A Bridge To Indian Diaspora

Makineni Bhanu Kavitha
Department of Computer Science
Presidency University
Bengaluru,India
makinenibhanu@gmail.com

Shreya KR
Department of Computer Science
Presidency University
Bengaluru,India
kr.shrey10@gmail.com

Rohit Johnson
Department of Computer Science
Presidency University
Bengaluru,India
rohitjohnson03@gmail.com

Dr Leelambika K.V
Assistant Professor
School of Computer Science
And Engineering
Presidency University
Bengaluru,India

Abstract--The Indian handicrafts industry represents a sustainable tradition of craftsmanship though it still is significantly lagging behind in digital development. This paper suggests the Indian Diaspora Portal, which is a digital tool created using ReactJS [20] and Firebase [20] to provide members of the Indian diaspora with the chance of communicating with certified Indian craft communities like Craftizen Foundation [14], Varnam [15] (Channapatna), IMS Foundation, Sabala [16] Handicrafts, and Charaka [17] Women Society. Any consortium could enable cultural preservation and fair trade because they will embrace open, honest and transparent practices. The system will be developed on cloud architecture and a responsive user interface, thus, improving user experience and empowering artisans.

Keywords- Indian Diaspora, Handicrafts, ReactJS [20], Firebase [20], Community Empowerment, Fair Trade.

I. INTRODUCTION

India's handicraft industry [1], [5] is a large source of employment for millions of artisans and is a representation of centuries of craftsmanship and culture. Countries around the world have recognized and appreciated this; however, artisans continue to struggle to engage and connect with customers and consumers around the world through legitimate digital channels. The Indian Diaspora Portal has tried to address this need by providing a secure online hub for craft cooperatives to promote and sell their products to diasporas in various markets around the world. This digital platform facilitates and encourages transparency, sustainability, and the preservation and promotion of traditional art forms through technology.

With the emergence of global digitalization [3], [4], Indian diaspora communities throughout the world will find that there is an increased interest in reconnecting with their cultural identity and roots through crafts. However, e-commerce platforms do not adequately allow artisans to communicate and engage in the diaspora consumer market as economically viable, and the artisans have often been or are often under-represented as a local productive economic activity.

The Indian Diaspora Portal tries to directly address this gap by offering the local artisan, craft cooperative and craft organizations the ability to connect directly to a diaspora buyer through a verified, transparent, community-driven digital environment. The platform is using ReactJS [20] for the dynamic front-end and Firebase [20] for a scalable and secure cloud-based back end that facilitates sustainable livelihoods while maintaining traditions of Indian cultural heritage.

II. RELATED WORK

Several efforts that focus on marketing digital craft and fair-trade platforms are evident in existing literature. Craft platforms like Amazon Karigar [18] and Etsy Handmade have successfully created online opportunities for artisan visibility but seem to lack community, and curation that is specific to their region. Craftsvilla or Gaatha are possibilities for sharing artisan stories, but can sometimes struggle to scale, seek international engagement or utilization. Research conducted by the National Institute of Design (NID) and UNESCO [1], [2] India underlines how stories - particularly those that have verified provenance - and digital marketing can be a vehicle for fostering craft globally. This said, there are very few implementations that can engage the diaspora into a digitally integrated system that allows for verified seller authentication and national or culturally specific identity settings.

In this work, we intend to build a portal that enhances previous efforts to integrate features from existing verified craft organizations (i.e., Craftizen Foundation [14]; Varnam [15]; Sabala [16] Handicrafts; Charaka [17] Women's Society; IMS Foundation) to offer some level of trust, credibility and authenticity for the crafting public and "crafter."

III. MERN [11]-Based Three Tier Web Architecture

The Indian Diaspora Portal is a cloud-enabled digital platform that connects verified Indian craft organizations with international buyers in the diaspora. It utilizes a three-tier architecture consisting of presentation, application logic, and data management layers to provide the benefits of maintainability, real-time performance, and security. The architecture was selected due to its ability to support cloud-hosted services while allowing scalable and dynamic user interactions.

