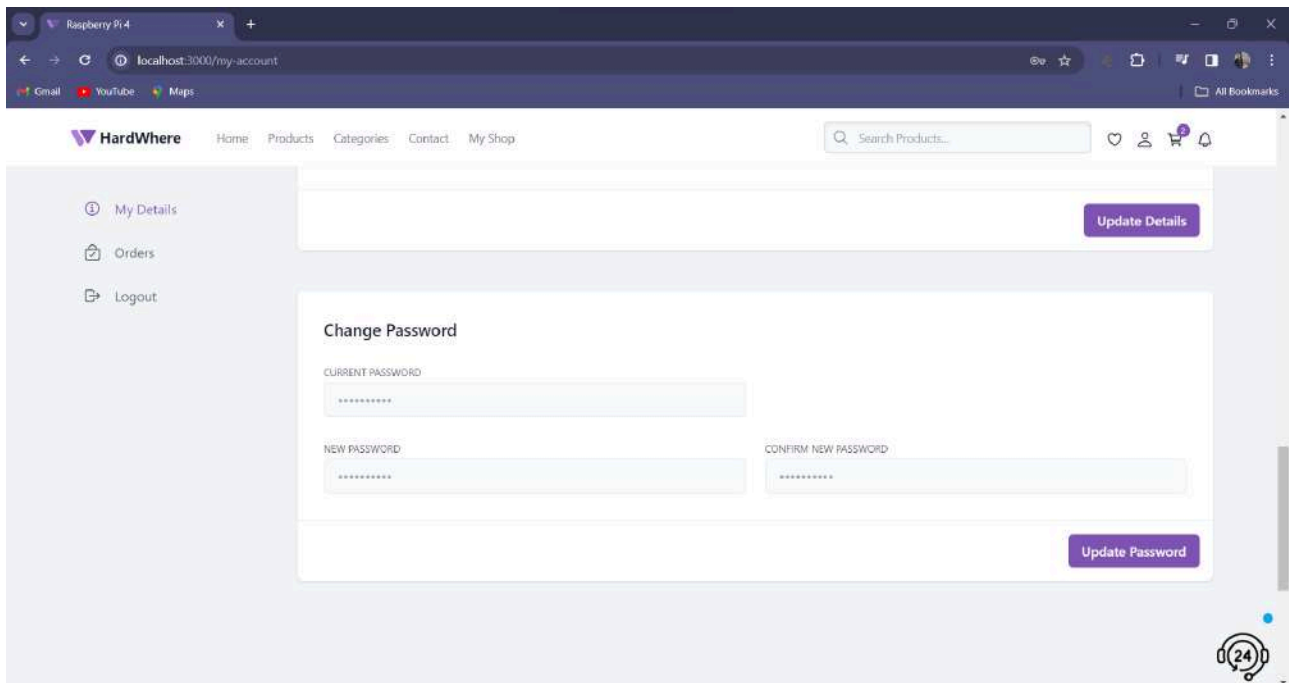
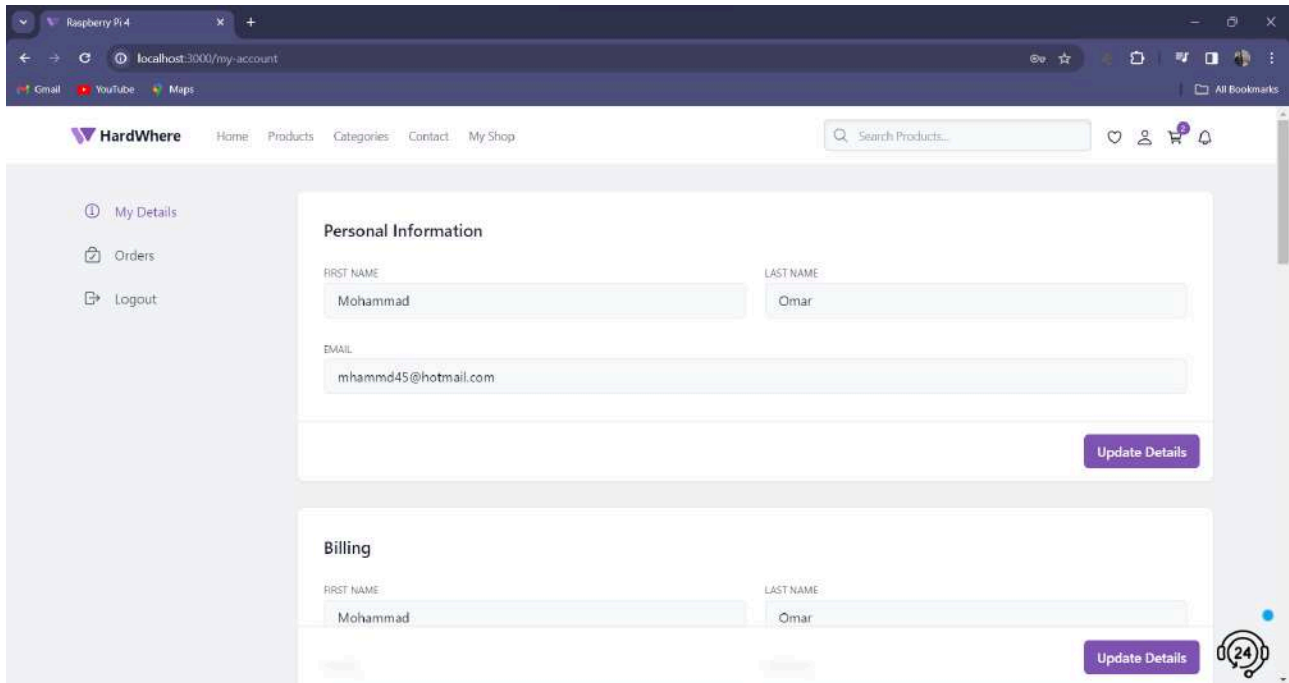
## Messaging and support system:



## Messages List:

- The Messages service is built using SignalR Library also, for introducing a real time chatting system.
- The user can send messages to the Support system.
- The user can scroll up in messages and see his older messages.
- The support system represents the Admin, who can access the admin dashboard and talk to users.

