NATIONAL UNIVERSITY

SamutSari: A Barangay Digital Marketplace and Skills Hub

Submitted by:
Aceveda, Ben Christian G.
Bautista, Charles Denzel A.
Gabion, Christian Jeff L.
Morales, Aaron Gabriel D.
Valencia, Hans Aaron N.

Submitted to:
Ms. Anjelica Castillo
October 18, 2025

Mission and Vision

Mission:
Our mission is to empower local communities by creating a simple, accessible, and affordable digital marketplace that connects residents with vendors, freelancers, and service providers. We aim to provide a trusted platform where people can buy, sell, and trade goods and services with ease, while also promoting digital literacy, encouraging entrepreneurship, and fostering stronger community connections.

Vision:
Our vision is to build a sustainable, community-led platform that strengthens local economies and opens new opportunities for collaboration and growth in every barangay. We aim to uplift small business owners, give residents the tools to succeed in the digital space, and promote wider participation by creating a trusted system where communities can progress together and secure lasting prosperity.

Objectives

General Objective
The general objective of the team is to develop a community-based e-commerce web platform that connects local vendors, freelancers, and service providers with residents, while promoting secure transactions, affordable logistics, and inclusive digital participation.

Specific Objectives
The specific objectives of the project are as follows:

1. To develop a user-friendly e-commerce website that functions smoothly on both desktop and mobile devices.
2. To provide a central hub where vendors and service providers can showcase products, services, availability, and customer feedback.
3. To integrate secure and convenient digital payment methods such as GCash, Maya, and Cash-on-Delivery.
4. To introduce affordable and efficient logistics support, including subscription options for essentials, and real-time tracking.
5. To promote digital inclusion through basic training modules that enhance residents' skills in online selling, customer service, and digital literacy.

SWOT Analysis

Strengths:

1. Localized approach – Designed specifically for barangays, making it highly relevant.
2. Affordable & accessible – Low-cost hosting and minimal fees keep it sustainable.
3. All-in-one platform – Combines product selling, services/skills hub, and logistics in one place.
4. Community trust – Familiarity within the barangay encourages adoption.
5. User-friendly design – Lightweight interface, accessible even with limited mobile data.
6. Cash & e-wallet options – Flexibility through GCash, Maya, or COD.

Weaknesses:

1. Limited technical literacy – Some residents may struggle with digital selling.
2. Small initial user base – Adoption might be slow at the start.
3. Dependence on internet connectivity – Unstable mobile data could affect usage.
4. Limited funding – Budget constraints may restrict features and marketing reach.
5. Maintenance & support – Volunteers may lack long-term resources for upkeep.

Opportunities:

1. Digital inclusion – Helps barangays join the digital economy.
2. Job creation – Residents can monetize skills (repairs, tutoring, tailoring, freelancing).
3. Partnerships – Possible collaboration with LGUs, cooperatives, or local businesses.
4. E-commerce growth trend – Taps into the growing Filipino habit of online buying and selling.
5. Training & education – Opportunity to teach digital skills and entrepreneurship.

Threats:

1. Competition – Larger platforms (Facebook Marketplace, Lazada, Shopee) may overshadow adoption.
2. Cybersecurity risks – Fraud, scams, or privacy concerns could harm trust.
3. Resistance to change – Some residents may prefer traditional selling/buying methods.
4. Sustainability challenges – Without ongoing funding or barangay support, the project may stall.

System Architecture Diagram

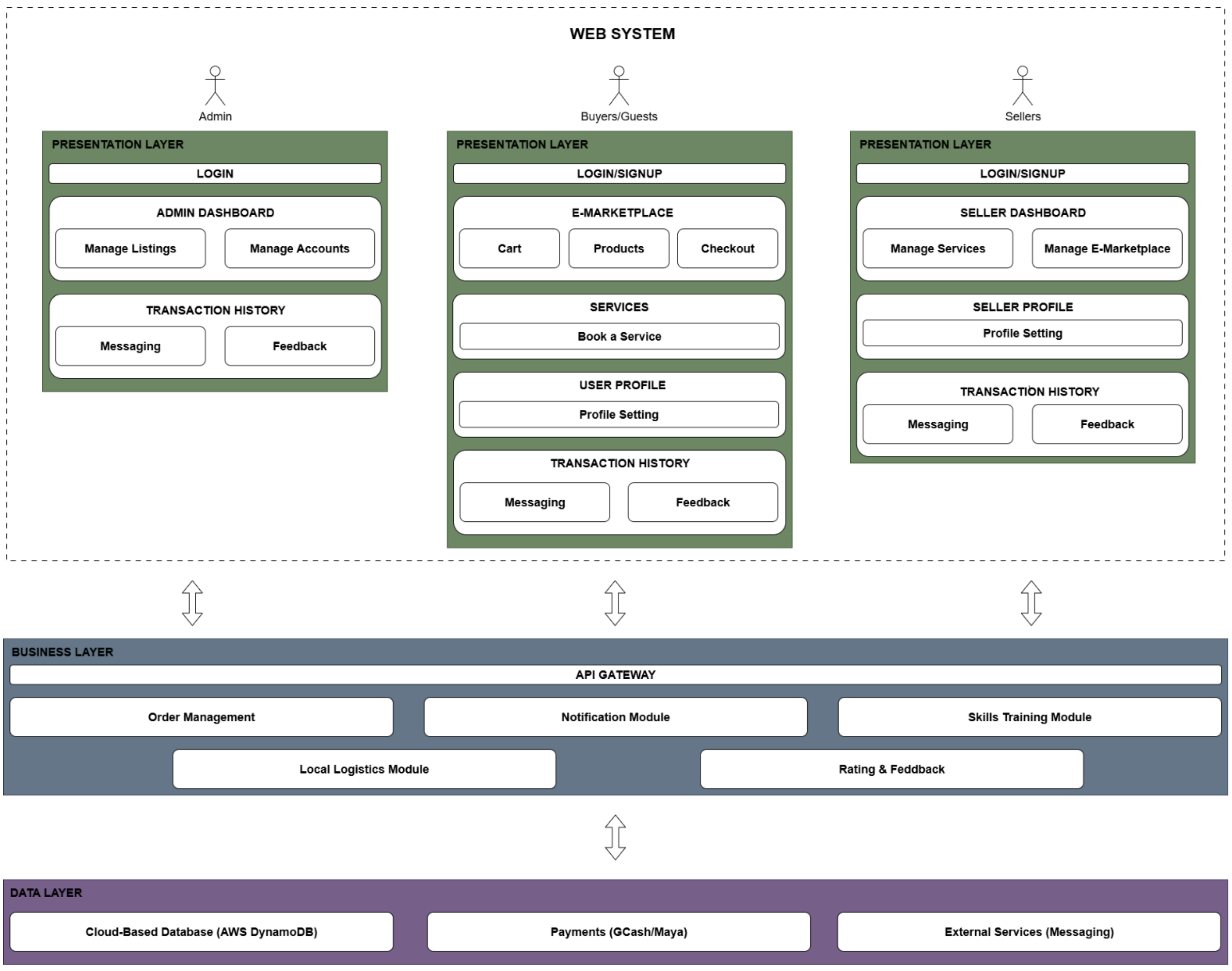
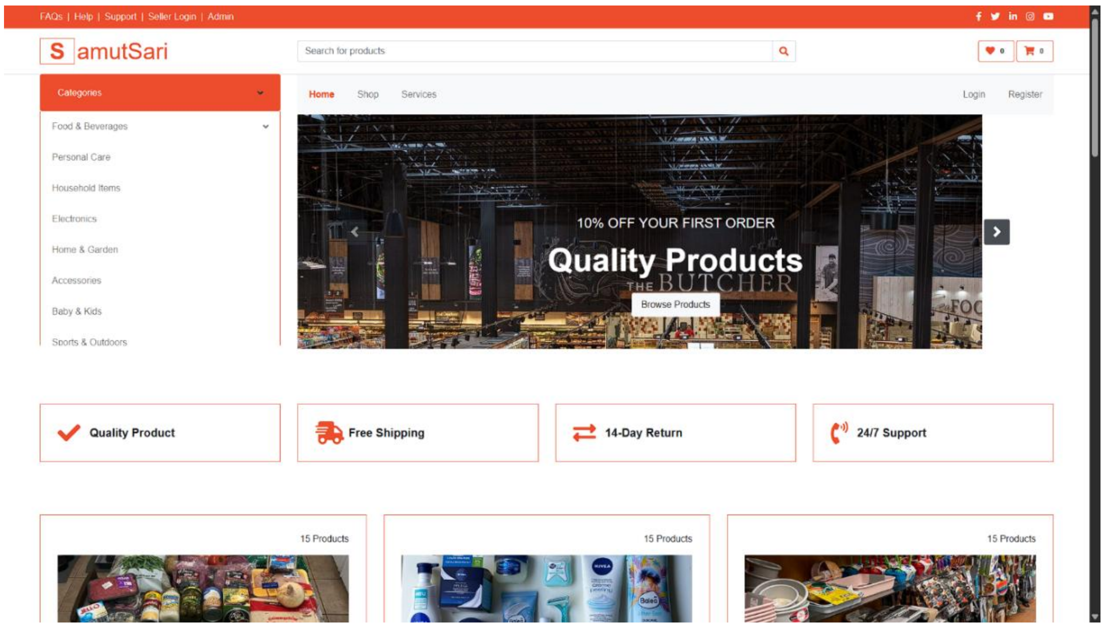


Figure: Overall System Architecture

The Figure shows the system architecture of the Barangay Web System, composed of three main layers: the Presentation Layer, Business Layer, and Data Layer. The Presentation Layer provides differentiated access for Admins, Buyers/Guests, and Sellers through their respective dashboards, enabling management of listings, accounts, marketplace activities, services for freelance work, and transaction history. The Business Layer, powered by an API Gateway, handles order management, notifications, logistics, skills training, and ratings/feedback. The Data Layer integrates cloud-based databases, payment gateways (GCash/Maya), and external messaging services to ensure secure data storage, reliable transactions, and communication support.

