Abstract

This project presents the design and implementation of **ShopNest**, a responsive e-commerce web application that enables end-to-end online retail operations. The system provides core user features—including product browsing and search, category filtering, detailed product pages, a persistent shopping cart, and a secure checkout flow—alongside an administrative dashboard for product, inventory, and order management. The solution adopts a modular three-tier architecture: a **Bootstrap 4** front end for accessible, mobile-first UI; a **PHP** application layer handling authentication, session management, form validation, and server-side business logic; and a **MySQL** database responsible for durable storage of users, roles, products, categories, orders, and payments. Security measures include password hashing, prepared statements to mitigate SQL injection, and role-based access control. Performance and UX are improved through client-side validation, caching of static assets, and semantic markup for SEO. Functional testing validates user registration/login, cart and checkout workflows, and admin operations; usability checks confirm consistency, clarity, and responsiveness across devices. The prototype demonstrates a scalable foundation suitable for small to medium retailers, with proposed future enhancements such as payment gateway integration, order tracking, analytics dashboards, and microservice-ready APIs to support mobile apps and third-party integrations.

Table of the content

Chapter 1 – Introduction

1. Introduction

The rapid growth of online shopping has transformed how businesses sell products and how customers interact with brands. An e-commerce website serves as a digital storefront, allowing users to browse products, compare options, and make purchases from anywhere at any time. This project focuses on developing **ShopNest**, a modern e-commerce web application designed to provide a seamless and secure online shopping experience for both customers and administrators.

ShopNest integrates essential features for an online retail platform, including product listing and categorization, search and filter functionalities, user authentication, shopping cart management, and order processing. It also provides an administrative interface for managing products, categories, inventory, and customer orders. The application is designed with a **three-tier architecture**: a **front end** built with Bootstrap 4 for responsive and mobile-friendly UI, a **back end** developed in PHP for server-side logic and data processing, and a **MySQL database** for persistent storage of user, product, and order information.

The goal of this project is to develop a fully functional e-commerce system that emphasizes usability, security, and scalability. By implementing features such as password hashing, role-based access control, and prepared statements for database queries, the system ensures data security while providing a smooth and intuitive user experience. ShopNest demonstrates a practical approach to creating web-based retail solutions suitable for small to medium-sized businesses.

* 1. Problem Statement

E-commerce platforms are increasingly essential for retail businesses, but building an efficient, secure, and user-friendly system remains challenging. The goal of this project is to identify gaps in existing online shopping solutions and develop **ShopNest**, a web-based e-commerce application that addresses these issues.

* + 1. Existing Challenges in ShopNest

Before the development of this project, online shopping systems—including preliminary versions of ShopNest—faced several challenges:

1. **Lack of User-Friendly Navigation**
   * Users often struggle to browse products efficiently due to poorly organized categories and missing search/filter options.
2. **Security Vulnerabilities**
   * Existing systems may not properly secure user data, with weak password handling, absence of role-based access control, and risks of SQL injection attacks.
3. **Inefficient Cart and Order Management**
   * Shopping carts may not persist across sessions, and order processing lacks administrative oversight, causing delays or errors.
4. **Limited Administrative Control**
   * Administrators often have minimal tools to manage products, inventory, categories, and orders, making it difficult to scale operations.
5. **Lack of Responsive Design**
   * Many platforms are not optimized for mobile devices, resulting in poor user experience for smartphone or tablet users.

By addressing these challenges, the project aims to provide a secure, responsive, and fully functional e-commerce system that improves usability for customers while enhancing management capabilities for administrators.

* + 1. Need for Software-Based Solution

With the rapid growth of online shopping and digital retail, traditional manual or semi-digital methods for managing sales and customer interactions are no longer sufficient. A software-based solution for ShopNest is essential to address the limitations of existing systems and provide a seamless online shopping experience.

Key reasons for implementing a software-based solution include:

1. **Automation of Operations**
   * Automates product management, inventory updates, and order processing, reducing human error and saving time.
2. **Enhanced User Experience**
   * Provides a responsive and intuitive interface for browsing, searching, and purchasing products, improving customer satisfaction.
3. **Data Security and Privacy**
   * Ensures sensitive user data, such as login credentials and payment information, is securely stored and processed using encryption and role-based access control.
4. **Scalability**
   * A software solution can easily accommodate growing numbers of users, products, and transactions without compromising performance.
5. **Real-Time Management and Reporting**
   * Allows administrators to monitor inventory, orders, and sales analytics in real time, enabling informed decision-making.
6. **Integration Capabilities**
   * Facilitates integration with third-party services such as payment gateways, delivery partners, and marketing tools to enhance the overall system functionality.
     1. Summary of the Problem

The existing online shopping systems, including earlier versions of ShopNest, face multiple challenges that hinder efficiency, usability, and security. Customers struggle with poor navigation, limited search and filtering options, and non-responsive designs, which reduce overall satisfaction. Administrators face difficulties in managing products, inventory, and orders effectively, while security vulnerabilities put sensitive user data at risk. Manual or semi-automated processes lead to errors, delays, and limited scalability.

