



SCHOOL OF TECHNOLOGY

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

BIT: BIT 04205

KCAMARKET WEB APPLICATION

BY

AWANZI HASSAN MUNENE - 21/04609

PRESENTED TO: Prof. PATRIC OGAO

SUBMISSION DATE: 28/03/2024

Introduction	3
The system installation	3
To install the application, follow these steps:	3
Running the system	3
Welcome Page	4
Sign-up page	5
Account Verification	5
Accessing the system	6
Best selling page.	8
Products page	8
Events page	8
Become Seller Page	10
Shop register page	10
Shop Dashboard	11
All orders page.	12
Create product page	12
View the product.	13
Buying the product.	14
Checkout page.	15
Payment Page.	15
Payment with card.	16
Payment using paypal.	17
Withdraw money	23
Adding withdrawal method(s)	24
Coupon creation and use.	27
Create a coupon page.	27
Using the coupon code	29
Profile Edit.	31
Change Password.	32
Add address	33
Logout.	33
Logout page.	33
Database	35
Integration with Cloudinary	35
Users document	36
Shops document	36
Products Document	37
Orders Document	37
Messages Document	38
Events Document	38
Coupon codes document	39

Conversations Document	39
Cloudinary for image storage and optimization.	40
Error Message	40
Troubleshooting- Problems	40
Troubleshooting	41
Back up	41
Conclusions and Recommendations	41

USER MANUAL

Introduction

The user manual serves as a guide to the use of the KcaMarket ecommerce web application, built using the MERN (MongoDB, Express.js, React.js, Node.js) stack. It provides assistance to users in navigating the system and making the most of its features. This manual outlines the system's functions, installation procedures, operation guidelines, and troubleshooting methods.

The system installation

The multi vendor ecommerce web application can be installed on any operating system that supports the following:

Node.js and npm (Node Package Manager)

MongoDB (NoSQL database)

A web browser (e.g., Google Chrome, Mozilla Firefox)

To install the application, follow these steps:

Clone the repository from GitHub.

Navigate to the project directory in the terminal.

Install dependencies using the command `npm install`.

Start the server using the command `npm start`.

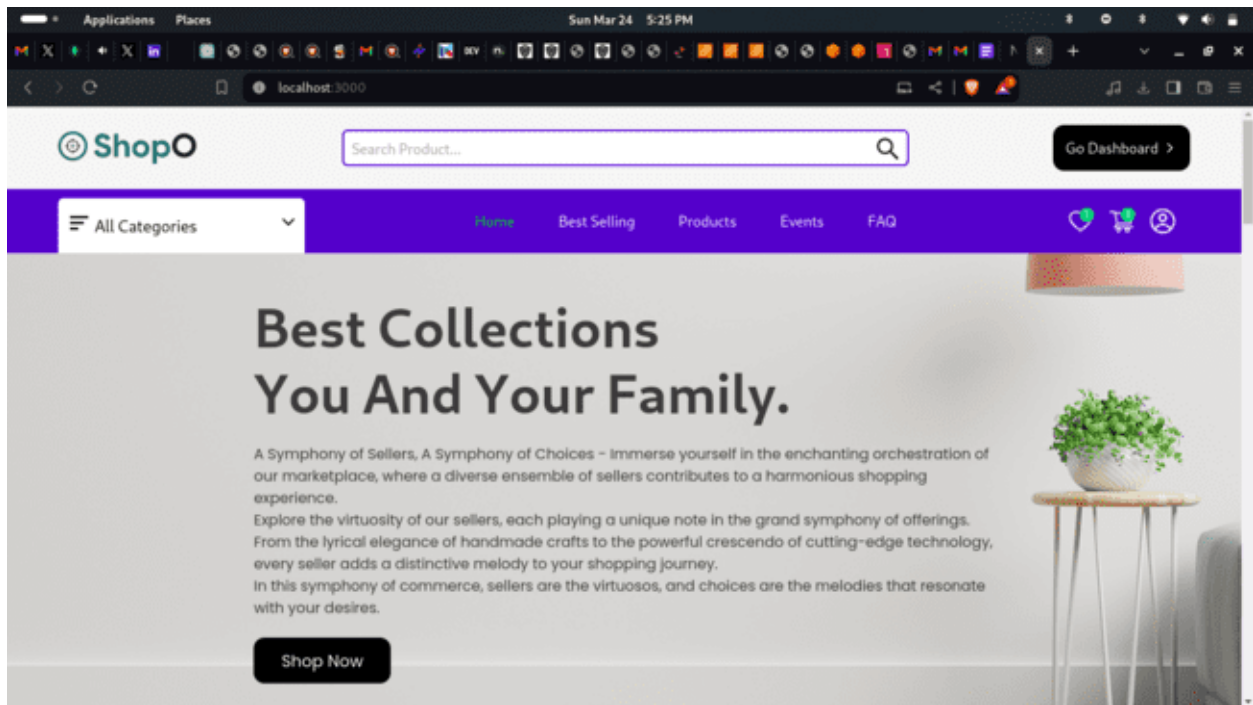
Open a web browser and navigate to `http://localhost:3000` to access the application.

Running the system

Upon launching the application, users will be presented with a login screen where they must enter their credentials to access the system. New users must register before logging in to gain access. Once logged in, users can navigate through the various forms and features of the application.

Welcome Page

On opening the web browser and navigating to port 3000 you will be welcomed to a welcome page.



Sign-up page

This is a page/part where the user is able to register themselves to the system by supplying their Full Name, Email Address, Password and Profile Picture/Avatar then

Applications Places Sun Mar 24 5:30 PM

localhost:3000/sign-up

Register as a new user

Full Name
ADONIJAH KIPLIMO

Email address
adonjakiplimo8@gmail.com

Password

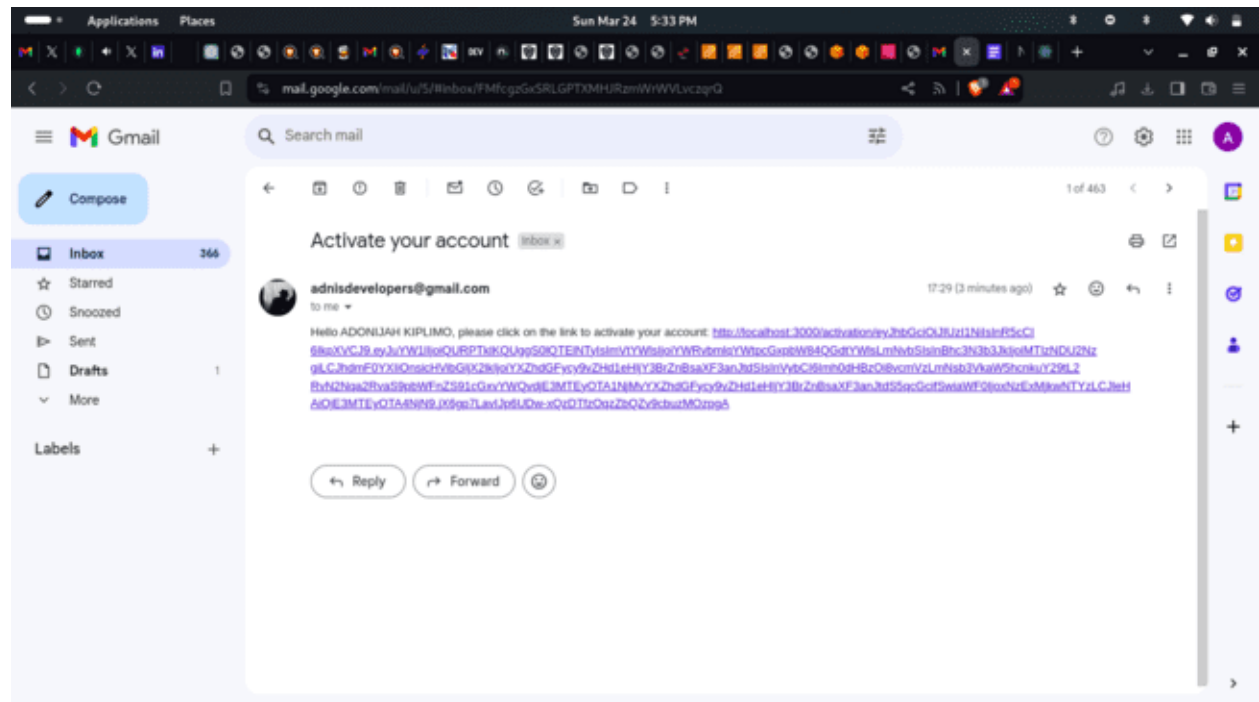
Upload a file

Submit

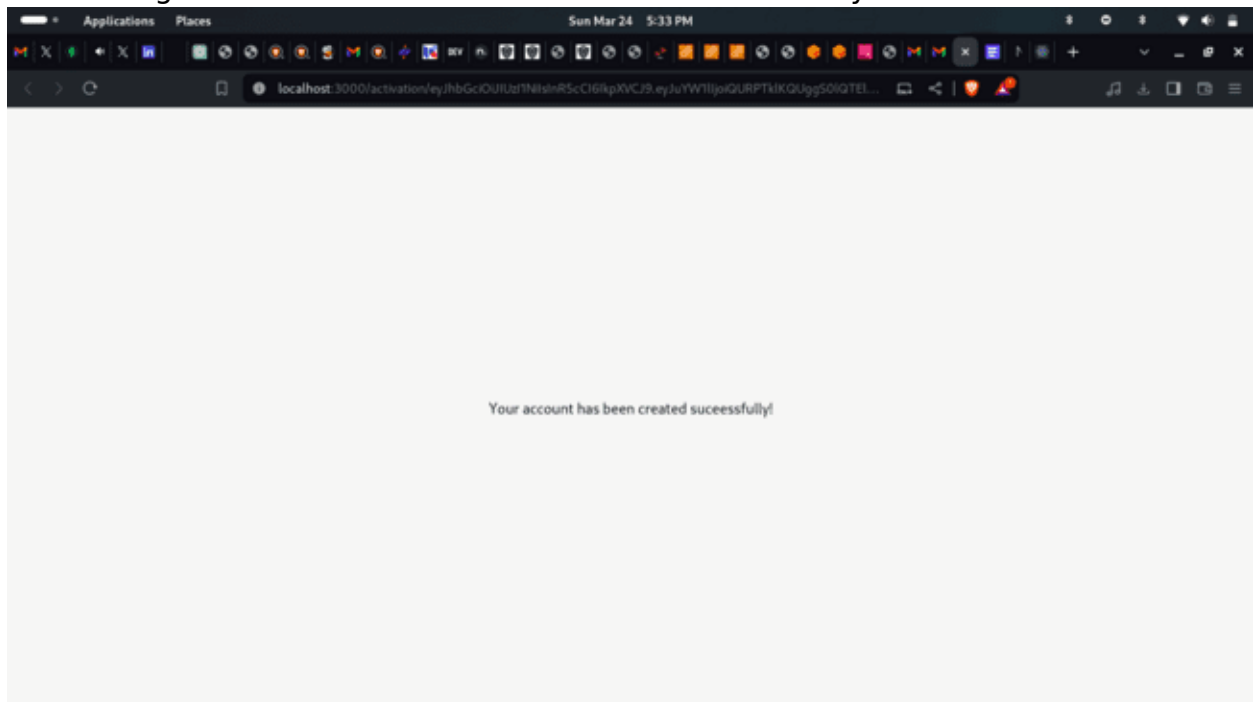
Already have an account? [Sign In](#)

localhost:3000/login

After submitting the credentials an email will be sent to the supplied email for account verification.

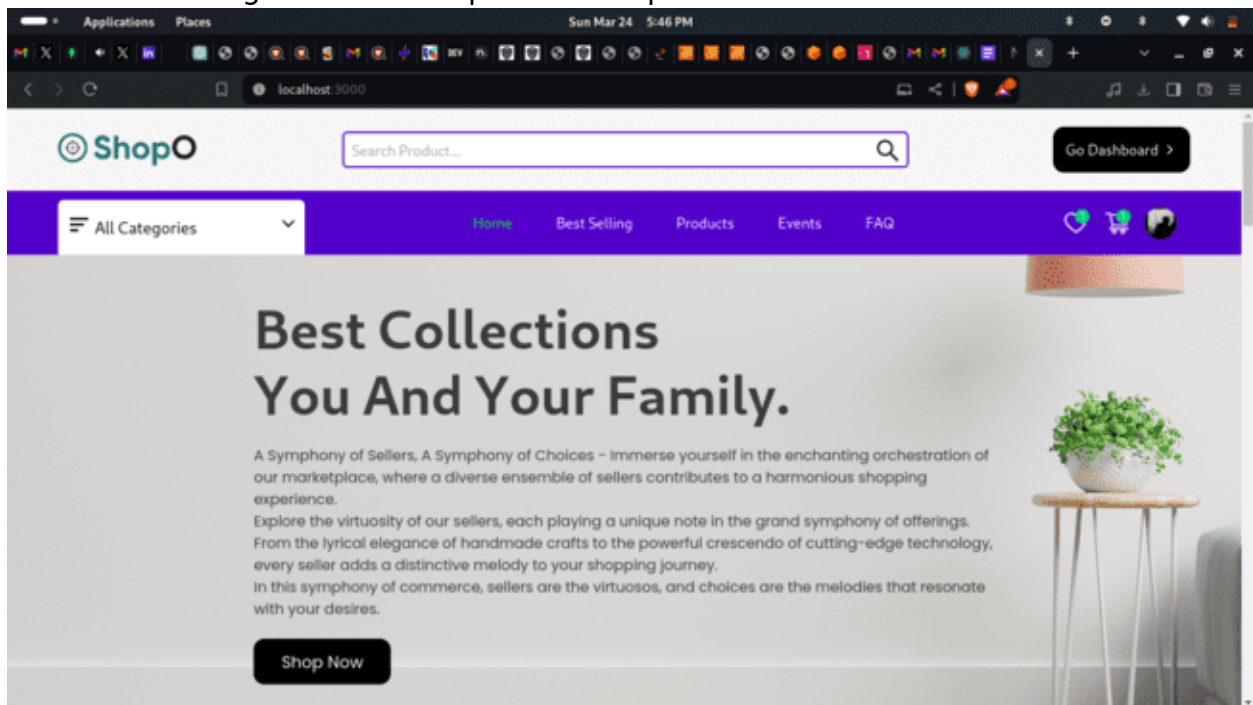


On Clicking the verification link the account is successfully created.

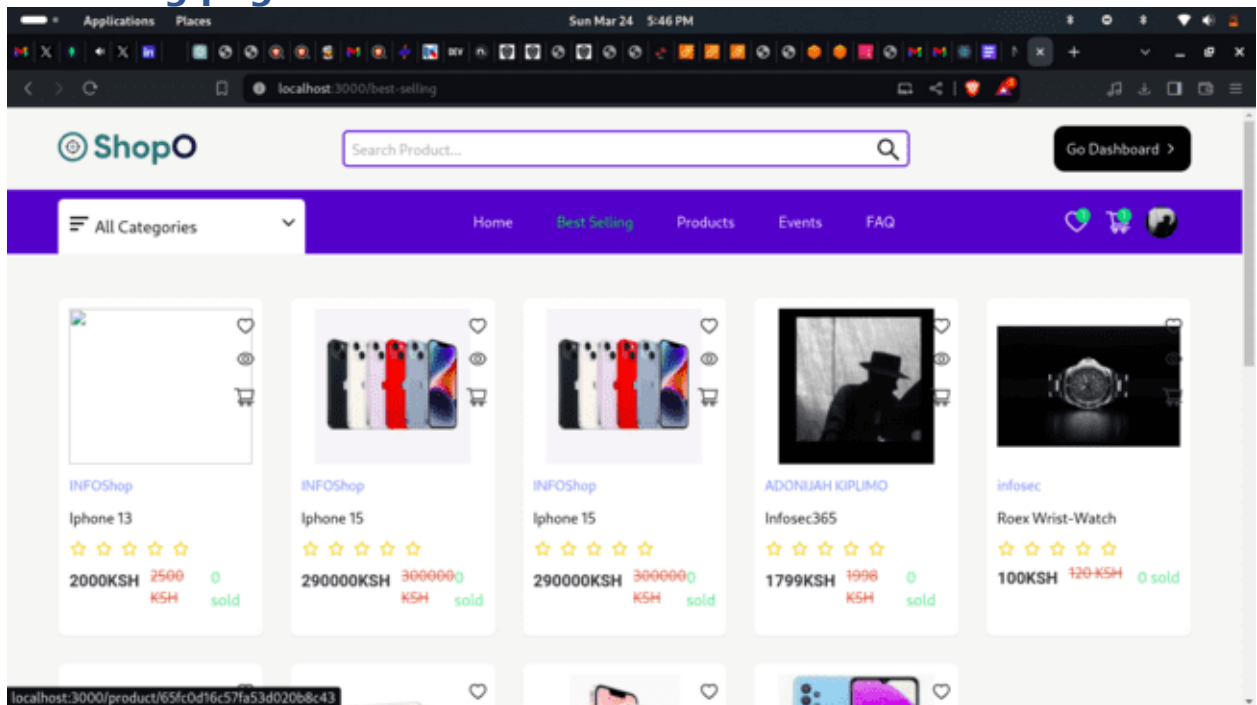


Accessing the system

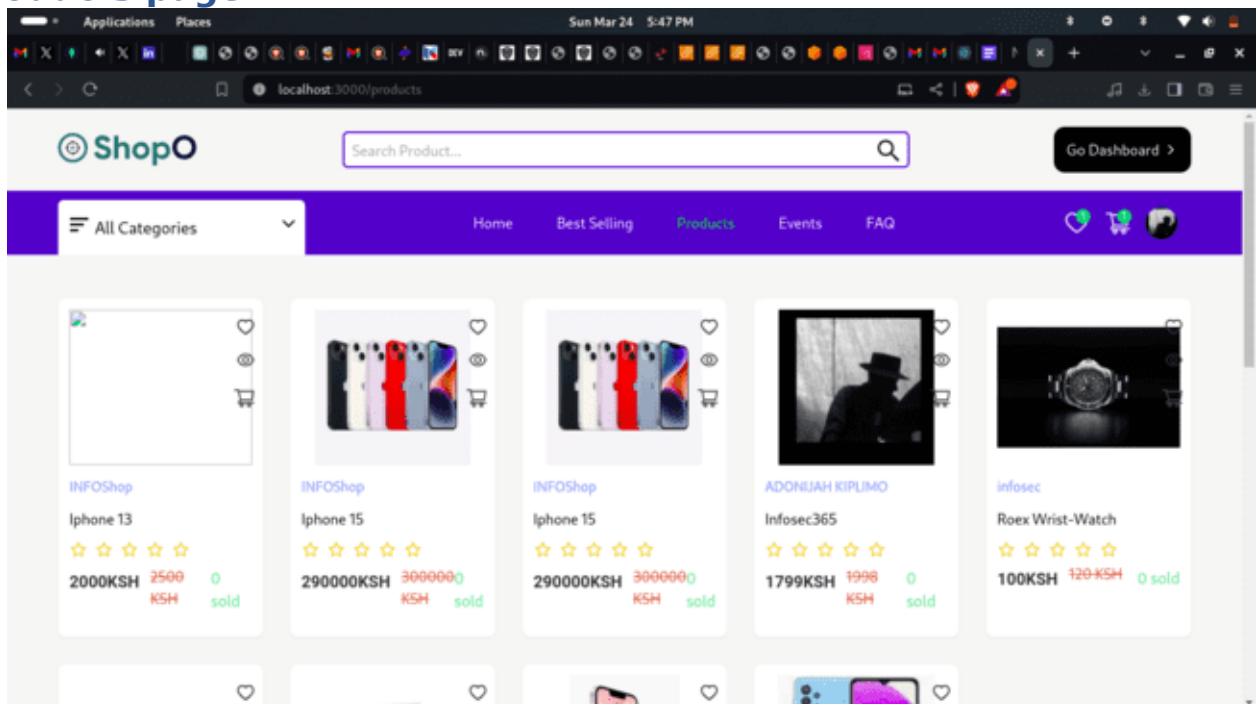
When the user logs in successfully he/she is welcomed to the welcome page with a welcome message and a description of the platform on the home tab.