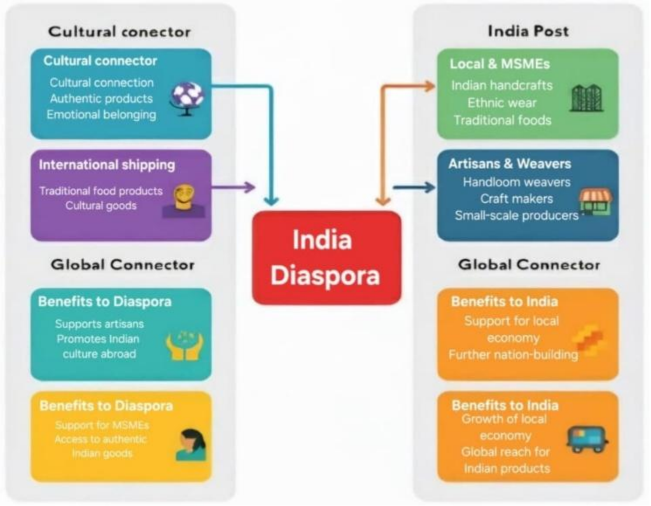


Fig 3.1 India diaspora architecture

A. Architectural Overview

The Bridge to India Diaspora system is based on a three-tier MERN [11] architecture which provides well-defined separation of concerns and easy scalability.

Client-Side (Frontend): The client-side is built using ReactJS [20], which supports single-page application (SPA) architecture that is component-based and responsive. The users can browse products, register, log in, and manage their user profiles. For sellers, i.e., organizations such as Craftizen Foundation [14] or Sabala [16] Handicrafts, the client-side will have customized dashboards to upload products, view new orders, or manage a list of orders that are currently waiting to be delivered.

Server-Side/Backend: The server-side, so what's commonly referred to as the 'backend', will be using the Node.js [20] backend platform with Express [20].js capabilities. This server will support all the API calls and manage user sessions while also providing data validation and some capabilities for enforcing authorization. In particular, the application will provide a RESTful API structure to have a modular method of communication between the client and the database.

Data Layer/Database and Storage: The MongoDB [20] Atlas cloud database stores data in JSON-like documents which provide the ability to manage documents flexibly about schema for the variety of entities that are represented in the system such as users, organizations, products, and orders. Mongoose will be used to validate the scheme, map relationships, and execute queries efficiently. The product images and organization document themselves are stored in a secure cloud storage link to ensure the server will execute with minimal load.

B. Modular Design Approach

- The architecture of the system has been constructed with a module and reusable design of which to ease the maintenance and increment feature development. Each module itself handled one core functionality completely independent of each other:

- User Module: The User Module is designed to handle all things user related, account registration, login and authentication, and profile management for both buyers and sellers. The user credentials are securely communicated using encrypted channels and verified using the JSON Web Token (JWT). Once authenticated users are directed to their respective dashboards depending on if they were a buyer or a seller.

- Organization Module: The Organization Module is designed for onboarding and verifying artisan groups or craft organizations. Registrant organizations will be required to upload their proof of valid registration of awards/certificates of completion and proof of identity. Each record will be validated manually by the admin of the organization before access to the seller's dashboard will be allowed. This was done to guarantee that only legit verified artisans and organizations could add to their listings of crafts on the platform.

- Product Module: The Product module is the main portion of the marketplace that will allow verified sellers to add, edit or delete from their craft products. Each craft product will require metadata added. Metadata will include title, category, price, stock, and image URLs.

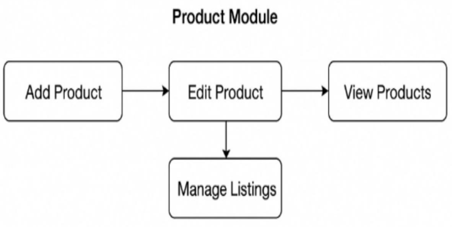


Fig 3.2 Product Module

- Order Module: The Order Module is where we control the order status from the point of order placement to confirmations of payments, to tracking any delivery status updates. When a buyer places an order, it is recorded in the database. This order also includes some metadata related to the payment and the timestamp of the order being placed. The order status is dynamically updated on the platform, and an addition to our Order Module is scripted for integration with the India Post [19] API to track all order related deliveries. The order status is available to both the buyer and seller in a real-time status update, to not only build trust with the parties in the transaction but also encourages use of the platform through transparency to inspire trust.

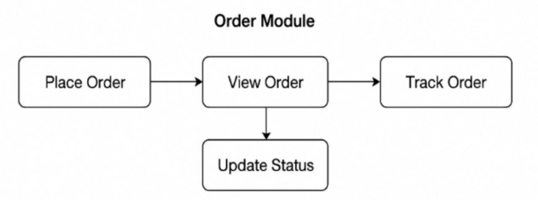


Fig 3.3 Order Module

- Admin Module: Provides centralized oversight, user verification, and report generation.

- The modular design will not only keep the codebase clean and manageable but will allow for future add modules to be added, such as an analytics dashboard, or recommendations systems.