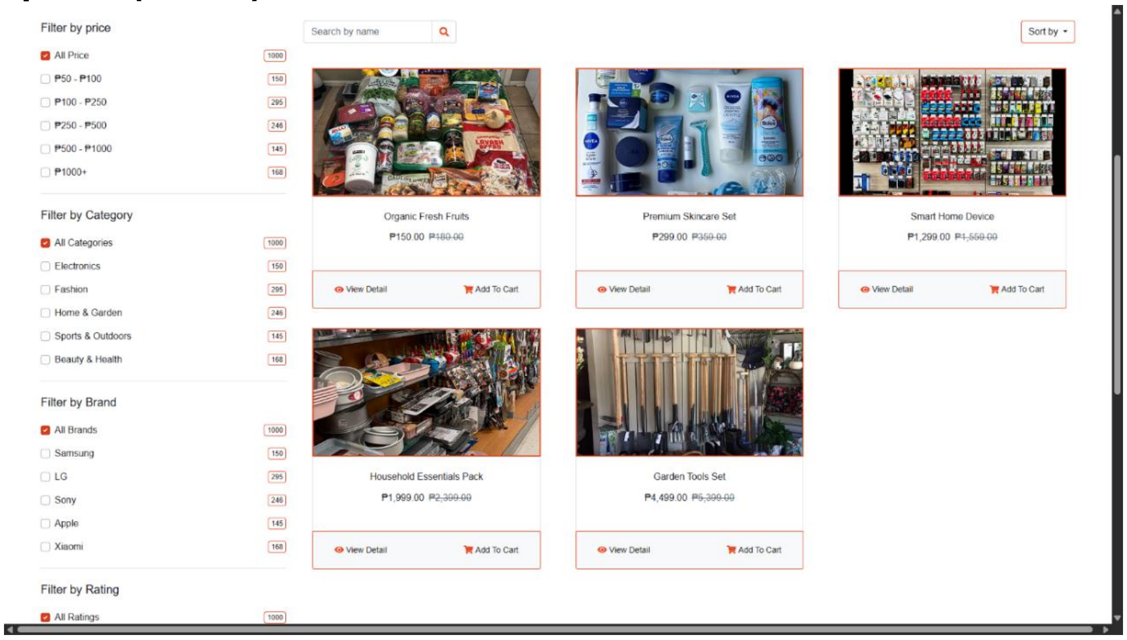
User Interface

Homepage (Guest)



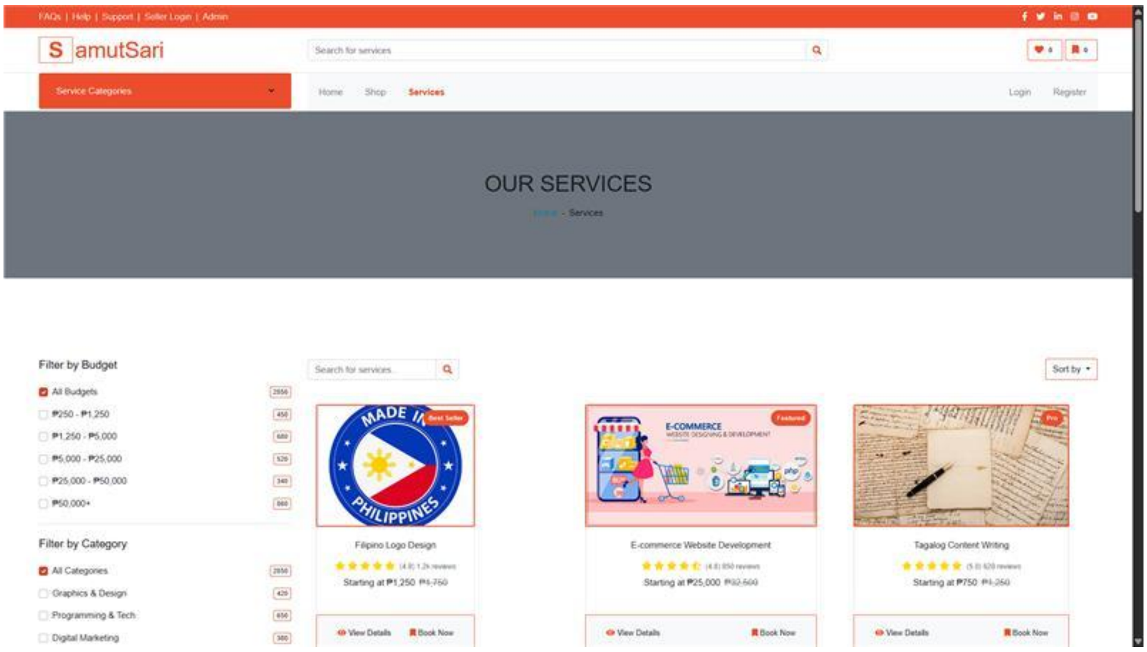
This page is the landing page when searching the website. This provides the guest users to view some products available in the shop. This page also allows the guest, the seller, and the admin to access their respective login pages.

Marketplace (Guest)



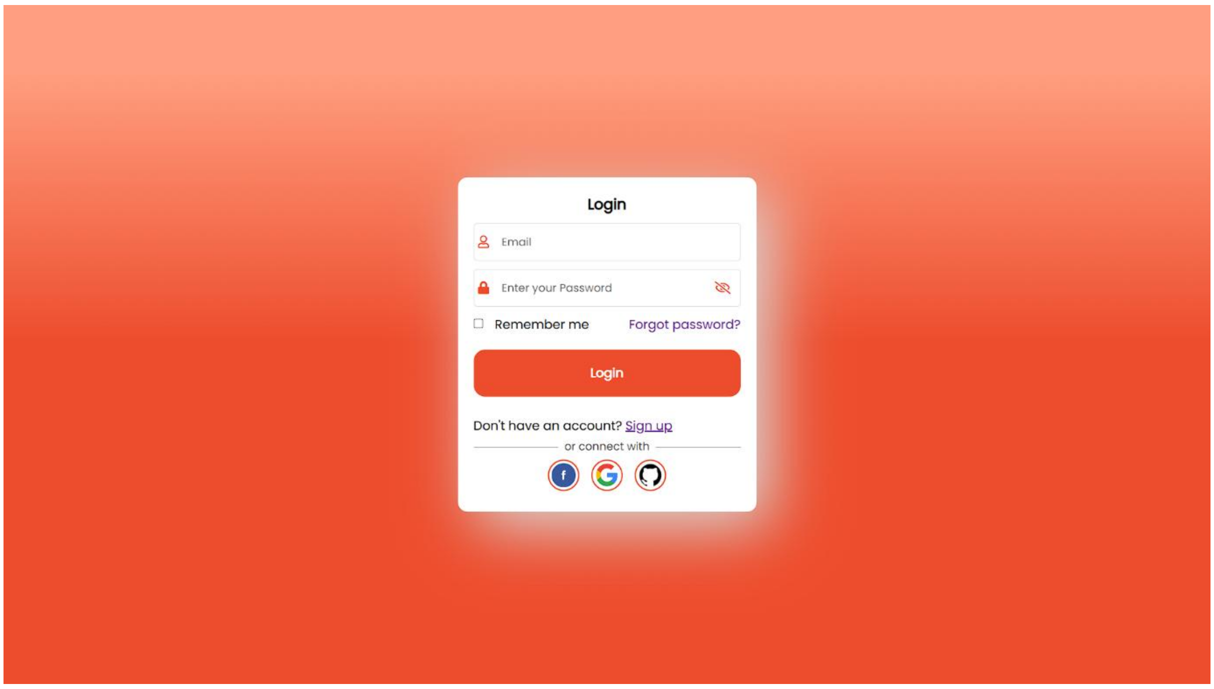
This page is the guest shop page that allows the users to view specific items. This page has filters to help the users find their specific item of choice. This page also allows the guest, the seller, and the admin to access their respective login pages.

Services (Guest)



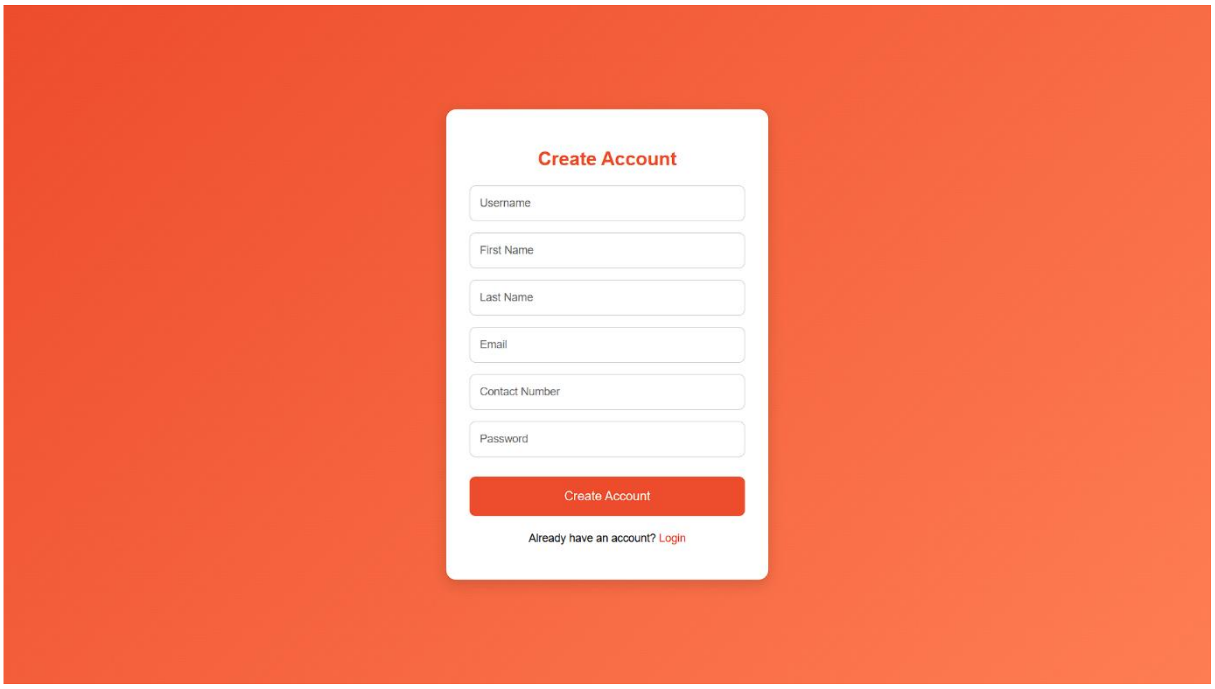
This page allows guest users to view the services offered in the app by service providers and freelancer.

Login



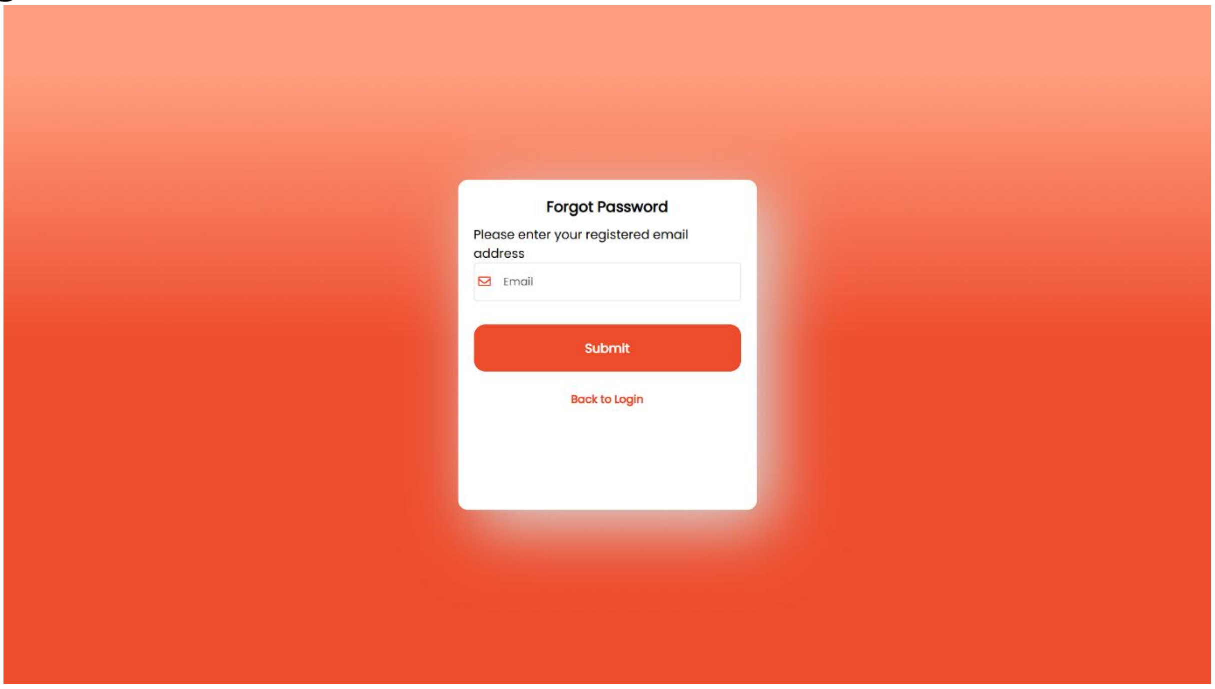
This page allows guest to log in to their account.This page also allows guest to change their password and to sign up.