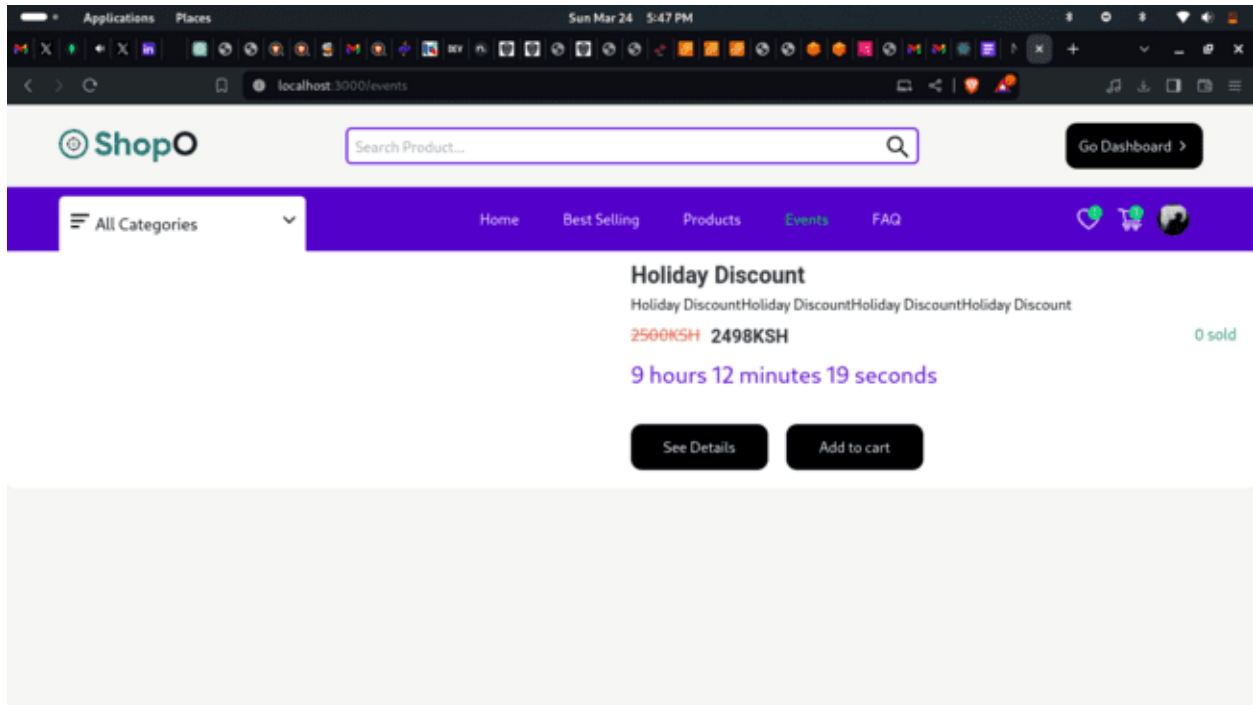
Best selling page.



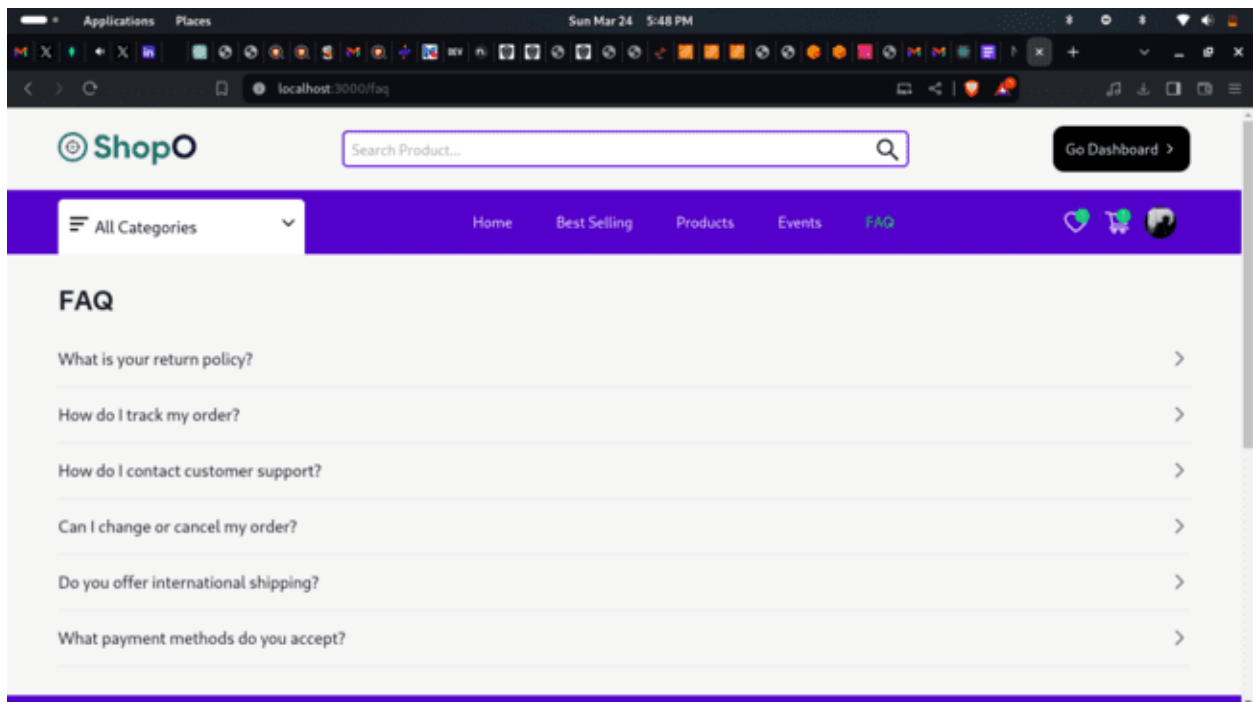
Products page



Events page

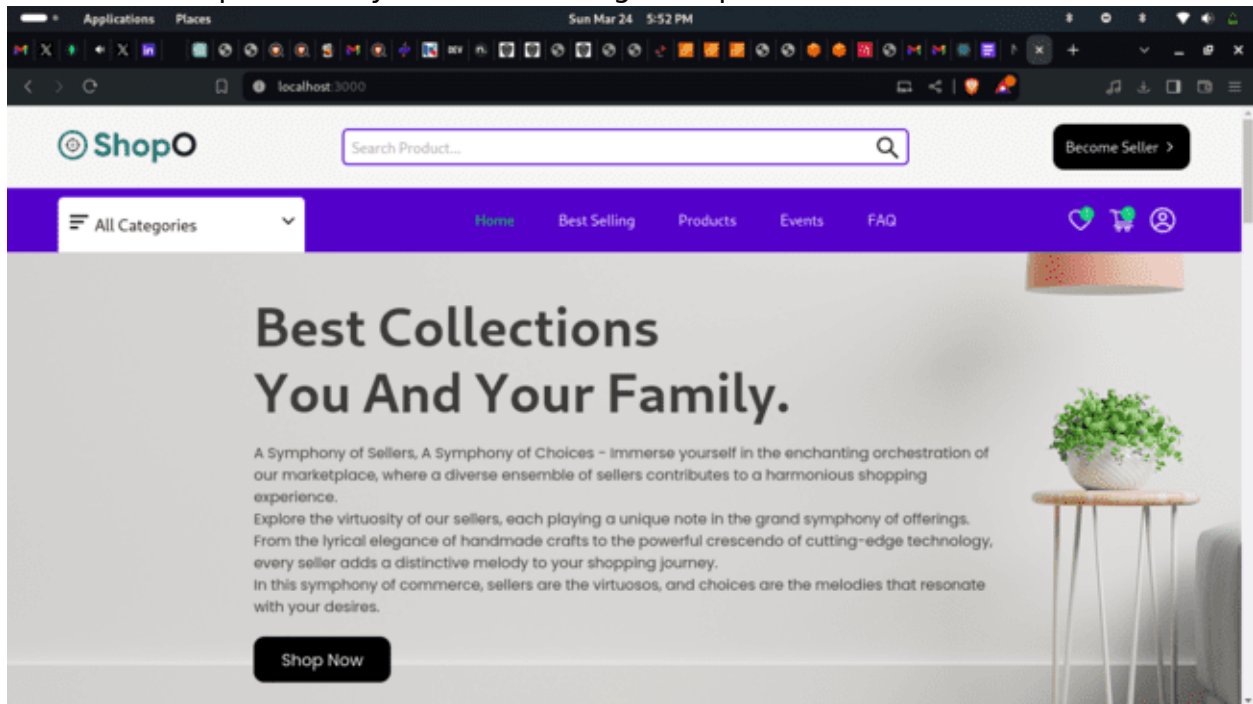


Frequently Asked Questions[FAQ]

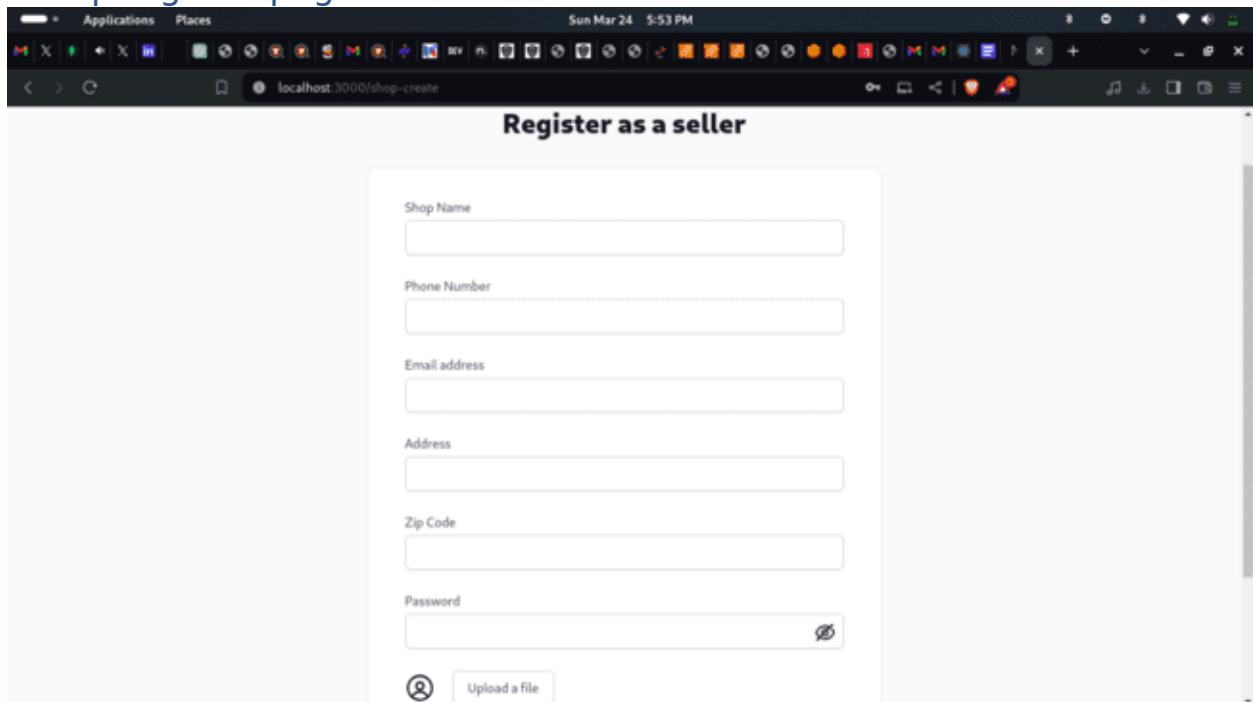


Become Seller Page

At the home page there's a button of "become seller" where the user can sign up for his/her shop then they can start selling their products.