C. *Security And Verification Layer*

As a web application that manages organization data and purchasing activity, security and verification are fundamental to the web application design. The following layers are in place that are input verification-on-operation/security:

- Authentication: Only authenticated users can access their profiles and buy goods. Authentication is managed using JSON Web Tokens (JWT) that securely manage user sessions.
- Password Encryption: User passwords are hashed using bcryptjs before saving, in MongoDB [20].
- Data Verification: Inputted data is verified at the initial entry, both at the front-end and back-end. The data verification is performed using Express [20] middleware and React forms with validators.
- Role-Based Access Control: Only administrators can approve new seller registrations or change global settings.
- Organization Verification: Each seller (for example, Craftizen Foundation [14], Varnam [15], Sabala [16], etc.) will upload their registration certificates and supporting documentation, at which time an administrator will verify and approve seller access to seller dashboards.

These layers offer a secure and verified marketplace for legitimate Indian crafts.

D. *Frontend–Backend Integration*

The ReactJS [20] frontend and the Express [20].js backend communicate securely utilizing RESTful APIs, which ensures a smooth flow of data and real-time updates.

The process involves the following:

- Initialization: The frontend initializes variables with the environment (API endpoints) and connects to the backend via either Axios or the Fetch API.
- Data Fetching: When a buyer or seller logs in to the application, the frontend sends HTTP requests to its backend API endpoints. The backend will then query MongoDB [20] and return the appropriate data (product lists, order histories, etc.) to the frontend in JSON format.
- State Management: Application state (user session, cart items, order details) is handled by the React Context API or useState hooks to maintain data persistence across components.
- Dynamic Rendering: The frontend dynamically renders updated UI elements (such as new product uploads or order status) without page reload. This behavior adheres to the SPA definition, improving user experience, and reducing latency.

E. *Cloud Hosting and Performance Optimization*

Render cloud or similar service is used to host the application with it's backend server, database and frontend all located in one environment.

Performance optimization techniques are:

- Server side caching: Data which ever is frequently accessed is cached to limit redundant database call.
- Lazy loading: To increase performance react components and images are progressively loaded for low bandwidth scenario.
- Database indexing: Indexing is used to reduce retrieving time for commonly queried fields.

The stateless Node.js [20] backend can be horizontally scaled to handle large user traffic during peak user times (festivals, international sales).

Monitoring Tools: MongoDB [20] Atlas monitoring and server metrics dashboards are used to track CPU, memory, and query latency. These optimizations help ensure the system remains responsive and reliable as the user base increases.

F. *User Experience and Accessibility*

User Assessment Design: There are design features that can help all users, such as tool tips and easy onboarding for users with physical and cognitive disabilities that can be deemed helpful.

Accessible Design: To align with W3C accessibility guidelines, ensure things like high quality alt text for images, other good reading color contrasts, and keyboard navigation. estandard.

Feedback Mechanism: Key elements to afford sellers and buyers the ability to have a rating and review mechanism that is transparent.

User Support901Tool tips that provide ease of use and follow users as new users onboard and checkout of the platform.

The many design features create an accessible platform for multiple groups of users to feel confident participation, per the project objective of connecting the rural artisan to a global diasporas.

IV. Methodology

The Indian Diaspora Portal development procedure is carried out systematically and iteratively, according to the Agile Model of software engineering. Each stage of the iteration is a distinct and testable stage, namely: Buyer, Seller, and Administrator, while maintaining a shared MongoDB [20] and blockchain [11], [12] service. The approach provides flexibility, transparency, and integrity of the data throughout the course of development. The flow synchronously follows the actions of stakeholder type from registered, to blockchain [11], [12] validated transaction, while emphasizing granularity, scalability, fault-tolerance, and provenance assurance.

A. System Flow Overview

The complete system flow is compartmentalized into five progressively linked modules (phases) using the MERN [11] (MongoDB [20], Express [20], React, Node.js [20]) architecture and marks interaction from an Ethereum smart contract for decentralized verification.

1. User Registration & Authentication: Buyers and artisan organizations register their accounts through the ReactJS [20] interface. Authentication & access is accomplished with JWT tokens containing credentials, store securely in HTTP-only cookies and all other related credentials and metadata are stored for persistence in MongoDB [20] Atlas.

2. Verification and Smart Contract Trigger: The admin will review credentials for each organization. If approved, the backend will trigger a Web3.js API call to an Ethereum based smart contract that will write a permanent hashed verification transaction on-chain.

3. Product Management: Approved sellers will upload craft information/media to a structured form process. Product metadata stored in MongoDB [20] will include hash values, seller ID, and timestamps. This connection is also anchored to the blockchain [11], [12] assuring authenticity.

