Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: CP	**Aim: System Design and Architecture**	
	Date: 25-09-2025	**Enrolment No: 92310133015**

Jetpur Silk Roots E-commerce Platform

Table of Contents

1. Introduction
2. Modular Design
3. Technology Stack
4. Scalability Plan
5. Conclusion

Introduction

The Jetpur Silk Roots platform is a modern, scalable e-commerce solution designed to showcase and sell premium silk sarees and textiles from Jetpur. This system architecture document outlines the comprehensive design approach, technology choices, and scalability considerations for building a robust, maintainable, and extensible web application that serves both domestic and international customers.

The platform addresses the unique requirements of the silk textile industry, including high-quality product visualization, manufacturer profiles, export guidelines, and multi-channel customer engagement. The architecture is designed to handle varying traffic loads, support multiple user types (customers, manufacturers, exporters), and provide seamless user experiences across different devices and platforms.

Modular Design

Architecture Overview

The Jetpur Silk Roots platform follows a modular architecture pattern that separates concerns into distinct, independent components. This design enables better maintainability, reusability, and extensibility while ensuring system reliability and performance.

Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: CP	**Aim: System Design and Architecture**	
	Date: 25-09-2025	**Enrolment No: 92310133015**

Core Modules

1. Frontend Presentation Layer

- Purpose: User interface and user experience management
- Components: React-based Single Page Application (SPA), responsive UI components (shadcn/ui), client-side routing with React Router, and state management with React Query
- Responsibilities: Rendering user interfaces, handling user interactions, managing client-side state, optimizing user experience with progressive loading

2. API Gateway Layer

- Purpose: Centralized entry point for all client requests
- Components: Supabase client integration, authentication and authorization, request routing and load balancing
- Responsibilities: Managing API endpoints, handling authentication tokens, rate limiting and security, request/response transformation

3. Business Logic Layer

- Purpose: Core application logic and business rules
- Components: React components with custom hooks, Zod for form validation, business rules
- Responsibilities: Product catalog management, user authentication flows, order processing logic, export guidelines management

4. Data Access Layer

- Purpose: Database operations and data persistence
- Components: Supabase PostgreSQL, type-safe client, data models and schemas
- Responsibilities: Data storage and retrieval, database optimization, data integrity, backup and recovery

5. External Services Layer

- Purpose: Integration with third-party services

- Components: Email services, analytics, (future: payment gateway)
- Responsibilities: External API management, service monitoring, error handling and fallbacks

Module Interactions and Data Flow

- User Request Flow: Frontend → API Gateway → Business Logic → Data Access → Database
- Authentication Flow: Frontend → API Gateway → Supabase Auth → Session Management
- Data Flow: Database → Data Access Layer → Business Logic → API Gateway → Frontend
- External Service Flow: Business Logic → External Services → Third-party APIs

Benefits of Modularity

- **Maintainability**: Independent development, easier debugging, version control
- **Reusability**: Shared UI components, business logic, schemas
- **Extensibility**: Easy to add new features, integrate new technologies, and scale individual modules

---

Technology Stack

Frontend

- **React 18.3.1**: Robust component-based framework with strong community support and performance features.
- **TypeScript 5.8.3**: Provides static type checking for reliability and maintainability.
- **Vite 5.4.19**: Fast development bundler with hot module replacement and efficient builds.
- **shadcn/ui + Radix UI**: Accessible, customizable component library following WCAG compliance.
- **Tailwind CSS 3.4.17**: Utility-first CSS for consistent, responsive UI design.

Backend & Database

- **Supabase**: Complete backend-as-a-service including PostgreSQL, authentication, real-time subscriptions, and APIs.
- **PostgreSQL**: Relational database with ACID compliance, advanced indexing, JSON support, and full-text search.

State Management & Forms

- **TanStack Query (React Query)**: Handles server state caching, synchronization, and background updates.
- **React Hook Form with Zod**: Efficient, type-safe form validation.

Tooling

- **ESLint**: Ensures consistent, high-quality code.
- **PostCSS with Autoprefixer**: CSS compatibility across browsers.