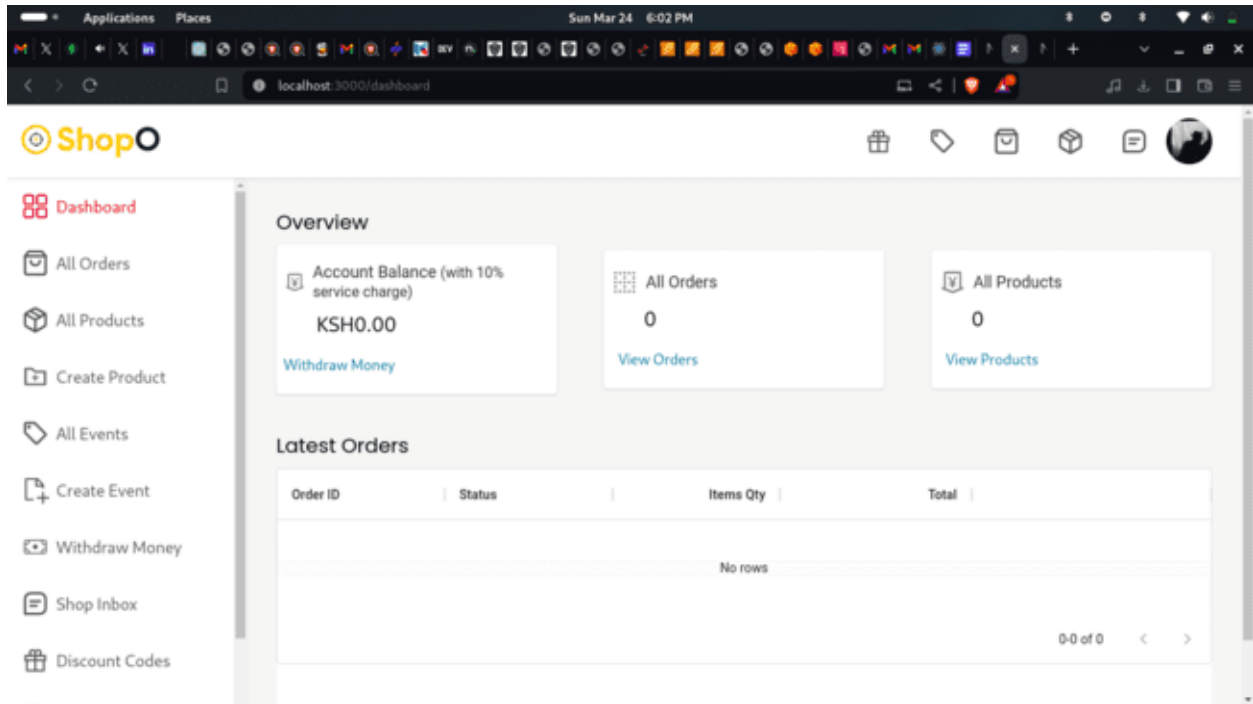


Shop register page

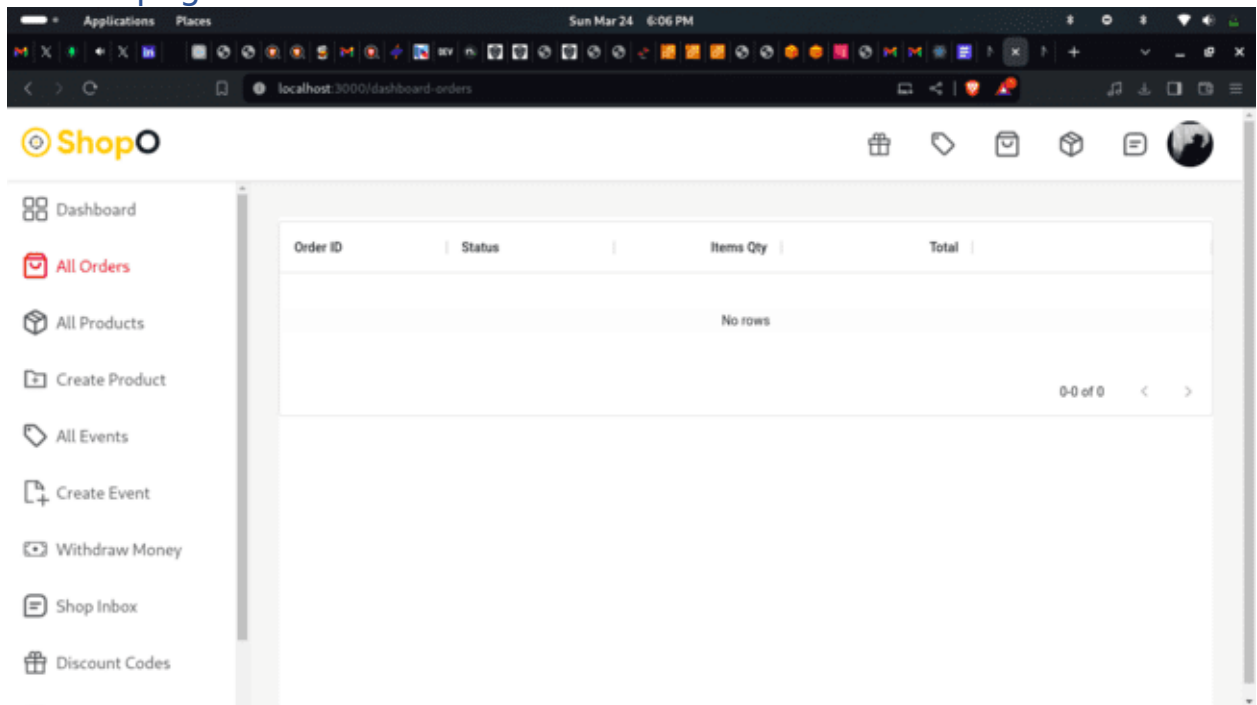


Shop Dashboard

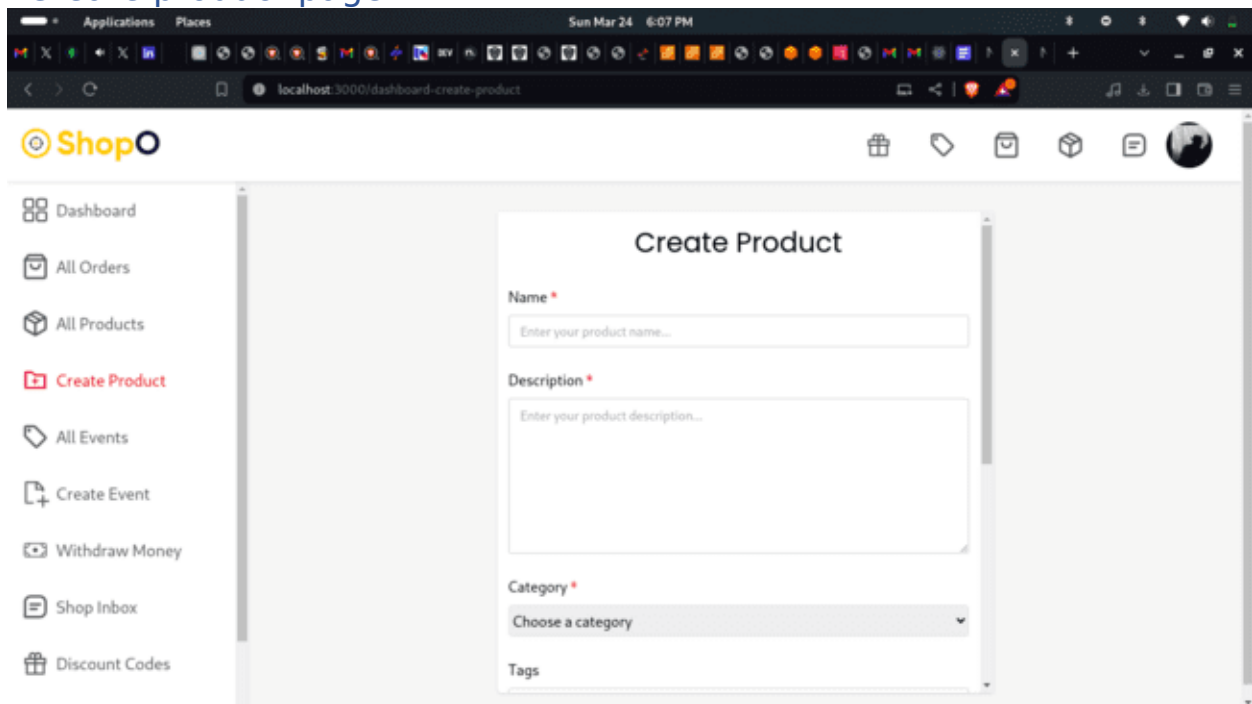
On login the dashboard shows all the orders, products, events and he/she can create products, create events and withdraw money.



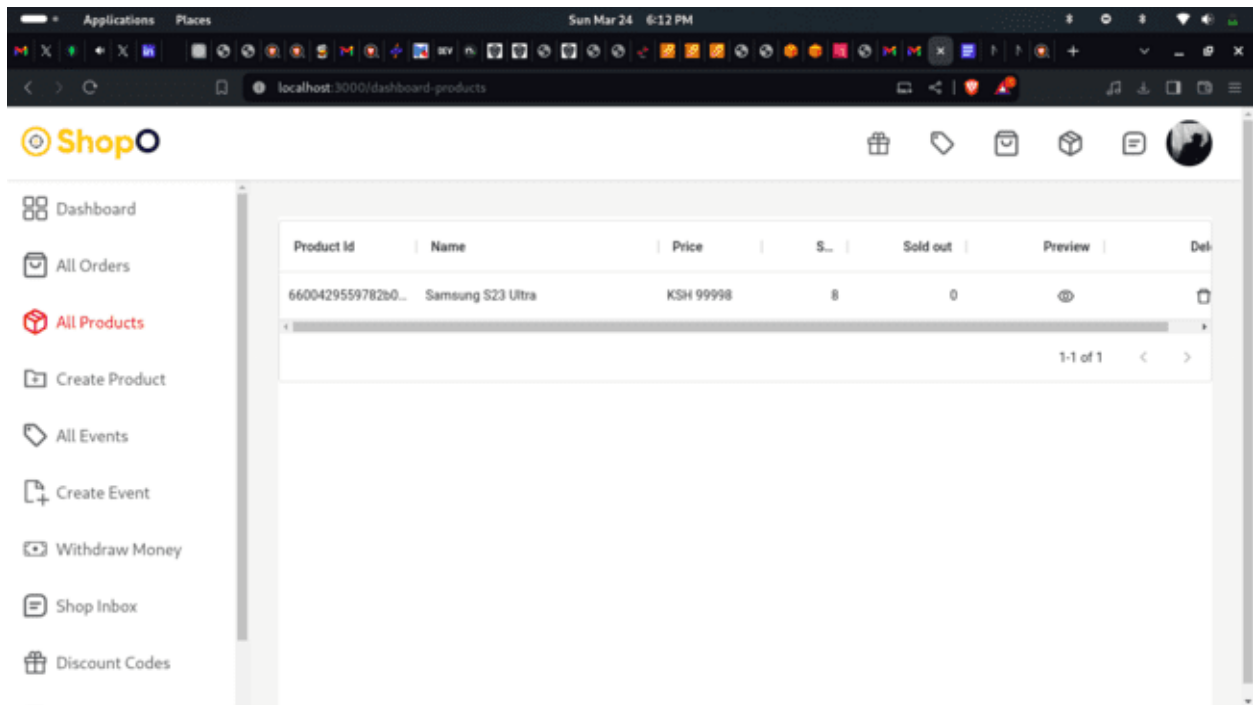
All orders page.



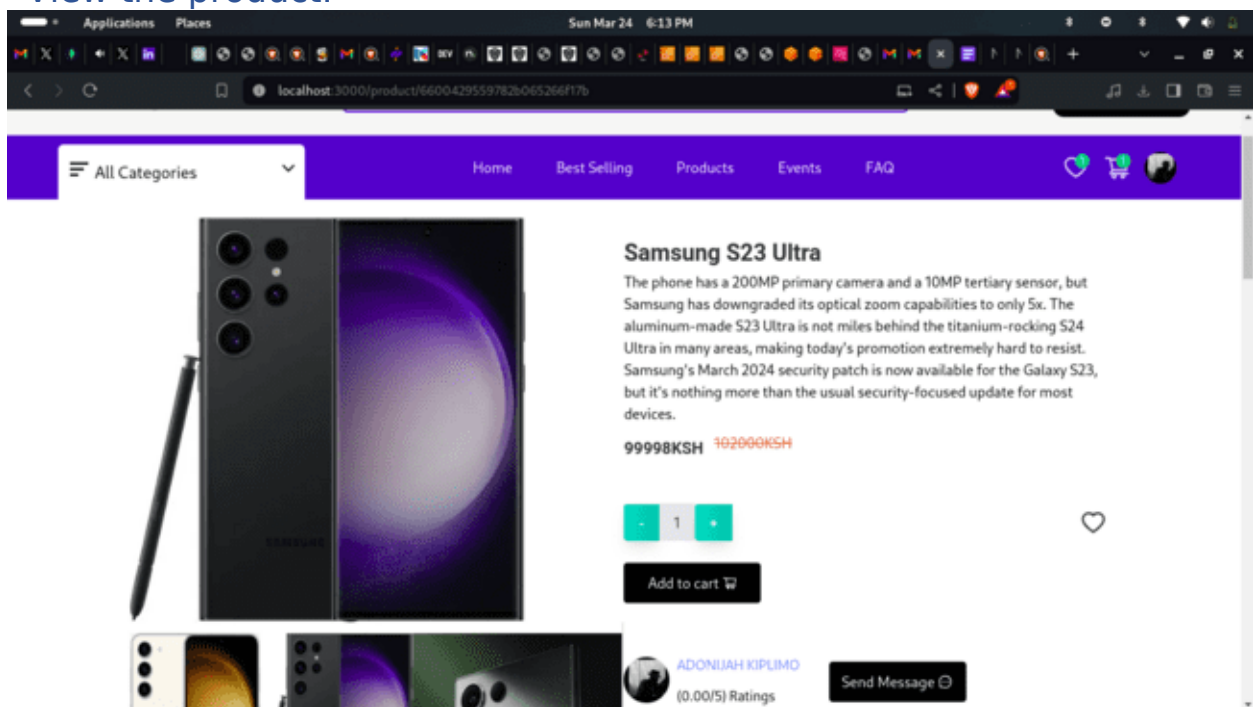
Create product page



After creating the products it will appear on the all products page.

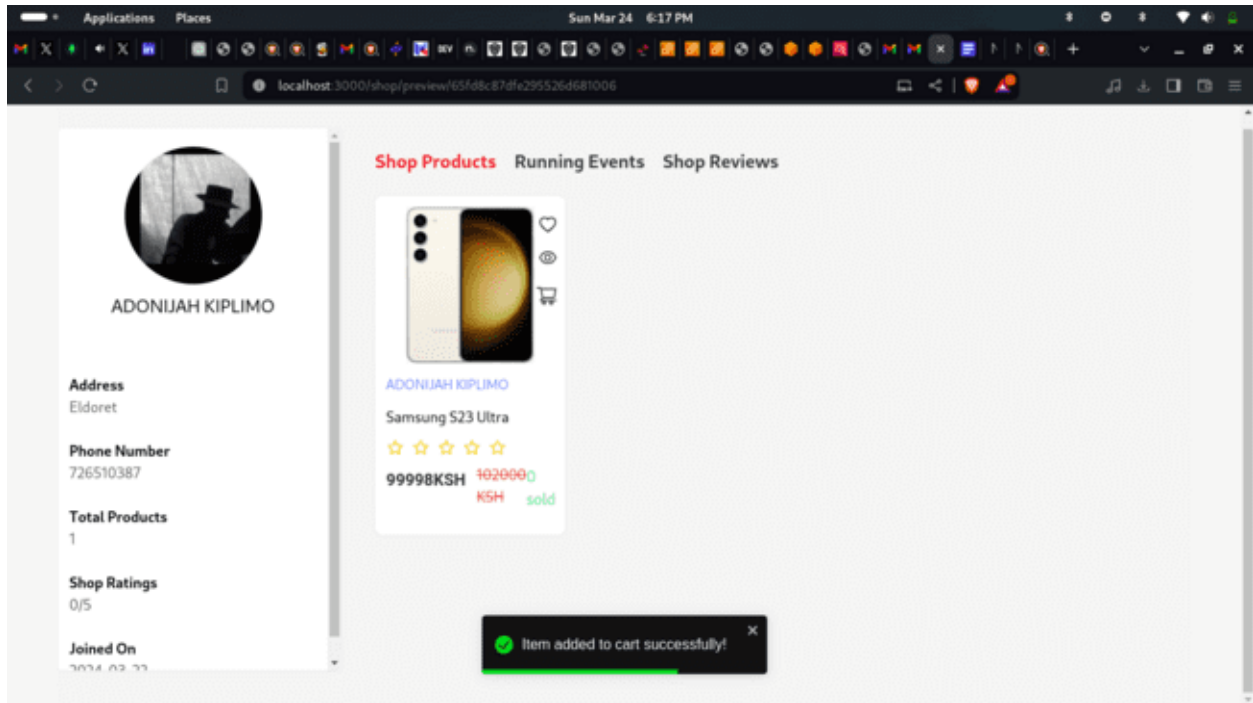


View the product.

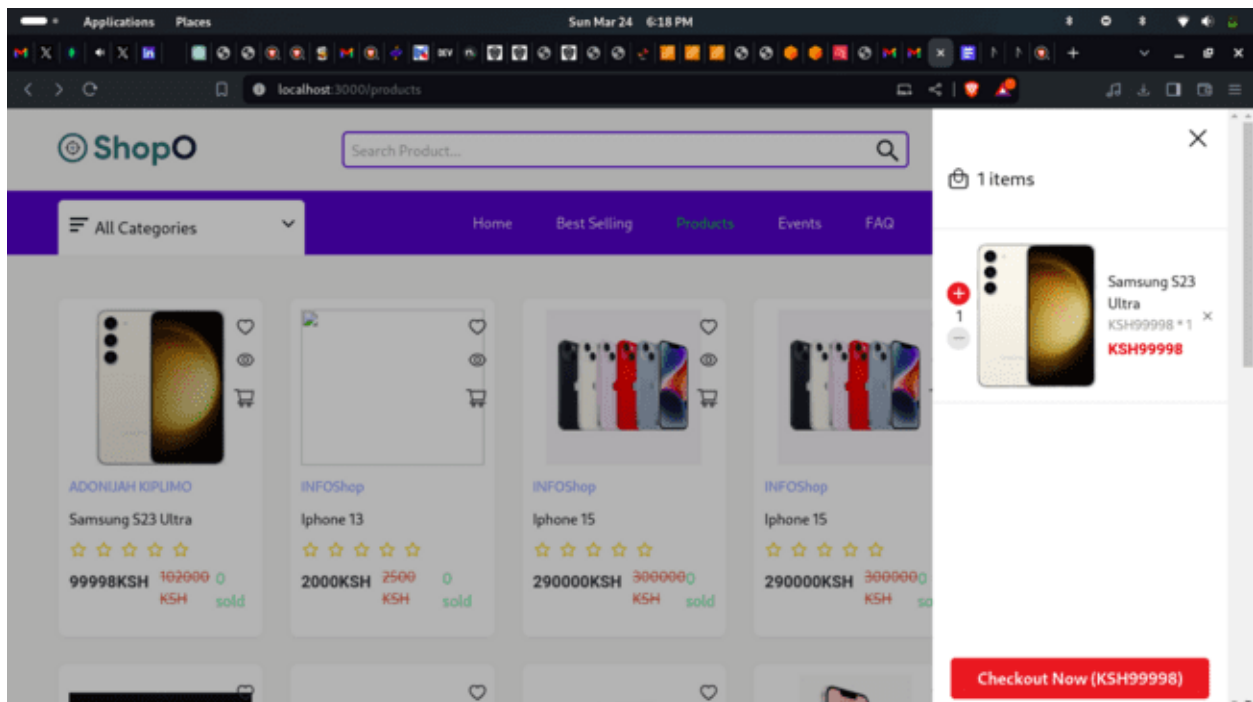


Buying the product.

To buy a product the user has to visit the homepage then select the product they want to buy and then add to cart.



Then proceed to the cart to check out.



Then click on the checkout button.

Checkout page.

Here the user have to fill in the shipping details that is Full name, email address, phone number, country, city and address.

The screenshot displays the ShopO checkout page in a web browser. The browser's address bar shows 'localhost:3000/checkout'. The page features a purple header with the ShopO logo, a search bar, and a 'Go Dashboard' button. Below the header is a navigation bar with links for 'All Categories', 'Home', 'Best Selling', 'Products', 'Events', and 'FAQ'. A progress indicator shows three steps: '1.Shipping' (active), '2.Payment', and '3.Success'. The main content area is divided into two columns. The left column, titled 'Shipping Address', contains form fields for 'Full Name' (filled with 'ADONIJAH KIPLIMO'), 'Email Address' (filled with 'adonijakiplimo8@gmail.com'), 'Phone Number', 'Zip Code', 'Country' (a dropdown menu with 'Choose your country' selected), 'City' (a dropdown menu with 'Choose your City' selected), 'Address1', and 'Address2'. The right column displays the order summary: 'subtotal: KSH99998', 'shipping: KSH9999.80', 'Discount: -', and a total of 'KSH109997.80'. There is a 'Coupon code' input field and an 'Apply code' button.