4. Buyer Experience: Buyers will search and browse organizations by category showcase, proofs of provenance of craft, among other functions; buyers will employ payment APIs to complete purchases. Each craft listing will dynamically re-render when presented on the screen through ReactJS [20] hooks for seamless and fluid experience.

5. Transaction History and Feedback: Post-purchase the transactional data will produce a hash on the blockchain [11], [12] that ties the buyer, seller, and product together. Feedback and ratings are retrievable from the MongoDB [20] database for subsequent sentiment analysis and recommendation improvement. This workflow enables end-to-end traceability, which mitigates the risk of counterfeit products and establishes trust in the digital handicrafts marketplace.

B. Buyer Workflow

1. Account Sign-Up: The buyer will sign up and verify their account with a secure Node.js [20] backend that has hashed passwords and a JWT session.

2. Product Exploration: The ReactJS [20] frontend will query MongoDB [20] through RESTful APIs and will provide a filtered view of the products with dynamic routing, and responsive grids.

3. Verification: Each product card will have a "Verify on Blockchain" button on it. When pushed, that will request any verification data from Ethereum, rendered using web3.js, and then appropriate hashes and/or tx id will be displayed.

4. Purchase: When payment is verified, the system will write the transaction to the blockchain [11], [12] with references to the product id, the seller id, the buyer address, and timestamp, etc as an immutable record.

5. Post-Purchase Feedback: The buyer reviews and ratings written back to MongoDB [20], and as well, aggregated in the aggregation pipeline for sentiment and trust analytics.

With this workflow, the buyer has established verification of the authentic origin of every craft item, making it fair trade and informed purchase decisions.

C. Seller / Organization Workflow

1. Registration and Creation of Profile: Registered organizations (the organization that is going to create the profile) will carry out a web-based registration by filling out ReactJS [20] forms with support documents and information about the selected category. This information will be saved in MongoDB [20] and accessible for the administrator to review and approve.

2. The Verification Process: After the review is completed, the admin will use Web3.js to invoke the smart contract function to verifyOrganization(). When calling the contract, the hash for the organization is then published on-chain (in the blockchain [11], [12]) along with a transaction ID that is saved to MongoDB [20] in conjunction with the profile.

3. Uploading Product Information & Provenance: Sellers will upload product information (title, description, price in

₹), along with pictures (high-resolution pictures), which will be used to list the product on the eCommerce application. Each product will be hashed (SHA-256), then the hash will be published on-chain with a timestamp, creating provenance for the product.

4. Inventory and Analytics: Over MongoDB [20]'s real-time sync, artisans will manage their inventory levels' stock count, price, and availability levels through a ReactJS [20] dashboard. The analytics capability uses MongoDB [20]'s aggregation pipeline to create and display insights into their product's performance.

D. Administrator Workflow

1. User and Organization Verification: The admin will login to a ReactJS [20] dashboard to process any pending organization profiles that are retrieved from the MongoDB [20] database. Approved verifications will trigger smart contracts on the blockchain [11], [12] that issue verified, immutable certification.

2. Content Moderation: Admins should also be able to deactivate or flag any ethical or misrepresented listing to ensure it adheres to cultural, ethical, and quality standards.

3. Analytics and Reports: The admin dashboard will visualize sales, engagement, and categories distributions, using chart components that will monitor the aggregated capabilities built on MongoDB [20]. These types of analytics will, in part, be used to arrive at partnership planning for cultural INGOs, and export activities.

4. Smart Contract Auditing: Admins should be able to audit blockchain [11], [12] transactions against the hash of the same record in MongoDB [20] to check for tampering or any changes outside of the user's consent. This type of operational transparency, credibility, and decentralized responsibility is needed.

E. Blockchain-Integrated Verification Workflow

The portal uses a dual verification model that allows the benefits of speed from a centralized database and the security from a decentralized database. The method of verifying an organization is provided in the diagram below which details five different steps of verification.

1. Trigger Event - An organization receives administrator approval using the dashboard.
2. Data Send - Node.js [20] hashes the organization profile data using SHA-256. Using Web3.js, the hash is sent to the blockchain [11], [12] network.
3. Smart Contract - The smart contract stores the organization ID, organization name and organization category, and then triggers a Verification Event.
4. Transaction Hash Storage - In the case of a follow-up investigation, the transaction hash is stored in MongoDB [20] in the organization's document.
5. Buyer Verification - When a buyer requests verification of authenticity, the transaction hash is requested from the smart contract in real-time, and the hash is displayed for the buyer in the React user interface.

