

Online Handmade Artisan Portal: A Web-Based E-Commerce Platform for Digital Empowerment of Local Artisans

Guggilla Shivani¹, Dr. M. Dhanalakshmi²

¹ Post Graduate Student, Department of Computer Science and Engineering, JNTUH, Hyderabad, Telangana, India

² Professor, Department of Computer Science and Engineering, JNTUH, Hyderabad, Telangana, India.

Article Info

Article History:

Published: 05 August 2026

Publication Issue:

Volume 3, Issue 8
August-2026

Page Number:

15-21

Corresponding Author:

Guggilla Shivani

Abstract:

Online shopping platforms today are largely designed for factory-made, mass-produced goods, leaving local artisans and small handmade-product creators with few dedicated channels to reach buyers and earn a fair income. This paper presents the design and implementation of the Online Handmade Artisan Portal, a web-based marketplace that connects artisans directly with customers who value authentic, handcrafted products. The portal allows artisans to register, build a personal storefront, and list products with images, descriptions, and prices, while customers can search and filter listings, add items to a cart, complete secure checkout, track orders, and post reviews. An administrative interface supports user management, content moderation, and reporting to maintain platform quality. The system is built using ReactJS for the front end, Spring Boot for backend services, and MySQL for persistent storage, communicating through RESTful APIs to ensure scalability, security, and maintainability. Role-based access control, encrypted credential storage, and SSL-secured data transmission protect user information, while a modular three-layer architecture (presentation, business logic, and data access) keeps the system easy to extend. Unit, integration, system, and user-acceptance testing confirm that the core workflows of registration, product management, cart handling, checkout, and order tracking function correctly under normal load. The resulting platform gives artisans a sustainable digital channel for their craft, gives buyers a trustworthy source of authentic handmade goods, and contributes to the preservation of local cultural heritage through technology-enabled commerce.

Keywords: handmade e-commerce, artisan marketplace, ReactJS, Spring Boot, MySQL, REST API, role-based access control

1. Introduction

Digital commerce has expanded rapidly over the last decade, but the majority of e-commerce platforms are built around standardized, factory-produced catalogues. This design assumption creates a structural disadvantage for local artisans and small-scale craftspeople, whose products are one-of-a-kind, culturally rooted, and produced in low volumes. Artisans who wish to sell online today typically rely on generic marketplaces or social media, both of which lack the tools needed to manage listings, orders, payments, and customer trust in an organized way. The result is that many skilled craftspeople remain confined to local exhibitions, craft fairs, or word-of-mouth referrals, limiting both their market reach and their income.

The Online Handmade Artisan Portal addresses this gap directly. It is a dedicated web platform that lets artisans register, create a public profile, and list handmade products with photographs, descriptions, and pricing, while customers can browse by category, search for specific items, place secure orders, and leave reviews. The system is implemented with ReactJS on the front end, Spring Boot on the back end, and MySQL as the data store, with security measures such as authenticated login, role-based access, and encrypted transactions built in from the start. By giving

artisans ownership of their storefront and giving customers a transparent, trustworthy shopping experience, the portal supports local livelihoods while preserving cultural craftsmanship.

The primary objectives of this work are to: (i) provide artisans a dedicated online platform to showcase and sell handmade products to a wider audience; (ii) offer customers an intuitive interface for browsing, searching, and purchasing across categories; (iii) enable secure transactions through safe payment handling, order tracking, and role-based access control; (iv) support efficient administration of products, orders, and user accounts through an integrated admin panel; and (v) bridge the gap between traditional artisans and digital commerce in a way that promotes cultural heritage alongside economic growth.

2. Literature Review

General-purpose e-commerce platforms are optimized for high-volume, standardized products and do not adequately serve artisans whose goods are unique, low-volume, and often tied to cultural context. Prior study of existing marketplaces highlights recurring limitations relevant to this project. Handmade products are frequently overshadowed by mass-produced listings on large marketplaces, reducing artisan visibility; there is little structured space to communicate the cultural or personal story behind a handcrafted item; order and inventory management for small sellers is often manual, relying on phone calls, informal messaging, and cash-based settlement; and the absence of verified profiles or structured reviews makes buyers hesitant to purchase handmade goods online. Existing generic platforms are also not tuned to the packaging, shipping, and pricing variability typical of handmade items.