Payment Page.

On this page the User chooses the payment method he/she prefers whether by card paypal or even by payment on delivery.

Applications Places Sun Mar 24 6:23 PM

localhost:3000/payment

ShopO Search Product... Go Dashboard >

All Categories Home Best Selling Products Events FAQ

1.Shipping 2.Payment 3.Success

☒ Pay with Debit/credit card

Name On Card Exp Date
ADONIJAH KIPLIMO MM / YY

Card Number CW
1234 1234 1234 1234 CVC

Submit

☐ Pay with Paypal

subtotal: KSH99998
shipping: KSH9999.80
Discount: -
KSH109997.80

Payment with card.

Applications Places Sun Mar 24 6:25 PM

localhost:3000/payment

ShopO Search Product... Go Dashboard >

All Categories Home Best Selling Products Events FAQ

1.Shipping 2.Payment 3.Success

☒ Pay with Debit/credit card

Name On Card Exp Date
ADONIJAH KIPLIMO MM / YY

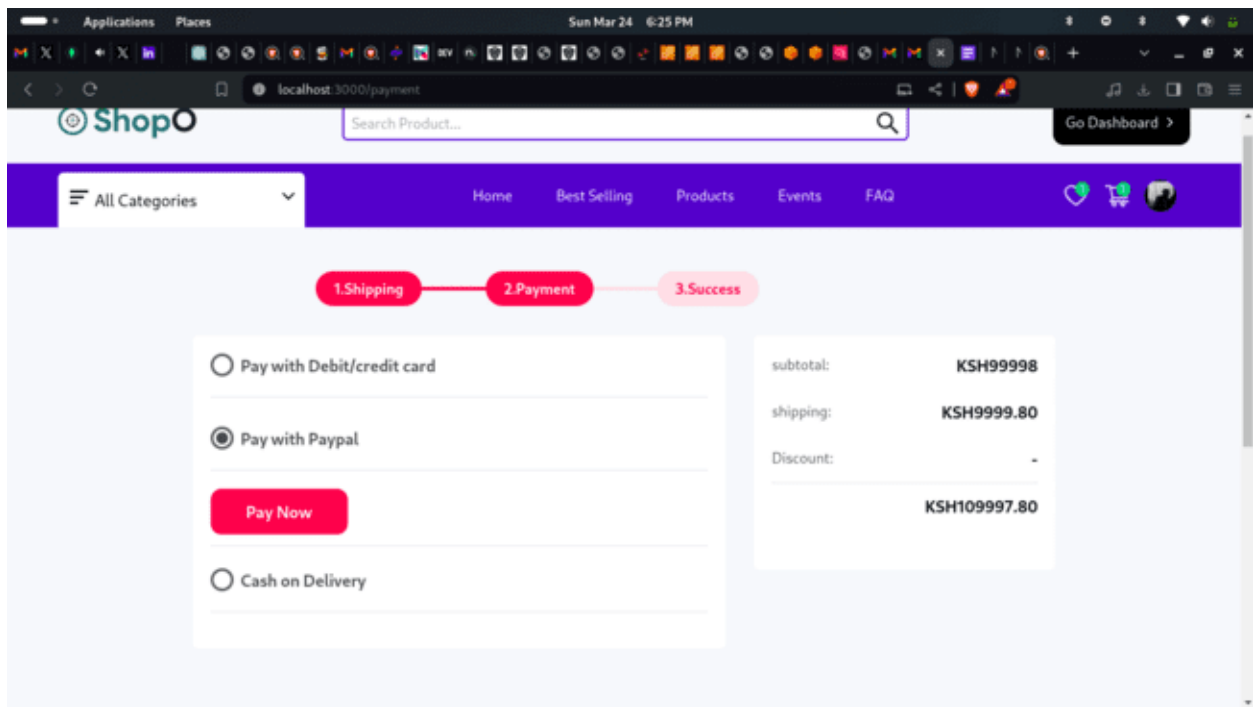
Card Number CW
1234 1234 1234 1234 CVC

Submit

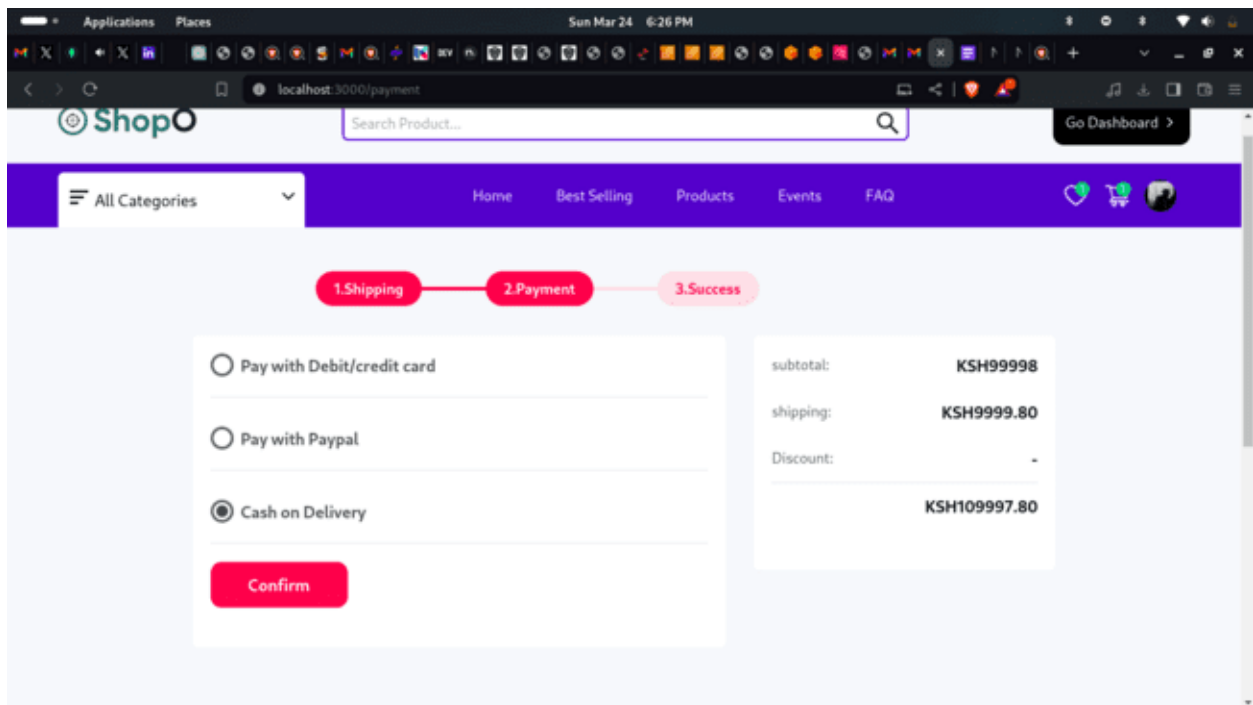
☐ Pay with Paypal

subtotal: KSH99998
shipping: KSH9999.80
Discount: -
KSH109997.80

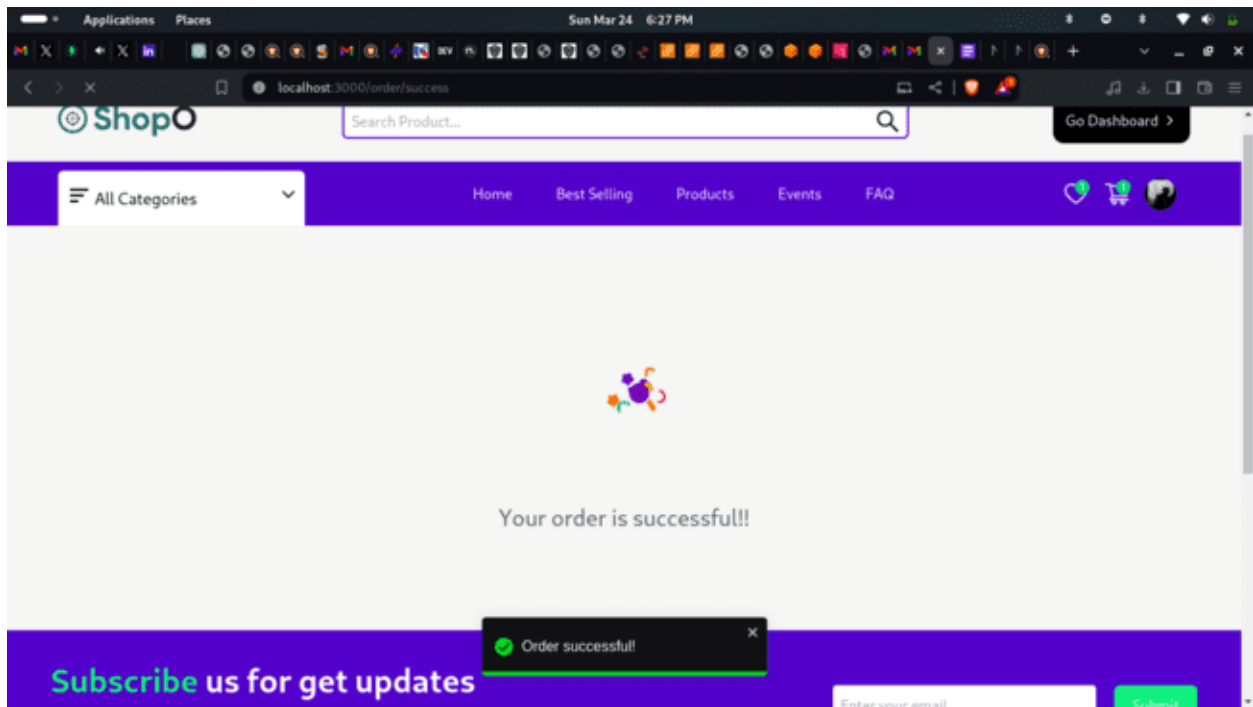
Payment using by paypal.



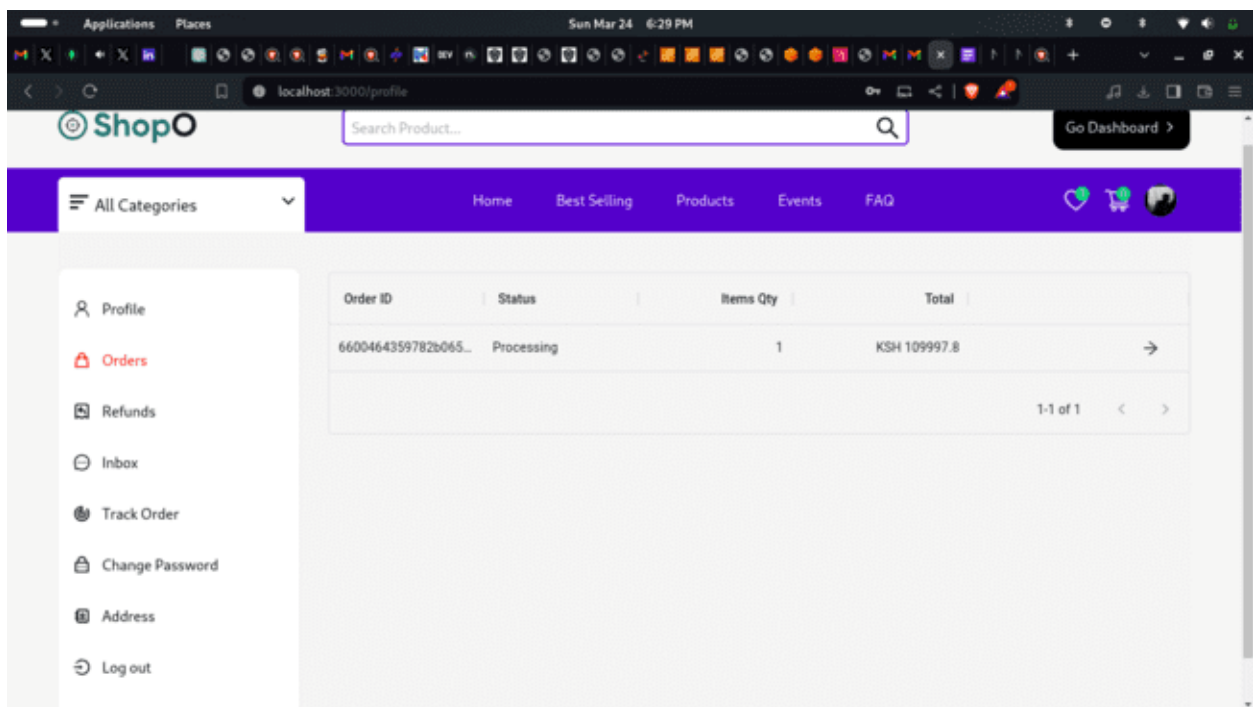
Payment on delivery



On successful payment the order will be processed as successfully placed.



On the buyers orders page the order will be in the processing stage awaiting approval by the seller.



On the sellers dashboard there will be new orders awaiting approval and delivery on the orders page.

Applications Places Sun Mar 24 6:31 PM

localhost:3000/dashboard

ShopO

Dashboard

- All Orders
- All Products
- Create Product
- All Events
- Create Event
- Withdraw Money
- Shop Inbox
- Discount Codes

Overview

Account Balance (with 10% service charge)

KSH0.00

[Withdraw Money](#)

All Orders

1

[View Orders](#)

All Products

1

[View Products](#)

Latest Orders

Order ID	Status	Items Qty	Total
6600464359782b06526...	Processing	1	KSH 109997.8

1-1 of 1

Applications Places Sun Mar 24 6:31 PM

localhost:3000/dashboard-orders

ShopO

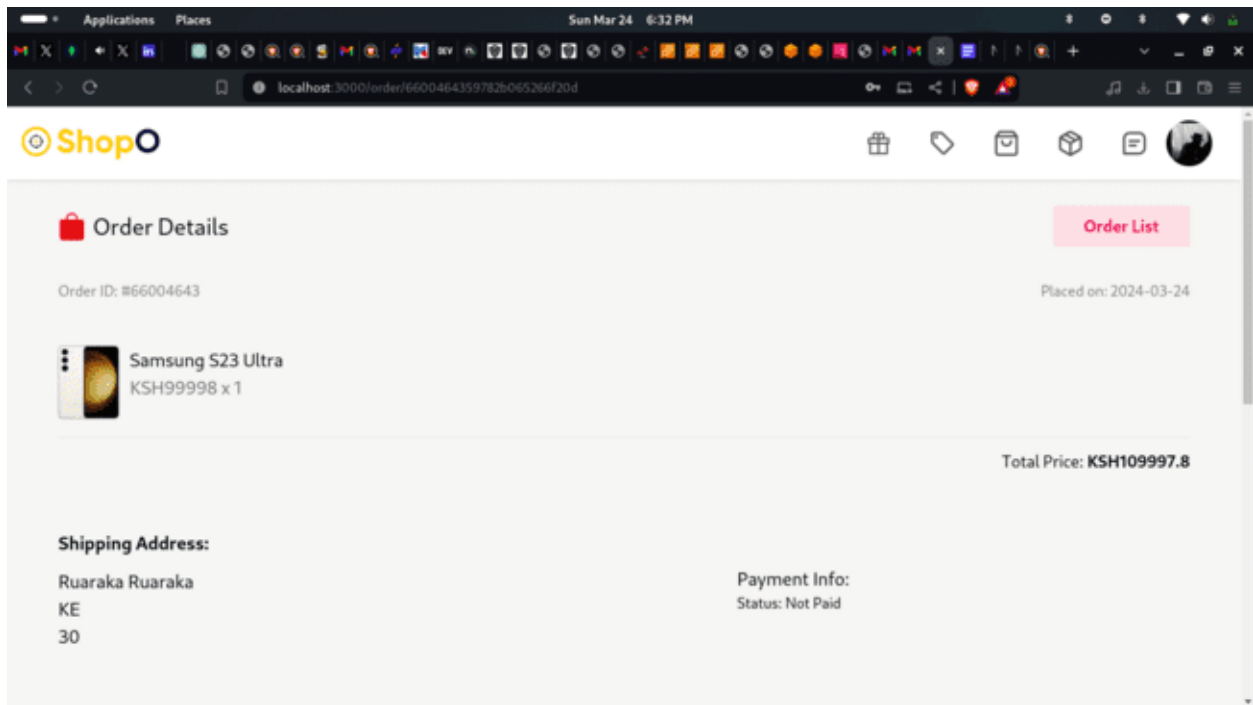
Dashboard

All Orders

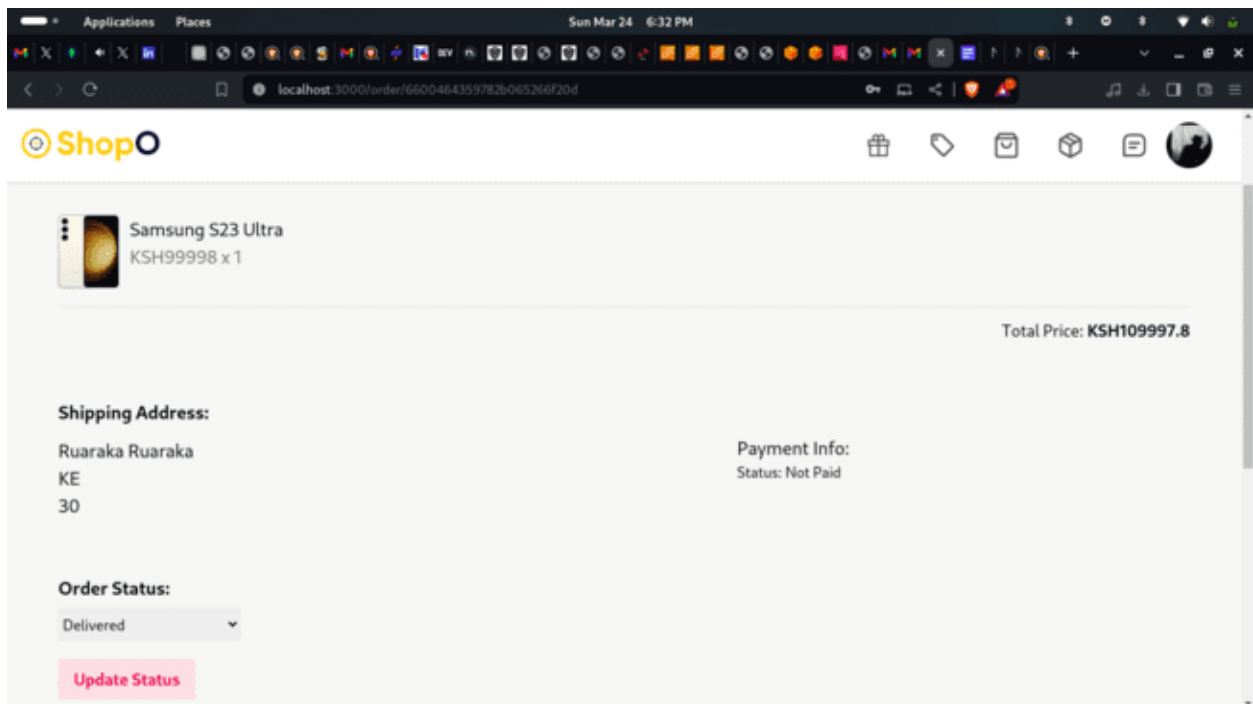
- All Products
- Create Product
- All Events
- Create Event
- Withdraw Money
- Shop Inbox
- Discount Codes