Signup



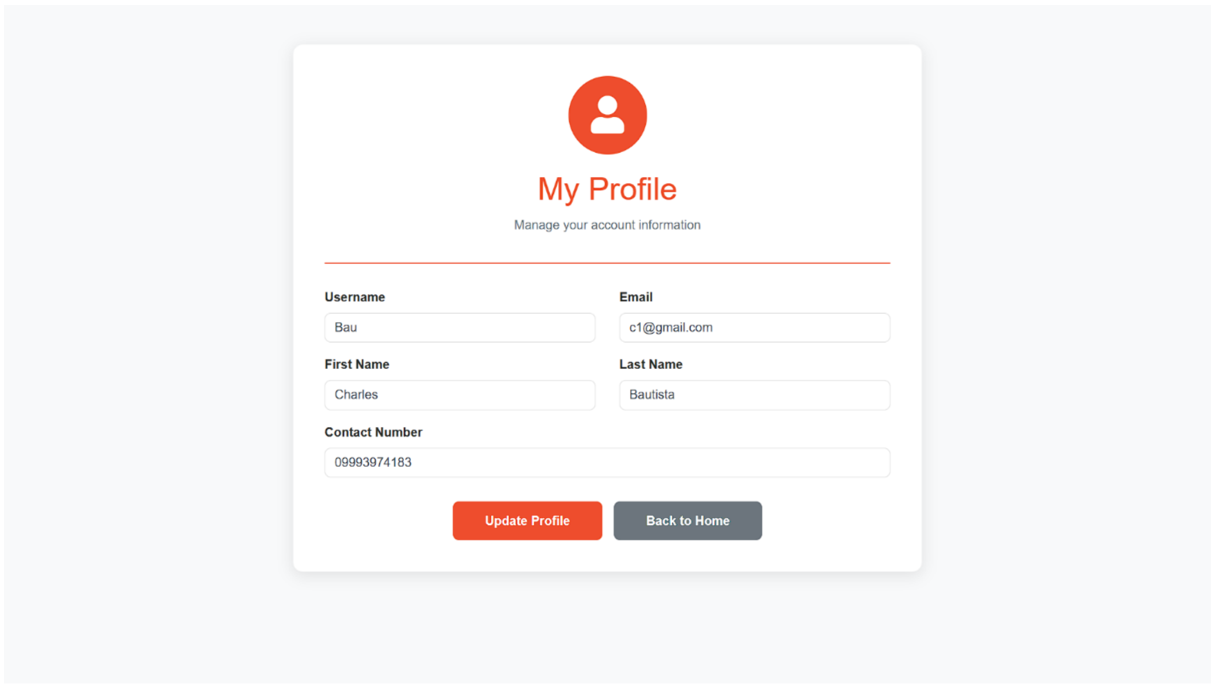
This page allows guests to create the SamutSari account

Forgot Password



This page allows guests to change the password of their SamutSari account using their email address.

User Profile


My Profile

Manage your account information

Username: Bau

Email: c1@gmail.com

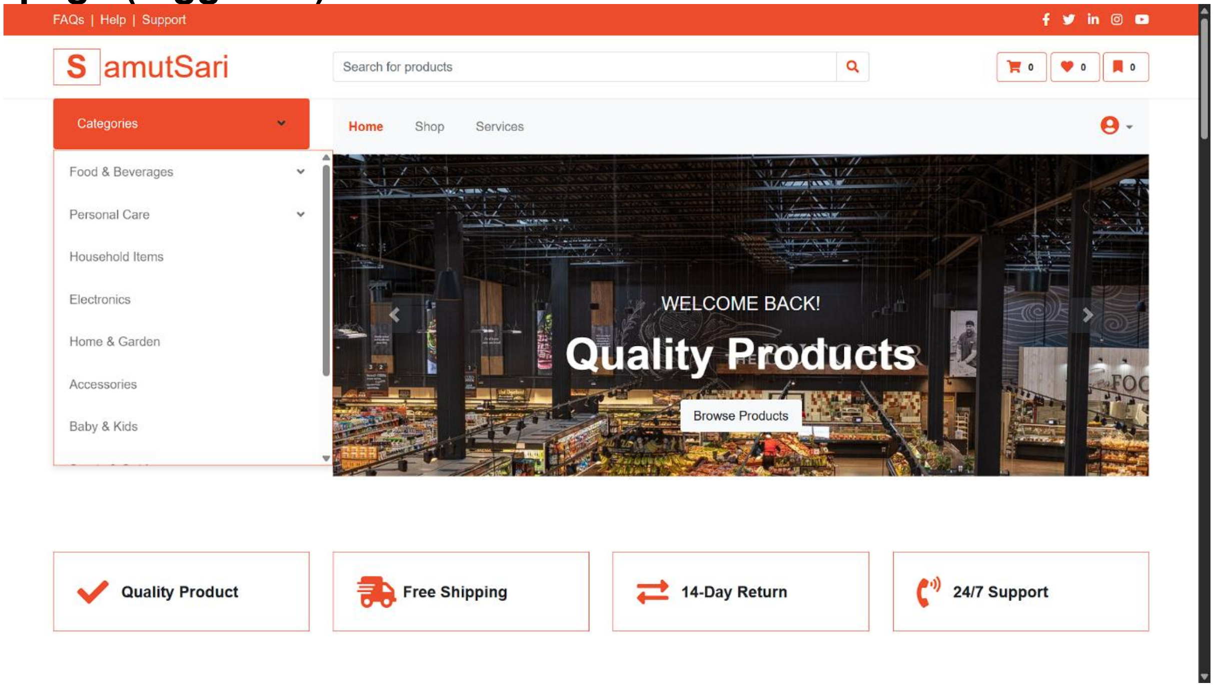
First Name: Charles

Last Name: Bautista

Contact Number: 09993974183

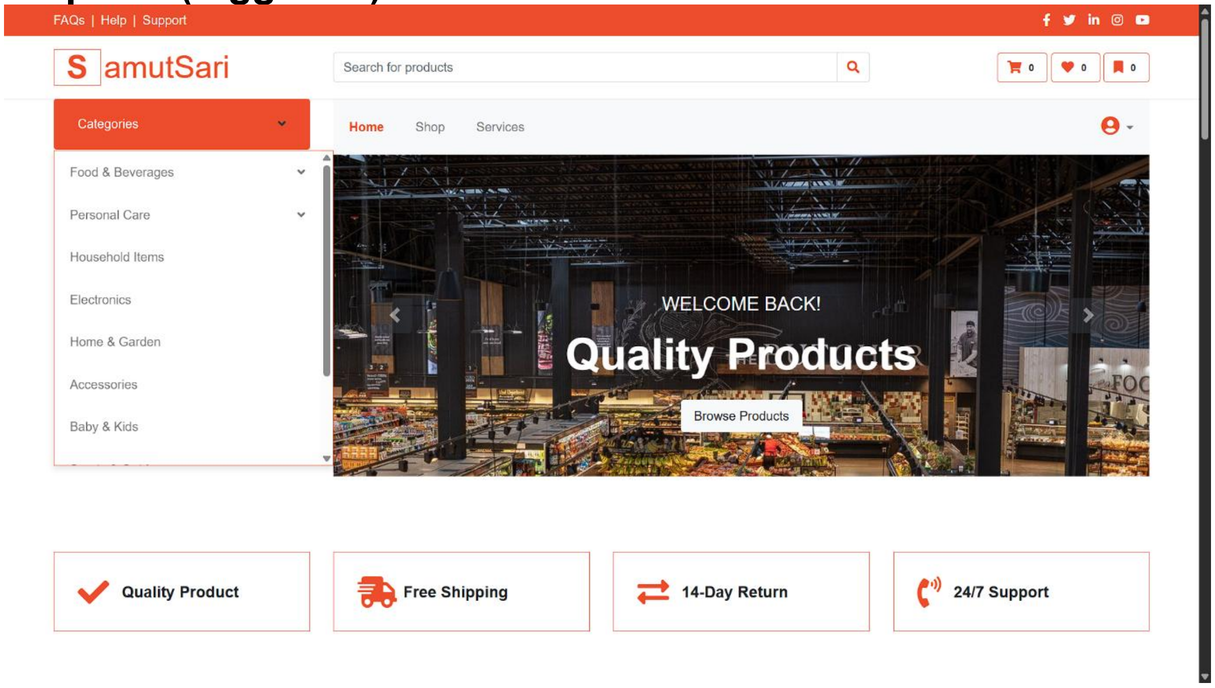
This page allows guests to view their profile information and be able to update their information.