A comprehensive software-based solution is required to overcome these issues by providing a **secure, user-friendly, and scalable e-commerce platform**. This system must automate core operations, enhance the shopping experience, ensure data protection, and provide real-time management tools for administrators. Addressing these problems will result in a reliable online retail system that meets the needs of both customers and business operators.

1. Project Objectives

Admin

The administrator of ShopNest is responsible for managing products, categories, inventory, and orders. The objectives for the admin module include:

1. **Product Management**
   * Add, update, and delete products with detailed information such as name, description, price, category, and images.
2. **Category Management**
   * Create, modify, and remove product categories to ensure proper organization and navigation.
3. **Inventory Management**
   * Monitor stock levels, update quantities, and receive alerts for low-stock items to prevent overselling.
4. **Order Management**
   * View and process customer orders, update order status (e.g., pending, shipped, delivered), and handle cancellations or returns.
5. **User Management**
   * Manage registered users, assign roles, and control access permissions to maintain security and order in the system.
6. **Reporting and Analytics**
   * Generate sales reports, track product performance, and monitor overall business metrics for informed decision-making.
7. **Security Management**
   * Ensure data security through password hashing, role-based access control, and protection against SQL injection or unauthorized access.

**Methnata ss dnn one ekin ekt**

1. User Roles and Functionalities

ShopNest supports multiple user roles, each with distinct functionalities to ensure security, usability, and efficient management of the e-commerce platform. The main roles include **Admin** and **Customer/User**.

**User Management**

User management is a critical aspect of ShopNest that ensures proper authentication, authorization, and engagement for all users. Key functionalities include:

1. User Registration
   * New users can create an account by providing personal information such as name, email, and password.
   * Validation checks ensure the email is unique and the password meets security standards.
2. User Login and Authentication
   * Registered users can securely log in using their credentials.
   * Passwords are stored securely using hashing mechanisms.
   * Role-based access ensures users only access functionalities intended for their role.
3. Profile Management
   * Users can view and update their personal information, including name, email, and contact details.
4. Password Management
   * Users can reset forgotten passwords via secure email verification.
   * Option to change passwords for enhanced security.
5. Session Management
   * Persistent login sessions ensure a seamless experience while preventing unauthorized access.
6. User Activity Tracking
   * Records user interactions such as login history, cart updates, and order history for better service and analytics.

By implementing robust user management, ShopNest ensures security, personalized experience, and smooth interaction between users and the platform, forming the foundation for all other user functionalities like browsing products, placing orders, and managing carts.

**Methentatth ekin ekt SS danna one**

1. Scope the Project

The **ShopNest e-commerce website** allows users to shop online and administrators to manage the store efficiently.

**User Features**

* Register and log in to the website.
* Browse and search products by categories.
* Add items to the cart and place orders.
* View order history.

**Admin Features**

* Add, update, or delete products and categories.
* Manage inventory and stock levels.
* Process and update customer orders.
* Manage user accounts.

**System Features**

* Works on desktop and mobile devices (responsive design).
* Secure login and data storage using a database.
* Handles errors and validates user input.

**Out of Scope**

* Payment gateway integration.
* Mobile app version.
* Advanced AI recommendations.

Chapter 2 – System Analysis

1.**User Requirements**

* Users need an easy way to browse products, search by categories, and place orders securely.
* Users should be able to register, log in, manage their profile, and view order history.

2. **Admin Requirements**

* Administrators need tools to manage products, categories, inventory, and orders.
* Admins should monitor user activity and generate reports to improve store performance.

3. **System Requirements**

* A secure login system with password encryption.
* A responsive web interface for desktop and mobile devices.
* A database to store users, products, categories, and orders.
* Proper validation and error handling to ensure smooth operation.

**Usecase diagram for ShopNest**

**Methanat ss ekak usecase digram eke**

Chapter 3 – Requirenment Specification

1. Functional Requirements

User Authentication & Roles

* + Secure sign-up/login with email/phone; password reset.
  + Roles: Admin, Seller/Vendor (optional marketplace), Customer, Delivery Partner (future).
  + Role-based dashboards and permissions.