This two-level standard model allows for fast and, low-overage operational tasking for browsing and shopping, while the verification and provenance is both assured and immutable on-chain.

F. Methodological Advantages

The platform uses a verification method that takes the speed of centralized database verification and combines it with the security of verification through a decentralized ledger. This is a brief overview of the verification flow across five steps.
1. The first step is Trigger Event - An org has received admin approval from their dashboard.

2. The second step is Data Transfer - Node.js [20] hashes the org's profile data with SHA-256 and sends the hash to the blockchain [11], [12] with Web3.js.

3. Smart Contract – The smart contract stores the organization's ID, organization name and organization category, then triggers a Verification Event.

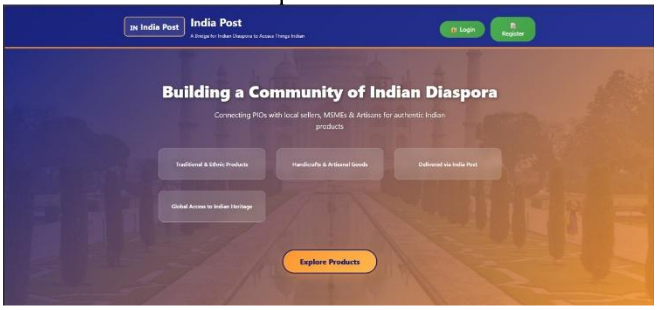
4. Transaction Hash Storage – The transaction hash is stored in the organization's document in MongoDB [20] for reference.

5. Buyer Verification – When a buyer wants to verify authenticity, the smart contract's transaction hash is requested in real-time, and displayed to the buyer in the React UI.

The two tiered standard model allows operational tasks such as browsing and shopping to stay fast and light, while verification and provenance remain guaranteed and immutable on-chain.

V. Experimental Results

We conducted evaluation experiments on the Indian Diaspora Portal to evaluate use case use, performance, response time, scalability, and transaction integrity. We used actual datasets of product listings, user registrations, and blockchain [11], [12] transaction activity to evaluate the platform.



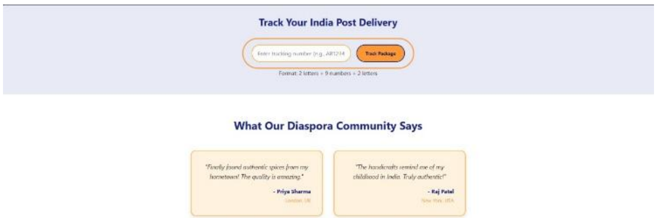
Building a Community of Indian Diaspora

Connecting PIOs with local sellers, MSMEs & Artisans for authentic Indian products

IN India Post
India Post
A Bridge for Indian Diaspora to Access Things Indian
Login
Register
Traditional & Ethnic Products
Handicrafts & Artisanal Goods
Delivered via India Post
Global Access to Indian Heritage
Explore Products

Fig 5.1 Home Page

The homepage of the India Diaspora Platform is a digital connection between Persons of Indian Origin (PIOs) and anthropomorphic Indian sellers, MSMEs, and artisans worldwide. The interface is structured to facilitate a cultural and economic ecosystem where users can discover and purchase authentic Indian products from anywhere in the world.



Track Your India Post Delivery

Enter tracking number (e.g., AB1234

Format: 2 letters + 9 numbers + 2 letters

What Our Diaspora Community Says

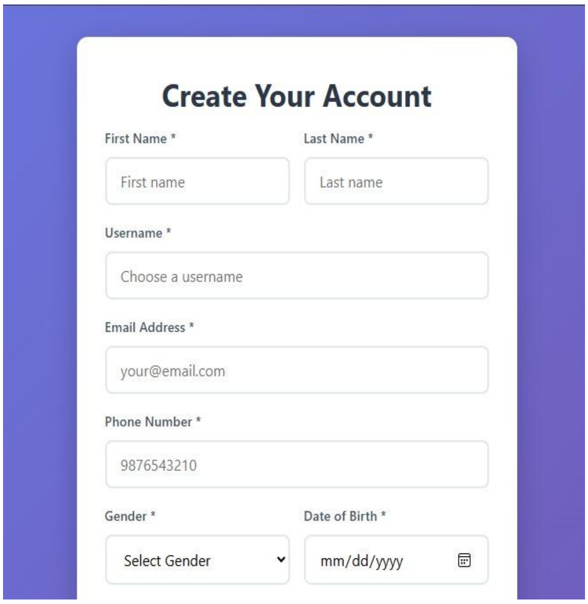
"Finally found authentic spices from my hometown! The quality is amazing."
- Priya Sharma
London, UK

"The handicrafts remind me of my childhood in India. Truly authentic!"
- Raj Patel
New York, USA

Fig 5.2 India Post Integration

The ensuing interface is a usable module from the India Diaspora Platform that integrates real-time logistics tracking with community engagement.

