🎨

🌿 VanVyaapaar – Tribal Handicraft E-Marketplace

🧭 Project Vision & Objective

🎯 **Vision**
To create a digital platform that connects tribal artisans directly to customers, eliminating middlemen and ensuring fair trade for traditional handicrafts, tribal art, and eco-friendly products.

💡 Project Objectives

- Promote **rural and tribal entrepreneurship**
- Enable artisans to **digitally showcase and sell** their products
- Ensure **secure transactions**, transparency, and easy order tracking
- Encourage **cultural preservation** through digital markets

---

🧩 Problem Statement

Tribal artisans face major challenges such as:

- Lack of awareness about e-commerce and online marketing
- Dependence on middlemen resulting in **low profit margins**
- Limited access to **digital payment systems** and **logistics support**
- Absence of platforms that focus on **tribal crafts** specifically

---

💡 Proposed Solution

The **VanVyaapaar** platform serves as a **centralized e-commerce system** where:

- **Sellers (tribal artisans)** can list and manage products easily
- **Buyers** can explore, purchase, and review authentic tribal products
- **Admins** ensure smooth functioning through monitoring and management

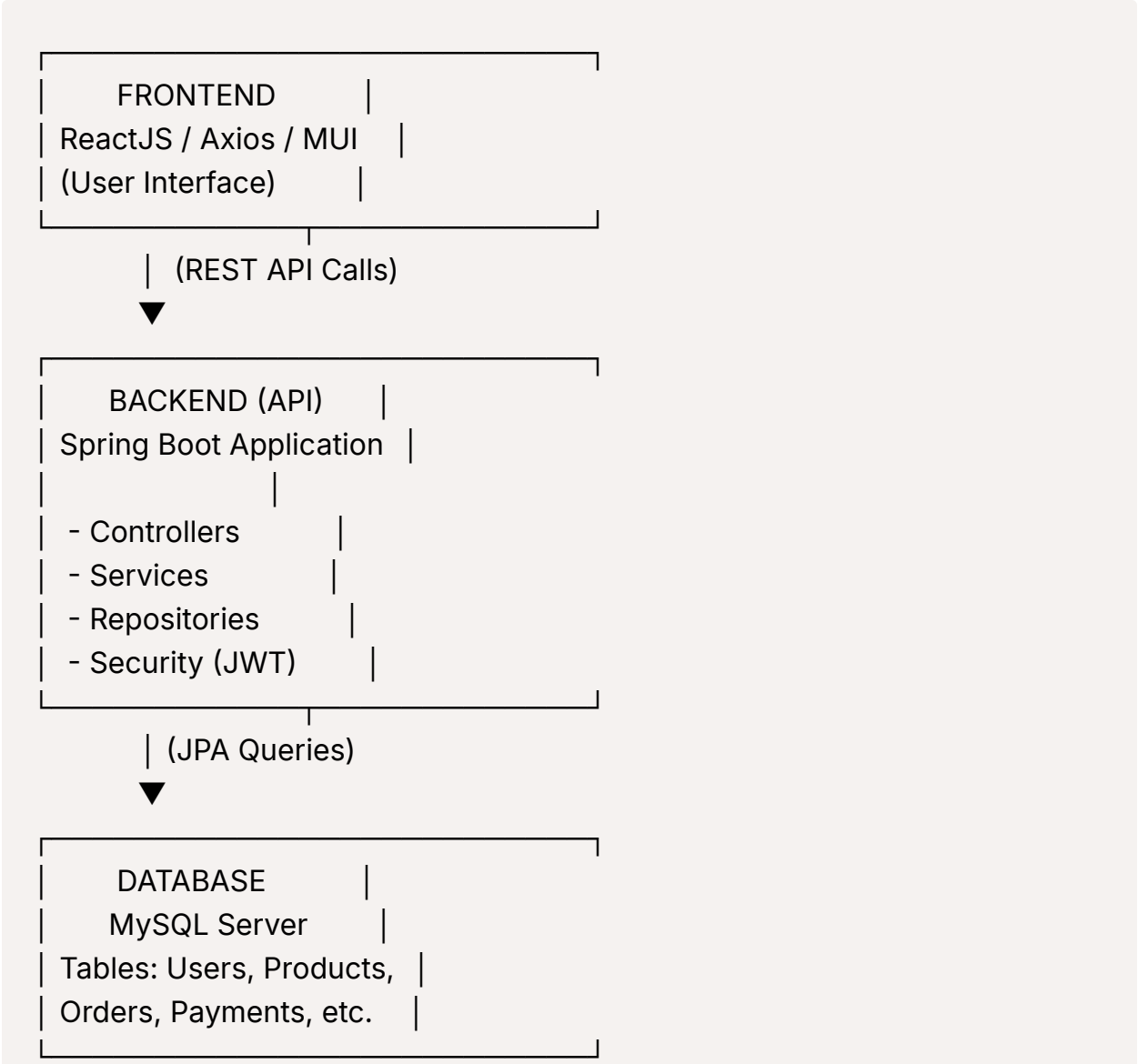
🔑 Key Features

- Role-based authentication (Admin, Seller, Buyer)
- Product management (CRUD)
- Order and cart management
- Review and rating system
- Payment and delivery tracking (future enhancement)
- RESTful API-based backend using Spring Boot