Catalog & Inventory

* + Admin/Sellers: create, edit, categorize products; upload images; set price, stock, variants (size/color); bulk import/export (CSV).
  + Real-time stock deduction on order; low-stock alerts.

Search & Discovery

* + Keyword search, category filters, sort by price/rating/newest.
  + Featured products, recommendations, and trending lists.

Cart & Checkout

* + Add/remove/update items; promo codes/coupons; shipping options.
  + Guest cart that persists after login.
  + Taxes & shipping fee calculation.

Payments

* + Integration with payment gateways (e.g., Stripe/PayPal/Local gateway); Cash-on-Delivery (optional).
  + Order confirmation after successful payment; refunds/voids.

Orders & Fulfillment

* + Order placement, status tracking (Placed → Packed → Shipped → Delivered → Returned/Refunded).
  + Admin/Sellers: print invoices & shipping labels; manage returns (RMA).

Customer Account

* + Profile & addresses; order history; re-order; saved cards (via gateway tokenization only).
  + Wishlist and Recently Viewed products.

Ratings & Reviews

* + Verified-buyer reviews with star ratings, photos (moderation tools).

Promotions

* + Coupons (percentage/fixed/BOGO), scheduled sales, free shipping rules.

Notifications

* + Email/SMS/in-app push for order & delivery updates, low stock, password reset.

Content & CMS

* + Manage homepage banners, static pages (About, FAQ, Return Policy), blog/news.

Reports & Analytics

* + Sales, revenue, conversion, top products, abandoned carts, vendor performance.

Delivery Partner Module (Future)

* + Delivery app/view: see assigned orders, update live status, proof-of-delivery (photo/OTP).

Internationalization (Future)

* + Multi-currency, multi-language, regional tax rules.

Chapter 4 – System Architecture

Class Diagram of the ShopNest Website

**Class diagram eke ss ekak**

Architecture

1.Key Classes and Details

**Admin**

**Purpose:** Handles all administrative operations in the system.

**Key Attributes:**

* admin\_id – Unique identifier for the admin.
* name – Admin name.
* email – Admin email.
* password – Encrypted password for login.

**Key Methods/Functions:**

1. login() – Allows the admin to securely log in.
2. addProduct() – Add new products to the store.
3. updateProduct() – Modify product details.
4. deleteProduct() – Remove a product from the store.
5. manageCategory() – Add, update, or delete product categories.
6. manageInventory() – Update stock levels and monitor inventory.
7. processOrder() – View and update customer orders.
8. generateReport() – Create sales and performance reports.

**User**

**Purpose:** Handles all customer-related operations in ShopNest.

**Key Attributes:**

* user\_id – Unique identifier for the user.
* name – Customer name.
* email – User email.
* password – Encrypted password for login.
* cart – Stores selected products.
* orderHistory – List of past orders.

**Key Methods/Functions:**

1. register() – Allows a new user to create an account.
2. login() – Enables user login securely.
3. browseProducts() – View all available products.
4. searchProducts() – Search products by name or category.
5. addToCart() – Add selected products to the shopping cart.
6. removeFromCart() – Remove products from the cart.
7. checkout() – Place an order and complete the purchase.
8. viewOrderHistory() – Check past orders and their status.

**Cart**

**Purpose:**

* To store products that a user wants to buy temporarily.
* To calculate the total cost of selected items before checkout.

**Key Attributes:**

* cart\_id – Unique identifier for the cart.
* user\_id – Links the cart to a specific user.
* products – List of products added to the cart.
* Quantity – Number of units for each product.
* total\_price – Total cost of all products in the cart.

**Key Methods/Functions:**

1. addToCart(product\_id, quantity)

Add a product to the cart with the desired quantity.

1. removeFromCart(product\_id)

Remove a product from the cart.

1. updateQuantity(product\_id, quantity)

Change the quantity of a product in the cart.

1. viewCart()

Display all products and their quantities in the cart.

1. checkout()

Proceed to place an order for all items in the cart.

**Items (Products)**

**Purpose:**

* To provide detailed information about products so users can browse, compare, and make purchases.

**Key Attributes:**

* product\_id – Unique identifier for each product.
* Name – Name of the product.
* Description – Short details about the product.
* Price – Cost of the product.
* category\_id – Links the product to a specific category.
* stock\_quantity – Number of units available in inventory.
* Image – Product image for display on the website.