Deployment & Hosting

- **Vercel**: Optimized for React, supports CDN distribution, automatic deployments, and serverless functions.

Scalability Plan

Current Capacity

- 1,000+ concurrent users
- 100,000+ monthly page views
- 50,000+ product records
- 10GB+ images and assets

Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: CP	**Aim: System Design and Architecture**	
	Date: 25-09-2025	**Enrolment No: 92310133015**

Horizontal Scaling

- **Frontend**: CDN distribution, code splitting, lazy loading, caching strategies
- **Database**: Read replicas, indexing, query optimization, connection pooling
- **API Layer**: Rate limiting, Redis caching, load balancing, microservices (future)

Vertical Scaling

- CPU, memory, and storage upgrades
- PostgreSQL tuning and optimization

Bottleneck Analysis

- **Database**: Indexing, caching, query optimization
- **Network Latency**: CDN, image optimization (WebP), compressed API responses
- **Frontend Performance**: Bundle splitting, lazy loading, virtual scrolling

Cost Optimization

- Serverless functions for variable workloads
- Auto-scaling resources
- Optimized use of Supabase tiers, CDN, and email services

Reliability & Monitoring

- Error handling with retry logic and error boundaries
- Real User Monitoring (RUM) + APM tools
- Automated backups and disaster recovery planning

Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: CP	**Aim: System Design and Architecture**	
	Date: 25-09-2025	**Enrolment No: 92310133015**

Future Scaling

- Microservices for product, user, order, and notification services
- Redis caching and edge computing
- Multi-region deployment with localization and compliance support

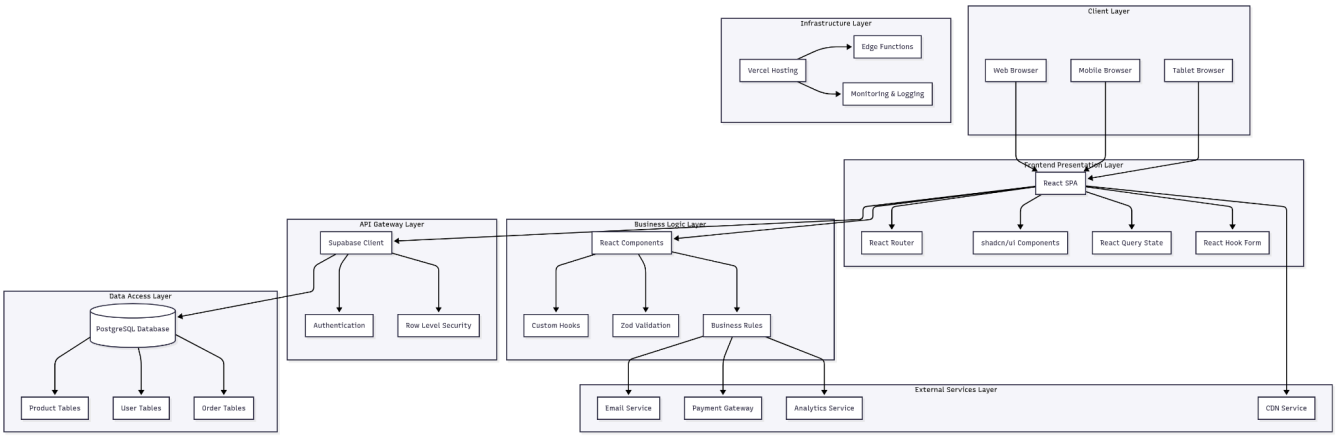
---

Conclusion

The Jetpur Silk Roots platform architecture offers a robust, scalable foundation tailored for the textile industry. Its modular design ensures maintainability and extensibility, while the chosen technology stack provides performance and reliability. The scalability plan prepares the platform for growth, international expansion, and evolving user demands.