Homepage (logged in)



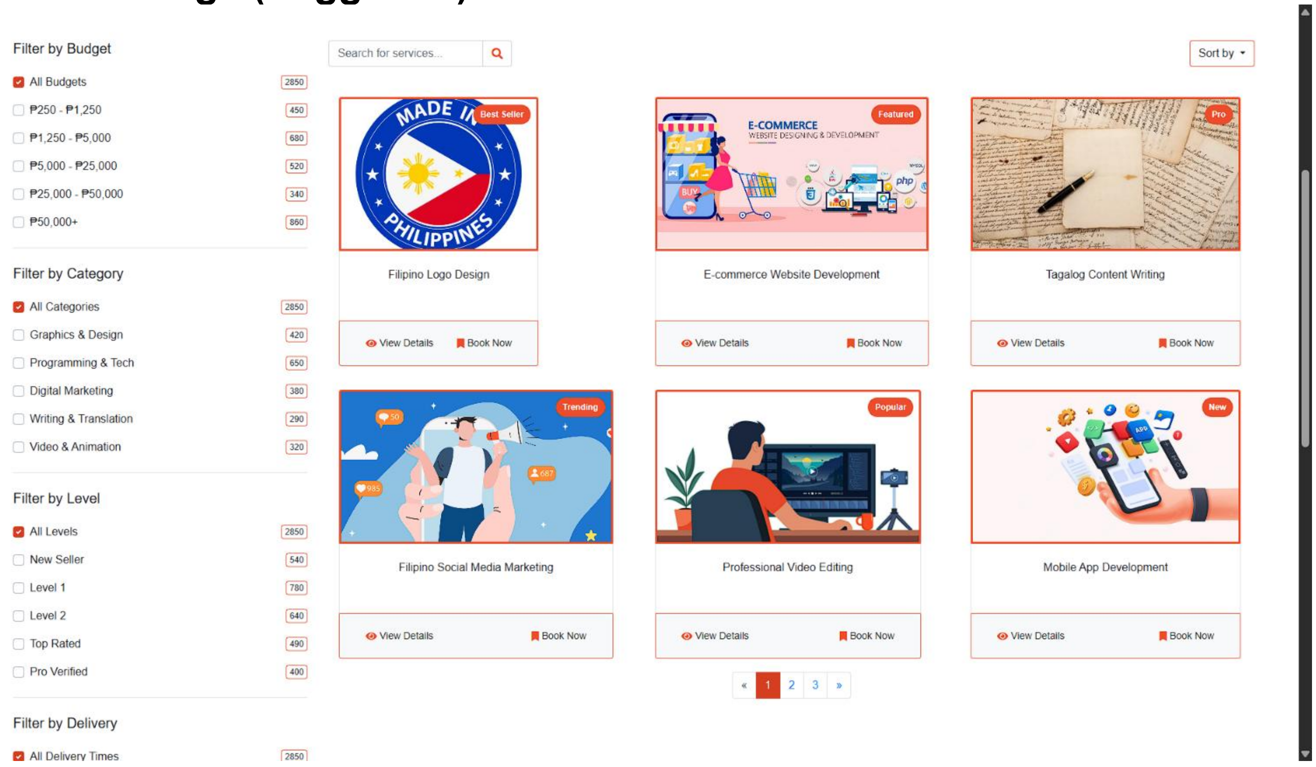
The loggedShop. This page is for SamutSari users which allows them to view shops and products and be able to directly add those products to their cart and make their purchase.

Marketplace (logged in)



The homepage for logged-in users allows SamutSari users to view, purchase, and book popular products and services.

Services Page (Logged in)



Filter by Budget

☑ All Budgets	2850
☐ ₱250 - ₱1,250	450
☐ ₱1,250 - ₱5,000	680
☐ ₱5,000 - ₱25,000	520
☐ ₱25,000 - ₱50,000	340
☐ ₱50,000+	860

Filter by Category

☑ All Categories	2850
☐ Graphics & Design	420
☐ Programming & Tech	650
☐ Digital Marketing	380
☐ Writing & Translation	290
☐ Video & Animation	320

Filter by Level

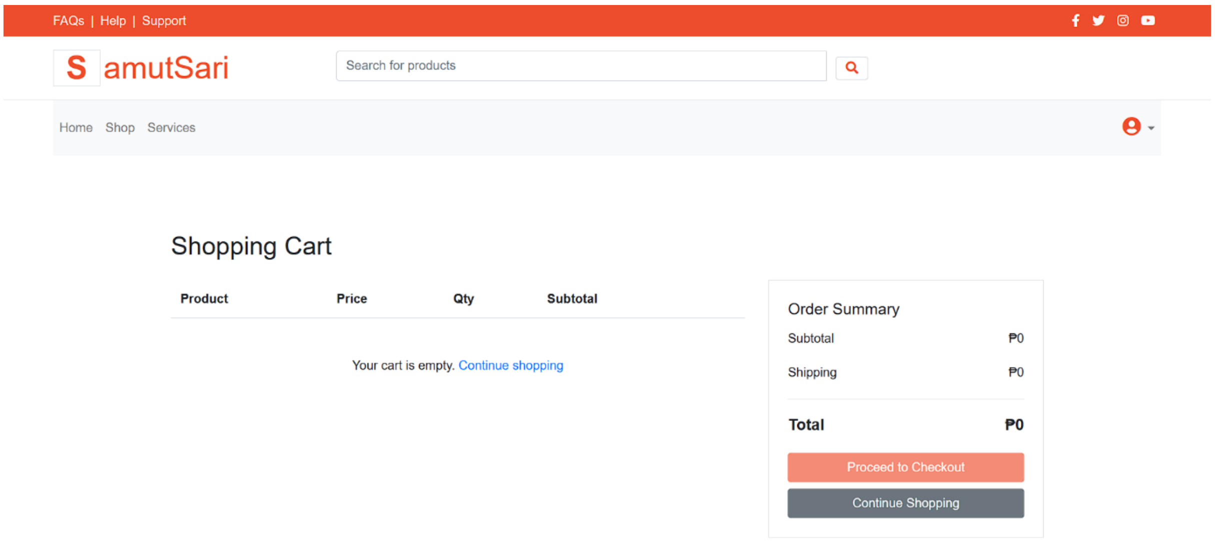
☑ All Levels	2850
☐ New Seller	540
☐ Level 1	780
☐ Level 2	640
☐ Top Rated	490
☐ Pro Verified	400

Filter by Delivery

☑ All Delivery Times	2850

This page allows SamutSari users to view the services offered in the app by service providers and freelancers.

Cart



Shopping Cart

Product	Price	Qty	Subtotal

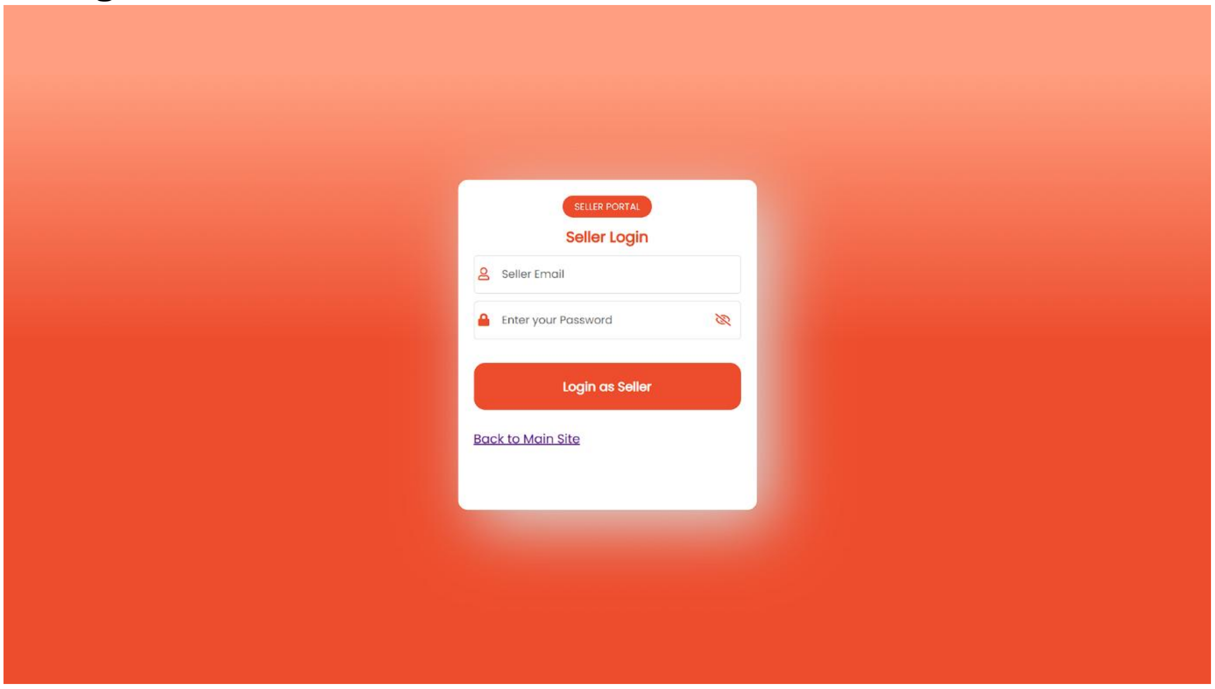
Your cart is empty. Continue shopping

Order Summary

Subtotal	₱0
Shipping	₱0
Total	₱0

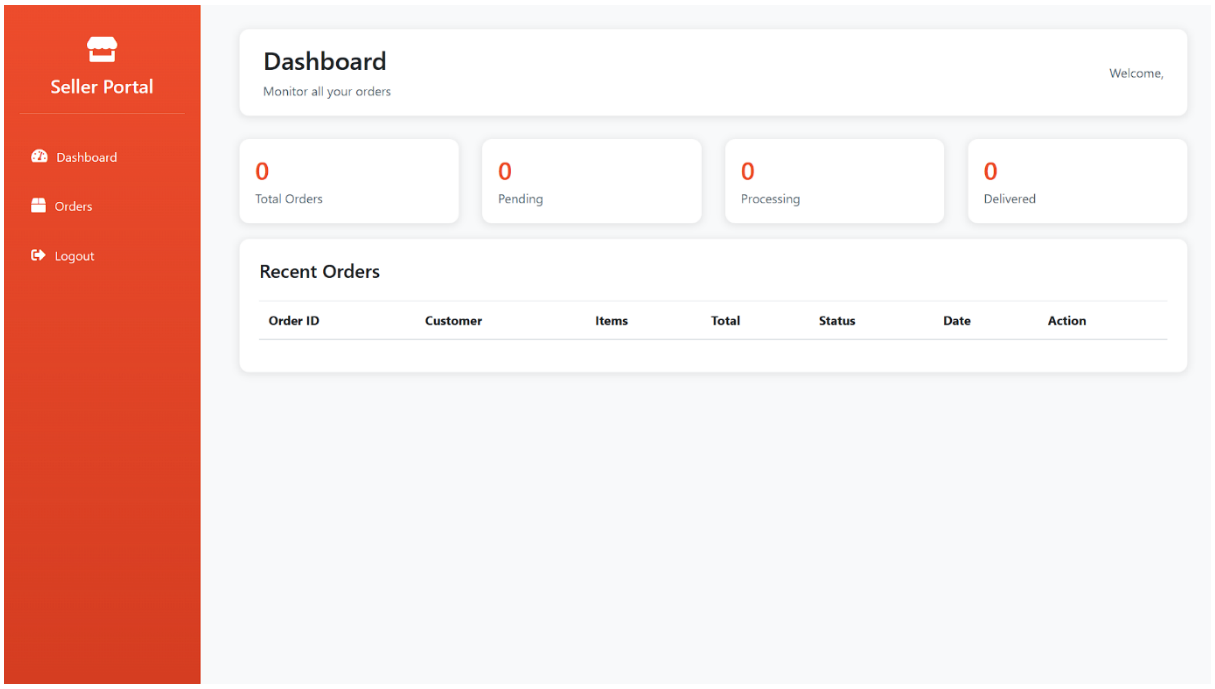
This page allows SamutSari users to view their shopping cart and check out their cart items.

Seller Login



This page allows the sellers to login into their account.