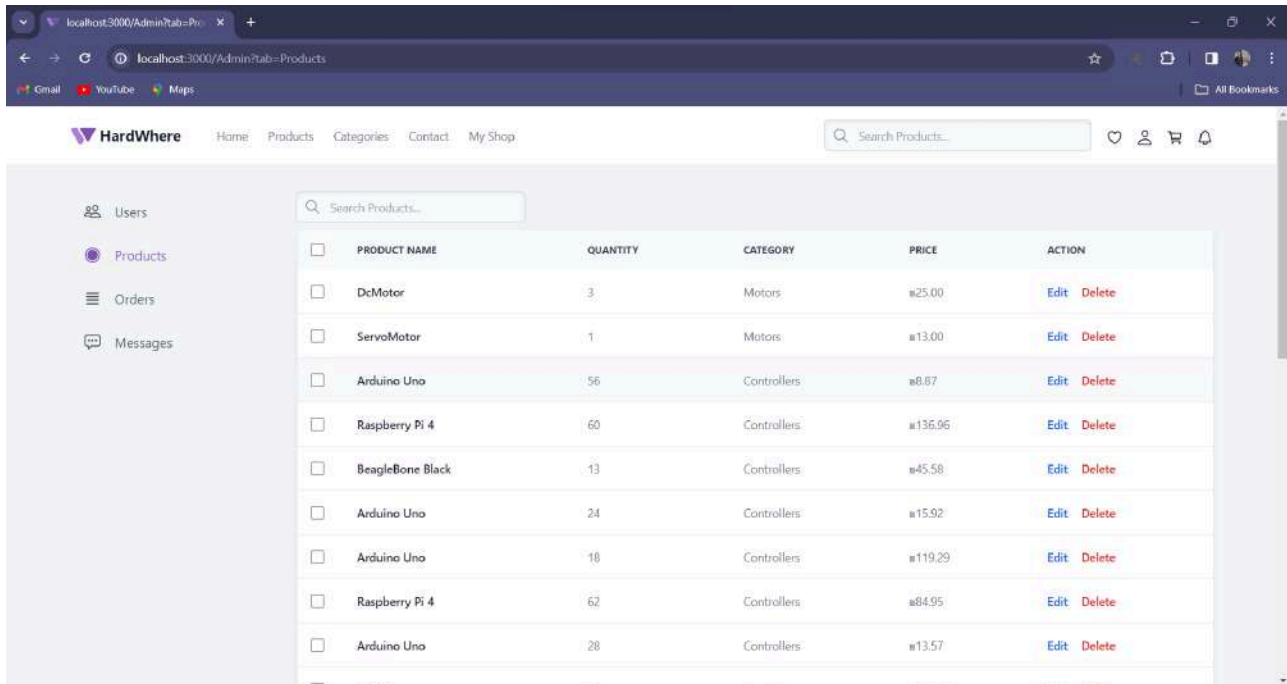
# User Profile Management:



## User Update Profile:

- The user can edit his informations like Personal Informations, Billing and Shipping and Password of the system by a simple user friendly interface as shown above.

## Admin Dashboard:



The screenshot shows the Admin Dashboard for the HardWhere application. The dashboard features a sidebar with navigation options: Users, Products, Orders, and Messages. The main content area displays a table of product listings. The table has the following columns: Product Name, Quantity, Category, Price, and Action. The data rows are as follows:

PRODUCT NAME	QUANTITY	CATEGORY	PRICE	ACTION
DcMotor	3	Motors	₺25.00	Edit Delete
ServoMotor	1	Motors	₺13.00	Edit Delete
Arduino Uno	56	Controllers	₺8.67	Edit Delete
Raspberry Pi 4	60	Controllers	₺136.96	Edit Delete
BeagleBone Black	13	Controllers	₺45.58	Edit Delete
Arduino Uno	24	Controllers	₺15.92	Edit Delete
Arduino Uno	18	Controllers	₺119.29	Edit Delete
Raspberry Pi 4	62	Controllers	₺84.95	Edit Delete
Arduino Uno	28	Controllers	₺13.57	Edit Delete

This screen presents an administrative interface for managing product listings in the HardWhere.ps application. The layout is a structured table format, comprising several columns for product data:

- **Product Name:** Displays names of items, such as DcMotor, ServoMotor, Arduino Uno, etc.
- **Quantity:** Lists the available stock quantity for each item.
- **Category:** Categorizes products under labels like Motors or Controllers.
- **Price:** Shows the price for each product in the local currency.
- **Action:** Provides "Edit" and "Delete" options, allowing the admin to modify or remove product listings.

On the left sidebar, there are navigation options for different admin panel sections such as Users, Products, Orders, and Messages. The top search bar allows filtering or searching the product list. This interface is crucial for inventory management and maintaining the product catalog.



<input type="checkbox"/>	ORDER ID	ORDER DATE	USER NAME	ORDER STATUS	CHANGE STATUS
<input type="checkbox"/>	13	27/01/2024	Sami Jamal	Completed	ChangeStatus ▾
<input type="checkbox"/>	16	27/01/2024	Sami Jamal	Completed	ChangeStatus ▾
<input type="checkbox"/>	27	28/01/2024	Sami Jamal	Completed	ChangeStatus ▾ ChangeStatus Pending Completed Processing Shipped

Showing 1- of 4 5 Next

The Orders tab is focused on transaction management and shows a list of orders with key details:

- **Order ID:** Unique identifier for each order.
- **Order Date:** The date when each order was placed.
- **User Name:** The name of the customer who placed the order.
- **Order Status:** Current status of the order, with an option to change it via a dropdown menu (ChangeStatus), showing options like Pending, Completed, Processing, and Shipped.

This screen is essential for tracking order processing, customer service, and overall sales management.

