

DETAILED REQUIREMENTS/ SPECIFICATIONS

Kalaanubhav.com – Multi-Vendor E-Commerce Marketplace

1. Project Overview

Kalaanubhav is a centralized digital marketplace for promoting and selling handicrafts and artwork created by artisans.

The solution shall be developed as a single, unified multi-vendor e-commerce platform using a custom-built technology stack. The application is built as a modular Node.js/NestJS backend API with a React/Next.js frontend, backed by a self-managed, audit-first PostgreSQL database, as documented in Appendix A (Technology Stack) and the accompanying Development Plan.

Each approved artisan shall operate as an independent vendor within this single application — with their own storefront, product catalogue, orders and earnings — without requiring a separate website, subdomain, or installation per vendor.

Website: kalaanubhav.com (single domain, single installation).

2. Objectives

- Create a unified, trusted marketplace for artisans.
- Facilitate direct-to-customer sales, eliminating intermediary dependency.
- Promote regional handicrafts, GI-tagged products, and traditional goods to wider markets.
- Enhance artisan visibility, digital literacy, and financial inclusion.
- Support seamless online payments and last-mile logistics.
- Enable data-driven ecosystem development and policy insights.
- Facilitate access to national and international markets.

3. Platform Structure

3.1 Marketplace Website (Single Site)

The website shall function as both the informational portal and the transactional marketplace.

Features:

- Home
- About Kalaanubhav
- Vision & Mission
- Government / Organization Information
- Contact Us
- Privacy Policy
- Terms & Conditions
- Artisan Directory (search/browse across all vendor storefronts)
- Product Catalogue
- Shopping Cart
- Checkout
- Customer Accounts
- Orders
- Payments

- Shipping
- Reports
- Banners
- Gallery
- Events

All products from all artisans shall be discoverable and purchasable from this single website. Each artisan's products, orders and earnings shall remain logically isolated to that artisan (vendor-level data isolation) even though there is only one website.

3.2 Vendor Storefronts

Each approved artisan shall receive a dedicated vendor storefront page within the single website (not a separate website), displaying:

- Storefront Banner / Logo
- Artisan Profile & Biography
- Product Catalogue (that artisan's products only)
- Ratings & Reviews
- Contact / Follow Options

4. User Roles

4.1 Administrator

A single, centralized Administrator role shall manage the entire marketplace. There shall be no separate site-level or regional administrators.

Responsibilities include:

- Manage Marketplace Website
- Manage Users
- Approve Artisan Registrations
- Approve Products
- Manage Orders (platform-wide)
- Manage Hubs
- Configure Payment Gateway
- Configure Shipping
- Manage Policies
- Configure Taxes
- Configure Commission Rules
- View System Reports
- Manage Featured Products / Featured Artists
- Manage entire CMS (see Section 25)

4.2 Artist / Vendor

Each approved artist shall function as an independent vendor within the single marketplace.

Artists shall be able to:

- Login
- Manage Profile & Storefront
- Upload Products
- Edit Products
- Manage Inventory
- View Orders

- Update Shipment Status
- View Sales Reports
- View Earnings

Artists shall only have access to their own information. No artist shall access another artist's information.

4.3 Customer

Customers shall be able to:

- Register / Login
- Browse Products
- Search Products
- Add Products to Wishlist
- Purchase Products
- Track Orders
- Download Invoices
- Submit Reviews
- Manage Addresses

5. Artist Registration

Artists may be registered through:

Admin Registration

Administrator creates artist accounts.

Frontend Registration

Artists submit registration requests.

Information captured:

- Name
- Mobile
- Email
- Address
- Assigned Hub
- Bank Details
- PAN
- Aadhaar (Optional)
- GST (Optional)
- Biography
- Profile Photo
- Portfolio Images

Workflow:

- Registration Submitted
- Pending Verification
- Approved / Rejected
- Login Credentials Generated

6. Artist Public Profile

Each approved artist shall receive a dedicated public profile / storefront page. The page shall display:

- Artist Photograph
- Biography
- Craft Information
- Assigned Hub
- Portfolio
- Published Products

7. Product Management

Artists shall manage:

- Add Product
- Edit Product
- Delete Draft
- Manage Inventory

Product Information:

- Product Name
- Description
- SKU
- Category
- Material
- Weight
- Dimensions
- Price
- Sale Price
- Stock
- Images
- Videos (Optional)

Workflow:

- Draft
- Submitted
- Admin Review
- Approved
- Published

Rejected products shall contain review comments.

8. Product Catalogue

Customers shall be able to:

- Browse Categories
- Search Products
- Filter Products

Filters:

- Artist
- Category
- Material

- Price
- Availability
- Latest Products

9. Shopping Cart

Features:

- Add to Cart
- Remove Product
- Update Quantity
- Coupon Support
- Shipping Estimate

10. Checkout

Supported Payments via RazorPay:

- UPI
- Debit Cards
- Credit Cards
- Net Banking
- Wallets

Checkout shall display:

- Billing Address
- Shipping Address
- Taxes
- Shipping Charges
- Grand Total

Customers shall accept the platform policies before placing an order.