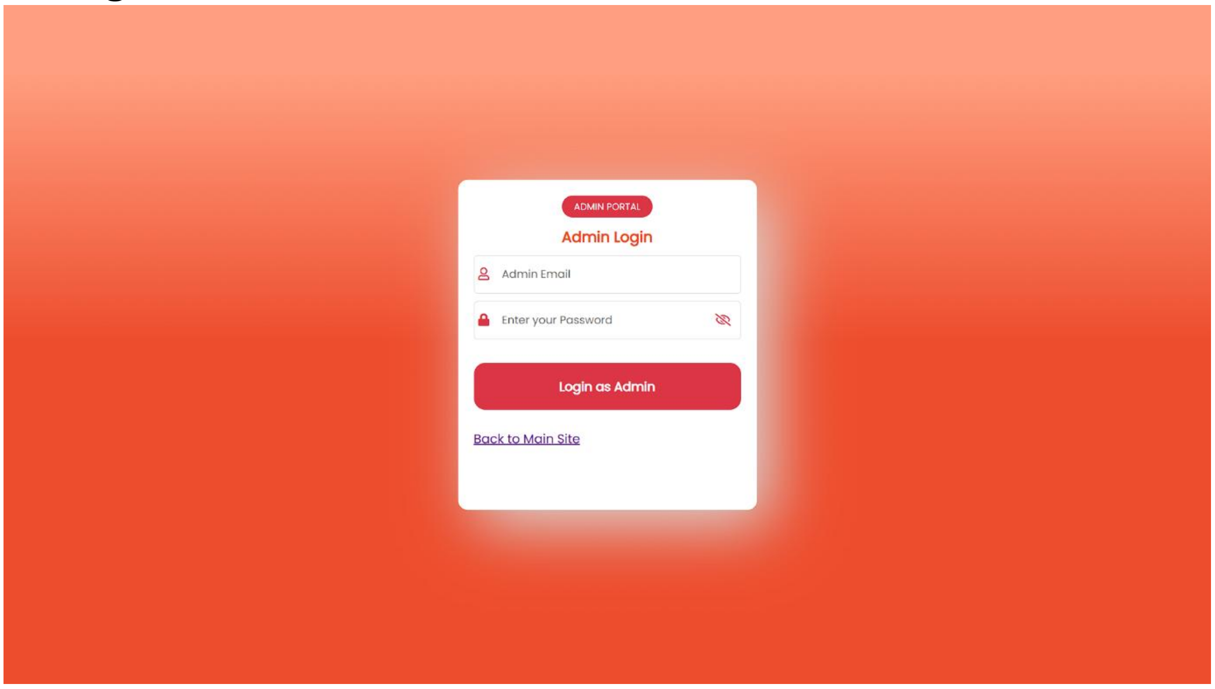
Seller Dashboard



Order ID	Customer	Items	Total	Status	Date	Action

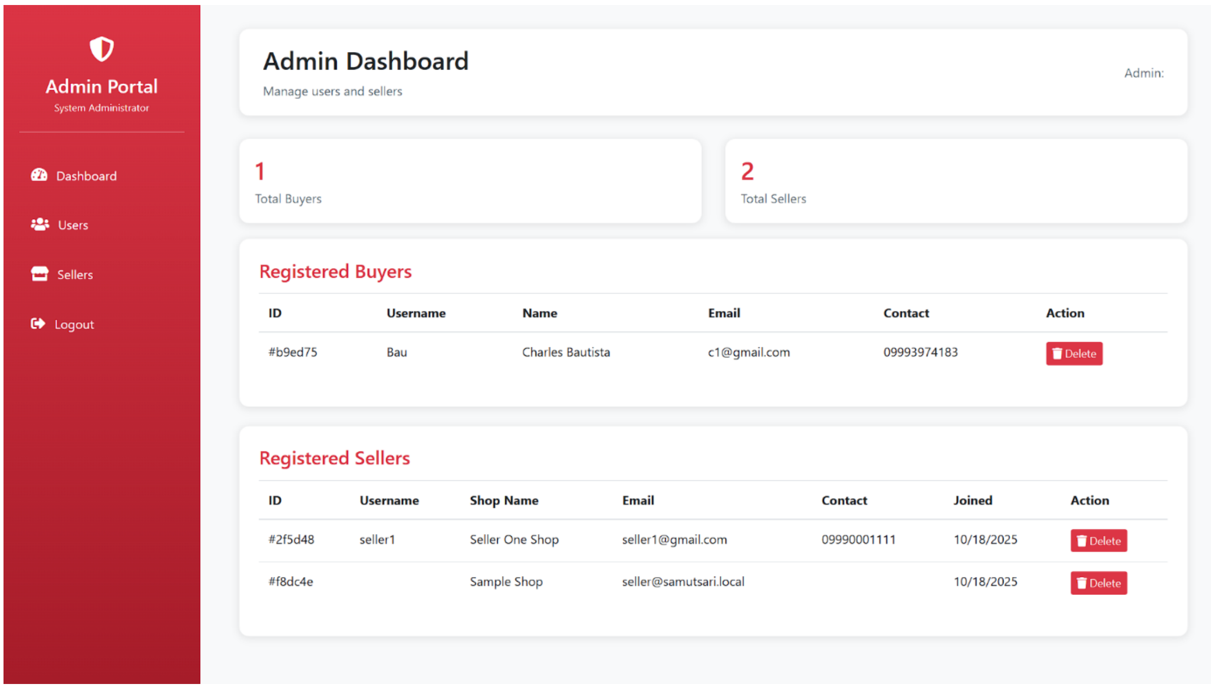
This page allows the sellers to view the dashboard information of their shop.

Admin Login



This page allows application admin to log and access their account.

Admin Dashboard



Admin Dashboard

Manage users and sellers

1 Total Buyers

2 Total Sellers

Registered Buyers

ID	Username	Name	Email	Contact	Action
#b9ed75	Bau	Charles Bautista	c1@gmail.com	09993974183	Delete

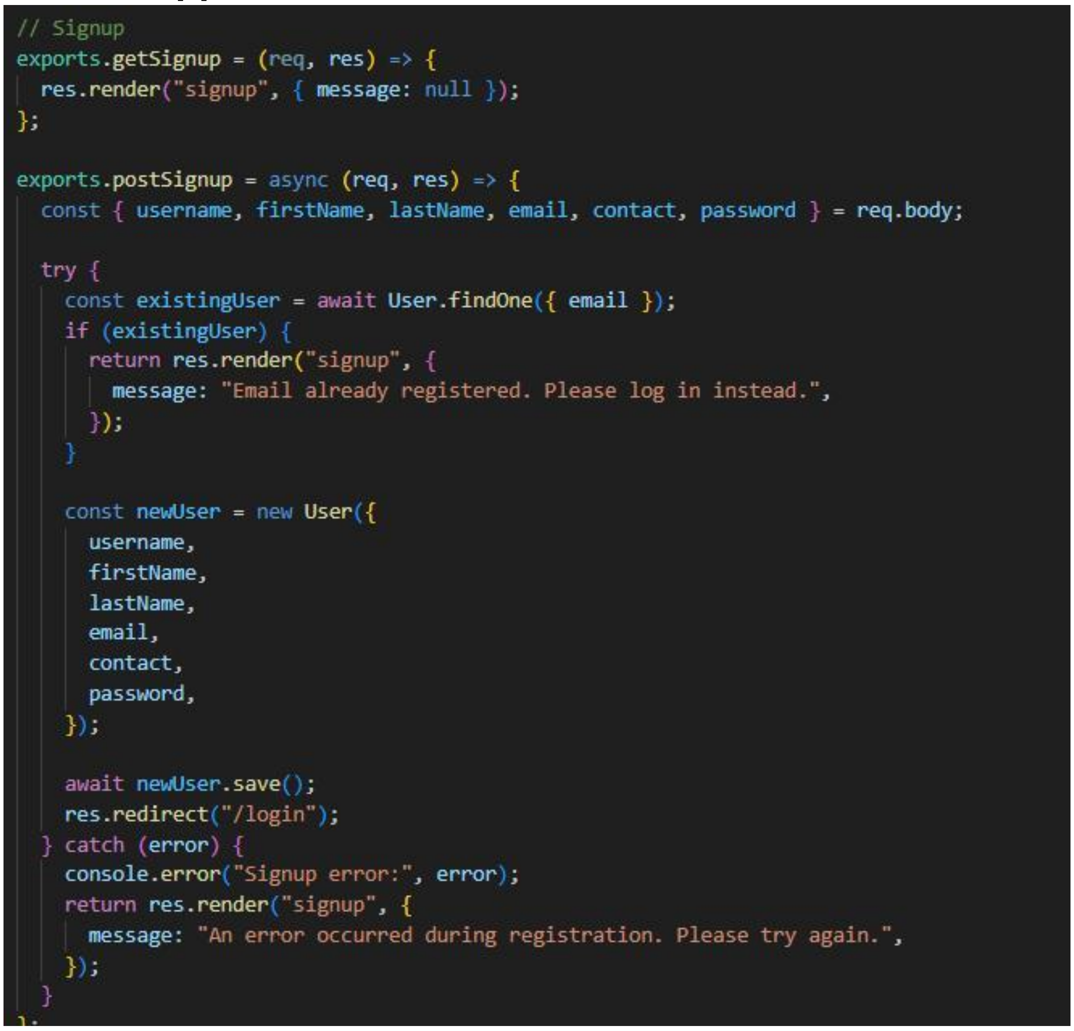
Registered Sellers

ID	Username	Shop Name	Email	Contact	Joined	Action
#2f5d48	seller1	Seller One Shop	seller1@gmail.com	09990001111	10/18/2025	Delete
#f8dc4e		Sample Shop	seller@samutsari.local		10/18/2025	Delete

The Admin Dashboard page allows the admin to view the registered accounts for both sellers and buyers.