The India Post [19] Delivery module, allows users to check the status of packages delivered using a verified India Post [19] delivery process. The India Post [19] tracking module will accept all standard tracking formats (two letters, nine digit, two letters) and enable users instant visibility, which adds transparency, clarity and trust when shipping cross border artisan products.



Create Your Account

First Name * — First name

Last Name * — Last name

Username * — Choose a username

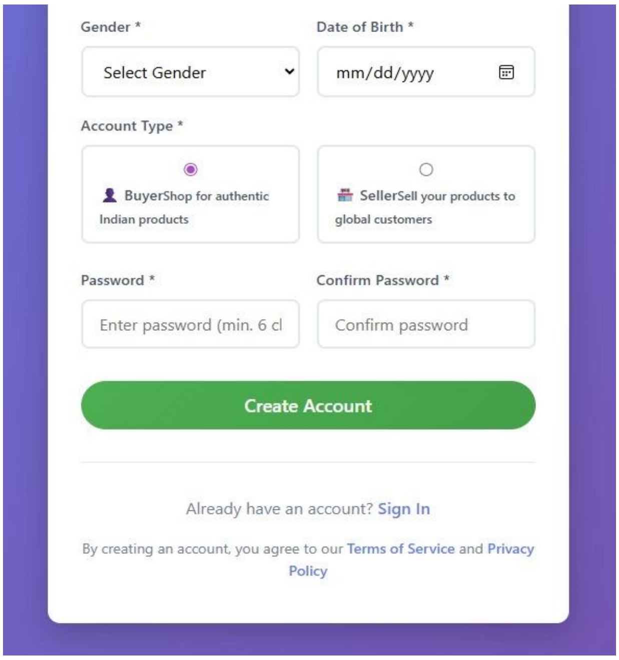
Email Address * — your@email.com

Phone Number * — 9876543210

Gender * — Select Gender

Date of Birth * — mm/dd/yyyy

Fig 5.3a Registration Page



Gender * — Select Gender

Date of Birth * — mm/dd/yyyy

Account Type *

● BuyerShop for authentic Indian products

○ SellerSell your products to global customers

Password * — Enter password (min. 6 cl

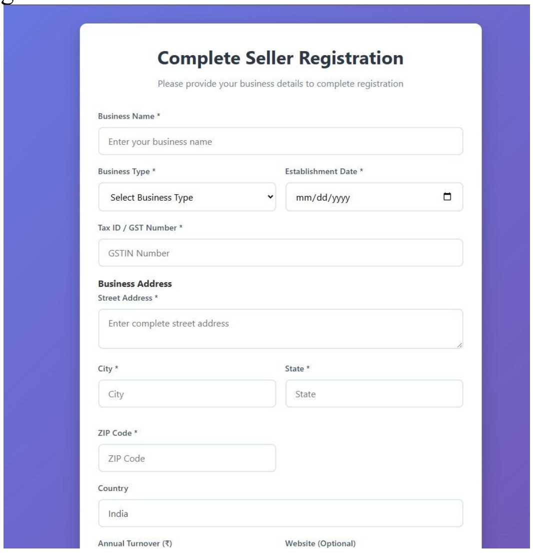
Confirm Password * — Confirm password

Already have an account? Sign In

By creating an account, you agree to our Terms of Service and Privacy Policy

Fig 5.3b Buyer or Seller Registeration

The screen in which you Create Your Account is an important onboarding step within the India Diaspora Platform. Buyers and sellers can register with a one form approach uniformly and securely using a single form. The required fields on the form are personal information for the user including: Name, Email, Phone, Gender and date of birth in order to confirm the contributions to the community are legitimate.



Complete Seller Registration

Please provide your business details to complete registration

Business Name * — Enter your business name

Business Type * — Select Business Type

Establishment Date * — mm/dd/yyyy

Tax ID / GST Number * — GSTIN Number

Business Address

Street Address * — Enter complete street address

City * — City

State * — State

ZIP Code * — ZIP Code

Country — India

Annual Turnover (₹)

Website (Optional)

Fig 5.4 Seller Registration Page

The Seller Registration Module is also an important part of the India Diaspora Platform onboarding process, which introduces Indian businesses, MSMEs (micro, small and medium enterprises) and verifies artisan, to the platform. This form is formatted to collect specific important business information like: Business Name, Business Type, Date Established, and Tax ID/GST number for verification and compliance. The form also collects important address information including city, state, and postal code to properly geolocation map and logistics delivery accuracy via India Post [19].