11. Order Management

Order Status:

- Pending Payment
- Processing
- Confirmed
- Packed
- Shipped
- Delivered
- Cancelled

Customers may:

- View Orders
- Track Orders
- Download Invoice

12. Hub Management

The platform may operate multiple hubs (regional pickup / fulfillment points) purely for logistics purposes — independent of any website or site structure.

Artists shall be mapped to one hub.

Administrators shall be able to:

- Create Hub
- Edit Hub
- Delete Hub
- Assign Artists
- Remove Artists
- View Orders by Hub

13. Delivery Management

The platform shall follow a Hub-Based Delivery Model.

Workflow:

- Customer Places Order
- Artist Receives Order
- Artist Packs Product
- Product Delivered to Assigned Hub / Ready for Pickup
- Courier Pickup from Hub
- Shipment Tracking
- Delivered

14. Shipping Integration

The platform shall integrate with Shiprocket supporting:

- Shipment Creation
- Pickup Scheduling
- AWB Generation
- Shipping Label Generation
- Shipment Tracking
- Courier Selection

Each Hub shall have its own pickup address. During shipment creation, the shipment shall automatically use the pickup address of the hub assigned to the selling artist.

15. Multi-Vendor Marketplace

Each artist shall function as an independent vendor within the single marketplace website.

Artists shall only access:

- Their Products
- Their Orders
- Their Earnings
- Their Inventory

No artist shall access another artist's information.

16. Payment Gateway

The platform shall support RazorPay payment gateway integration.

17. Split Payments (Future Enhancement)

Solutions such as Razorpay Marketplace/Route provide automatic settlement capabilities. However, implementation requires:

- Marketplace approval from the payment gateway.
- KYC verification of every artist.
- Linked merchant account for every artist.
- Compliance with gateway onboarding requirements.

Considering the onboarding effort, the recommended implementation for Phase 1 (Beta) is:

- Customer pays the platform.
- Artist commissions are calculated automatically.
- Administrator performs manual payouts on a periodic basis.

Automatic split settlements may be implemented in a future phase after completion of the required onboarding and approvals.

18. Inventory

Artists shall manage inventory.

Inventory Status:

- In Stock
- Low Stock
- Out of Stock

Administrators may update inventory if required.

19. Reports

Administrator Reports:

- Sales
- Revenue
- Orders
- Artist Sales
- Hub Sales

Customer Reports

Artist Reports:

- Sales
- Orders
- Inventory
- Earnings

20. Notifications

Customer:

- Registration
- Order Confirmation
- Shipment
- Delivery

Artist:

- Registration Approval

- Product Approval
- Product Rejection
- New Order

Administrator:

- New Artist Registration
- Pending Product Approval

21. Reviews & Ratings

Customers may:

- Rate Products
- Submit Reviews

Reviews shall be moderated before publication if enabled.

22. Wishlist

Customers shall be able to:

- Add Products
- Remove Products
- Move Products to Cart

23. Coupons & Promotions

Administrator shall manage:

- Percentage Discounts
- Flat Discounts
- Festival Campaigns
- Free Shipping Campaigns

24. Business Policies

The platform shall support configurable workflows for **product returns, exchanges, cancellations and refunds**, subject to Kalaanubhav's applicable business policies. Customers shall acknowledge these policies before placing an order.

25. CMS Management

The Administrator shall manage all content of the single marketplace website, including:

- Homepage
- About
- Contact
- News
- Privacy Policy
- Terms & Conditions
- Shipping Policy
- Cancellation Policy
- Return Policy
- Footer Content
- Gallery

- Events
- FAQs
- Featured Artists
- Featured Products
- Testimonials
- Downloadable Documents
- Homepage Banners
- Navigation Menu