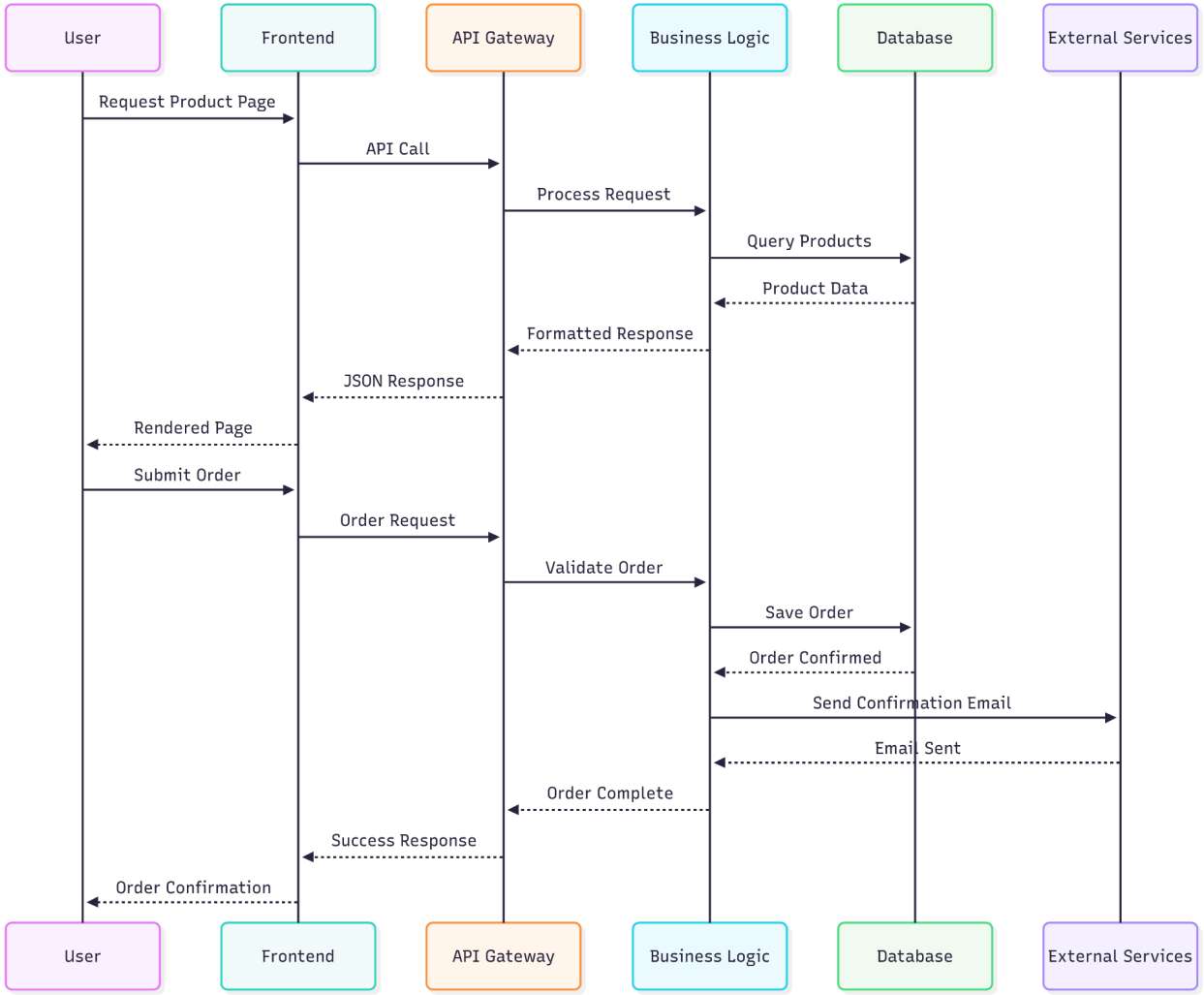
Diagrams

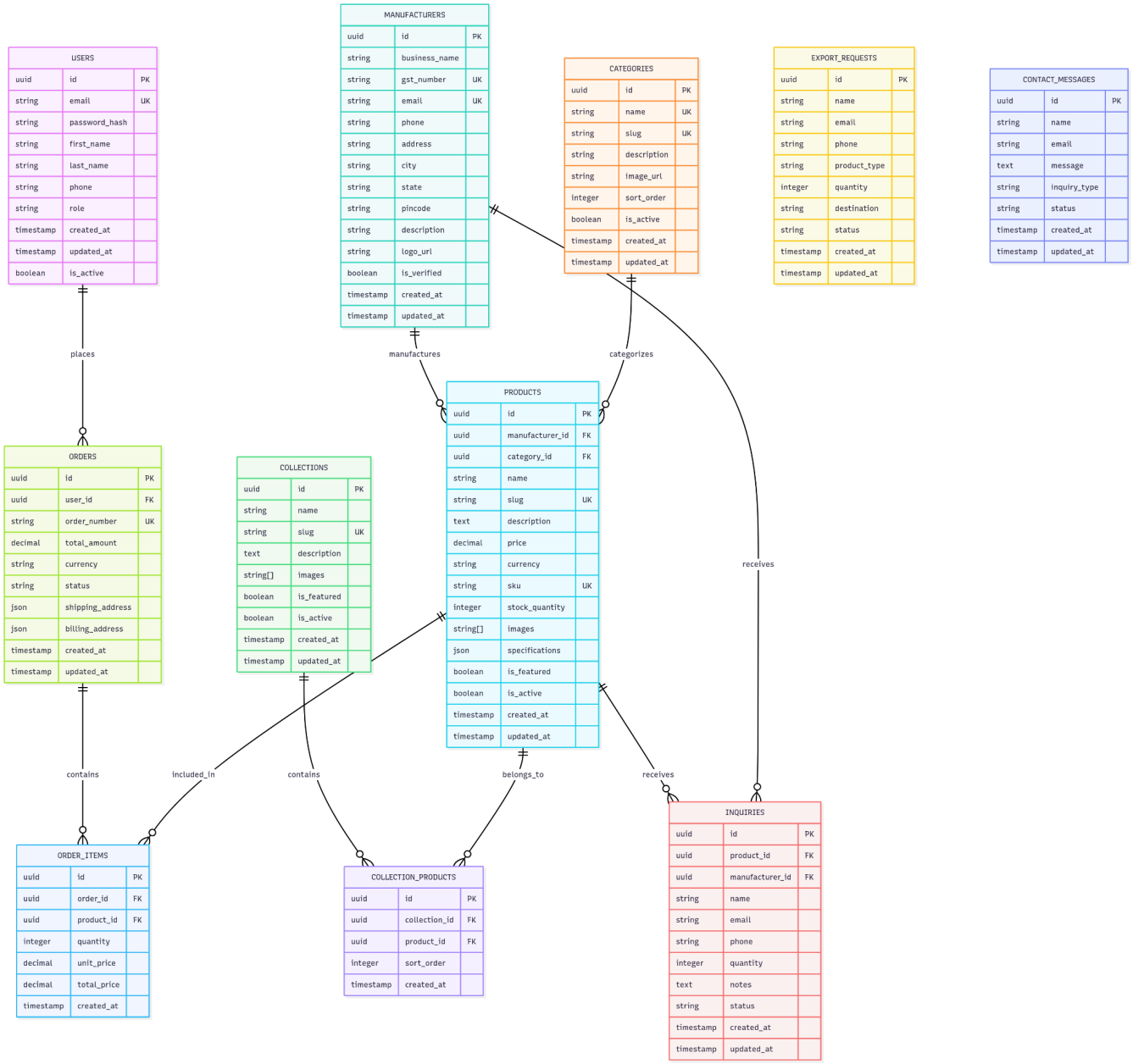
1.System Architecture



Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: CP	**Aim: System Design and Architecture**	
	Date: 25-09-2025	**Enrolment No: 92310133015**

2. Sequence Diagram



3. ERD Diagram

Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: CP	**Aim: System Design and Architecture**	
	Date: 25-09-2025	**Enrolment No: 92310133015**