⚙️ Technology Stack

Layer	Technology Used	Description
Frontend	ReactJS, Axios, React Router, Tailwind CSS / Material UI	Dynamic, responsive UI
Backend	Spring Boot (Spring Web, Spring Data JPA, Spring Security, Validation)	REST API development
Database	MySQL	Data storage for users, products, and orders
Authentication	Spring Security + JWT	Role-based secured access
ORM	Hibernate / JPA	Object-Relational Mapping
Dev Tools	Maven, Lombok, Postman, Swagger UI	API testing and documentation
Version Control	Git & GitHub	Collaborative code management
Deployment	Docker / AWS EC2 / Render	Cloud-based hosting

🧠 System Architecture Overview



👤 User Roles and Access

Role	Description	Example Actions
Buyer	Customer who browses and buys items	Search, add to cart, place order
Seller (Artisan)	Registered tribal seller	Add/edit/delete products

Admin	System manager	Manage users, products, and reports

🛒 Detailed Workflow

🔷 Landing Page

- Showcases trending tribal products, art, and regional collections
- Sections like: *Handicrafts*, *Jewelry*, *Paintings*, *Wood Art*
- Navigation buttons:
  - "Login / Sign Up" → Buyer
  - "Join as Seller" → Seller registration
  - "Admin Login" → Admin panel

👤 Authentication & Authorization

Login/Signup Flow (JWT-Based):

1. User signs up with their role-specific details
2. Backend validates and stores data in MySQL
3. Upon login → Spring Security verifies credentials
4. If valid → JWT Token is issued
5. Token used for all further requests (Authorization header)

Example Request:

```
{
  "email": "user@example.com",
  "password": "secure123"
}
```

Backend Response:

```
{
  "token": "eyJhbGciOiJIUzI1NiIs...",
  "role": "SELLER",
  "name": "Prasad Ingole"
}
```


👨‍💼 Buyer Module

▼ Features

- Browse & search products
- Filter by category, price, or seller
- Add to cart and checkout
- View order history
- Post reviews

▼ Workflow

1. Buyer logs in → receives JWT token
2. Fetch product list via `/api/products`
3. Add items to cart via `/api/cart/add/{productId}`
4. Place order → `/api/order/create`
5. Order stored in DB and confirmation shown

👩‍🎨 Seller Module (Tribal Artisan)

▼ Features

- Seller Dashboard: Order & product summary
- Add / Edit / Delete products
- View buyer reviews
- Track order status (Pending, Delivered)

▼ Workflow

1. Seller logs in → `/api/seller/login`
2. Dashboard shows stats (total sales, revenue)
3. Seller adds product via `/api/seller/add-product`
4. Product visible to all buyers after approval

👨‍💻 Admin Module

▼ Features

- Approve / Block sellers
- Add or remove products
- Monitor transactions
- View user analytics
- Manage discount campaigns

▼ Workflow

1. Admin logs in with predefined credentials
2. Accesses `/admin/dashboard`
3. Can perform CRUD on users, sellers, and products
4. Generates reports via `/api/admin/reports`

🗃️ Database Design

Simplified Schema

Table	Columns
users	id, name, email, password, role, address, phone
sellers	id, name, shop_name, tribe_name, bank_details
products	id, name, category, price, description, image, seller_id
orders	id, buyer_id, total_price, order_date, status
order_items	id, order_id, product_id, quantity
reviews	id, buyer_id, product_id, rating, comment

🔐 Security Implementation

🔒 Security Measures

- Uses **Spring Security** for authentication
- **JWT (JSON Web Token)** for session-less security
- Passwords hashed using **BCrypt**
- **Role-based access control:**
  - `/admin/**` → ADMIN only
  - `/seller/**` → SELLER only
  - `/buyer/**` → BUYER only

🧪 Testing & Documentation

- **Postman**: For endpoint testing
- **Swagger UI**: Integrated to auto-generate API docs
- **JUnit + Mockito**: Unit testing for services and controllers

> Swagger UI accessible via: http://localhost:8080/swagger-ui.html

🚀 Future Enhancements

Feature	Description
Payment Integration	Razorpay or Stripe for secure payments
Delivery Integration	Link with India Post or third-party APIs
AI Recommendation Engine	Suggest products based on user behavior
Language Support	English, Hindi, Marathi, etc.
Seller Analytics Dashboard	Monthly income, top-selling items

Cloud Storage	Product images on Cloudinary or AWS S3
Mobile App (React Native)	Expand reach to smartphone users

📥 Development Workflow (GitHub Collaboration)

Initial Setup (One-time)

```
git clone https://github.com/your-repo/vanpaayaar-backend
cd vanpaayaar-backend
```

Daily Workflow (Both Collaborators)

```
# Check for updates
git pull origin main

# Make your changes
git add .
git commit -m "Added Seller Controller"

# Push changes
git push origin main
```

Best Practice

- Create feature branches: `feature/seller-module`, `feature/order-service`
- Merge via Pull Requests for review
- Always pull before starting new work
- Write clear commit messages

🌍 Impact and Sustainability

🌱 Social Impact

- Promotes **rural employment** and **financial independence**
- Encourages **eco-friendly product promotion**
- Contributes to **Digital India** initiative
- Connects artisans directly to a **global market**
- Preserves **traditional crafts** and **cultural heritage**

🏁 Conclusion

VanVyaapaar is more than an e-commerce site — it's a **bridge between tradition and technology**, empowering tribal communities through digital transformation.

With a robust **Spring Boot + ReactJS architecture**, secure **JWT-based system**, and scope for **AI & cloud integration**, this project stands strong for academic, SIH, or startup-level deployment.

> *Empowering artisans, one click at a time.* 🎨✨