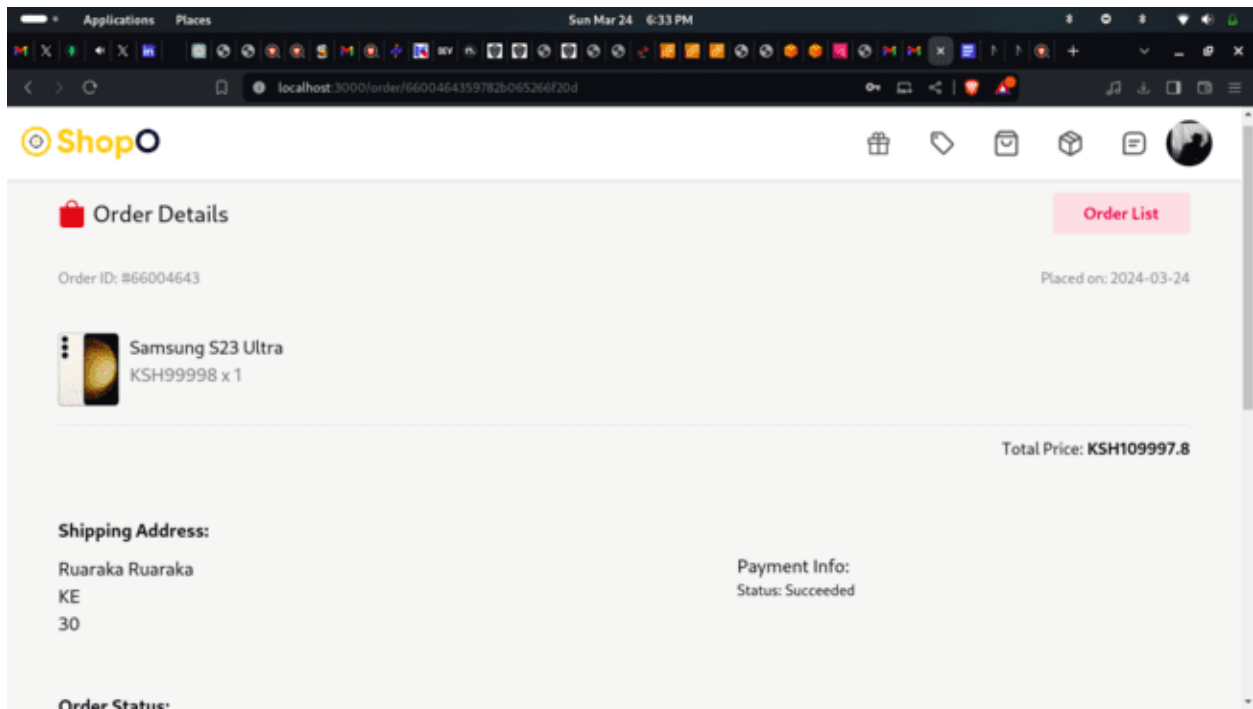
Order ID	Status	Items Qty	Total
6600464359782b06526...	Processing	1	KSH 109997.8

1-1 of 1

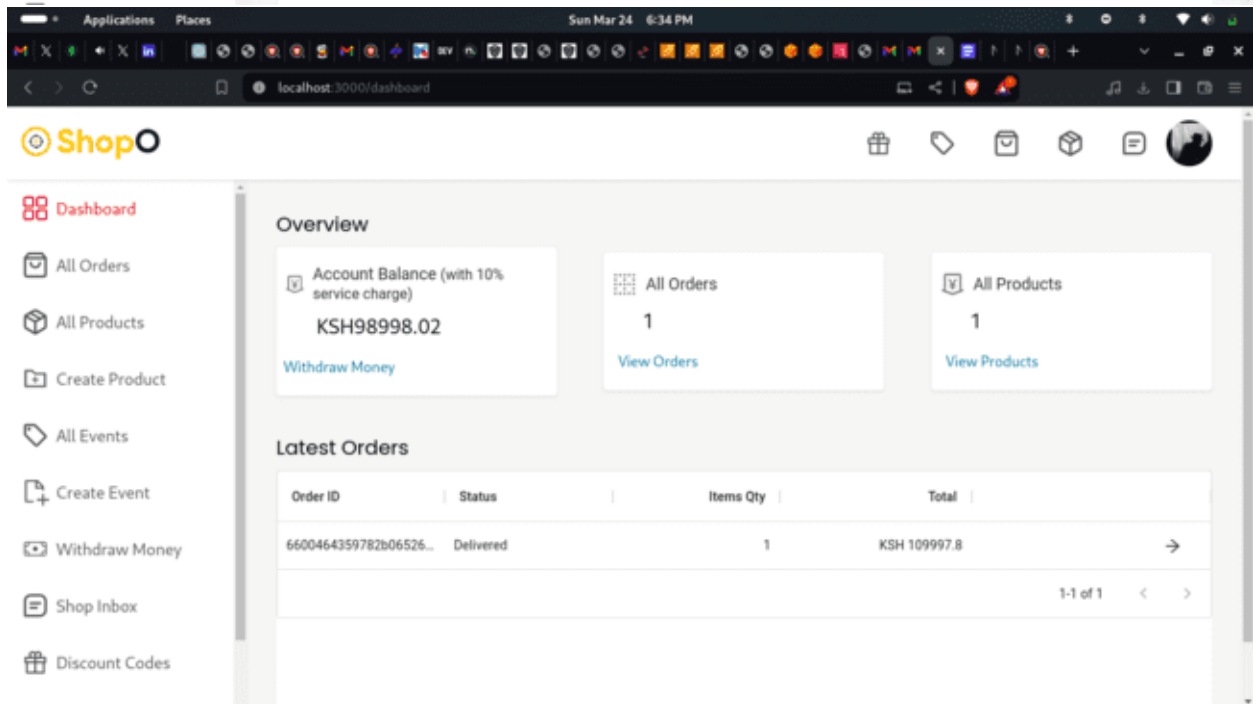
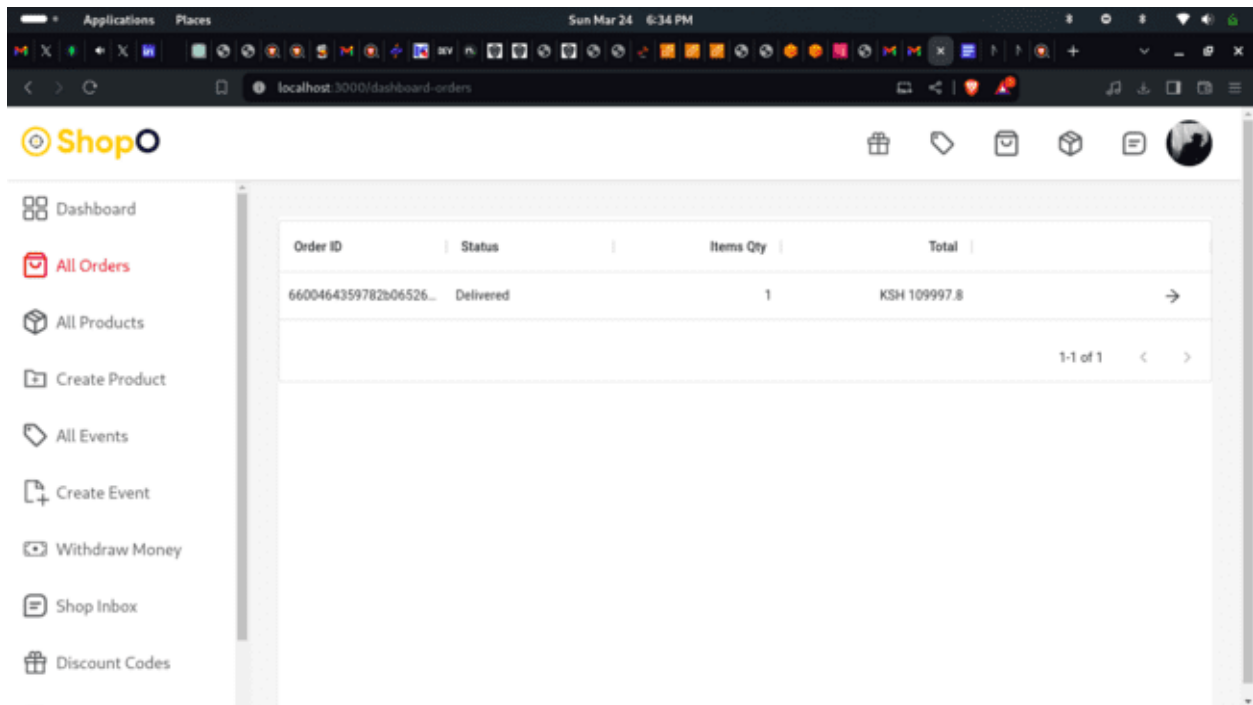


The seller can then update the status of the order.





Then back on the dashboard the orders will be updated and the balance as well updated.



Withdraw money

The seller can then withdraw money from the shop wallet.

withdraw money page.

The screenshot shows the ShopO dashboard overview page. The browser address bar indicates the URL is localhost:3000/dashboard. The page features a sidebar with navigation links: Dashboard, All Orders, All Products, Create Product, All Events, Create Event, Withdraw Money, Shop Inbox, and Discount Codes. The main content area is titled 'Overview' and contains three summary cards: 'Account Balance (with 10% service charge)' showing KSH98998.02 with a 'Withdraw Money' link, 'All Orders' showing 1 order with a 'View Orders' link, and 'All Products' showing 1 product with a 'View Products' link. Below these is a 'Latest Orders' table with one entry: Order ID 6600464359782b06526..., Status Delivered, Items Qty 1, and Total KSH 109997.8. The table has pagination controls showing 1-1 of 1.

Order ID	Status	Items Qty	Total
6600464359782b06526...	Delivered	1	KSH 109997.8

The screenshot shows the ShopO withdraw money page. The browser address bar indicates the URL is localhost:3000/dashboard-withdraw-money. The sidebar is identical to the dashboard overview page. The main content area displays the 'Available Balance: KSH98998.02' and a large black 'Withdraw' button.

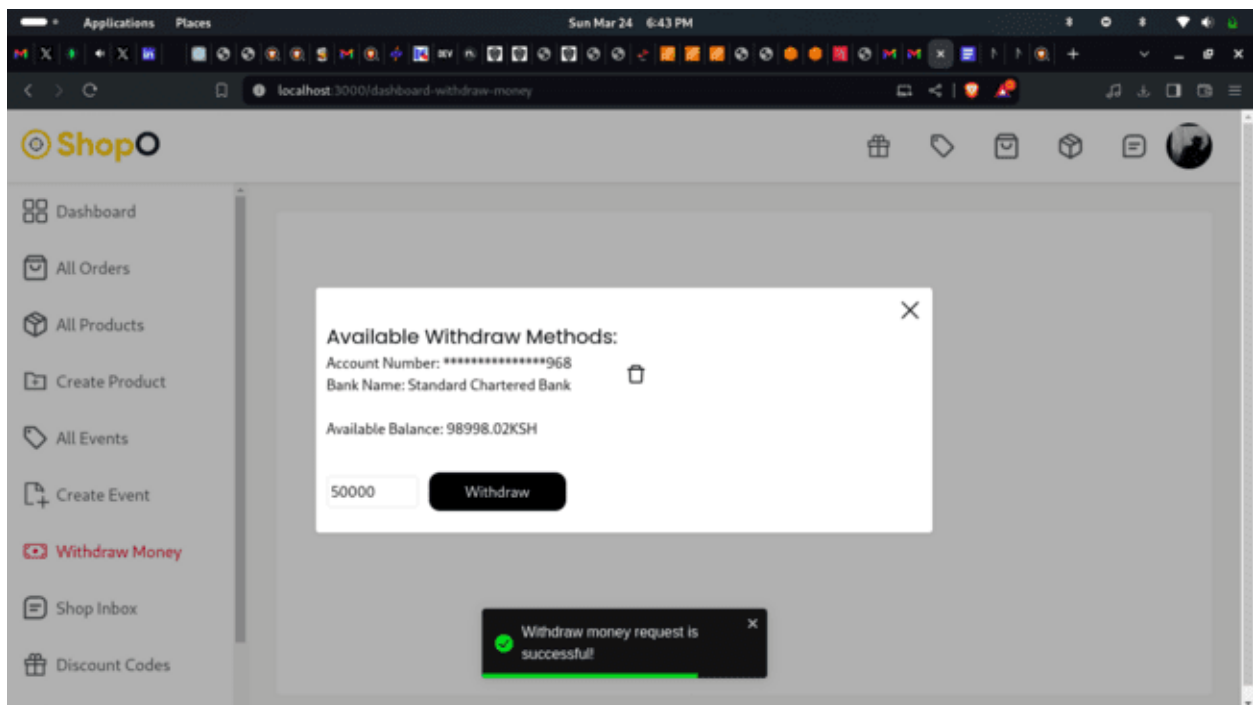
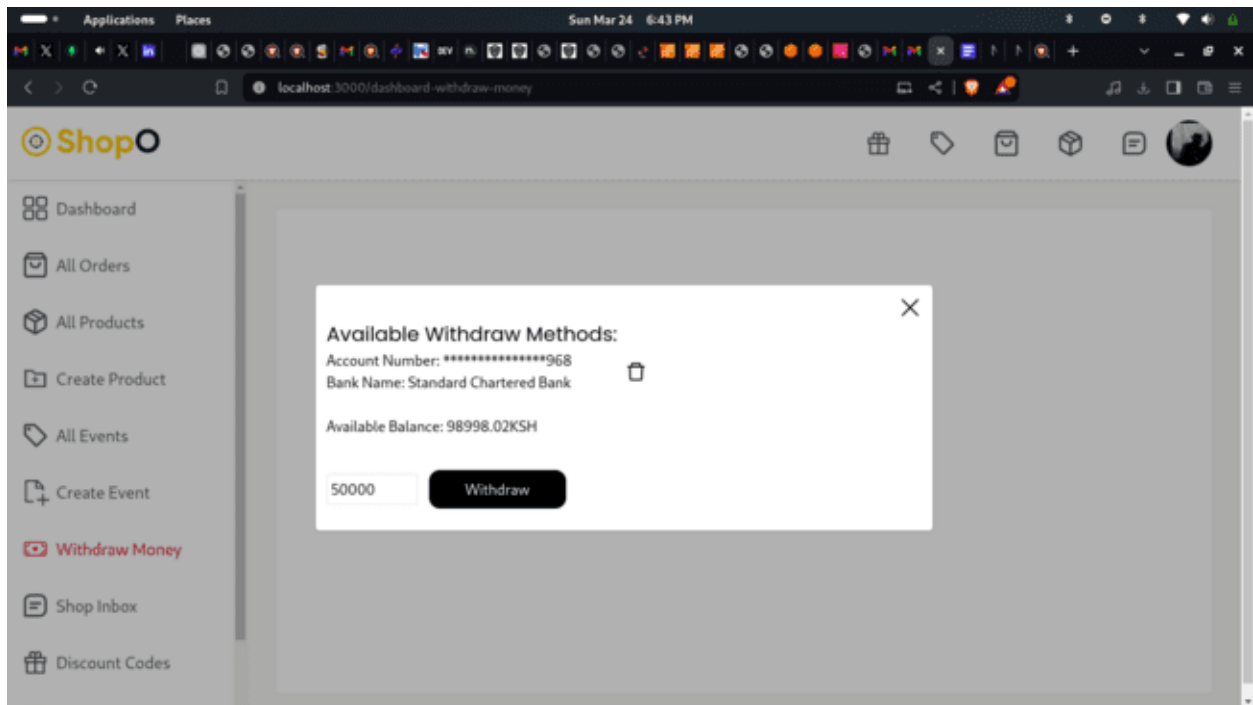
Adding withdrawal method(s)

The image shows two screenshots of the ShopO dashboard. The top screenshot displays the 'Add new Withdraw Method' form with the following fields:

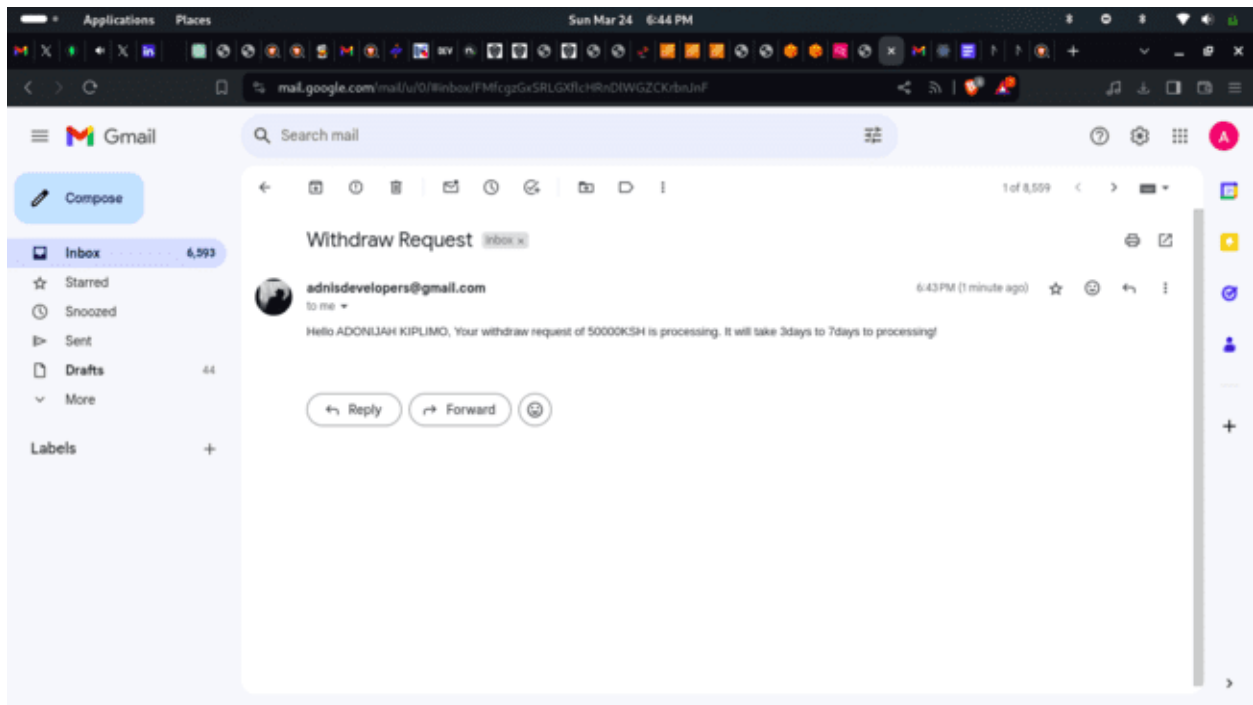
- Bank Name: Standard Chartered Bank
- Bank Country: Kenya
- Bank Swift Code: 3100
- Bank Account Number: 475236986532363968
- Bank Holder Name: Adonijah Kiplimo
- Bank Address: Nairobi Kenya

The bottom screenshot shows the dashboard after the withdrawal method has been added successfully. The available balance is KSH98998.02, and a 'Withdraw' button is visible. A success message is displayed at the bottom: 'Withdraw method added successfully!'.

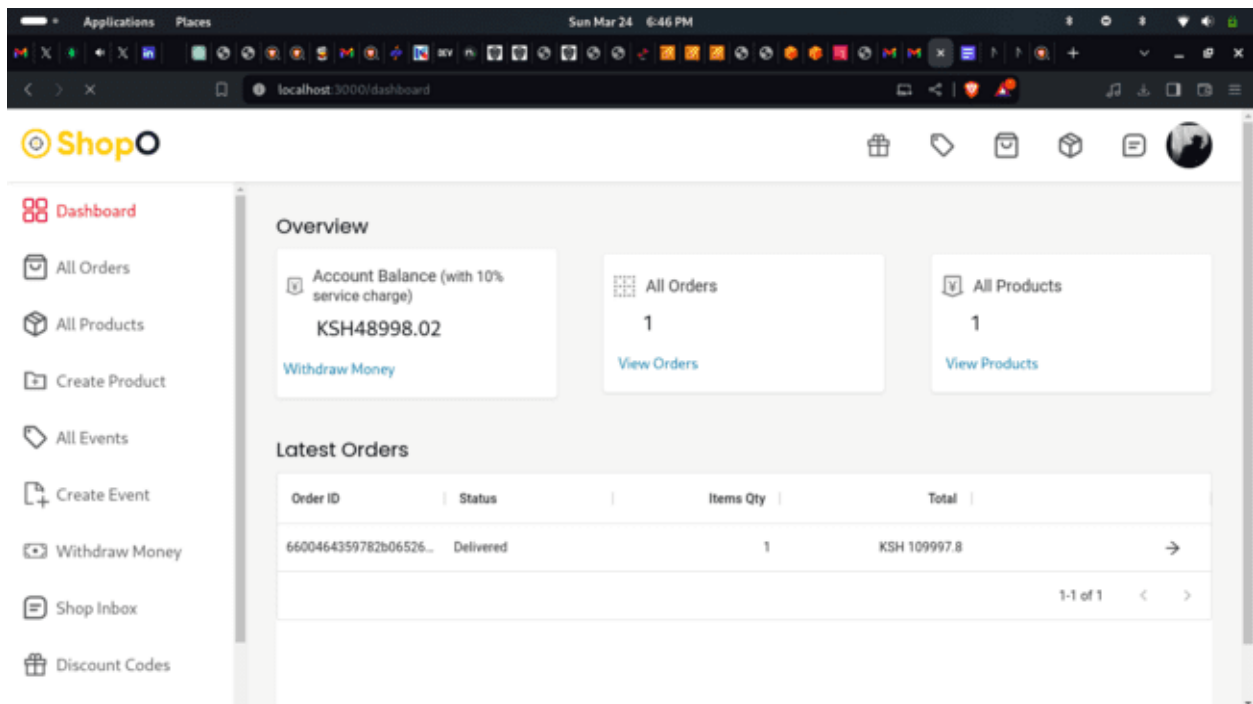
Then Proceed to withdraw the amount by specifying the amount to withdraw.



An email for withdrawal will be sent to the registered email for to let the owner that the amount withdrawn is processing.



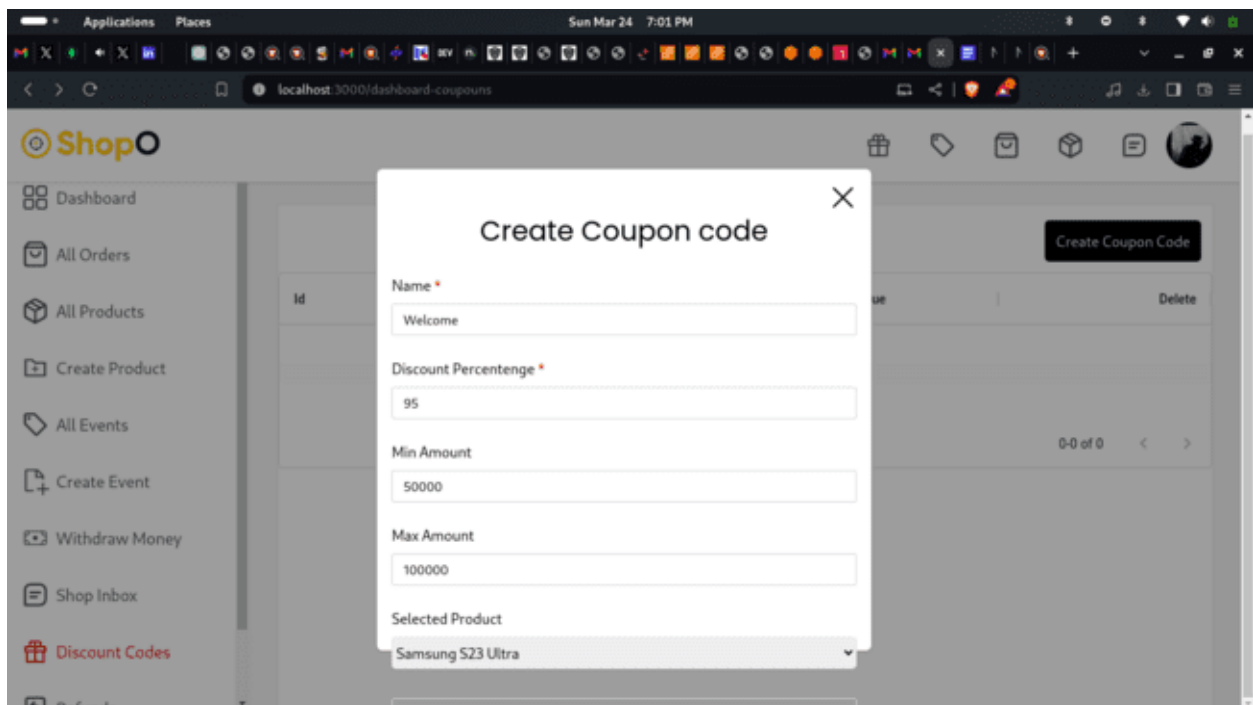
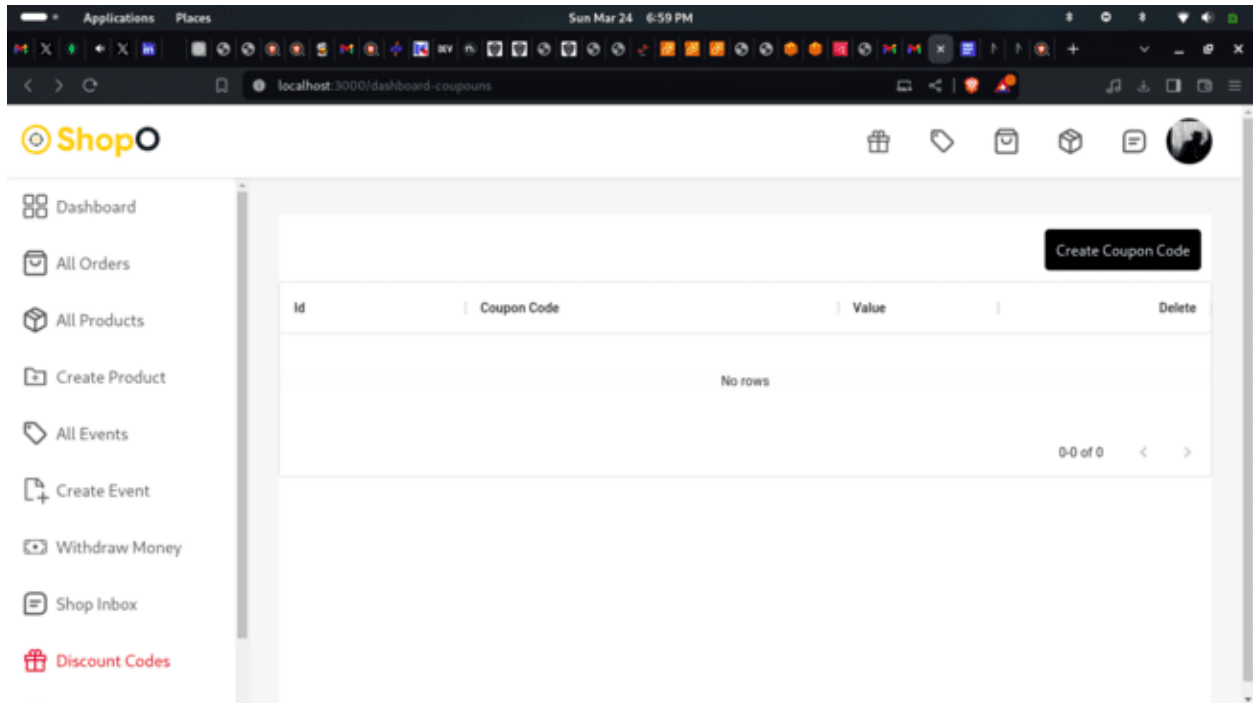
Upon successful withdrawal the balance will reduce with the amount withdrawn.



Coupon creation and use.

The seller can create coupon codes on special days to cut down the cost of specific products for its users.

Create a coupon page.



Applications Places Sun Mar 24 7:01 PM

localhost:3000/dashboard-coupons

Shop

Dashboard

All Orders

All Products

Create Product

All Events

Create Event

Withdraw Money

Shop Inbox

Discount Codes

Dashboard

Create Coupon Code

Id	Coupon Code	Value	Delete
No rows			

0-0 of 0

Coupon code created successfully!

Applications Places Sun Mar 24 7:01 PM

localhost:3000/dashboard-coupons

Shop

Dashboard

All Orders

All Products

Create Product

All Events

Create Event

Withdraw Money

Shop Inbox

Discount Codes

Dashboard

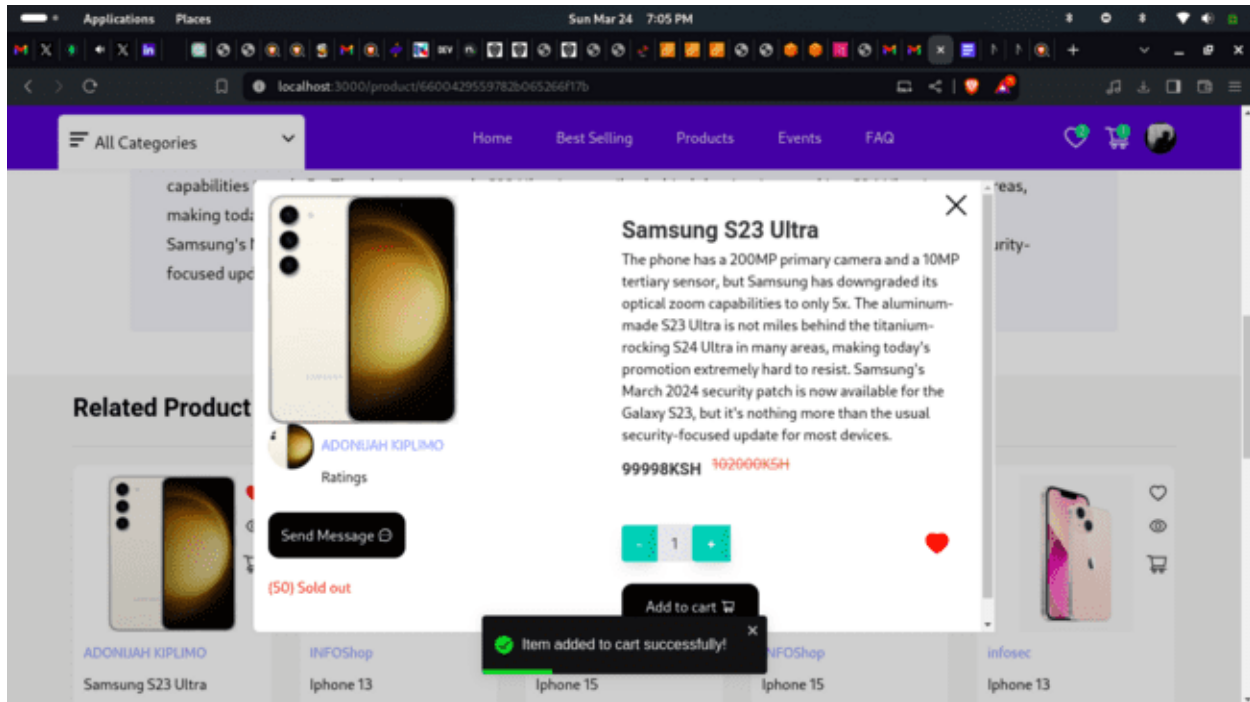
Create Coupon Code

Id	Coupon Code	Value	Delete
66004e5a59782b065266f3...	Welcome	95 %	

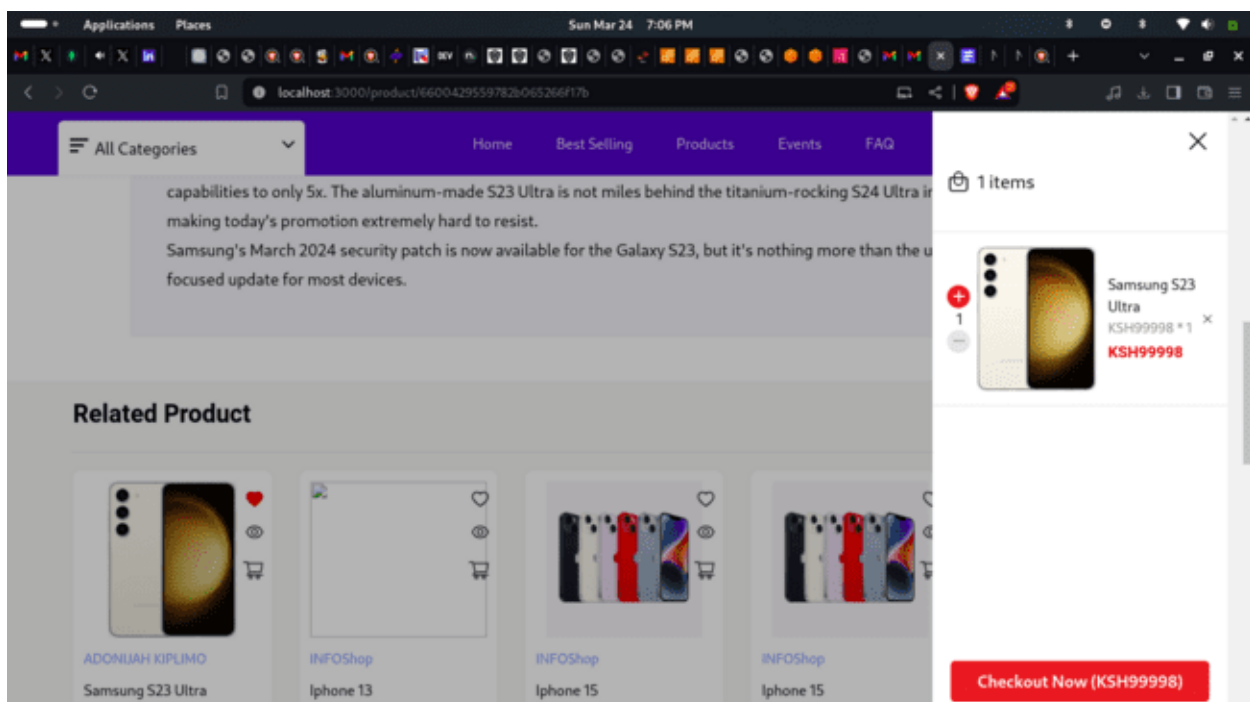
1-1 of 1

Using the coupon code

The user navigates to the products page then selects the product with the coupon code eligibility the proceeds to payment and on the payment page he/she chooses apply coupon then the amount will be deducted according to the amount the coupon covers. Here I used Samsun A23 Ultra product.