Appendix A – Confirmed Technology Stack

Component	Confirmed Technology / Requirement
Backend / API	Node.js + NestJS — modular architecture organised around bounded contexts such as catalogue, vendors, orders, hubs, payments and users. PostgreSQL shall be accessed through parameterised SQL
Web Frontend	React + Next.js , with SSR/ISR for SEO and performance and a mobile-first responsive design.
Mobile Application	React Native + TypeScript , supporting Android and iOS from a common cross-platform codebase. The mobile application shall consume the same versioned REST APIs and business logic used by the web platform.
Multi-Vendor Marketplace	Custom-built multi-vendor architecture supporting independent artisan storefronts, product catalogues, orders, inventory and earnings, with vendor-level data isolation.
Database	PostgreSQL 16 , self-managed, with audit logging and PostgreSQL Row-Level Security (RLS) to enforce vendor-level data isolation.
Caching / Background Jobs	Redis , with a background job queue such as BullMQ for asynchronous processing.
Infrastructure / Hosting	AWS India Region , using Dockerised application containers. Production infrastructure shall support load balancing, auto-scaling and secure deployment.
Object Storage / CDN	AWS S3 for product, artisan and CMS media and backups, with AWS CloudFront for production content delivery.
Payment Gateway	Razorpay , with server-side integration and webhook verification. Phase 1 shall support the agreed payment modes and manual commission/payout tracking.
Shipping / Logistics	Shiprocket API , supporting hub-based pickup, shipment creation, AWB generation, label generation, courier selection and tracking.
Notifications	SMS and Email, with Firestore Cloud Messaging (FCM) for mobile push notifications. WhatsApp may be enabled as an additional channel subject to implementation requirements.
Authentication	JWT-based authentication. Mobile application shall support biometric authentication where supported by the device.
Security	TLS, MFA, RBAC, WAF, audit logging, secure credential management, SAST/DAST and third-party VAPT prior to Full Go-Live.
Multilingual / i18n	Database-driven multilingual architecture supporting English, Hindi and Punjabi at launch, with provision for Dogri, Kashmiri and Pahari.
Analytics	Dashboard-based analytics covering sales, revenue, orders, products, artisans, customers, hubs and regional performance.
Future Integrations	Architecture shall remain extensible for AI recommendations, AI agents/MCP or equivalent standards, ONDC, GeM, social commerce, international commerce and B2B functionality.

Note: This technology stack is detailed further in the Development Plan and the Infrastructure Requirements & Setup Guide. Post-launch support and maintenance from Phase 2 (Full Go-Live) onward shall be governed by the Annual Maintenance Contract (AMC), as specified in the RFQ.

Appendix B – Development Approach

The Kalaanubhav platform shall be developed and delivered as an integrated **web + mobile + API-based multi-vendor marketplace solution**, in accordance with the Detailed Requirements / Specification and the RFQ.

B.1 Development Approach & Responsibilities

The Technology Partner shall undertake the complete development lifecycle, including:

1. Requirement validation and stakeholder consultations.
2. Functional and technical architecture.
3. UI/UX design and design system development.
4. Web application development.
5. Android and iOS mobile application development.
6. Backend and versioned API development.
7. Database design and implementation.
8. Integration of payment, logistics, notification and other approved third-party services.
9. Implementation of security, access control and audit mechanisms.
10. Cloud infrastructure setup, deployment and configuration.
11. User Acceptance Testing (UAT), defect resolution and production readiness.
12. Production deployment and Full Go-Live.
13. Documentation, training and knowledge transfer.
14. Post-launch support and maintenance as specified in the RFQ.

The development shall follow a **modular, scalable and API-first architecture**, enabling the web and mobile applications to use common backend services and business logic.

B.2 Mobile Application Delivery

The mobile application shall be developed using **React Native and TypeScript** for Android and iOS through a common cross-platform codebase.

The mobile application shall provide the core customer marketplace journey, including:

- Registration and login
- Product discovery and search
- Artisan/vendor discovery and storefronts
- Product details
- Wishlist
- Shopping cart
- Checkout and Razorpay payments
- Address management
- Order history and tracking
- Invoice access
- Reviews and ratings
- Push notifications
- Profile management

The application shall consume the same versioned REST APIs and business logic as the web platform.

The Technology Partner shall clearly identify any functionality proposed to be unavailable on mobile in the Technical Bid. Any such exclusion shall require PI-RAHI approval.

B.3 API & Integration Approach

The web portal and mobile applications shall use a common backend API layer.

The Technology Partner shall ensure consistent handling of key platform data and transactions across channels, including:

- Customer accounts
- Artisan/vendor information
- Product catalogue and availability
- Wishlist and cart
- Orders and order status
- Payments
- Shipping and logistics
- Promotions
- Notifications
- Reviews and ratings

Third-party integrations shall be implemented through secure APIs and shall include appropriate error handling, authentication and transaction/status verification.

B.4 Testing & Quality Assurance

The Technology Partner shall undertake testing throughout the development lifecycle, including:

- Functional testing
- Integration testing
- API testing
- Web application testing
- Android and iOS device testing
- Cross-browser and cross-device testing
- Performance and load testing
- Security testing
- Regression testing
- User Acceptance Testing (UAT)

A third-party VAPT shall be completed and identified security vulnerabilities shall be addressed prior to Full Go-Live.

The Technology Partner shall maintain appropriate test cases, defect logs and test reports as part of the project documentation.

B.5 Documentation, Source Code & Handover

The Technology Partner shall maintain and provide, as applicable under the contract:

- Source code repositories
- API documentation
- Database schema and documentation
- System and architecture documentation
- Deployment documentation
- Infrastructure configuration documentation
- Test cases and test reports
- Security/VAPT reports
- User manuals
- Administrator manuals
- Mobile application build and release documentation
- Training and knowledge-transfer material

All required documentation and project assets shall be handed over to PI-RAHI as part of project closure and/or as otherwise specified in the contract.