**Key Methods/Functions:**

1. addProduct() – Admin adds a new product to the store.
2. updateProduct() – Modify existing product details.
3. deleteProduct() – Remove a product from the store.
4. viewProduct() – Display product information to users.
5. searchProduct(keyword) – Search products by name or description.

**Orders**

**Purpose:**

* To record and manage user purchases from cart to delivery.
* To allow both users and admins to track order status.

**Key Attributes:**

* order\_id – Unique identifier for each order.
* user\_id – Links the order to the user who placed it.
* order\_date – Date when the order was placed.
* total\_amount – Total price of all items in the order.
* Status – Current status of the order (e.g., Pending, Shipped, Delivered, Cancelled).

**Key Methods/Functions:**

1. placeOrder(cart\_id)

Converts a user’s cart into an order.

1. updateStatus(order\_id, status)

Admin updates the order status.

1. viewOrder(user\_id)

Users can see their order history and details.

1. cancelOrder(order\_id)

Users or admins can cancel orders under certain conditions.

2.Relationships in ShopNest

The ShopNest system has several key relationships between its entities.

**Admin – Product / Category / Order**

**Type: One-to-Many**

**Explanation:** One admin can manage multiple products, categories, and customer orders.

**User – Cart**

**Type: One-to-One**

**Explanation:** Each user has one cart to store products before checkout.

**User – Order**

**Type: One-to-Many**

**Explanation:** A user can place multiple orders over time.

**Product – Category**

**Type: Many-to-One**

**Explanation:** Multiple products belong to a single category (e.g., “Watches”).

**Cart – Product**

**Type: Many-to-Many**

**Explanation:** A cart can contain multiple products, and a product can appear in multiple carts.

**Order – Product**

**Type: Many-to-Many**

**Explanation:** Each order can include multiple products, and each product can appear in multiple orders.

**Implementation:** Typically handled via an Order\_Details table.

**Admin – User**

**Type: One-to-Many**

**Explanation:** Admins can manage multiple users, including viewing, editing, or blocking accounts.

Entity Relationship Diagram of the ShopNset

Er ekk adala ss ekk

3.Entitiees and their Attributes

ER EKE VIDIYATA ME TIKA DANNA ONE

4.Technologies Stack

The **ShopNest** e-commerce website uses modern technologies to ensure a secure, responsive, and efficient platform for both users and administrators.

**1. Frontend (Client-Side)**

* **HTML5**

Structure of web pages.

* **CSS3 / Bootstrap 4**

Styling and responsive design for desktop and mobile devices.

* **JavaScript / jQuery**

Interactive features, form validation, and dynamic content.

**2. Backend (Server-Side)**

* **PHP**

Server-side scripting and handling user requests.

* **MySQL**

Database management for storing users,products, orders, and categories.

**3. Tools and Libraries**

* **Bootstrap Icons**

For icons in buttons, menus, and navigation.

* **Popper.js**

Supports tooltips and dropdowns in Bootstrap.

* **VS Code / PHPStorm**

Code editor and development environment.

**4. Web Server**

* **Apache / PHP Built-in Server**

To run the application locally or on a hosting server.

**5. Security & Utilities**

* **Password Hashing (PHP password\_hash)**

Secure storage of user passwords.

* **Session Management**

For login and user session control.

* **Form Validation**

Prevents invalid data and SQL injection.

**Summary:** This technology stack allows ShopNest to be fully functional, secure, and scalable, while providing a smooth experience for users and administrators.

High-Level Architecture Diagram

Methnatth photo ekk danna one

Database Schema

Methenta table tika danna one

5.User Interface Design

The **ShopNest** website is designed to provide a smooth and intuitive experience for both users and administrators. The interface focuses on simplicity, responsiveness, and easy navigation.

**Home Page**

* Displays featured products, categories, and promotions.
* Navigation bar includes links to About, Contact, Category, Cart, Sign In, and Login.
* Responsive design works well on desktop, tablet, and mobile devices.

**User Pages**

1. **Login / Signup** – Users can securely create accounts and log in.
2. **Product Listing** – Products are displayed with images, names, prices, and categories.
3. **Product Details** – Detailed description, price, and “Add to Cart” option.
4. **Cart** – Users can view, update, or remove items before checkout.
5. **Checkout** – Secure form to confirm orders and view total price.
6. **Order History**  – Users can track past orders and their status.