These observations motivate three design priorities carried into the proposed system: (a) trust-building mechanisms such as verified artisan profiles and customer reviews; (b) a centralized, artisan-owned storefront model rather than an anonymous product feed; and (c) an integrated, digitized workflow for listings, orders, and payments so that sellers are not dependent on ad-hoc manual processes. Publicly available technical documentation and community resources for ReactJS, Spring Boot, MySQL, and RESTful API design (see References) were consulted to select an architecture that is scalable, secure, and maintainable while remaining approachable for a small development team.

3. Proposed System and Methodology

The proposed system is a purpose-built digital marketplace that replaces the disconnected, manual methods used by artisans today with an integrated, artisan-centric platform. Its design centres on a centralized marketplace in which each artisan owns a customizable storefront showing their products, biography, and customer feedback; independent product management so artisans can add, update, and remove listings with real-time availability; advanced search and discovery by category, price range, material, or cultural origin; secure authentication with distinct roles for customers, artisans, and administrators; trust-building features such as verified-artisan badges, reviews, and transparent return policies; and an administrative dashboard for monitoring usage, moderating content, and generating reports.

3.1 System Architecture

The system follows a three-layer architecture. The presentation layer, built with ReactJS, HTML5, CSS3, and Bootstrap, renders a responsive interface for customers, artisans, and administrators. The business logic layer, implemented with Spring Boot, exposes RESTful APIs that handle authentication, product management, order processing, and reporting, and enforces role-based access control. The data access layer uses MySQL to persist users, products, orders, and reviews in a normalized relational schema. Communication between layers occurs entirely through stateless REST endpoints using standard HTTP methods, which keeps the front end and back end loosely coupled and independently deployable. Figure 1 shows the entity-relationship diagram underlying the platform's core data model.

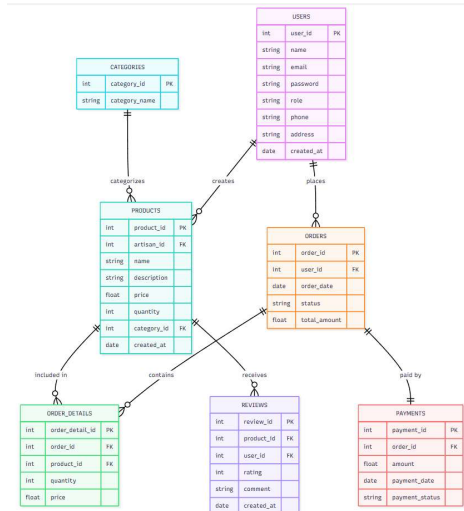


Fig. 1. Entity-Relationship Diagram of the Online Handmade Artisan Portal

3.2 Functional Modules

- User Module: handles registration, login, logout, and password recovery for both customers and artisans.
- Artisan Module: allows artisans to create and update profiles and manage their storefront.
- Product Management Module: supports adding, editing, and removing product listings with images, descriptions, and stock levels.
- Search and Navigation Module: enables keyword search and category/price/rating filters across listings.
- Shopping Cart and Checkout Module: manages cart contents, shipping details, payment, and order confirmation.
- Order Management Module: lets customers track order status and artisans update fulfilment progress.
- Admin Module: provides user management, content moderation, and platform-wide reporting.
- Notification Module: sends email and in-app alerts for order confirmations, shipping updates, and account changes.
- Security Module: enforces encrypted password storage, SSL-protected transmission, and role-based access control.

3.3 Technology Stack

The front end is implemented in ReactJS with HTML5, CSS3, and Bootstrap for a mobile-responsive, component-based user interface. The back end uses Java with the Spring Boot framework to expose secure, stateless REST APIs and to implement business logic such as order processing, validation, and role enforcement. MySQL serves as the relational database, chosen for its reliability, structured schema support, and wide tooling ecosystem. All client-server communication uses standard HTTP methods (GET, POST, PUT, DELETE) over RESTful endpoints, allowing the system to integrate future services such as logistics providers or a companion mobile application without redesigning the core architecture.