## Messaging :

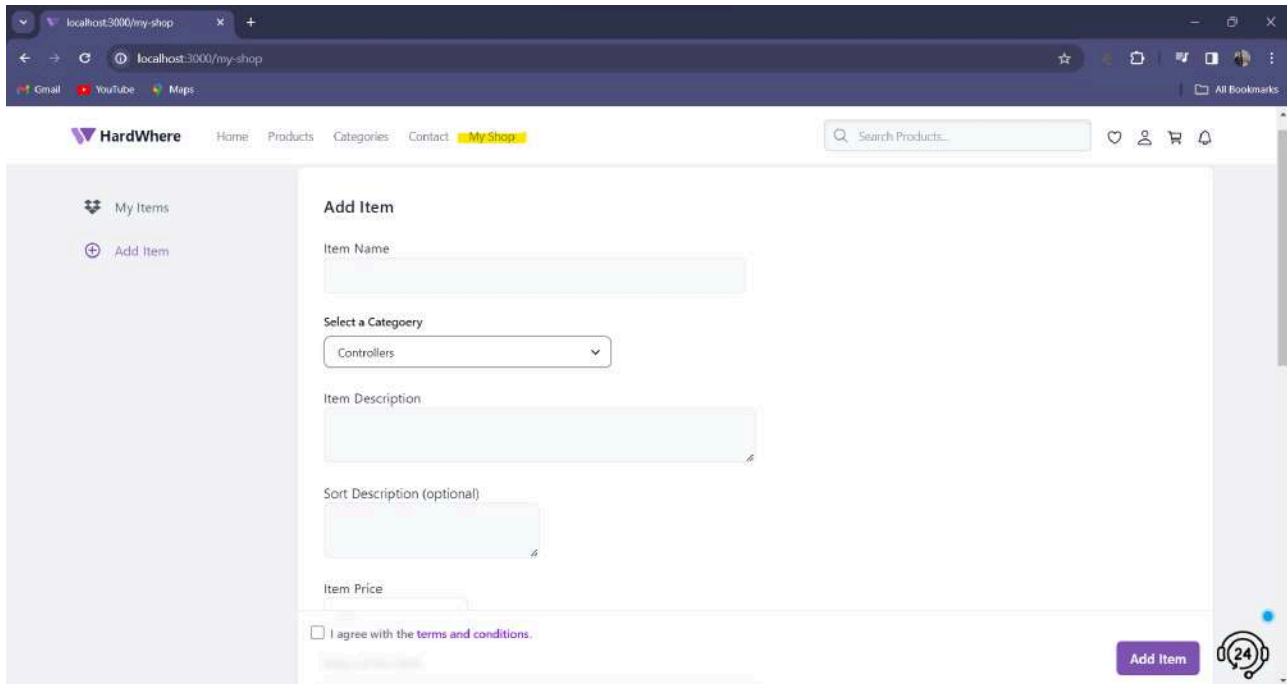


The Messages section of the admin panel appears to be a communication hub where the admin can interact with users of the platform:

- **Sidebar:** Shows a list of conversation threads with different users.
- **Chat Interface:** A standard messaging layout with speech bubbles showing the conversation between the admin and a selected user (Mohammad Omar in this case).
- **Input and Send:** At the bottom is a text input field for typing messages and a "Send" button to dispatch messages.

The chat interface indicates message status (e.g., "Delivered"), and time stamps for when messages were sent or received, providing an efficient way for the admin to handle inquiries and support requests directly within the admin panel.

## My shop:



The screenshot shows a web browser window with the URL `localhost:3000/my-shop`. The page title is "HardWhere" and the navigation menu includes "Home", "Products", "Categories", "Contact", and "My Shop". A search bar is located at the top right. The main content area is titled "Add Item" and contains the following form fields:

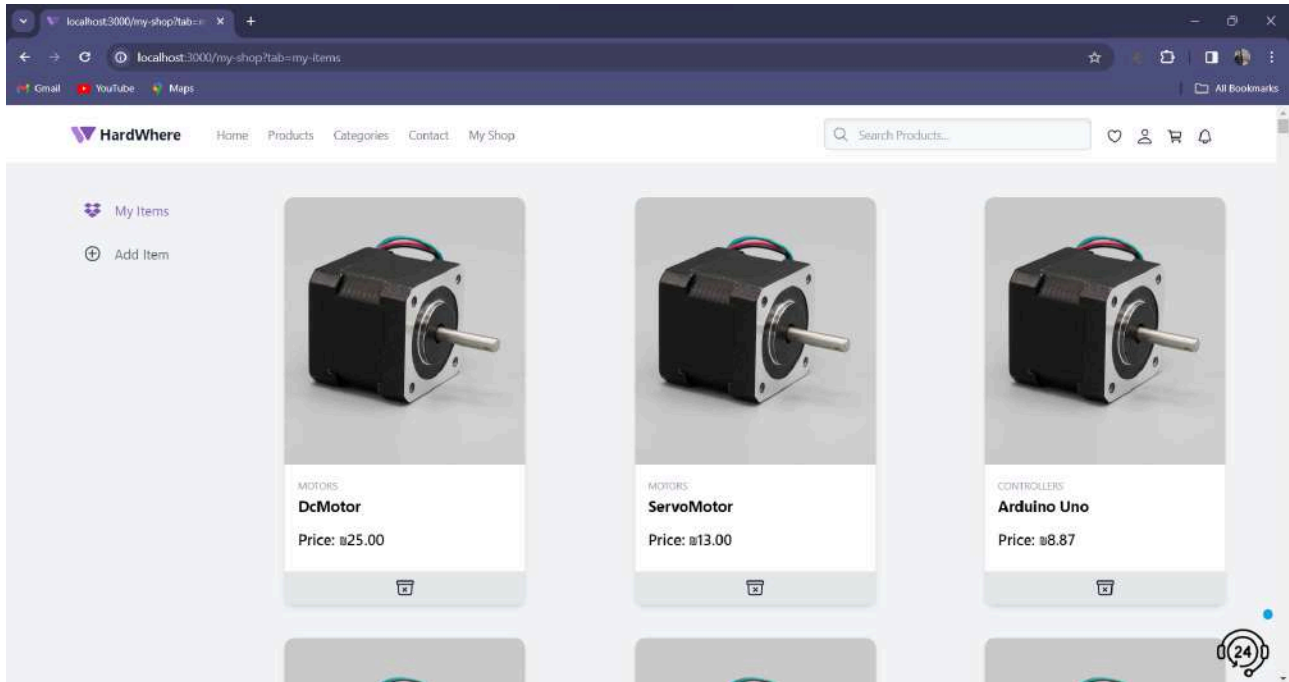
- Item Name:** A text input field.
- Select a Category:** A dropdown menu with "Controllers" selected.
- Item Description:** A large text area for a detailed description.
- Sort Description (optional):** A smaller text area for a concise description.
- Item Price:** A text input field.