Code Snippet

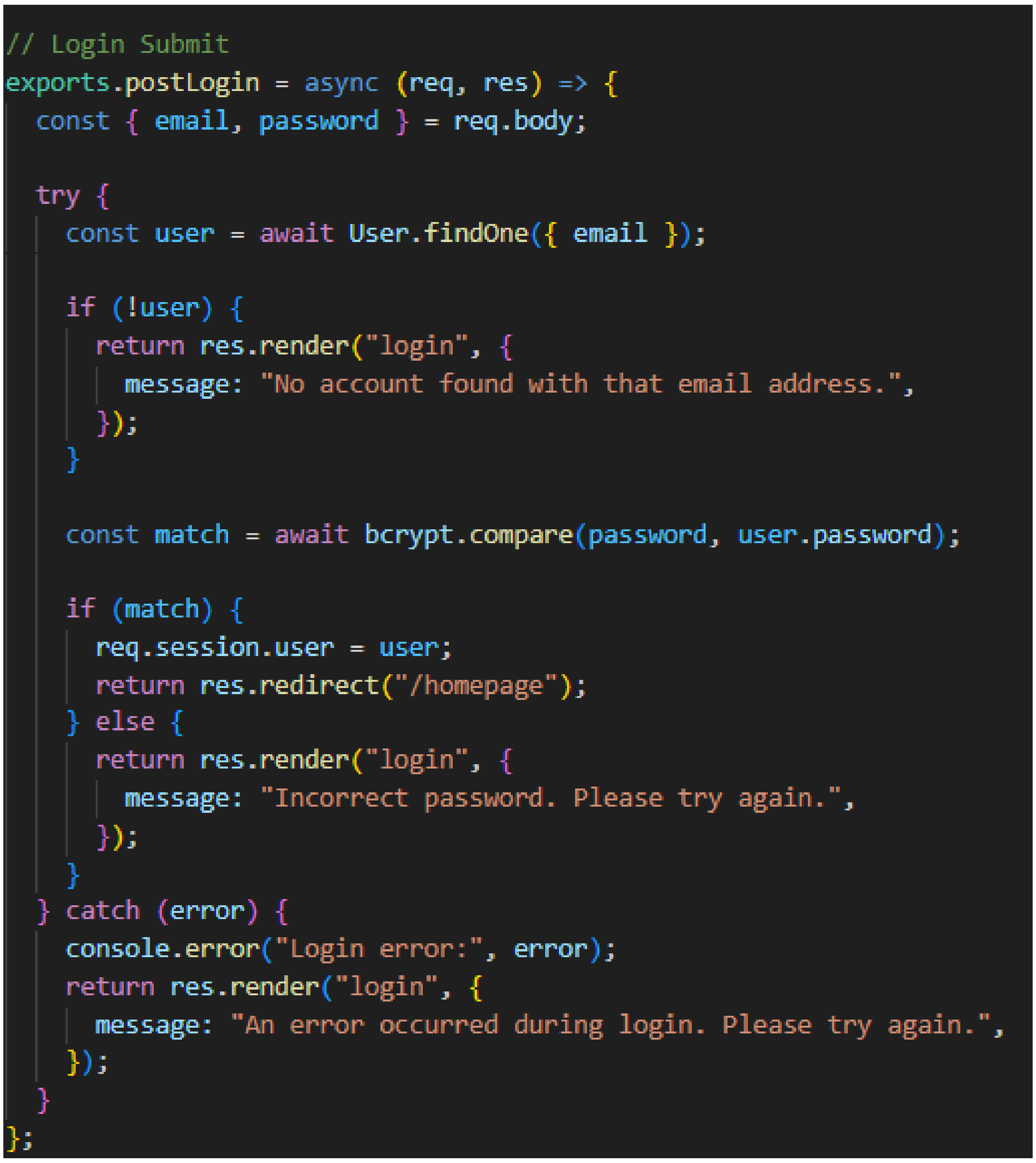
Sign-up Code Snippet



```
// Signup
exports.getSignup = (req, res) => {
  res.render("signup", { message: null });
};

exports.postSignup = async (req, res) => {
  const { username, firstName, lastName, email, contact, password } = req.body;

  try {
    const existingUser = await User.findOne({ email });
    if (existingUser) {
      return res.render("signup", {
        message: "Email already registered. Please log in instead.",
      });
    }

    const newUser = new User({
      username,
      firstName,
      lastName,
      email,
      contact,
      password,
    });

    await newUser.save();
    res.redirect("/login");
  } catch (error) {
    console.error("Signup error:", error);
    return res.render("signup", {
      message: "An error occurred during registration. Please try again.",
    });
  }
};
```

This sign-up code snippet retrieves the user’s information and saves that information to the database.

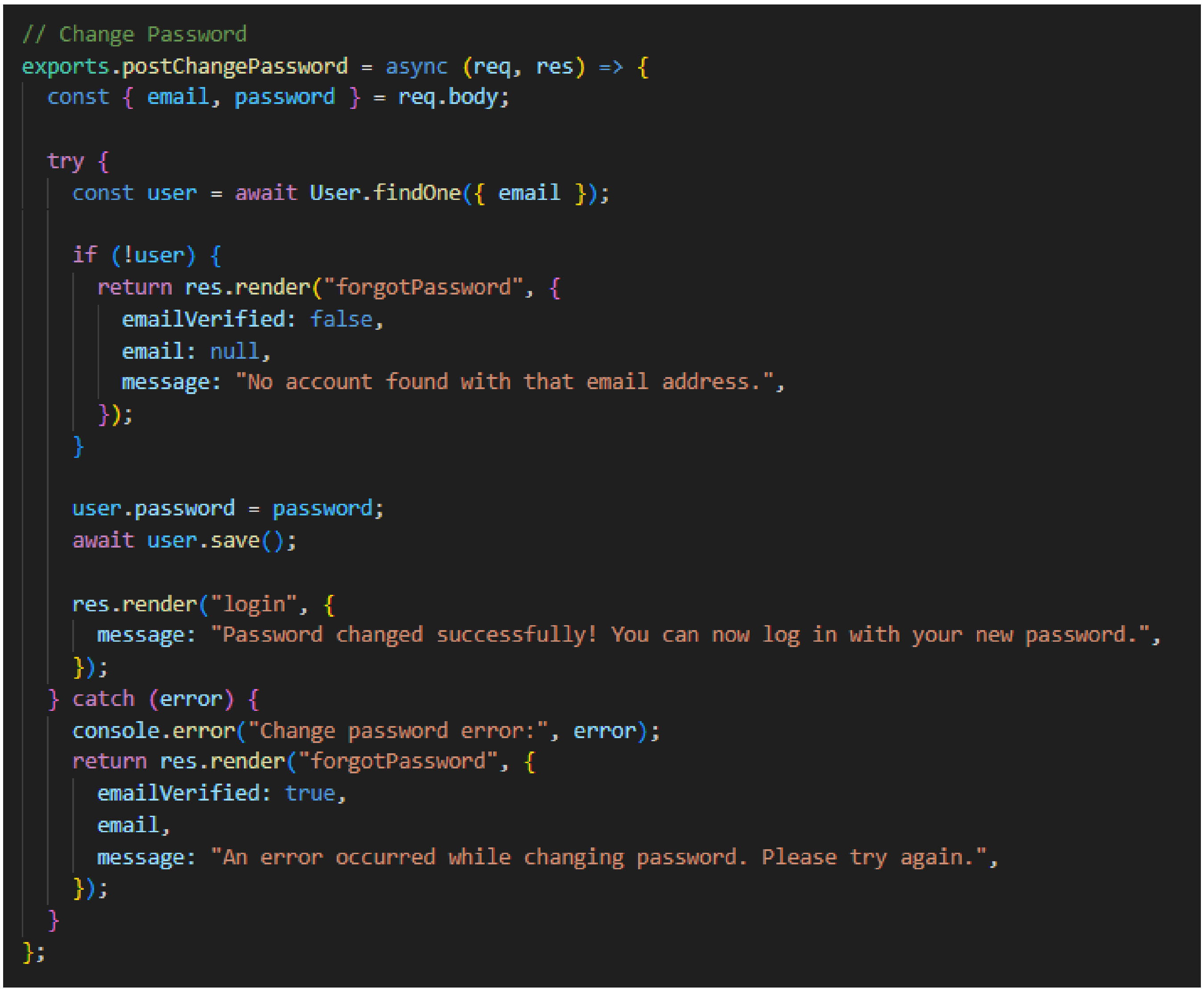
Login Code Snippet



```
// Login Submit
exports.postLogin = async (req, res) => {
  const { email, password } = req.body;

  try {
    const user = await User.findOne({ email });

    if (!user) {
      return res.render("login", {
        message: "No account found with that email address.",
      });
    }

    const match = await bcrypt.compare(password, user.password);

    if (match) {
      req.session.user = user;
      return res.redirect("/homepage");
    } else {
      return res.render("login", {
        message: "Incorrect password. Please try again.",
      });
    }
  } catch (error) {
    console.error("Login error:", error);
    return res.render("login", {
      message: "An error occurred during login. Please try again.",
    });
  }
};
```

The login code snippet retrieves the user's information from the database and validate the current user input to allow login.

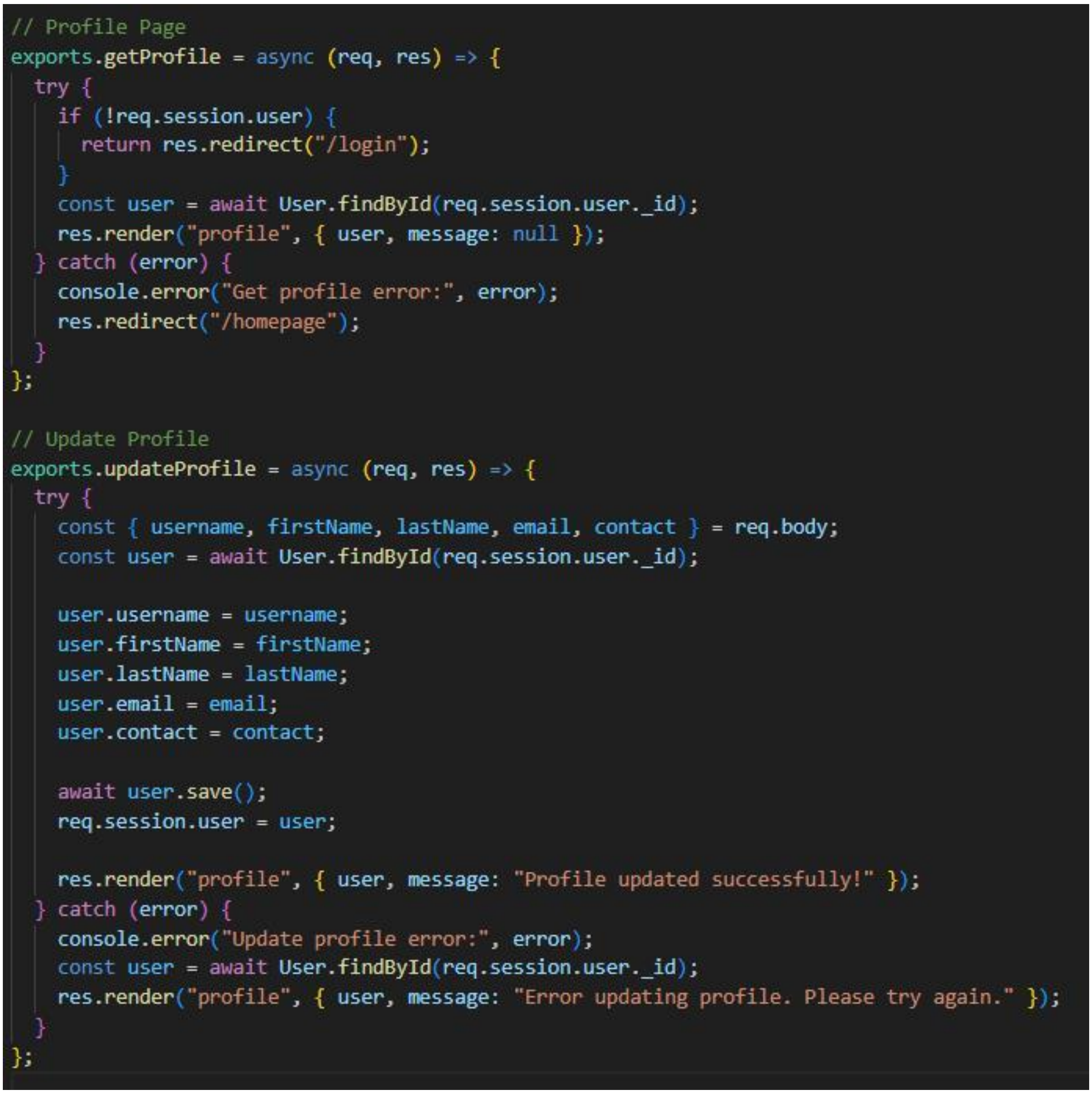
Change Password Code Snippet



```
// Change Password
exports.postChangePassword = async (req, res) => {
  const { email, password } = req.body;

  try {
    const user = await User.findOne({ email });

    if (!user) {
      return res.render("forgotPassword", {
        emailVerified: false,
        email: null,
        message: "No account found with that email address.",
      });
    }

    user.password = password;
    await user.save();

    res.render("login", {
      message: "Password changed successfully! You can now log in with your new password.",
    });
  } catch (error) {
    console.error("Change password error:", error);
    return res.render("forgotPassword", {
      emailVerified: true,
      email,
      message: "An error occurred while changing password. Please try again.",
    });
  }
};
```

