Aryan Saxena & Sumit Kant

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING (DevOps)

Department of Computer Science and Engineering
Chandigarh University, Gharuan, Mohali (Punjab), India

BONAFIDE CERTIFICATE

This is to certify that the project report entitled “APNADUKAAN – E-COMMERCE PLATFORM FOR LOCAL SELLERS” is the bonafide work of Mr. Aryan Saxena (UID: 23BDO10015) and Mr. Sumit Kant(23BDO10006), who carried out the project work under my/our supervision in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering (DevOps) at Chandigarh University, Gharuan.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

Hardware / Software Requirements Hardware

Requirements:

- **Processor: Intel Core i5 or higher**
- **RAM: 8 GB or above**
- **Hard Disk: Minimum 256 GB SSD**
- **Internet Connectivity: Required Software**

Requirements:

- **Operating System: Windows / macOS / Linux**
- **Frontend: React.js, Tailwind CSS**
- **Backend: Node.js, Express.js**
- **Database: MongoDB Atlas**
- **AI Integration: Google Gemini API via @google/generative-ai**

- **Tools Used:**
  - **Visual Studio Code**
  - **Postman**
  - **Git & GitHub**

1. Overview

ApnaDukaan is a **community-driven e-commerce platform** created to uplift local vendors and small-scale shopkeepers by bridging the gap between **traditional markets** and **digital commerce**. The main idea is to provide an easy, affordable, and user-friendly solution that allows sellers to showcase their products online without depending on expensive third-party platforms.

While large e-commerce players often impose strict policies and commissions, **ApnaDukaan focuses on inclusivity and simplicity.** It allows even a non-technical user to create an online store, manage inventory, view customer orders, and track sales—all from a single dashboard.

The **Customer Portal** complements this by providing a seamless shopping experience. Customers can explore products, add them to the cart, and place secure orders using digital payment gateways.

The system is built using a **modern and modular architecture**:

- **Frontend:** React.js for a dynamic and interactive user experience.
- **Backend:** Spring Boot for handling APIs, authentication, and order processing.
- **Database:** MySQL for structured and reliable data storage.

Through **ApnaDukaan**, we aim to empower small businesses to thrive in the digital age, ensuring that the benefits of online commerce reach every level of society.

---

2. Problem Statement

Despite therapiddigitalization of commerce, **local sellers remain digitally marginalized.** Most e-commerce ecosystems favor large-scale vendors due to their logistics, branding, and technical capabilities. Small shopkeepers, however, face numerous challenges:

- Lack of digital skills
- High commission fees on popular e-commerce sites
- Complicated onboarding processes

- Limited access to online customers

This digital divide prevents thousands of micro and small-scale vendors from competing fairly. Customers, too, are limited to big brands and online marketplaces, missing out on **trustworthy local businesses** that may offer better products or services.

ApnaDukaan addresses this gap by offering a **dedicated, easy-to-use, and cost-effective** platform designed especially for local sellers. By simplifying the technical and financial barriers, it promotes **digital inclusion and local economic growth.**

---

3. Objectives

1. **Seller Empowerment:**
    - o Allow sellers to register easily, create their online store, and manage products, inventory, and orders.
    - o Simplify product uploads with images, prices, and stock information.
2. **Customer Convenience:**
    - o Provide customers with a clean and intuitive interface for browsing local products.
    - o Enable secure checkout with multiple payment options.
3. **Reliable Backend System:**
    - o Use a scalable and secure backend architecture to handle large datasets and concurrent transactions.
4. **Transparency & Trust:**
    - o Maintain transparency between sellers and customers through clear communication and order tracking.
5. **Promotion of Local Commerce:**
    - o Support local business growth by bringing them into the digital marketplace.

---

4. Requirements Specification

4.1 Functional Requirements

- **Seller Features:**
  - o Register, log in, and authenticate via secure JWT-based system.
  - o Manage product listings (add/edit/delete).
  - o Track inventory and update stock in real time.
  - o View and fulfill customer orders.
- **Customer Features:**
  - o Browse products based on category or seller.
  - o Add products to cart and proceed to checkout.
  - o Make payments securely via integrated payment gateways.
  - o View order history and track status.
- **Admin Features:**
  - o Manage user accounts and sellers.
  - o Review transactions and resolve disputes.
  - o Monitor platform performance and ensure compliance.