At the bottom of the form, there is a checkbox labeled "I agree with the terms and conditions." and a purple "Add Item" button. A customer support icon (headset with "24") is visible in the bottom right corner.

This screen provides a user interface within the "My Shop" section for an individual user to list a new item for sale. The form includes several fields for item details:

- **Item Name:** A text box for entering the name of the item.
- **Select a Category:** A dropdown menu to classify the item into predefined categories, such as Controllers.
- **Item Description:** A text box for a detailed description of the item.
- **Sort Description (optional):** An additional text box for a shorter, concise description.
- **Item Price:** A field to enter the price of the item.
- **Photos:** two fields to upload photos, one for thumbnail and the other for gallery.

At the bottom, there is a checkbox for agreeing to the terms and conditions, which is a standard requirement before an item can be listed. A button labeled "Add Item" allows the user to submit the form and list their new product.



This screen serves as a user's personal product management dashboard within "My Shop" on the website. It provides an overview of the items they have listed for sale. Each product is presented on individual cards, displaying key information at a glance:

- **Category:** Labeled on top of the card to categorize the item, such as "MOTORS" or "CONTROLLERS".
- **Item Name:** The specific product name is listed, for example, "DcMotor" or "Arduino Uno".
- **Price:** Each card shows the price of the item.

The user can interact with this interface in two ways:

**Deleting an Item:** At the bottom of each card is a trash can icon, indicating the functionality to delete the item from the listing.

**Editing Product Details:** By clicking on the card itself, the user is presumably taken to a separate page where they can edit the product's details, allowing for updates to the description, price, or other attributes.

This interface is designed to provide the user with straightforward control over their product listings, making the management of their online shop inventory intuitive and efficient.

# Edit item:

The screenshot shows a web browser window with the URL `localhost:3000/editItem/14`. The page title is "Edit Item" and the breadcrumb is "Edit Item". The form contains the following fields:

- Item Name:** A text input field containing "DcMotor".
- Item Description:** A text area containing "Fielect 5Pcs DC Motor 1.5-6v 5000-20000RPM Micro Motor Mini Motor Electric Motor Round Shaft for RC Boat DIY Car Toys Model".
- Sort Description (optional):** A text input field containing "Short Description For Motor".
- Item Price:** A text input field containing "0.0" and a label "₹25.00" below it.
- State of the Item:** A dropdown menu.

An "Add Item" button is located at the bottom right of the form.

The screenshot shows a web browser window with the URL `localhost:3000/editItem/15`. The page title is "Edit Item" and the breadcrumb is "Edit Item". The form contains the following fields:

- Main Image:** A "Choose File" button with the text "No file chosen".
- Images:** A "Choose Files" button with the text "No file chosen" and a note "\*choose at most 6".

A preview image of a blue servo motor is displayed below the image upload fields. An "Update Photos" button is located at the bottom right of the form.

The edit page is an interface that allows a user to modify the details of a product they have listed for sale. Here's an overview of the functionalities provided by the page::

- **Item Name:** A field where the user can update the name of the product.
- **Item Description:** This text area contains a detailed description of the product, mentioning specifics such as voltage range.
- **Sort Description (optional):** An additional field for a concise description, making it easier for customers to quickly understand what the product is about.
- **Item Price:** A numeric field where the user can set or change the price of the item.
- **State of the Item:** A dropdown menu likely provides options to indicate the condition of the product (new, used).
- **Main Image:** A button to upload or change the primary image that appears in product listings.
- **Images:** Options to upload additional images, with a note indicating a limit of six. These likely offer multiple views or details of the product.
- **Update Photos:** A button to finalize the uploading of new images or changes to existing ones.

The current main image of the product, a "ServoMotor," is displayed at the bottom.

This page makes up a comprehensive edit interface, enabling the user to maintain the accuracy and appeal of their product listings by updating vital information and visual presentation.

Mobile application:

## **User Interface and Experience**

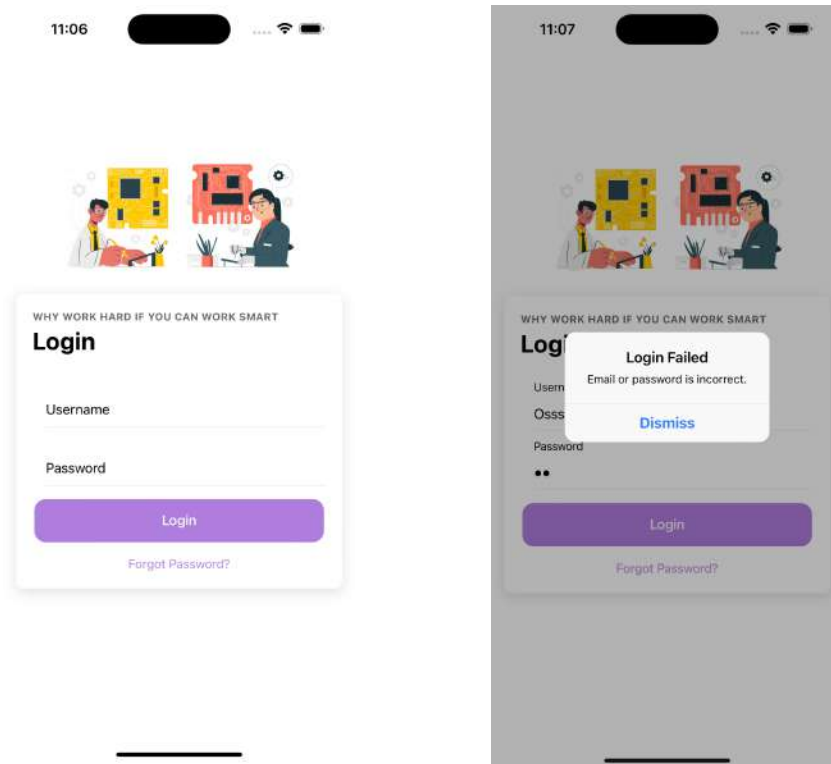
### **Design Principles**

The interface of HardWhere.ps adheres to the principles of simplicity, intuitiveness, and consistency, ensuring that users can navigate the app with ease. Aesthetically, the app employs a minimalist design with a focus on readability and ease of use. The design is responsive, ensuring that the app is accessible across a variety of devices and screen sizes.

### **Navigation Overview**

Navigation through the app is streamlined with a bottom navigation bar that provides access to the primary sections: Shop, Search, Cart, Help, and Settings. This allows for quick switching between different parts of the app without losing context. Additionally, breadcrumbs and clear back-button functionality ensure that users can trace their steps within more complex navigation paths.

## Screenshots and Explanations of Each Key Screen: Login page:



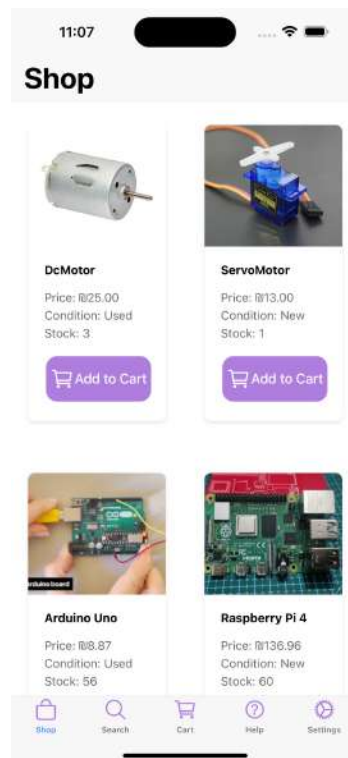
The login page of the "HardWhere.ps" app presents a clean and user-friendly interface designed for straightforward user access. Here's a detailed description:

- **Top Section:** The top section features a bright graphic that aligns with the theme of the application. It depicts circuit boards and a person working on a laptop, which reinforces the app's focus on electronics and hardware.
- **Slogan:** Below the graphic, a motivational slogan, "WHY WORK HARD IF YOU CAN WORK SMART," encourages users to consider the efficiency that the app offers.
- **Login Form:** The central part of the page is dedicated to the login form. It has two fields:
  - **Username:** A field where users input their unique identifier, which could be their email or a chosen username.
  - **Password:** A secure field for users to enter their password, ensuring privacy and security.
- **Action Button:** Below the input fields is a prominent 'Login' button in a contrasting color, making it clear where to tap to proceed.

- **Password Recovery:** There's a thoughtful inclusion of a 'Forgot Password?' link for users who need to reset their password, indicating an understanding of user needs and common issues.
- **Overall Aesthetics:** The page's design is minimalistic, with a focus on functionality and ease of use. The color scheme is balanced, with the use of white space enhancing readability and focus on the login form.

This page sets a professional and inviting tone for users as they begin their interaction with the HardWhere.ps application.

## Home/Shop screen:

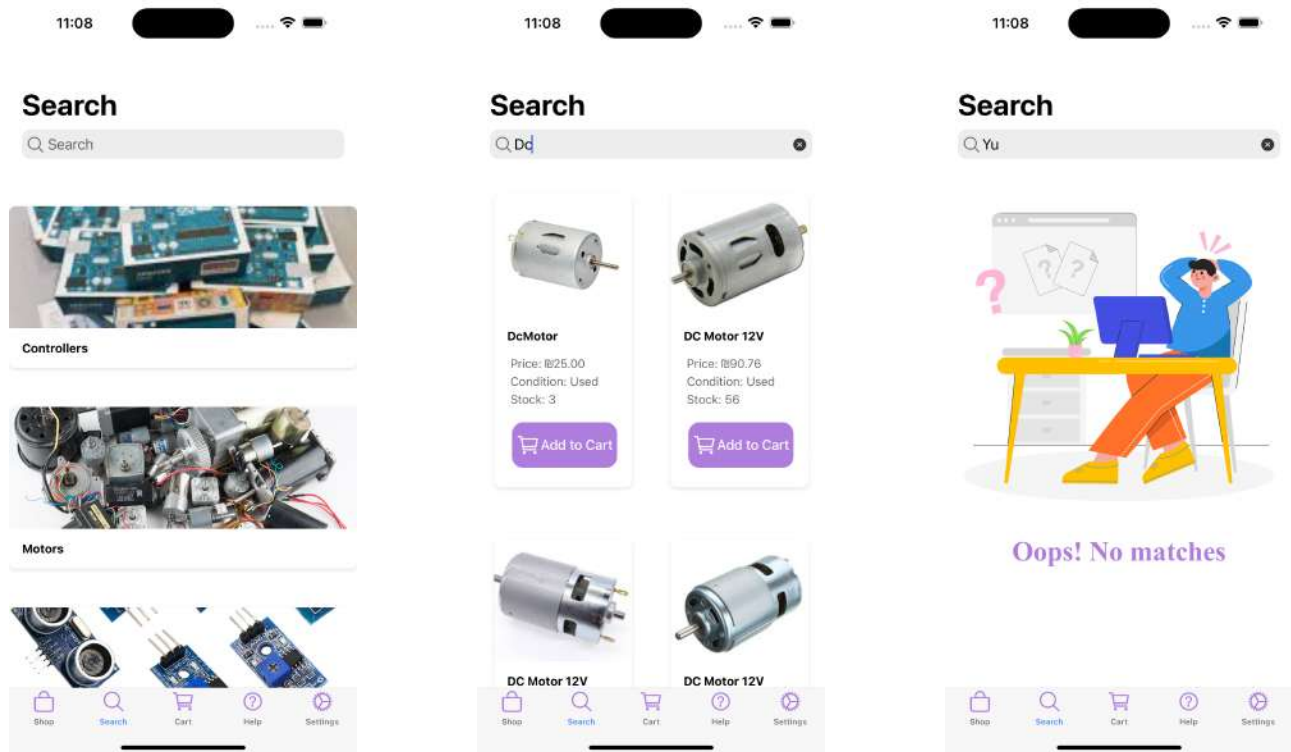


This screen appears to be the "Shop" section of the HardWhere.ps mobile application, where various electronic components are listed for sale. Here's a detailed breakdown of the page:

- **Product Listings:** There are four distinct products displayed, each with a photo and a brief description:
  - **DC Motor:** The first product is a DC Motor, labeled as used, priced at ₱25.00, and there are 3 units in stock. There is a purple "Add to Cart" button below the description, suggesting a simple, one-tap action to purchase the item.
  - **Servo Motor:** Next, there is a Servo Motor, new condition, priced at ₱13.00, with 1 unit in stock. Similar to the DC Motor, it has an "Add to Cart" button for easy purchasing.
- **Aesthetics and Functionality:** The page has a clean layout, with a white background that makes the images and text easy to view. The consistent design of the "Add to Cart" buttons and the price and stock information suggests a user-friendly interface focused on shopping convenience.

- **Imagery and Descriptions:** The images are clear and appear to be well-sized for the mobile format, and the text descriptions are concise yet informative, giving users enough information at a glance.

## Search Functionality Screen



This screen is part of the "Search" functionality within the HardWhere.ps mobile application, which operates dynamically based on user input. Here's a description of its behavior:

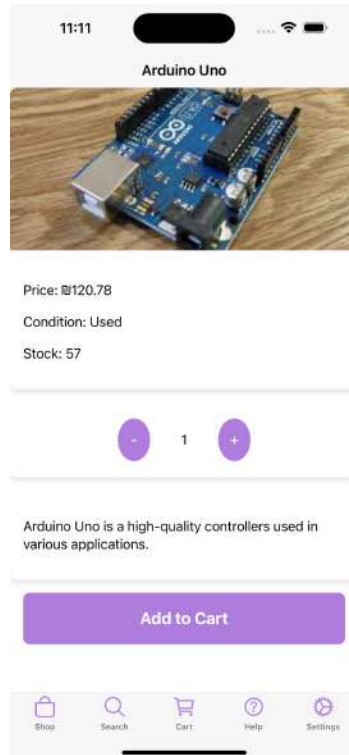
- **Initial State (Empty Search Bar):** When the search bar is empty, the app cleverly defaults to displaying categories of products available for purchase. In the provided screenshot, we can see two categories highlighted:
  - **Controllers:** A visual collection of various controller boards, likely Arduino or similar microcontroller platforms.
  - **Motors:** An assortment of different types of motors, suggesting a variety for users to browse through.

This initial state serves as a prompt for users to explore the items within these categories, enhancing the browsing experience and potentially encouraging discovery of items they might not have specifically searched for.

- **Active Search (Non-empty Search Bar):** When a user types into the search bar, the app dynamically updates to show the results. This can lead to two outcomes:
  - **No Matches:** If the search query does not match any products, the app displays an "Oops! No matches" screen, which includes a graphic of a person at a desk looking perplexed, signifying the lack of results. This friendly and illustrative feedback informs users that their search did not yield any results, and they may need to try different keywords or check the spelling.
  - **Matches Found:** If there are matches, the app will display the relevant products that fit the search criteria, allowing the user to view the results and select any item of interest to learn more or to add to their cart.

The "Search" page adapts to user interaction, making it a dynamic and responsive feature of the app, aimed at enhancing user experience by either guiding them through the product categories or directly showing the search results.

## Product Detail Screen

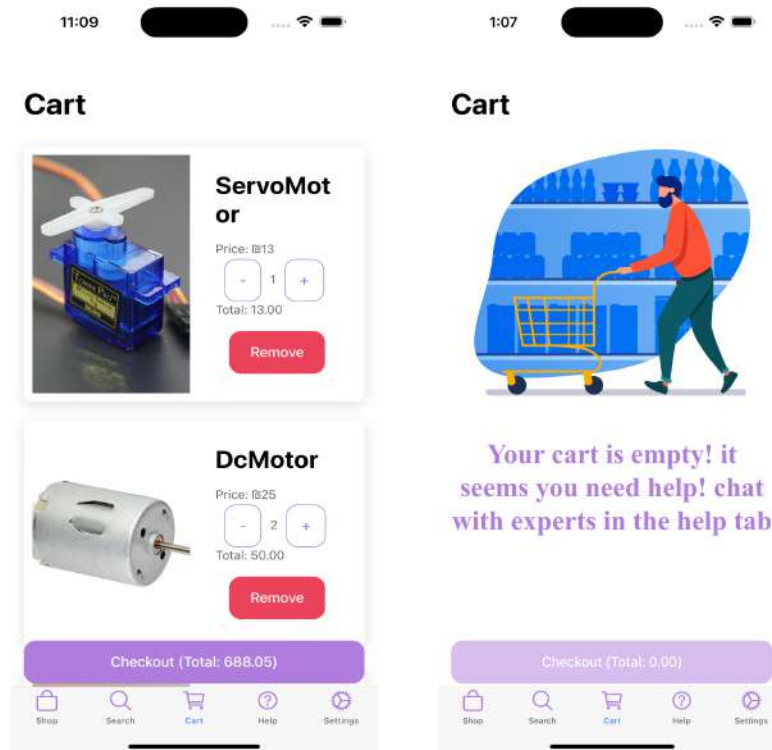


This screen is a product detail page within the HardWhere.ps mobile application, for example this an Arduino Uno board. Here's the breakdown of the elements on this page:

- **Product Image:** A clear, focused image of product takes prominence at the top, giving the user a good view of the item's condition and components.
- **Product Name:** Below the image, the name of the product is displayed, identifying the product.
- **Price and Details:** The price is displayed prominently. Below the price, the condition of the product which is used or new and the available stock of this product, providing the user with important purchasing information.
- **Quantity Selector:** There is a quantity selector with a minus "-" and plus "+" button, allowing the user to select the number of units they wish to purchase. The number in the middle indicates the currently selected quantity, which is set to 1 by default.
- **Product Description:** Beneath the quantity selector, there's a brief description for the product. This gives potential buyers a quick insight into the product's usage and quality.
- **Add to Cart Button:** A large purple "Add to Cart" button is positioned at the bottom of the product details, which suggests a call to action for the user to proceed with the purchase.