**Admin Pages**

1. **Dashboard**  – Overview of products, users, orders, and sales reports.
2. **Product Management**  – Add, update, or delete products and manage stock.
3. **Category Management** – Add or edit product categories.
4. **Order Management**  – View, update, or process customer orders.
5. **User Management** – View, edit, or block user accounts.

**Design Principles**

* **Consistency:** Uniform layout, buttons, and fonts throughout the website.
* **Clarity:** Clear labels, easy navigation, and readable text.
* **Responsiveness:** Works on multiple devices and screen sizes using Bootstrap.
* **Feedback:** Alerts and messages for successful actions, errors, or validations.

**Summary:**  
The UI design ensures that users can easily browse, shop, and manage their accounts, while admins can efficiently control the store, monitor orders, and manage inventory.

Chapter 6 – Development Tools and Technology

* 1. Development Methodology

The **ShopNest** website was developed using the **Waterfall Model**, a step-by-step approach. Each step is completed before moving to the next.

**Steps:**

1. **Requirement Analysis** – Collected all needs from users and admins.
2. **System Design**  – Planned the database, pages, and features.
3. **Implementation** – Coded the website using HTML, CSS, Bootstrap, JavaScript, PHP, and MySQL.
4. **Testing** – Checked that login, cart, orders, and other features work correctly.
5. **Deployment**  – Hosted the website and connected the database.
6. **Maintenance** – Fixed bugs and updated features when needed.
   1. Programming Tools

The following tools were used to develop the **ShopNest** e-commerce website:

**1. Code Editors / IDEs**

* **VS Code** – Used for writing HTML, CSS, JavaScript, and PHP code.
* **PHPStorm** – Optional IDE for PHP development and debugging.

**2. Web Technologies**

* **HTML5**  – Structure of web pages.
* **CSS3 / Bootstrap 4** – Styling and responsive design.
* **JavaScript / jQuery** – Interactive elements and form validation.

**3. Backend & Database**

* **PHP** – Server-side scripting to handle user requests and database operations.
* **MySQL** – Database management for storing users, products, orders, and categories.

**4. Tools / Libraries**

* **Bootstrap Icons**

Icons for buttons, menus, and navigation.

* **Popper.js**

Supports tooltips and dropdowns in Bootstrap.

* **XAMPP / WAMP / PHP Built-in Server**

To run the website locally.

**5. Version Control**

* **Git / GitHub** – To manage code versions and collaborate.

Chapter 7 - Future Work

The ShopNest e-commerce website can be improved and expanded in the future with the following features:

1. **Mobile Application** – Develop a mobile app for Android and iOS to allow users to shop on the go.
2. **Payment Gateway Integration** – Add support for online payments like credit cards, PayPal, and mobile wallets.
3. **Advanced Search & Filters** – Implement better search options with filters for price, category, and ratings.
4. **User Reviews & Ratings** – Allow customers to leave reviews and rate products.
5. **Notifications & Alerts** – Send email or push notifications for order updates and promotions.
6. **AI Recommendations** – Suggest products to users based on their browsing and purchase history.
7. **Multi-language Support** – Support multiple languages for a wider audience.
8. **Admin Analytics Dashboard** – Add advanced reports and charts to track sales, orders, and user activity.

**Reference for ShopNest**

1. W3Schools. *HTML, CSS, JavaScript, PHP, MySQL Tutorials.* Available at: <https://www.w3schools.com>

Links dnna one mewata

1. Bootstrap. *Bootstrap 4 Documentation.* Available at: https://getbootstrap.com/docs/4.3/getting-started/introduction/
2. PHP.net. *PHP Official Documentation.* Available at: [https://www.php.net/docs.php](https://www.php.net/docs.php?utm_source=chatgpt.com)
3. MySQL. *MySQL Documentation.* Available at: https://dev.mysql.com/doc/
4. Stack Overflow. *Programming and Development Solutions.* Available at: [https://stackoverflow.com](https://stackoverflow.com?utm_source=chatgpt.com)
5. GitHub. *Version Control and Project Management.* Available at: [https://github.com](https://github.com?utm_source=chatgpt.com)

Appendices

Github commit history and respository link

The project’s source code was hosted on Github, with the commit history demonstraiting colloborative efforts.

The respsity link is : <https://darshanachinthaka.github.io/web-project-EC/index.html>

**Team plan and Responsibilities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | NSBM student no | Name | Task carried out | Comments by the group Leader |
| 01. |  |  |  |  |
| 02. |  |  |  |  |
| 03. |  |  |  |  |
| 04. |  |  |  |  |
| 05. |  |  |  |  |
| 06. |  |  |  |  |
| 07. |  |  |  |  |