Desi-Etsy

Niche E-Commerce Platform for Indian Artisans

(A Full-Stack MERN Application)

Project Duration:
07/05/2025– 07/07/2025

Submission Date:
12/07/2025

Submitted By:
Rohit Kumar
22CSE163(22UG010296)
COMPUTER SCIENCE AND ENGINEERING
GIET UNIVERSITY, GUNUPUR,ODISHA

Introduction

Overview of the Project

Desi-Etsy is a full-stack e-commerce platform built specifically to support Indian artisans and their handmade products. The platform functions as a digital marketplace where artisans can list, manage, and sell their crafts directly to customers across the country. Developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack, Desi-Etsy combines technical efficiency with cultural sensitivity to deliver a user-friendly and secure shopping experience.

The application is designed with scalability and modularity in mind, supporting multiple user roles (customer, artisan, and administrator), secure payment gateways, and real-time order management. It aims to digitize the traditional handicrafts industry by removing geographical barriers and providing artisans with tools to enter the e-commerce space seamlessly.

Purpose and Motivation

The primary purpose of Desi-Etsy is to empower rural and urban Indian artisans who often lack access to digital platforms for selling their products. While global e-commerce is expanding rapidly, many local artisans are left behind due to limited exposure, technical skills, and marketing resources.

This project was motivated by the need to create an inclusive, tech-driven ecosystem that brings artisan products to the digital mainstream. By offering artisans the ability to independently manage their storefronts, Desi-Etsy aims to reduce reliance on intermediaries and improve their income potential.

Additionally, the project aligns with broader national goals such as “Digital India” and “Vocal for Local,” promoting indigenous talent and craftsmanship through technology.

Real-World Inspiration

The concept of Desi-Etsy draws direct inspiration from globally successful platforms like **Etsy**, which is known for its focus on handmade, vintage, and unique factory-free goods. Etsy has empowered creators around the world by giving them a place to sell their art and crafts online. Similarly, Desi-Etsy localizes this concept to the Indian context, considering the cultural richness and diversity of Indian handicrafts.

While platforms like Etsy are global, they often come with challenges like international payment gateways, high listing fees, and global competition—factors that may not suit Indian artisans. Desi-Etsy solves these issues by offering a tailored, India-first solution with INR payments, local language support (planned), and features suited to small-scale entrepreneurs.

Empowering Indian Artisans

Desi-Etsy provides Indian artisans with a self-service digital storefront that is both easy to use and powerful. Artisans can register, upload product details, manage stock, respond to orders, and track sales without any dependency on web developers or middlemen.

The platform enables artisans from even the remotest areas to access national buyers and expand their reach far beyond local exhibitions and craft fairs. With integrated Razorpay payments, automated email confirmations, and a responsive design, Desi-Etsy ensures that artisans experience a smooth transition into online commerce.

By equipping artisans with tools for visibility, digital literacy, and financial growth, Desi-Etsy becomes more than just a website—it becomes a catalyst for economic empowerment and cultural preservation.

Objectives & Goals

Project Objectives

The *Desi-Etsy* platform was conceptualized and developed with a strong focus on empowering Indian artisans by offering them a specialized digital marketplace. This initiative combines modern technology with cultural preservation to bridge the gap between traditional craftsmanship and online commerce. The primary aim is to provide artisans, many of whom have limited access to the digital economy, with a reliable platform to sell their handmade goods directly to consumers.

The application is developed using the **MERN stack**—MongoDB, Express.js, React.js, and Node.js. This full-stack approach enables rapid development, seamless integration between frontend and backend, and a scalable architecture to support growth. One of the most significant technical decisions was to implement **Role-Based Access Control (RBAC)** to manage user roles and permissions. Customers, artisans, and administrators all interact with the platform in unique ways, with interfaces tailored to their specific needs.

A central goal is to support artisans in gaining financial independence and visibility. By enabling artisans to register, list products, and manage orders without any technical expertise, the platform fosters inclusivity and economic empowerment. Furthermore, customer features like cart management, checkout, and real-time order tracking ensure a smooth shopping experience. The integration of **Razorpay** ensures secure and convenient payments, while **Nodemailer** handles email communications for OTP verification and order updates.

Project Objectives

Develop a Full-Stack MERN Application

Build a scalable, modular, and performant web application using MongoDB, Express.js, React.js, and Node.js.

Implement Role-Based Access Control (RBAC)

Separate dashboards for customers, artisans, and administrators, each with specific features and access permissions.

Empower Indian Artisans

Provide a platform where artisans can independently manage product listings, inventory, and orders without intermediaries.

Enable Secure Payments

Integrate Razorpay to handle online transactions using UPI, debit/credit cards, and net banking securely.

Real-Time Order and Email Notifications

Use Nodemailer to send email updates for account verification, order confirmation, and shipping status.

Authentication and Authorization

Secure user sessions and protect sensitive routes using JSON Web Tokens (JWT).

Simplified Product Management

Allow artisans to add products with images, descriptions, prices, and categories in a simple and guided format.