This product detail page is designed to provide all the necessary information a user needs to make an informed purchase decision, with a straightforward and clean interface that emphasizes clarity and ease of use.

## Cart Screen



These two screens represent the shopping cart interface within the HardWhere.ps mobile application.

### First Screen - Cart with Items:

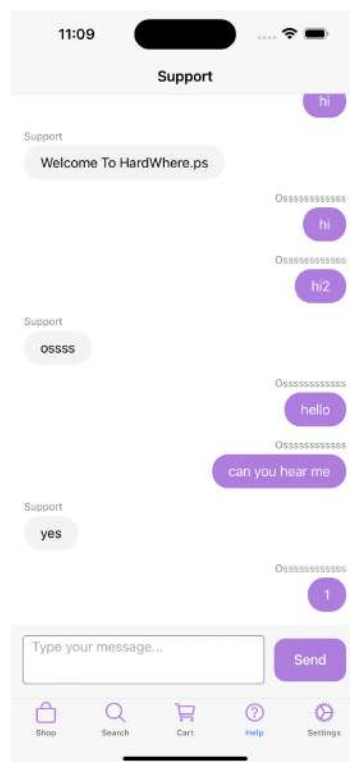
- **Product Listings:** for example two products out of many are shown in the cart:
  - **Servo Motor:** The first item is a Servo Motor, priced at ₱13.00, with a quantity selection interface allowing the user to adjust the quantity right from the cart. There's also a "Remove" button in red, suggesting an option to delete this item from the cart if the user changes their mind.
  - **DC Motor:** The second item is a DC Motor, with a total price of ₱50.00 for 2 units. It similarly has quantity controls and a "Remove" button.
- **Checkout Button:** At the bottom, there's a prominent "Checkout" button with the total cost of ₱688.05, calling the user to proceed to the payment process.

## Second Screen - Empty Cart:

- **Empty State:** When the cart is empty, instead of listing products, the app shows an illustration of a person with a shopping cart, accompanied by the message "Your cart is empty! It seems you need help! Chat with us in the help tab" This friendly nudge encourages users to return to the shop to add items.
- **Checkout Button:** Despite the empty cart, the checkout button is still visible, but it displays a total of ₦0.00, indicating there's nothing currently to check out.

Both screens maintain consistency with the app's overall design language and provide clear cues for user action, whether that's modifying the cart contents or prompting to fill an empty cart.

## Support Chat



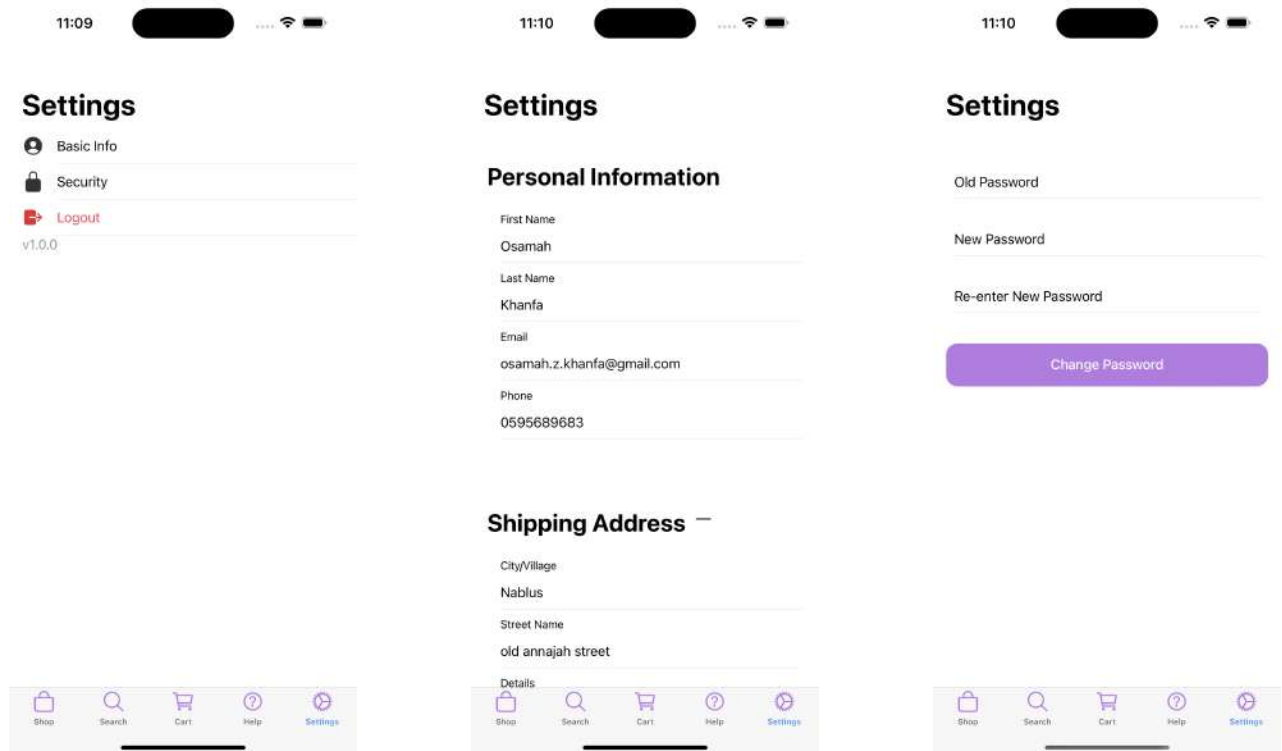
This screen is the support chat interface within the HardWhere.ps mobile application, designed for real-time communication between the user and customer support experts.