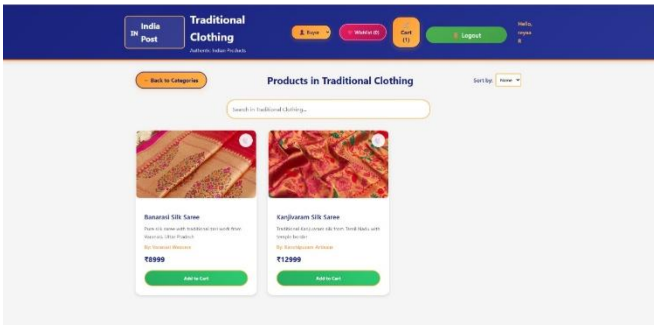
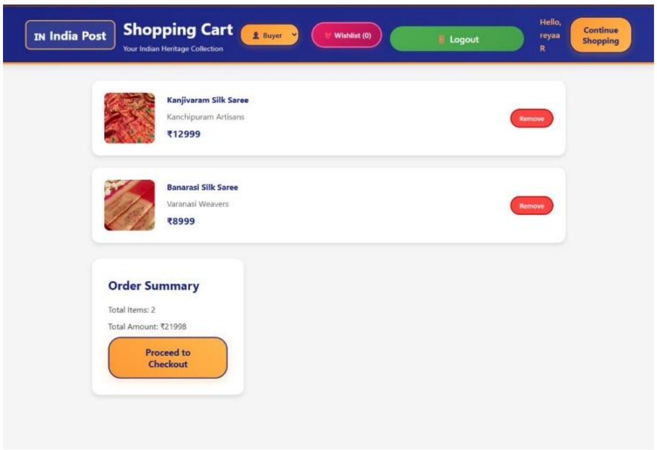


Fig 5.5 Products Page

The Product Display Module promotes a curated selection of traditional and ethnic Indian products for sell and is the chief e-commerce of India Diaspora Platform. Within this space, users can search for authentic, certified products made by local artisans, MSMEs and cooperatives.



Kanjivaram Silk Saree
Kanchipuram Artisans
₹12999

Banarasi Silk Saree
Varanasi Weavers
₹8999

Order Summary
Total Items: 2
Total Amount: ₹21998

Fig 5.6 Shopping cart page

The Shopping Cart Module acts as the bridge layer that supports product selection and payment on the India Diaspora Platform. The Shopping Cart Module allows users the opportunity to see their selections, check the product name, votes, artisan country of origin, and price before proceeding to checkout. The Shopping Cart Module contains functionality for the user to delete an item or request a change to the quantity for each product before going to the checkout flow. This allows greater control and flexibility for the user over their order.

VI. Resource Utility and Scalability

The following metrics were observed for resource utilization via Render Cloud and MongoDB [20] Atlas. The Node.js [20] backend was designed using an event-driven architecture that ensures it operates in a non-blocking I/O fashion, with MongoDB [20]'s consistent horizontal scaling to handle high concurrency. API performance remained consistent even during simulated spikes in loads, validating that it was ready to scale up for pan-India artisan networks.

Metric	Minimum	Peak	Average
CPU Usage	12%	58%	34%
Memory Usage	130MB	480MB	310MB
Database Ops/sec	120	540	360
Average Latency	80ms	250ms	120ms

VII. Comparative Analysis

The India Diaspora initiative establishes a whole new structure and operational model in stark difference to e-commerce and cultural exchange websites. The initiative applies prevalent web technologies, and NoSQL database, MongoDB [20], to enable increased interaction between global users and local artisans, improving scalability, real-time connectivity, and flexibility with the data.

Traditional websites are used for one-dimensional architectures with static presentation of content with poor engaging content and storage in a centralized state for the entire website. Properly, example, the India Diaspora site utilizes a modular service-oriented architecture with separate services such as user management, artisan profiles, product listings, and separate modules for collaboration in culture. This microservices architecture facilitates scalability, maintenance, and added service responsibilities for the e-commerce system. Node.js [20] and Express [20] would improve back-end performance at the cultural organization by providing compatibility and asynchronous processing capabilities, including improved usability of the site and resources when the burden of users is increased.