Key Goals

Deliver a Responsive and Accessible UI

Use React.js with Tailwind CSS to create mobile-first designs for smooth navigation across devices.

Ensure Real-Time Communication (future scope)

Plan for future integration of real-time customer-artisan chat for direct support and engagement.

Deploy with Modern CI/CD Workflows

Use Vercel for the frontend and Render for the backend to ensure zero-downtime deployments and continuous integration.

Ensure Security and Data Privacy
Encrypt passwords, validate tokens, and implement secure coding practices for all backend operations.

Promote Cultural and Economic Inclusion
Focus on Indian currency, regional product categories, and future multilingual support for diverse audiences.

Design for Extensibility and Scalability
Architect the application in a modular way to support future features like reviews, analytics dashboards, and delivery tracking.

Modules & Key Features

The Desi-Etsy platform is organized into various modules to ensure clarity, separation of concerns, and ease of maintenance. Each module addresses a specific functionality that contributes to a seamless and user-friendly e-commerce experience. Below are the main modules and their key features:

1. User Module

- Supports registration and login for three roles: Customer, Artisan, and Admin.
- Uses JWT-based authentication to secure user sessions.
- Implements Role-Based Access Control (RBAC) to show different dashboards and functionalities based on user type.
- Artisans require admin approval before listing products.

2. Product Module

- Artisans can add, edit, and delete products from their personal dashboard.
- Each product includes title, image, price, description, and category.
- Customers can browse products by category, artisan name, or price.
- Real-time updates allow products to appear instantly after upload.

3. Cart Module

- Allows users to add products to cart, modify quantity, or remove items.
- Cart data is persistent across sessions using local storage or context.
- Dynamically updates total cost and shows a summary during checkout.

- Designed to be responsive and fast for a smooth shopping experience.

4. Order Module

- Handles checkout process for both Razorpay (online) and Cash on Delivery (COD).
- Collects delivery address and payment details.
- Generates a new order and updates product stock if required.
- Enables real-time order tracking through the dashboard.
- Order status is managed by artisans and visible to customers.

5. Admin Panel

- Admins can view and approve artisan registrations.
- Manage all products on the platform, including editing or removing inappropriate listings.
- Monitor customer orders, update statuses, or handle refund requests.
- View and manage all registered users (customers and artisans).
- Provides a centralized control panel to ensure platform integrity.

6. Email Notifications

- Sends OTP verification emails during registration/login.
- Order confirmation emails are sent upon successful payment or order placement.
- Order status updates (processed, shipped, delivered) are sent automatically.
- Implemented using Nodemailer with Gmail SMTP for reliable delivery.

7. Payment Integration

- Integrated with Razorpay to handle secure online transactions.
- Supports multiple payment methods: UPI, Credit/Debit Cards, Net Banking.
- Automatically updates payment status and order confirmation upon success/failure.
- Displays real-time confirmation on both the UI and via email.

8. Responsive UI

- Built using React.js and Tailwind CSS for mobile-first design.
- Fully responsive across mobile, tablet, and desktop devices.

- Smooth navigation with minimal clicks and clear CTA buttons.
- Optimized for accessibility and ease of use, even for non-technical users.

Technology Stack

The Desi-Etsy platform uses a modern, efficient, and scalable tech stack to ensure performance, security, and ease of development. The technology choices align with best practices in full-stack web development and support modular, extensible architecture.

Frontend Technologies

- **React.js:**
  A powerful JavaScript library used to build a dynamic and responsive user interface with component-based architecture. It ensures fast rendering and seamless user experience.
- **Tailwind CSS:**
  A utility-first CSS framework used for styling the frontend. It helps in creating modern, responsive designs with minimal custom CSS.
- **React Router:**
  Used for managing navigation across pages without reloading the app. It enables a smooth Single Page Application (SPA) experience.

Backend Technologies

- **Node.js:**
  A runtime environment that allows server-side execution of JavaScript. It handles concurrent requests efficiently and forms the backbone of the backend.
- **Express.js:**
  A lightweight web application framework built on Node.js. It simplifies routing, middleware configuration, and handling HTTP requests and responses.

Database

- **MongoDB:**
  A NoSQL database that stores data in JSON-like documents. It offers flexibility in schema design and is ideal for handling product listings, users, and order data.
- **Mongoose ORM:**
  An Object Data Modeling (ODM) library for MongoDB. It provides schema validation, easier querying, and structured models to interact with the database.

Authentication & Security

- **JWT (JSON Web Tokens):**
  Used for secure authentication and session management. Ensures that only verified users can access protected routes and perform specific actions.
- **Bcrypt.js:**
  A password-hashing library used to securely store and compare passwords in the database.

Payment Gateway

- **Razorpay:**
  Integrated to handle online transactions securely. Supports UPI, debit/credit cards, and net banking. Provides real-time payment confirmation and error handling.

Email Service

- **Nodemailer:**
  A Node.js module used for sending automated emails like OTPs, order confirmations, and shipping updates using Gmail's SMTP service.