The change password code snippet retrieves the email of the account and changes and updates the password in the database

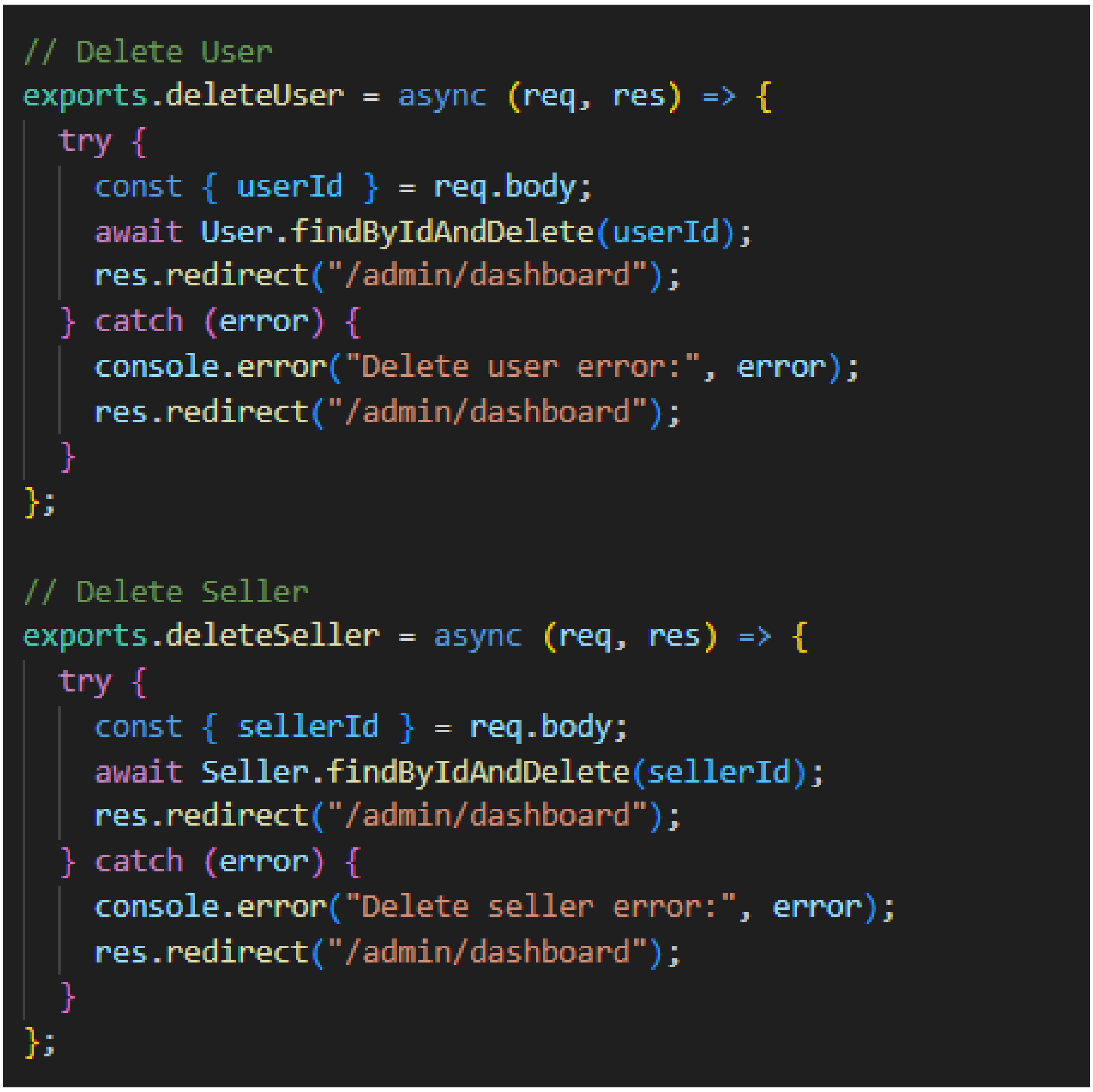
Update Profile Code Snippet



```
// Profile Page
exports.getProfile = async (req, res) => {
  try {
    if (!req.session.user) {
      return res.redirect("/login");
    }
    const user = await User.findById(req.session.user._id);
    res.render("profile", { user, message: null });
  } catch (error) {
    console.error("Get profile error:", error);
    res.redirect("/homepage");
  }
};

// Update Profile
exports.updateProfile = async (req, res) => {
  try {
    const { username, firstName, lastName, email, contact } = req.body;
    const user = await User.findById(req.session.user._id);

    user.username = username;
    user.firstName = firstName;
    user.lastName = lastName;
    user.email = email;
    user.contact = contact;

    await user.save();
    req.session.user = user;

    res.render("profile", { user, message: "Profile updated successfully!" });
  } catch (error) {
    console.error("Update profile error:", error);
    const user = await User.findById(req.session.user._id);
    res.render("profile", { user, message: "Error updating profile. Please try again." });
  }
};
```

The update profile code snippet retrieves all of the information that the user filled up when creating their account. It then allows the user to change information of their profile.

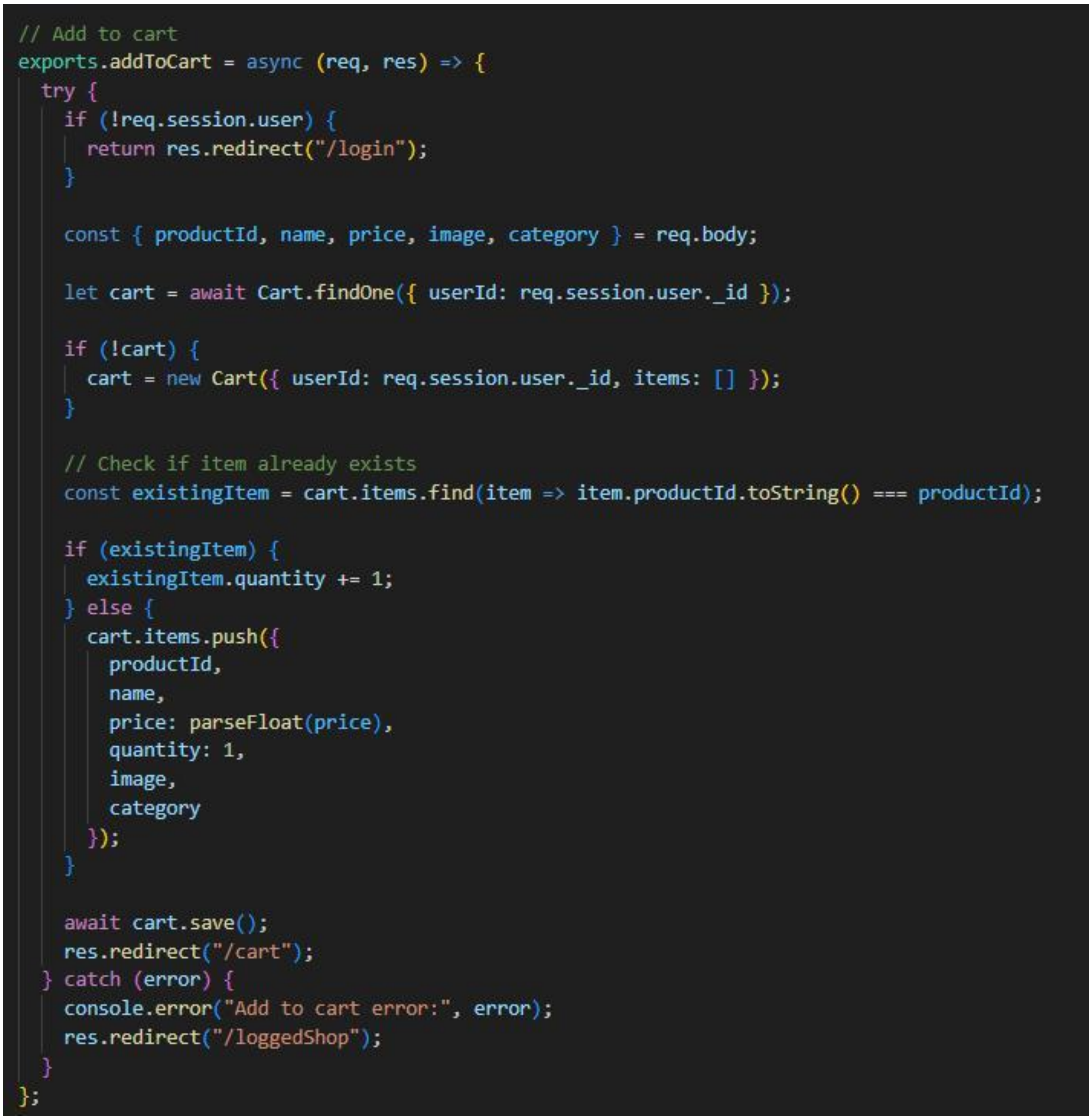
Delete User (Admin) Code Snippet



```
// Delete User
exports.deleteUser = async (req, res) => {
  try {
    const { userId } = req.body;
    await User.findByIdAndDelete(userId);
    res.redirect("/admin/dashboard");
  } catch (error) {
    console.error("Delete user error:", error);
    res.redirect("/admin/dashboard");
  }
};

// Delete Seller
exports.deleteSeller = async (req, res) => {
  try {
    const { sellerId } = req.body;
    await Seller.findByIdAndDelete(sellerId);
    res.redirect("/admin/dashboard");
  } catch (error) {
    console.error("Delete seller error:", error);
    res.redirect("/admin/dashboard");
  }
};
```

This delete user code snippet retrieves all of the accounts in the system first, then the admin will have the ability to delete both user-type and seller-type accounts.