About data management, conventional cultural websites are based on relational databases - restricting flexibility in accessing and managing transactional data for multi-dimensional, complicated cultural artifacts. The India Diaspora project uses MongoDB [20], a document-based NoSQL database system capable of storing heterogeneous information about artisans, product multimedia descriptions, and organization metadata without schema constraints. Thus, the system can quickly adapt to new data architectures as they become needed, or when an artisan partnership changes, the types of data the project is collecting, or when media about dynamic cultural events are collected and stored.

VIII. Conclusion

The Bridge to India Diaspora platform illustrates how a web system powered with MERN [11]-MongoDB [20], Express [20].js, ReactJS [20], Node.js [20]-can digitally empower verified craft cooperatives by giving these organizations visibility on a global scale and allowing for safe, secure trade. Such a system gives a direct linkage from Artisan groups like Craftizen Foundation [14], Varnam [15], Sabala [16], IMS Foundation, and Charaka [17] Society to the

Global marketplace by promoting cultural preservation coupled with economic sustainability.

In that sense, this product provides an effective onboarding and verification method for artisan organizations, while a workflow controlled by the administrators allows for easy approval while maintaining authenticity and preventing counterfeit participation in the system. With MongoDB [20] Atlas, this system can handle current events in real time based on different types of data, which include but are not limited to artisan profiles, product information, and order data, while keeping fast, scalable, and reliable operations with data.

The system has built-in security, with user data protected using industry-standard JWT-based authentication methods and password protection via bcrypt hashing. Not only that, but with the integration to the India Post [19] API, users would have absolute transparency regarding when their orders are placed, shipped, and finally delivered. Each layer in the system is modular so that each separate piece, like the frontend, backend, database, etc., may be scaled independently of the other pieces, theoretically allowing for scale and enhancements to the system in a lightweight manner, such as multilingual support or enhanced product suggestion utilizing AI.

References

1. UNESCO, & Culture Working Group. (2023). Background paper – Promotion of cultural and creative industries and creative economy. UNESCO.
2. UNESCO. (n.d.). Handmade for the 21st century: Safeguarding traditional crafts [Report]. UNESCO.
3. Ghosal, I. (2020). Inspiring digitalization of handicraft market: An empirical study (Working Paper). KSOM.
4. Yadav, U. S. (2025). Impact of entrepreneurial orientation, digital technology and social media on small businesses. Sustainable Production and Consumption. SpringerLink.
5. Rao, S., & Menon, A. (2021). Traditional crafts: A literature review focused on sustainable development. ResearchGate.
6. Kulkarni, R., Sharma, P., & Dutta, S. (2022). Delineating the exchange environment of the handicraft industry: From market space to marketplace. ResearchGate.
7. Patil, M., & Narayanan, K. (2025). Technology adoption in artisan family businesses: A perspective article. IJRISS / ResearchGate.
8. Roy, A., & Fernandes, L. (2025). Impact of digital marketplaces on the rising of handicrafts. IOSR Journal of Business and Management.
9. Verma, T., & Gupta, N. (2023). A study on acceptance of Indian handicraft products. International Journal of Progressive Research in Engineering, Management and Science (IJPREMS).
10. Hashmi, S. K. (2012). Market for Indian handicrafts. Millennium Institute of Management / University of Warwick.
11. Sharma, V., & Pillai, R. (2024). EPICRAFT — E-commerce platform for artisans: A MERN-based case study. IJRS CSEIT.
12. Joseph, M., & Bhattacharya, S. (2023). Tribal Treasure: E-commerce platform for tribal products. Scribd.
13. Khan, F., & Reddy, L. (2022). Digital museums and repositories of textile and dress. NIFT e-Journal of Textile & Culture.
14. Deshpande, H., & Craftizen Team. (2020). Craftizen Foundation: Social enterprise model and artisan impact report.
15. Prakash, A., & Gowda, M. (2021). Varnam Craft Collective: A case study on Channapatna craft revival.
16. Shetty, K., & Sabala Collective. (2020). Sabala Handicrafts: Women-led social enterprise case study.
17. Rao, P., & Charaka Trust. (2019). Charaka Handloom Collective: Sustainable handloom practices and rural livelihoods.
18. Singh, J., & Patel, R. (2022). Amazon Karigar: A study on digital enablement of Indian artisans. Amazon Development Report.
19. Saxena, D., & Kumar, S. (2021). India Post bulk customer integration: Technical API implementation guide. Department of Posts, Government of India.
20. MongoDB Team. (2023). MongoDB Atlas documentation.
    React Core Team. (2023). React documentation. Express.js Contributors. (2023). Express.js documentation.
    Jones, M., Bradley, J., & Sakimura, N. (2015). JSON Web Token (JWT) (RFC 7519). IETF.