- **Chat Window:** The main part of the screen shows a chat window where the conversation takes place. It displays messages from both the user and the support representative.

- The user has initiated the conversation with greetings like "hi" and "hello" and a query "can you hear me," indicating they are seeking acknowledgement from the support team.
- The support representative responds with "Welcome To HardWhere.ps," which is a standard greeting message, and confirms their presence with a "yes" to the user's inquiry.
- **User Input Field:** At the bottom, there is an input field with the placeholder "Type your message..." where the user can type their questions or responses.
- **Send Button:** Adjacent to the input field is a "Send" button in a distinct purple color, which the user can tap to dispatch their message to the support team.

This support chat feature is crucial for e-commerce platforms as it provides immediate assistance to users, helping them with product inquiries, technical issues, or guidance through the buying process. The interface here is intuitive and familiar, mirroring common messaging applications, which helps reduce any learning curve and facilitates easy communication.

## Account Settings



The screens provided are part of the "Settings" functionality within the HardWhere.ps mobile application, detailing user account management and security features.

### **First Screen - Settings Tab:**

- **Header:** The screen is headlined with "Settings," clearly indicating that users are in the app's configuration area.
- **Menu Options:**
  - **Basic Info:** An option that likely allows users to view and edit basic personal information such as name, email, and contact details.
  - **Security:** This option suggests a section where users can manage their account security, possibly including password changes and privacy settings.
  - **Logout:** A distinct red button is available for users to log out of their account, which is a standard security and convenience feature.

### **Second Screen - Personal Information Page:**

- **User Details:** Various fields display the user's personal information, including:
  - First Name: Osama
  - Last Name: Khanfa
  - Email: A google address
  - Phone: A phone number is provided, likely for contact purposes.
- **Shipping Address:** There is also a section for the user's shipping address, including city/village and street name, which is essential for e-commerce transactions.

### **Third Screen - Security Page:**

- **Security Settings:** This page is clearly dedicated to security, allowing the user to update their password.
  - Fields for "Old Password" and "New Password" are present, along with a confirmation field for the new password to ensure it is entered correctly.
- **Action Button:** A "Change Password" button is prominently displayed, which users can tap to save the new password, completing the update process.

These screens are designed to provide a user-friendly experience for managing account settings and security, ensuring users have easy access to update their personal information and maintain the security of their account.

## Technologies

Developing an application with Ionic, Vue.js, and TypeScript involves a combination of modern web technologies and frameworks designed to create high-performance, cross-platform mobile and web apps. Below is a description of the process and how these technologies come together in application development.

### Ionic Framework

Ionic is an open-source UI toolkit for building high-quality, cross-platform native and web apps from a single codebase. It leverages web technologies such as HTML, CSS, and JavaScript, making it accessible to developers with web development experience.

- **UI Components:** Ionic provides a rich library of pre-designed components that mimic native elements, enabling the creation of a native-like user interface.
- **Plugins:** Through Cordova or Capacitor, Ionic offers access to native device features like the camera, GPS, and more.
- **Themes and Styling:** Ionic supports theming and customization, allowing developers to apply custom styles globally or locally to components

### Vue.js

Vue.js is a progressive JavaScript framework used for building user interfaces. It focuses on the view layer, offering a reactive and composable data binding system.

- **Reactivity:** Vue's core feature is its reactive data binding that ensures the UI updates efficiently when the underlying data changes.
- **Components:** Vue utilizes a component-based architecture, enabling developers to build encapsulated and reusable components.

# TypeScript

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript. It provides static typing, which can help catch errors early in the development process.

- **Type Safety:** Using TypeScript ensures type safety, improved code quality, and predictability.
- **Advanced Features:** TypeScript supports modern JavaScript features, including ES6+ syntax, and provides advanced features like interfaces, generics, and namespaces.
- **Tooling:** The TypeScript toolchain includes features like auto-compilation and advanced autocompletion, enhancing developer productivity.

## Development Process with Ionic, Vue, and TypeScript

1. **Project Setup:** Initialize a new project using the Ionic CLI, selecting Vue as the framework of choice and enabling TypeScript support.
2. **Design UI:** Use Ionic components to design the app's user interface, customizing them with Vue's template syntax and CSS.
3. **Implement Logic:** Write the application logic in TypeScript, taking advantage of Vue's reactivity and component lifecycle hooks.
4. **Integrate APIs:** Connect to backend APIs or services using Vue's reactive data properties to fetch, display, and manage data.
5. **Native Features:** Implement native functionality by installing Ionic Native plugins and injecting them into Vue components.
6. **Testing:** Write and run unit and end-to-end tests, leveraging TypeScript's type system for more robust testing.
7. **Build and Deploy:** Compile the TypeScript code, and use Ionic Appflow or native build tools to build and deploy the application on different platforms.

By combining these technologies, developers can create a maintainable and scalable application with a rich user interface, robust features, and native integrations.

# Features and Functionalities

## Product exploring and Search Capability

- **Exploring:** Users can find their needed electronic parts from the marketplace through a straightforward shop page, which includes product details and adding to their carts
- **Search:** The search function is designed with filters to refine results by category, and other specifications, allowing users to quickly find the items they need.

## User Account Management

- **Login:** Users can create an account using their email address, facilitating a personalized shopping experience and enabling order tracking.
- **Profile Customization:** Within their account, users can manage personal information, change passwords.

## Shopping Cart and Checkout Process

- **Cart Management:** Users can review items in their cart, adjust quantities, or remove products before proceeding to checkout.
- **Secure Payment:** The checkout process includes secure payment options, with clear summaries of charges and the option to save payment methods for future transactions.

## Real-time Communication with Support

- **Chat Feature:** Users have access to a real-time chat service, allowing them to communicate with support staff directly for inquiries or assistance.