Then proceed to the cart.

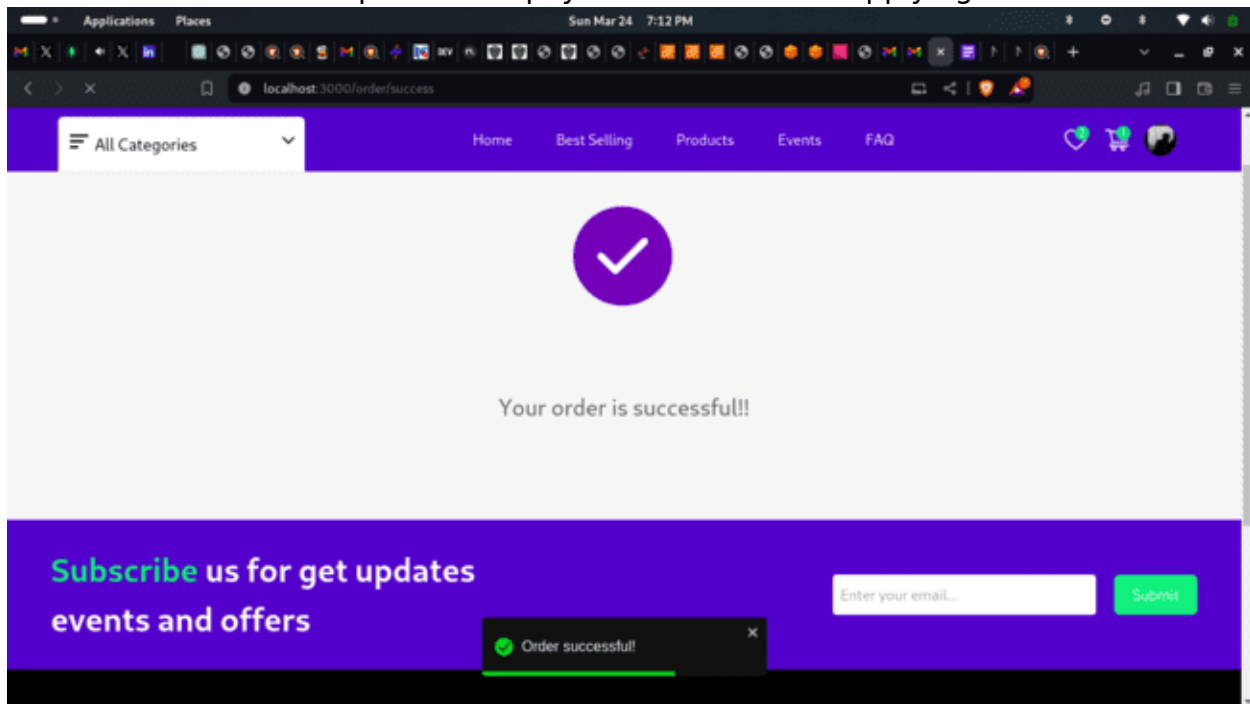


Then on the checkout page I select the Apply coupon button Note the price before I apply coupon code.

The screenshot shows a web browser at localhost:3000/checkout. The page has a purple header with navigation links: Home, Best Selling, Products, Events, and FAQ. On the left, there's a dropdown menu for 'All Categories'. The main content area is divided into two columns. The left column is for the 'Shipping Address' and contains form fields for Full Name (ADONIJAH KIPLIMO), Email Address (adonijakiplimo8@gmail.com), Phone Number (0725796075), Zip Code (00100), Country (Kenya), City (Nairobi City), Address1 (Ruaraka), and Address2 (Ruaraka). Below these fields is a link 'Choose From saved address'. The right column shows the price breakdown: subtotal: KSH99998, shipping: KSH9999.80, and Discount: -. Below this is a coupon code input field with a placeholder 'Please fill out this field.' and an 'Apply code' button. At the bottom center, there is a black button labeled 'Go to Payment'.

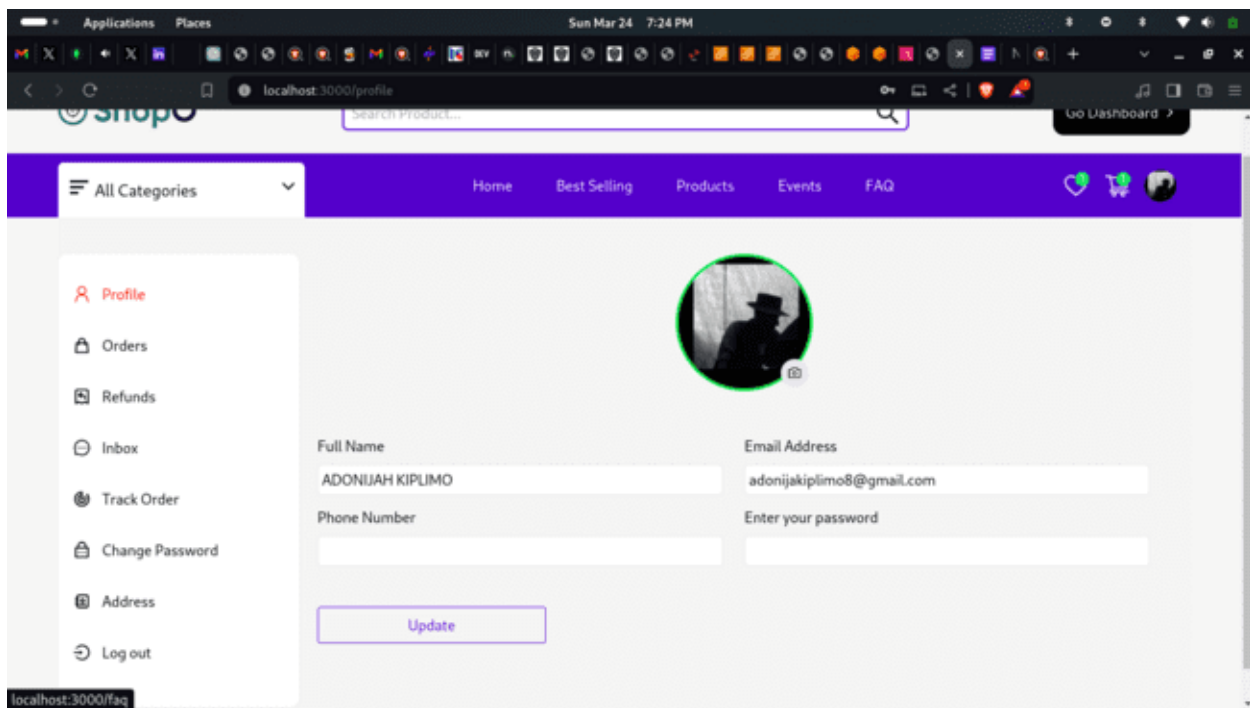
This screenshot shows the same checkout page but with a progress bar at the top indicating the current step: '1.Shipping' (highlighted in red), '2.Payment', and '3.Success'. The 'Shipping Address' form fields are identical to the previous screenshot. The price breakdown on the right is updated: subtotal: KSH99998, shipping: KSH9999.80, and Discount: - KSH94998.1, resulting in a new subtotal of KSH14999.70. The 'Apply code' button remains visible. The 'Go to Payment' button is still present at the bottom.

Then the user can then proceed to pay the amount after applying the discount.



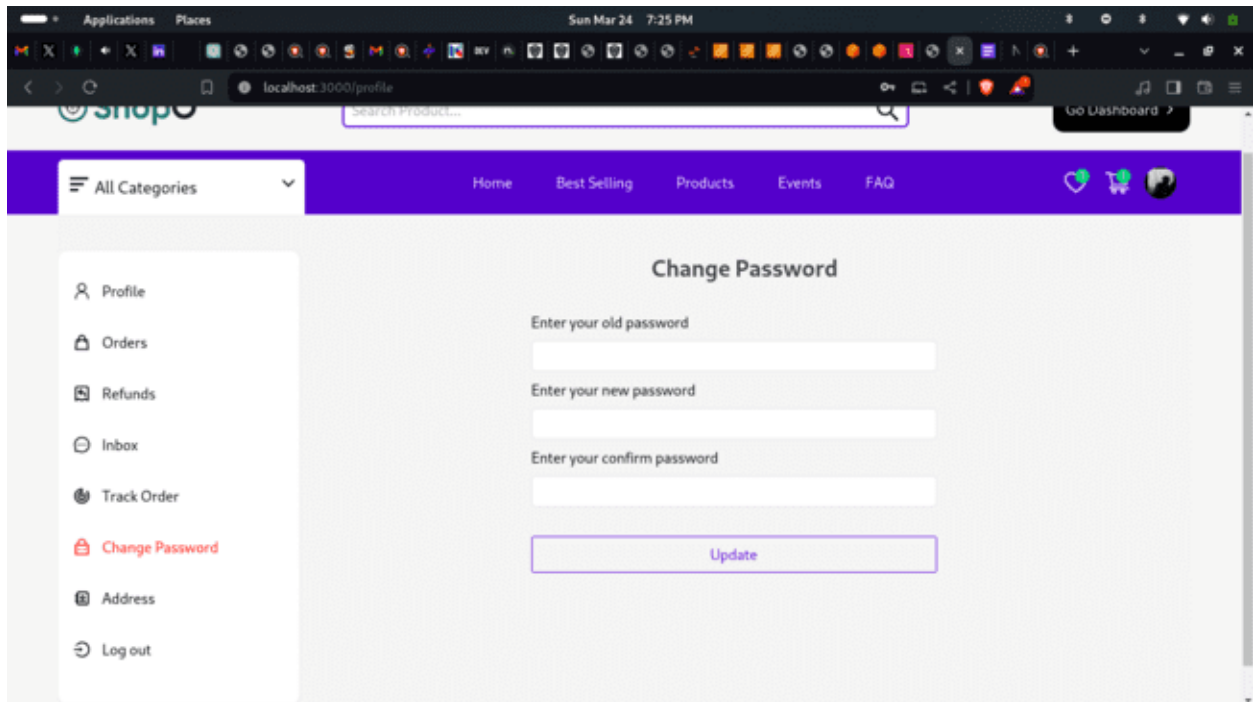
Profile Edit.

Here the user can edit their usernames, emails and profile pictures/avatars.



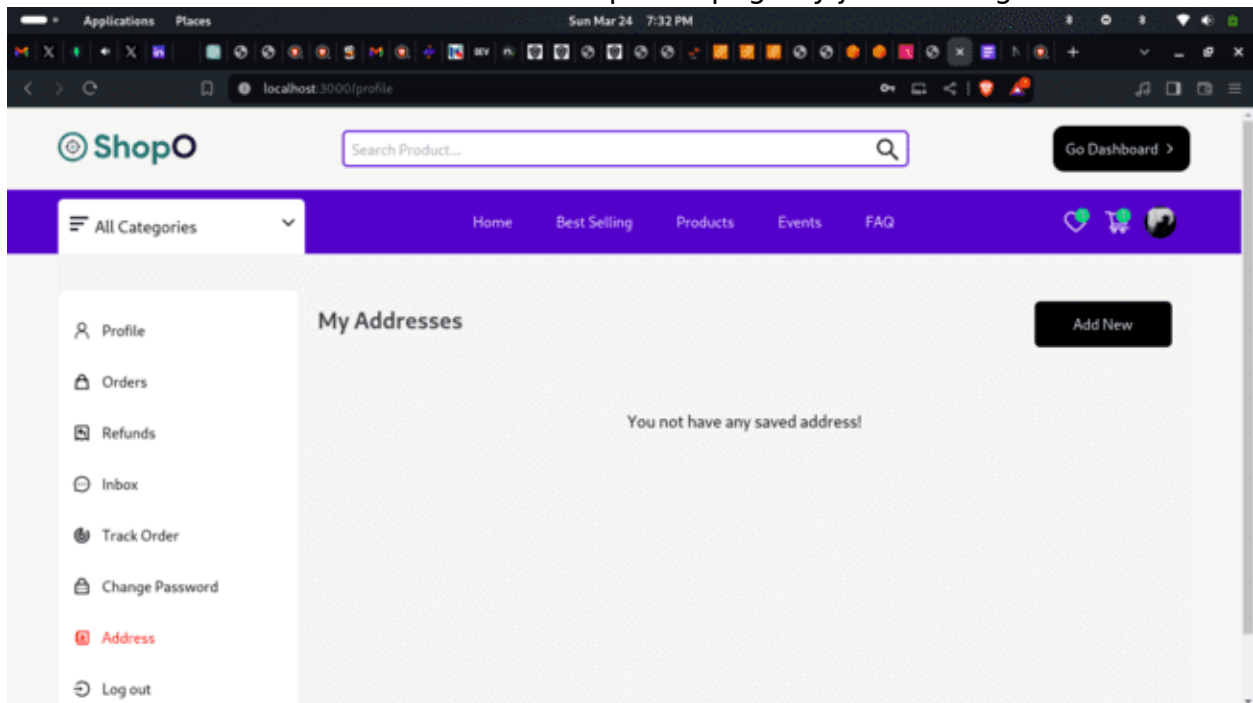
Change Password.

The user can change their passwords as well on the profile section by entering their old password and new password respectively.



Add address

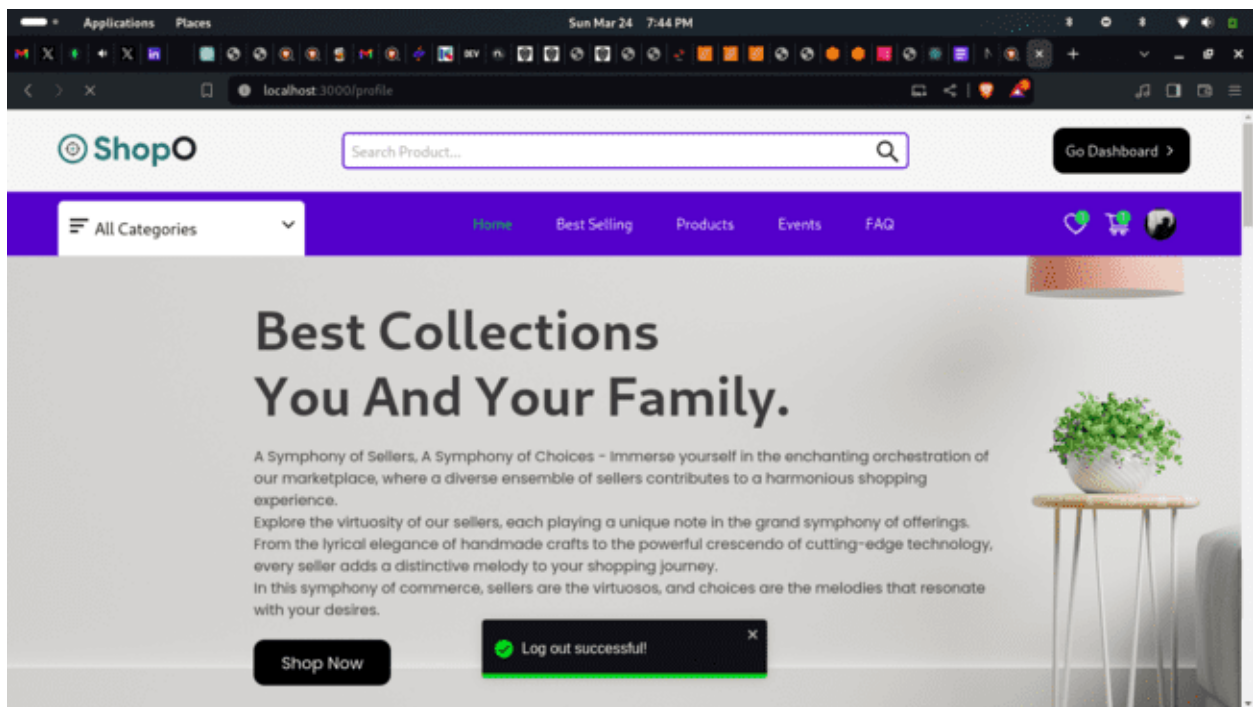
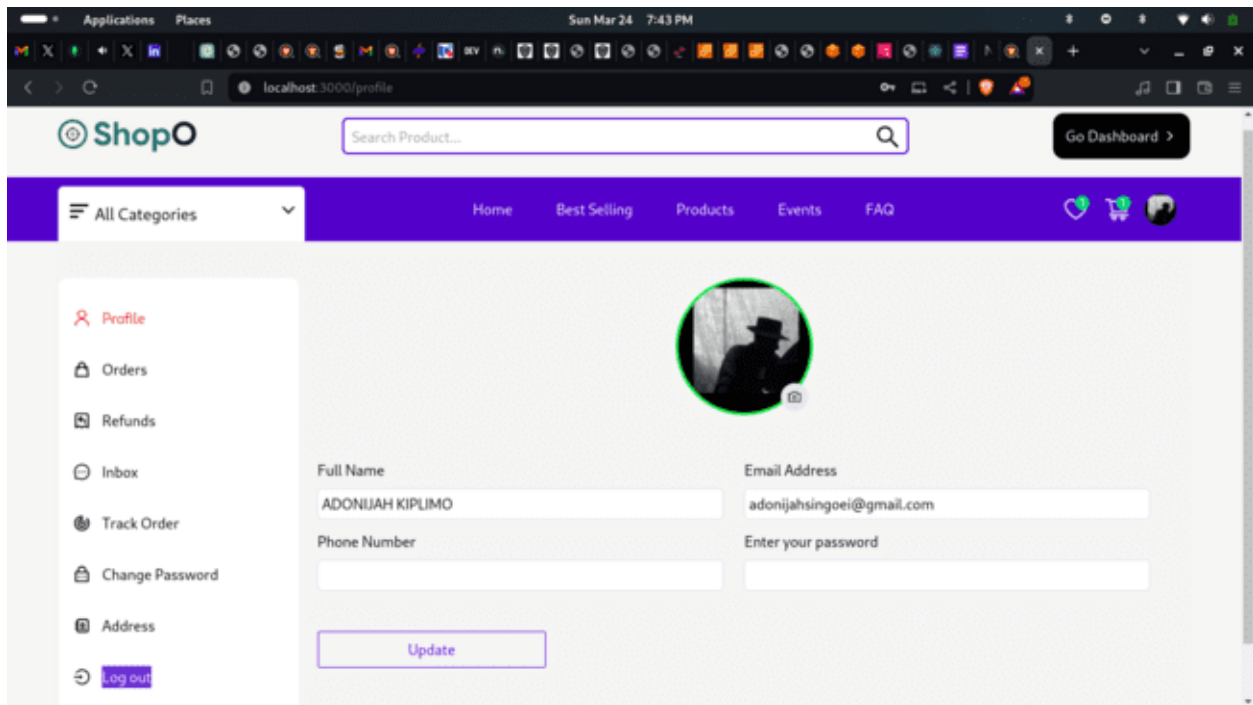
The user can add their address from the profile page by just clicking add address



Logout.