4. Implementation and Results

The platform was implemented and evaluated across its core user journeys: registration and authentication, product listing and storefront management, browsing and cart handling, and order tracking. The following figures illustrate representative screens from the deployed system.

Fig. 2. User Registration Form

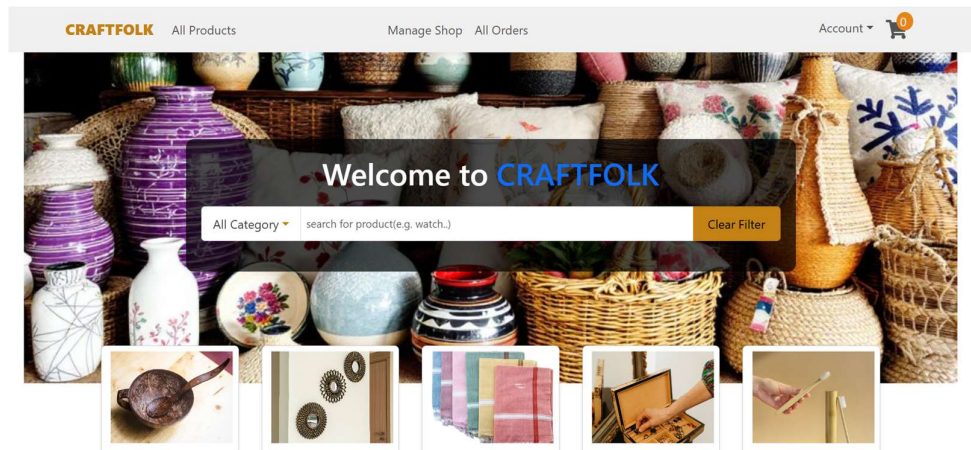


Fig. 3. Login Page

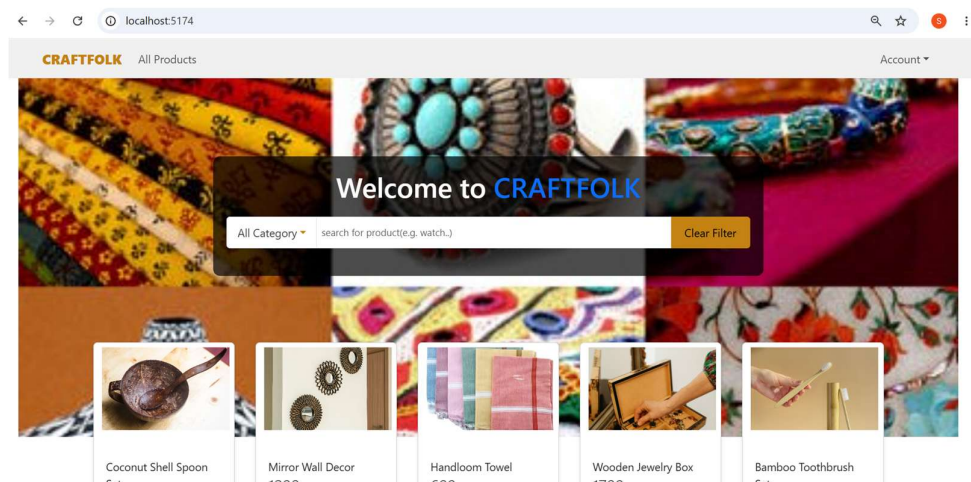


Fig. 4. Artisan Home Page with Manage-Shop Access

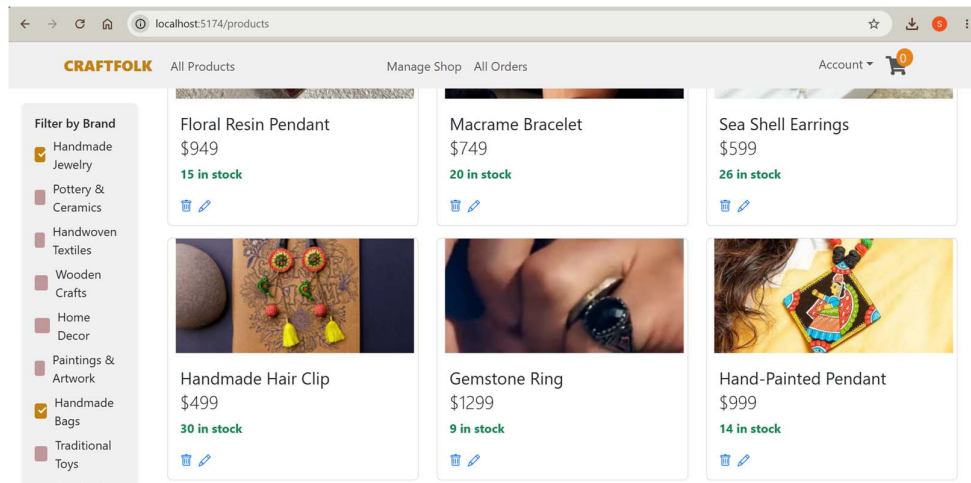


Fig. 5. Manage Shop Module for Product Entry

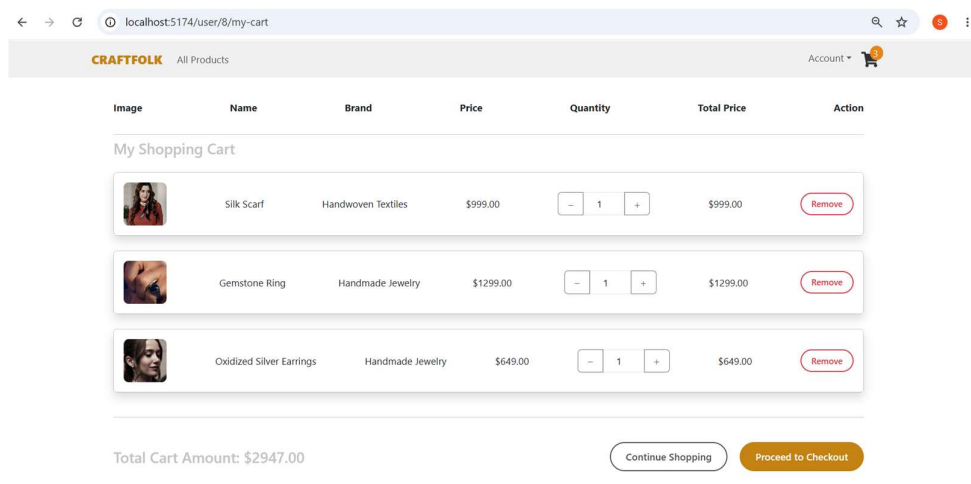


Fig. 6. Shopping Cart Module

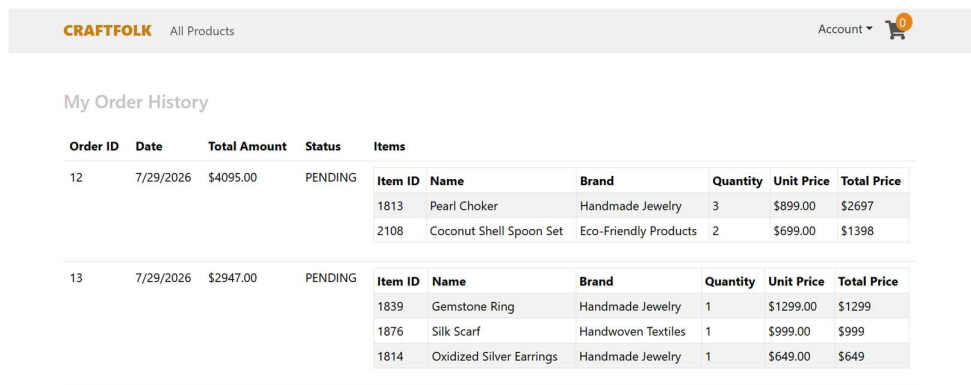


Fig. 7. My Orders Module

As shown in Figures 2 and 3, new users can register and log in through a validated form that separates artisan and customer roles at account creation, which drives the role-based views used throughout the rest of the system. Figure 4 shows the artisan home page, which exposes a Manage Shop option that is not visible to customer accounts, reflecting the role-based access control implemented in the business logic layer. Figure 5 shows the product-entry interface an artisan uses to add a new handmade item, such as a hand-made gemstone necklace, together with images, description,

and price; once submitted, the listing becomes immediately visible to customers, as reflected in Figures 6 and 7, which show the cart and order-tracking modules customers use to complete and monitor a purchase.