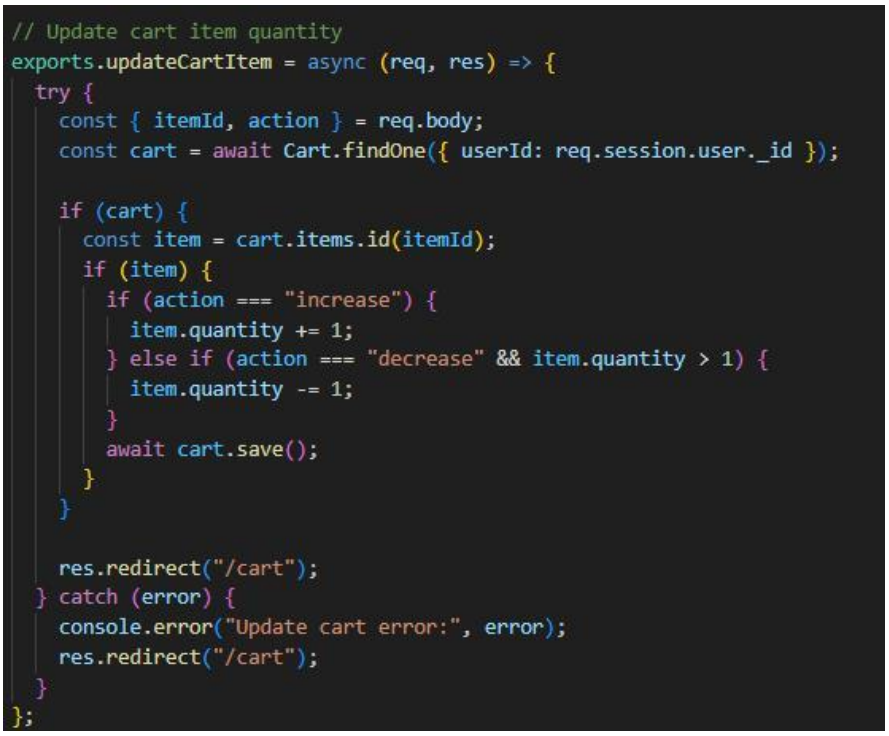
Add to Cart Code Snippet



```
// Add to cart
exports.addToCart = async (req, res) => {
  try {
    if (!req.session.user) {
      return res.redirect("/login");
    }

    const { productId, name, price, image, category } = req.body;

    let cart = await Cart.findOne({ userId: req.session.user._id });

    if (!cart) {
      cart = new Cart({ userId: req.session.user._id, items: [] });
    }

    // Check if item already exists
    const existingItem = cart.items.find(item => item.productId.toString() === productId);

    if (existingItem) {
      existingItem.quantity += 1;
    } else {
      cart.items.push({
        productId,
        name,
        price: parseFloat(price),
        quantity: 1,
        image,
        category
      });
    }

    await cart.save();
    res.redirect("/cart");
  } catch (error) {
    console.error("Add to cart error:", error);
    res.redirect("/loggedShop");
  }
};
```

This code snippet retrieves the data of the product and then displays that information in a designated cart page.

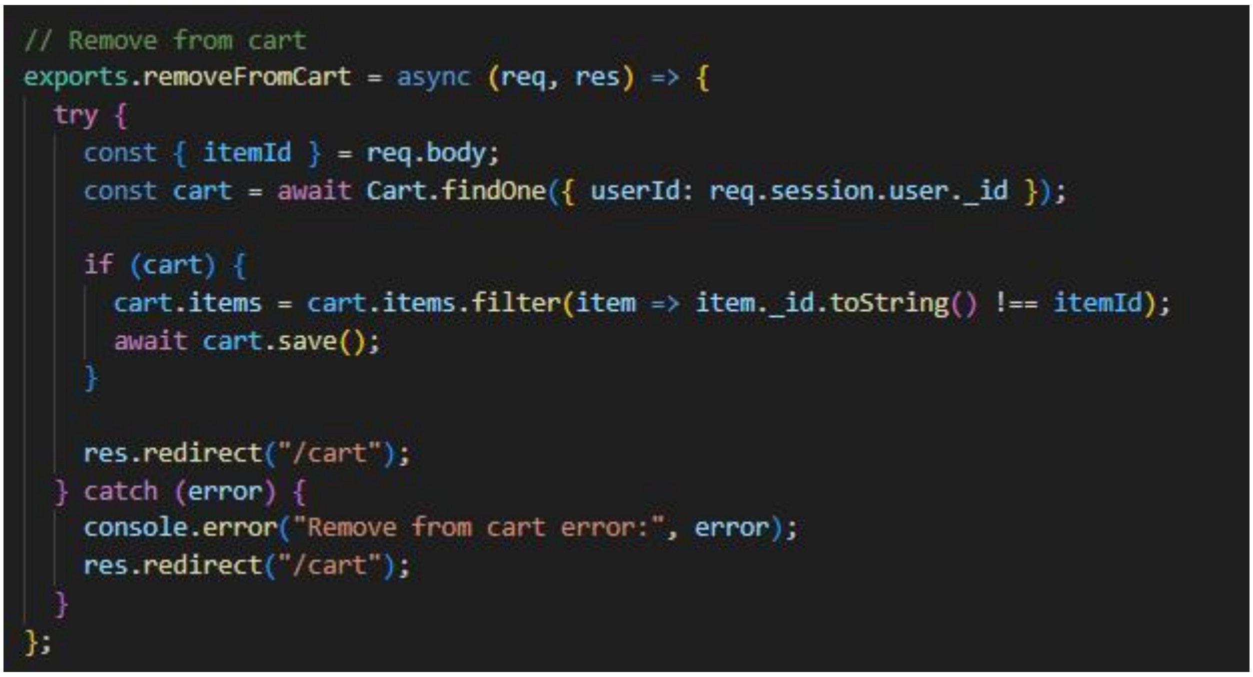
Update Cart Code Snippet



```
// Update cart item quantity
exports.updateCartItem = async (req, res) => {
  try {
    const { itemId, action } = req.body;
    const cart = await Cart.findOne({ userId: req.session.user._id });

    if (cart) {
      const item = cart.items.id(itemId);
      if (item) {
        if (action === "increase") {
          item.quantity += 1;
        } else if (action === "decrease" && item.quantity > 1) {
          item.quantity -= 1;
        }
        await cart.save();
      }
    }

    res.redirect("/cart");
  } catch (error) {
    console.error("Update cart error:", error);
    res.redirect("/cart");
  }
};
```

The update cart code snippet allows the user to add or deduct quantity from a specific item in their cart.

Remove Cart Code Snippet



```
// Remove from cart
exports.removeFromCart = async (req, res) => {
  try {
    const { itemId } = req.body;
    const cart = await Cart.findOne({ userId: req.session.user._id });

    if (cart) {
      cart.items = cart.items.filter(item => item._id.toString() !== itemId);
      await cart.save();
    }

    res.redirect("/cart");
  } catch (error) {
    console.error("Remove from cart error:", error);
    res.redirect("/cart");
  }
};
```

This code snippet allows the user to remove a specific item from their cart.

NATIONAL UNIVERSITY

NATIONAL UNIVERSITY COLLEGE OF COMPUTING AND INFORMATION TECHNOLOGIES

Project Rubric

Course Code	CTWBCOML	**Course/Subject**	Web Commercialization and E-Commerce
Term/Academic Year	Term 1 - AY '25-'26		
Group Name	Alrism		
Members	**Surname, First Name MI. (Alphabetical)**	**Signature**	
	Aceveda, Ben Christian G.	[illegible]	
	Bautista, Charles Denzel A.	[illegible]	
	Gabion, Christian Jeff L.	[illegible]	
	Morales, Aaron Gabriel D.	[illegible]	
	Valencia, Hans Aaron N.	[illegible]	

Project Component	SO	Unsatisfactory (0)	Needs Improvement (1)	Satisfactory (2)	Proficient (3)	Exceptional (4)	PTS
User Interface, Design & Navigation	SO6	Unattractive, outdated design; inconsistent layout; non-responsive; confusing navigation.	Basic design with minor inconsistencies; limited responsiveness; partially intuitive navigation.	Visually acceptable, consistent layout; responsive on most devices; easy navigation.	Attractive, modern design; fully consistent across all pages; highly responsive; intuitive navigation.	Professional, modern UI with excellent visual appeal, seamless responsiveness, and flawless navigation experience.	
Functionality & Core Features		Major features missing or non-functional (e.g., browsing, search, cart, checkout).	Some core features functional but with major bugs or incomplete implementation.	All core features functional (browsing, search, cart, checkout) with minor issues.	Fully functional features with smooth interactions and proper error handling.	Exceptional functionality with additional enhancements beyond requirements; flawless performance.	
Security & Data Management		No HTTPS; weak or no authentication; poor data protection; vulnerable to attacks.	Basic security measures implemented but with noticeable vulnerabilities.	Secure authentication, basic encryption, and partial protection against attacks.	Strong security with HTTPS, encryption, and defenses against common attacks; efficient database handling.	Advanced security, full compliance with best practices, optimized database queries, high data integrity, and large dataset handling.	
Product & Order Management		Admin cannot manage products/orders; product details incomplete; poor categorization.	Basic product/order management with limited functionality.	Functional product and order management with correct details and category organization.	Efficient product and order management with smooth updates and well-organized categories.	Advanced product and order management with bulk operations, automation, and optimal category/tagging.	
Documentation, Code Quality & Presentation	SO3	No or poor documentation; messy, unreadable code; unclear presentation.	Basic documentation; code readable but lacks structure; weak presentation skills.	Adequate documentation; code mostly organized; clear presentation with minor gaps.	Good documentation; clean, well-structured code; confident and clear presentation.	Comprehensive technical/user documentation; excellent code quality; engaging and professional presentation with confident Q&A.	

Total Score and Feedback				
☐ Exceptional	20	Outstanding performance in all project components with minimal to no issues	**Total Points Earned**	
☐ Proficient	16-19	Good performance in most project components with minor issues.		
☐ Satisfactory	12-15	Acceptable performance in basic aspects with several issues.		
☐ Needs Improvement	8-11	Below-average performance with many issues.		
☐ Unsatisfactory	0-7	Poor performance with critical issues in most project components.		

Evaluated by: ____________________ Name of Course Instructor/Date	**Remarks/Comments**

NATIONAL UNIVERSITY

NATIONAL UNIVERSITY COLLEGE OF COMPUTING AND INFORMATION TECHNOLOGIES

Individual and Self Evaluation Rubric

Course Code	CTWBCOML	**Course/Subject**	Web Commercialization and E-Commerce
Term/Academic Year	Term 1 – AY '25-'26		
Group Name	Alrism		

Student Name(s) ***All members including the evaluator.***	**Contribution to Team Effort**	**Communication Skills**	**Meeting Deadlines and Reliability**	**Quality of Work**	**Collaboration and Teamwork**	**PTS**
	Contributes meaningfully to group discussions.	Demonstrate excellence in written and verbal communication skills	Completes group assigned task(s) on time.	Prepares work in a quality manner	Demonstrate a cooperative and supportive attitude.	
	Using scale 0-4 (0=Unsatisfactory; 1=Needs Improvement; 2=Satisfactory; 3=Proficient; 4=Exceptional)					
	SO5					
Aceveda, Ben Christian G.	4	4	4	4	4	20
Bautista, Charles Denzel A.	4	4	4	4	4	20
Gabion, Christian Jeff L.	4	4	4	4	4	20
Morales, Aaron Gabriel D.	4	4	4	4	4	20
Valencia, Hans Aaron N.	4	4	4	4	4	20

Peer Evaluation Interpretation

☒ Exceptional	20	Outstanding performance across all indicators.
☐ Proficient	16-19	Strong performance with minor areas for improvement.
☐ Satisfactory	12-15	Adequate performance with several areas for improvement.
☐ Needs Improvement	8-11	Below-average performance with significant areas for improvement.
☐ Unsatisfactory	0-7	Poor performance across most or all indicators.

Remarks/ Comments
All of my team members actively collaborated and contributed throughout the entire process, from ideation to development, documentation, and presentation. It was a smooth and productive journey working with them, and I look forward to the opportunity to work together again in the future.

Evaluated By:

[illegible]

Aaron Gabriel D. Morales
Signature over Student Name