Logout page.

The logged in user can just logout by clicking on the logout button at the bottom of the profile page.



Database

The multi vendor ecommerce web application utilizes MongoDB, a NoSQL database, to store and manage data. The database schema includes collections such as:

Users Collection: Stores information about registered users, including usernames, passwords, and roles (vendor or customer).

Products Collection: Contains details of all products available for sale, including product names, descriptions, prices, and inventory levels.

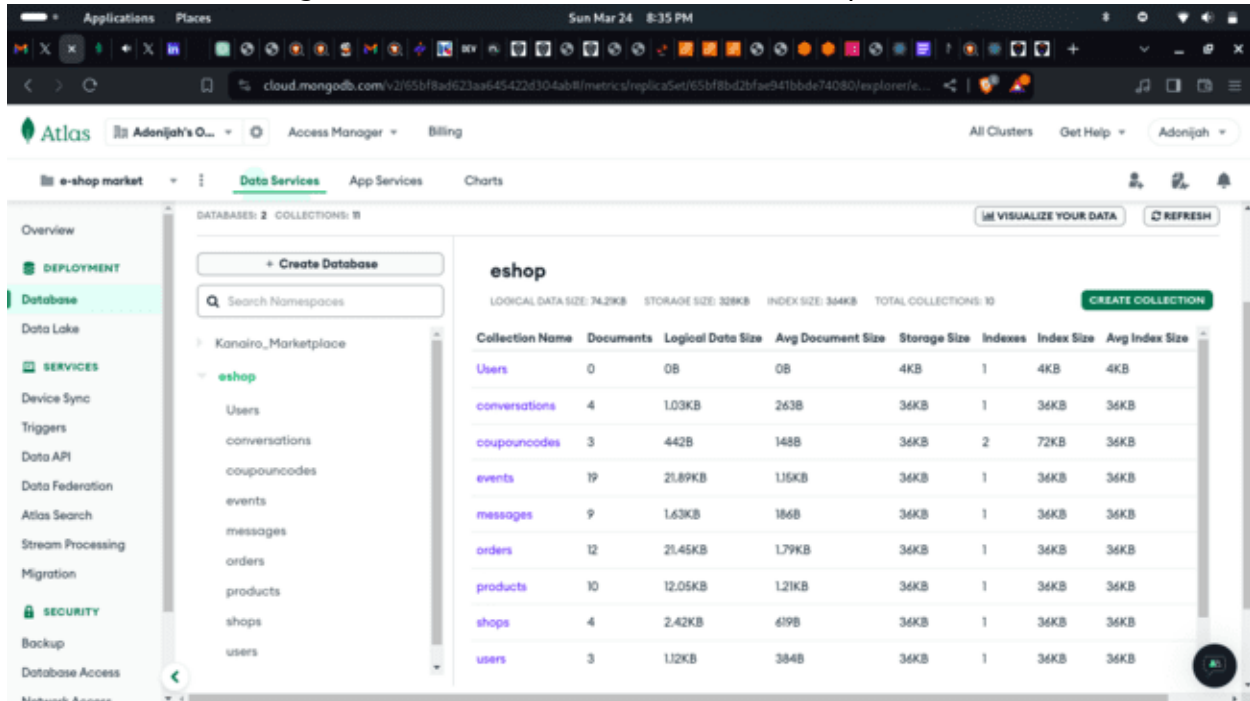
Orders Collection: Tracks orders placed by customers, including order numbers, customer details, product details, and order status.

Reviews Collection: Stores customer reviews and ratings for products.

Integration with Cloudinary

Cloudinary is used for efficient and reliable image storage and management. Product images are uploaded to Cloudinary and referenced within the Products Collection by their respective URLs. This allows for seamless integration of images into the ecommerce platform, enhancing the visual appeal of product listings and improving user engagement.

By leveraging Cloudinary alongside MongoDB, the multi vendor ecommerce web application ensures robust data storage and efficient handling of multimedia content, contributing to a smooth and immersive user experience.



Users document

The screenshot shows the MongoDB Atlas web interface. The left sidebar contains navigation options: Overview, DEPLOYMENT, Database (selected), Data Lake, SERVICES, Device Sync, Triggers, Data API, Data Federation, Atlas Search, Stream Processing, Migration, SECURITY, Backup, Database Access, and Network Access. The main panel displays the 'eshop' database with a list of collections: Users, conversations, couponcodes, events, messages, orders, products, shops, and users (highlighted). The 'users' collection is selected, and the 'Find' tab is active. A filter bar shows a query: `{ field: 'value' }`. Below the filter, the query results show 1-3 of 3 documents. The first document is a user with the following fields: `_id`, `name`, `email`, `password`, `role`, `avatar`, `createdAt`, and `addresses`. The `role` field is highlighted with a red box.

Shops document

The screenshot shows the MongoDB Atlas web interface. The left sidebar is the same as the previous screenshot. The main panel displays the 'eshop' database with a list of collections: Users, conversations, couponcodes, events, messages, orders, products, shops, and users. The 'shops' collection is selected, and the 'Find' tab is active. A filter bar shows a query: `{ field: 'value' }`. Below the filter, the query results show 1-4 of 4 documents. The first document is a shop with the following fields: `_id`, `name`, `email`, `password`, `address`, `phoneNumber`, `role`, `avatar`, `zipCode`, `availableBalance`, and `createdAt`. The `role` field is highlighted with a red box.

Products Document

The screenshot shows the MongoDB Atlas web interface. The left sidebar contains navigation options: Overview, DEPLOYMENT, Database (selected), Data Lake, SERVICES, Device Sync, Triggers, Data API, Data Federation, Atlas Search, Stream Processing, Migration, SECURITY, Backup, Database Access, and Network Access. The main panel displays the 'eshop.products' collection. It includes a search bar, a list of collections (Users, conversations, couponcodes, events, messages, orders, products, shops, users, withdraws), and a document viewer. The document viewer shows a single document with the following fields: `_id`, `name`, `description`, `category`, `tags`, `originalPrice`, `discountPrice`, `stock`, `images`, `shopId`, `shop`, and `sold_out`.

Orders Document

The screenshot shows the MongoDB Atlas web interface. The left sidebar contains navigation options: Overview, DEPLOYMENT, Database (selected), Data Lake, SERVICES, Device Sync, Triggers, Data API, Data Federation, Atlas Search, Stream Processing, Migration, SECURITY, Backup, Database Access, and Network Access. The main panel displays the 'eshop.orders' collection. It includes a search bar, a list of collections (Users, conversations, couponcodes, events, messages, orders, products, shops, users, withdraws), and a document viewer. The document viewer shows a single document with the following fields: `_id`, `cart`, `shippingAddress`, `user`, `totalPrice`, `status`, `paymentInfo`, `paidAt`, `createdAt`, and `__v`.

Messages Document

The screenshot shows the MongoDB Atlas web interface. On the left, a sidebar contains navigation links for Overview, Deployment, Database, Data Lake, Services, and Security. The 'Database' section is expanded, showing a list of collections under the 'eshop' namespace: Users, conversations, couponcodes, events, messages (highlighted), orders, products, shops, users, and withdraws. The main panel displays the 'eshop.messages' collection. At the top, it shows metadata: STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 143KB, TOTAL DOCUMENTS: 9, and INDEXES TOTAL SIZE: 36KB. Below this are tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, and Search Indexes. A search bar with the placeholder 'Type a query: { field: 'value' }' is present. Below the search bar, the 'QUERY RESULTS: 1-9 OF 9' are displayed. The first document is shown in a JSON format:

```
{
  "_id": ObjectId('65c8ad2642f5402530b569b6'),
  "conversationId": "65c8ad2342f5402530b569aa",
  "text": "hello",
  "sender": "65bfa3e0c5a989b9b68cd541",
  "createdAt": 2024-02-09T09:41:01.151+00:00,
  "updatedAt": 2024-02-09T09:41:01.151+00:00,
  "__v": 0
}
```

Events Document

The screenshot shows the MongoDB Atlas web interface, similar to the previous one, but displaying the 'eshop.events' collection. The sidebar and navigation are the same. The main panel shows the 'eshop.events' collection with metadata: STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 218KB, TOTAL DOCUMENTS: 19, and INDEXES TOTAL SIZE: 36KB. The 'Find' tab is selected. A search bar is present. Below it, the 'QUERY RESULTS: 1-19 OF 19' are displayed. The first document is shown in a JSON format:

```
{
  "_id": ObjectId('65fd96b395b2f9c34c7e032d'),
  "name": "Holiday Discount",
  "description": "Holiday DiscountHoliday DiscountHoliday Discount",
  "category": "Others",
  "start_date": 2024-03-22T00:00:00.000+00:00,
  "finish_date": 2024-03-25T00:00:00.000+00:00,
  "status": "Running",
  "tags": "Holiday DiscountHoliday Discount",
  "originalPrice": 2500,
  "discountPrice": 2498,
  "stock": 2500
}
```

Coupon codes document

The screenshot shows the MongoDB Atlas web interface. The left sidebar contains navigation options: Overview, DEPLOYMENT, Database (selected), Data Lake, SERVICES, and SECURITY. Under 'Database', the 'eshop' namespace is expanded, showing collections: Users, conversations, couponcodes (highlighted), events, messages, orders, products, shops, users, and withdraws. The main panel displays the 'eshop.couponcodes' collection. It shows metadata: STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 442B, TOTAL DOCUMENTS: 3, INDEXES TOTAL SIZE: 72KB. Below this are tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, and Search Indexes. A 'Filter' bar contains the query: { field: 'value' }. The 'QUERY RESULTS: 1-3 OF 3' section shows a single document:

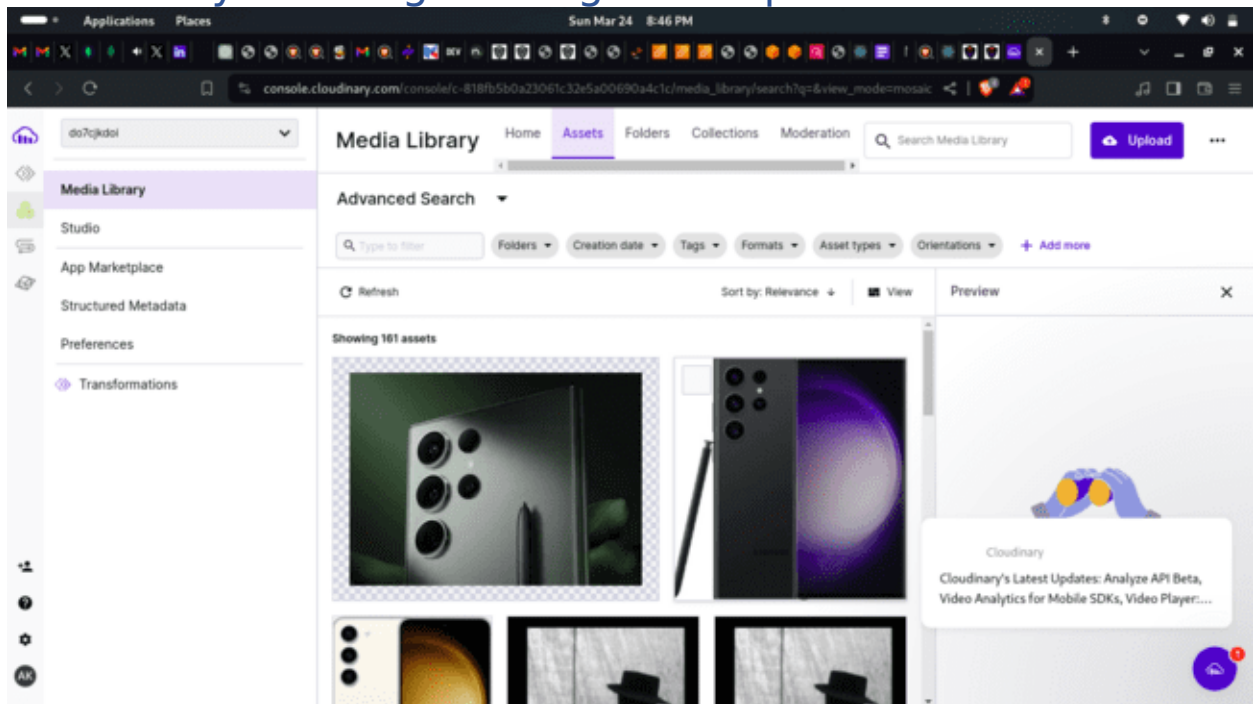
```
{
  "_id": ObjectId("65fd7fb7094f031b2c4225e"),
  "name": "Discount",
  "value": 50,
  "minAmount": 48,
  "maxAmount": 490,
  "shopId": "65d2e9d87355f4c7b57a4ed",
  "createdAt": 2024-03-22T12:34:19.344+00:00,
  "__v": 0
}
```

Conversations Document

The screenshot shows the MongoDB Atlas web interface. The left sidebar is the same as the previous screenshot. The main panel displays the 'eshop.conversations' collection. It shows metadata: STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 100KB, TOTAL DOCUMENTS: 4, INDEXES TOTAL SIZE: 36KB. Below this are tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, and Search Indexes. A 'Filter' bar contains the query: { field: 'value' }. The 'QUERY RESULTS: 1-4 OF 4' section shows a single document:

```
{
  "_id": ObjectId("65c8ad2342f54025300569aa"),
  "groupTitle": "65c8acb442f54025305698865bfa3eeca909b9b68cd541",
  "members": Array (2),
  "createdAt": 2024-02-05T09:40:51.964+00:00,
  "updatedAt": 2024-02-19T06:29:40.705+00:00,
  "__v": 0,
  "lastMessage": "yohhhh",
  "lastMessageId": "65bfa3eeca909b9b68cd541"
}
```


Cloudinary for image storage and optimization.



Error Message

Error messages may appear when users encounter issues such as incorrect login credentials, invalid inputs, or system errors. Users should follow the instructions provided in the error messages to resolve the issues.

Troubleshooting- Problems

The table below shows the error or problems that may occur, possible causes and measures to be taken.

NO.	PROBLEMS	POSSIBLE CAUSES	MEASURES
1	Software not working	Too much system overload insufficient memory or using a corrupt operating system	Repair and activate the operating system Check memory capacity and run disk cleanup wizard.

2	PC shutdown or logoff	Power failure which may be due to faulty power supply or power loss	Ensure there is power on the main supply. Check your power supply unit. Check power cables.
3	Display of irrelevant message on the screen or deletion of data spontaneously	Attack of virus to the system	Run anti-virus software and keep the systems up to date.

Troubleshooting

In case of system errors or malfunctions, users can follow the troubleshooting steps below:

Check System Requirements: Ensure that your system meets the minimum requirements for running the application, including hardware specifications and software dependencies.

Clear Browser Cache: Clearing the browser cache can help resolve display issues or loading errors caused by cached data.

Check Internet Connection: Verify that your internet connection is stable and functioning properly to ensure seamless interaction with the application.

Contact Support: If the issue persists, users can contact technical support for assistance. Provide detailed information about the problem encountered to facilitate troubleshooting.

Back up

It is highly recommended that you have to backup copies of the program files and system reports to ensure security of all data. This is done by copying the original disk to another disk or to an external storage using the copy facility. Contact the system developer in case of the system failure.

Conclusions and Recommendations

The multi vendor ecommerce web application built using the MERN stack is a powerful tool for facilitating online commerce. Moving forward, it is recommended to continuously improve the application's features and user experience based on

feedback from users. Additionally, regular maintenance and updates should be performed to ensure optimal performance and security of the application.