4.1 Testing

The system was validated using a layered testing strategy. Unit testing verified individual components such as form validation, price calculation, and authentication logic in isolation. Integration testing confirmed that the ReactJS front end, Spring Boot services, and MySQL database exchange data correctly across module boundaries, particularly for the registration-to-storefront and cart-to-order flows. System testing exercised complete end-to-end journeys, including registration, product listing, search, checkout, and order tracking, under representative usage conditions. Finally, user-acceptance testing gathered feedback from representative artisan and customer users to confirm that the interface was intuitive and that the core workflows matched real-world expectations. Across these test levels, the platform consistently responded within the target 2-3 second interaction window and correctly enforced role-based permissions between customer, artisan, and admin accounts.

4.2 Discussion

The results indicate that a dedicated, role-aware marketplace architecture directly addresses the gaps identified in the literature review. Centralizing listings under artisan-owned storefronts, rather than an anonymous product feed, gives handmade goods the visibility and cultural context that generic marketplaces do not provide. Digitizing the order and inventory workflow removes the manual bottlenecks artisans previously faced when relying on phone calls or social media for sales. The layered Spring Boot/MySQL back end also keeps the system maintainable: because business rules and data access are isolated from the ReactJS presentation layer, new features -- such as additional payment gateways or a mobile client -- can be added with minimal disruption to existing functionality.

5. Conclusion and Future Scope

This paper presented the Online Handmade Artisan Portal, a web-based marketplace built with ReactJS, Spring Boot, and MySQL that gives local artisans a dedicated channel to showcase and sell handmade products while giving customers a trustworthy way to discover authentic, handcrafted goods. Through role-based storefronts, integrated order and inventory management, secure authentication, and an administrative dashboard for oversight, the platform overcomes the visibility, trust, and workflow limitations of generic e-commerce sites and informal sales channels. Testing across unit, integration, system, and acceptance levels confirmed that the core registration, listing, cart, checkout, and order-tracking workflows function reliably.

Future work will extend the platform with integrated payment gateway support for multiple providers, AI-assisted product recommendations based on browsing and purchase history, a dedicated mobile application to complement the responsive web interface, and logistics-provider integration for real-time shipment tracking. Expanding analytics for artisans -- such as sales trends and customer engagement metrics -- would further help sellers make data-driven decisions about their offerings, strengthening the platform's role as a sustainable digital channel for handmade commerce.

Funding Details

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Disclosure Statement

The authors report no financial interest or benefit arising from the direct application of this research.

References

- [1] Spring Boot Official Documentation, Comprehensive reference for Spring Boot setup, configuration, and deployment best practices, [Spring.io](https://spring.io).
- [2] React Official Documentation, Authoritative guide to React concepts, APIs, and advanced component patterns, react.dev.
- [3] MySQL Documentation, Reference manuals for MySQL installation, data modeling, and query optimization, Oracle Corporation.
- [4] W3Schools React Tutorial, Tutorials and examples to learn core ReactJS principles, W3Schools.com.
- [5] W3Schools HTML & CSS, Reference material on HTML5 and CSS3 standards and best practices, W3Schools.com.
- [6] Mozilla Developer Network (MDN) Web Docs, Extensive guides on JavaScript, web APIs, HTML, and CSS, developer.mozilla.org.
- [7] Baeldung, Spring Boot Guides: high-quality articles on Spring Boot features, REST APIs, and security configuration, baeldung.com.
- [8] Stack Overflow, Community-driven Q&A platform used to resolve specific technical implementation challenges, stackoverflow.com.
- [9] Bootstrap Documentation, Reference for designing responsive front-end layouts and styling components, getbootstrap.com.
- [10] GitHub, Version control repository hosting and open-source project examples used for reference, github.com.
- [11] Unsplash, Source of high-quality royalty-free images used in product listing and interface design mock-ups, unsplash.com.
- [12] OWASP Foundation, Security standards and best practices for protecting sensitive data and application integrity, owasp.org.