Deployment Platforms

- **Vercel (Frontend Deployment):**
  Used for hosting the React.js frontend. Supports automatic builds and continuous deployment from GitHub.
- **Render (Backend Deployment):**
  Hosts the Node.js and Express backend. Supports environment variables, automatic server restart, and scalable cloud infrastructure.

Database Schema Overview

The Desi-Etsy platform uses MongoDB as its primary database, managed through Mongoose ORM. The database is designed with flexibility and scalability in mind, supporting multiple user roles, dynamic product listings, and real-time order tracking. Below is a breakdown of the key collections (tables) and their attributes:

1. User Collection

Stores details of all users including customers, artisans, and administrators.

- name: Full name of the user.

- email: Unique identifier used for login and communication.
- password: Encrypted using bcrypt for security.
- role: Defines user type (admin, artisan, or customer).
- isApproved: Boolean flag to approve artisan accounts by the admin.
- createdAt: Timestamp of account creation.

2. Product Collection

Stores data for all products listed by artisans.

- title: Name of the product.
- image: URL of the product image.
- price: Price in INR.
- description: Detailed product information.
- category: Product category (e.g., clothing, jewelry, home decor).
- artisanId: Reference to the artisan who listed the product.
- createdAt: Date when the product was listed.

3. Order Collection

Contains information about all customer orders.

- user: Reference to the customer placing the order.
- items: Array of objects containing:
  - productId: Reference to the product.
  - artisanId: Reference to the seller (artisan).
  - quantity: Quantity ordered.
- total: Total amount paid for the order.
- address: Delivery address details (street, city, pincode, etc.).
- status: Current status of the order (e.g., processing, shipped, delivered).
- paymentStatus: Payment status (e.g., paid, pending, failed).
- paymentMethod: Selected payment method (Razorpay, COD).
- createdAt: Timestamp of order placement.

4. OTP / Verification Token (Optional)

Used to store email OTPs or password reset tokens.

- email: Email address of the user.
- otp: One-time password sent for verification.
- expiresAt: Expiry time for the OTP/token.

The schema design supports fast queries, secure data handling, and smooth integration between the frontend and backend. Mongoose models are used to define schema rules, relationships, and validation mechanisms across all collections.

Future Enhancements

While the current version of Desi-Etsy delivers a fully functional and user-friendly e-commerce platform for Indian artisans, there is significant scope for future improvements to enhance user engagement, platform scalability, and market relevance. These enhancements are designed to meet evolving user needs and align with the long-term vision of the platform.

Planned Future Enhancements:

- **Product Reviews and Ratings:**
  Introducing a rating and review system will allow customers to share their feedback and help others make informed purchase decisions. This will also boost artisan credibility and promote high-quality products.
- **Customer-Artisan Chat:**
  A real-time chat feature will allow buyers to interact directly with artisans, ask product-related questions, and request customization, improving customer satisfaction and trust.
- **Admin Revenue Dashboard:**
  Adding a powerful dashboard for administrators to monitor revenue, user engagement, and product trends through analytics and visual reports.
- **Real-Time Delivery Tracking:**
  Integration with delivery service APIs or use of WebSockets to show live tracking updates of dispatched products, enhancing transparency for customers.

- **Multi-Language Support:**
  To improve accessibility for non-English speaking users, the platform will be localized into Indian languages like Hindi, Tamil, Bengali, and more.
- **Mobile App Version:**
  A dedicated mobile app version for Android/iOS users to increase reach and improve performance on mobile devices.

Conclusion

Desi-Etsy stands as a complete and practical solution aimed at empowering Indian artisans through digital commerce. Built using the MERN stack, the project successfully integrates crucial components such as user authentication, product and order management, role-based access control, secure payments via Razorpay, and automated email notifications.

The platform not only fulfills technical objectives but also addresses social and economic challenges by offering local artisans a cost-free digital space to grow their businesses. With intuitive UI/UX design, robust backend architecture, and secure transaction systems, Desi-Etsy ensures both artisan empowerment and customer satisfaction.

Moreover, the project demonstrates strong proficiency in full-stack web development, cloud deployment, API integration, and e-commerce design principles. It highlights how modern technology can be leveraged to build platforms that deliver real-world impact while maintaining scalability and extensibility.

As Desi-Etsy continues to evolve, upcoming features like real-time chat, reviews, analytics, and multi-language support will further enrich the platform and strengthen its mission of promoting local talent in the digital marketplace.

Project Repository & References

Project GitHub Repository

Desi-Etsy: Niche E-Commerce Platform for Indian Artisans
🔗 https://github.com/Rohitsharma97714/Niche-E-commerce-Platform-for-Handmade-Products

Key References & Documentation

- MERN Stack Official Documentation
- MongoDB Best Practices – https://www.mongodb.com/best-practices
- Express.js Guide – https://expressjs.com
- React.js Docs – https://reactjs.org/docs
- Tailwind CSS – https://tailwindcss.com/docs
- Razorpay API Documentation – https://razorpay.com/docs
- Nodemailer Setup – https://nodemailer.com/about/