4.2 Non-Functional Requirements

- **Security:**
  Data encryption, authentication, and secure APIs using Spring Security.
- **Scalability:**
  Handle increasing numbers of sellers and customers without performance loss.
- **Availability:**
  Ensure system uptime with minimal maintenance downtime.
- **Performance:**
  Fast response times and optimized API performance.
- **Usability:**
  Mobile-responsive, intuitive design ensuring accessibility across all devices.

5.SystemArchitecture and Design

5.1 Architecture Overview

The system follows a **three-tier architecture**:

1. **Presentation Layer (React Frontend):**

   Manages UI and user interaction.

   - Seller Dashboard
   - Customer Storefront
   - Admin Panel

2. **Application Layer (Spring Boot Backend):**

   Contains business logic, authentication, and request handling.

   - RESTful APIs for communication
   - Order management and payment validation

3. **Data Layer (MySQL Database):**

   Stores all persistent data like users, sellers, products, orders, and transactions.

5.2 Technology Stack

- **Frontend:** React.js, Axios, React Router, Tailwind CSS
- **Backend:** Spring Boot, Spring Security (JWT), Spring Data JPA
- **Database:** MySQL
- **Tools & Services:** Postman (API testing), GitHub (Version Control), IntelliJ IDEA (Backend IDE), VS Code (Frontend IDE)

---

6. System Workflow

1. **User Registration:**

   Both customers andsellers create accounts via the frontend.

2. **Seller Product Management:**

   Sellers upload product details with price, description, and stock count.

3. **Customer Browsing:**

   Customers browse available items by category, seller, or search keyword.

4. **Cart & Checkout:**

   Selected items are added to the cart, followed by checkout using secure payment options.

5. **Order Processing:**

   The backend processes the order, updates inventory, and notifies both seller and customer.

6. **Admin Monitoring:**

   The admin reviews transactions and ensures smooth platform operation.

---

7. Scalability Plan

Database Scaling (MySQL)

- Implement **indexes** on commonly queried fields (product_id, seller_id, order_id).
- Use **replication and sharding** for large-scale data management.
- Backup and recovery mechanisms for data integrity.

Application Scaling (Spring Boot)

- Stateless design for horizontal scaling.
- Load balancing using Nginx or HAProxy.
- Caching frequently accessed data with Redis/Spring Cache.

Frontend Scaling (React)

- Deploy on **CDN-backed servers (Vercel, Netlify)** for global access.
- Implement lazy loading and code splitting to enhance performance.

Workflow Optimization

- Enable server-side pagination for product catalogs.
- Use asynchronous APIs for improved checkout performance.
- Employ transactional database operations for order consistency.

Deployment Strategy

- Initial deployment: Monolithic system.

- Future scalability: Migrate to **microservices** for sellers, orders, and payments.
- Cloud Hosting (AWS or Azure) with auto-scaling infrastructure.

---

8. Testing and Evaluation

Testing plays a critical role in ensuring that the ApnaDukaan platform performs reliably, securely, and efficiently under all conditions. A structured testing methodology was adopted to verify that each module works as expected and integrates seamlessly with others. The goal of testing was to detect bugs early, improve user experience, and validate that system requirements were fully met.

8.1 Unit Testing

Unit testing was carried out using JUnit for backend components and React Testing Library for frontend validation.

Each function or class, such as product management, user authentication, and order creation, was tested independently to ensure correctness.

Mockito was used to mock dependencies in the Spring Boot environment, which helped test the business logic in isolation.

Example Scenarios:

- Verifying login with valid and invalid credentials.
- Checking whether cart calculations (total amount, discounts) are correct.
- Ensuring proper validation for product input fields such as price, name, and stock count.
- Confirming that APIs return expected responses and error codes.

The successful execution of unit tests guaranteed that all building blocks of the application worked independently before integrating them.

8.2 Integration Testing

After unit testing, integration testing was conducted to validate the communication between different layers — React frontend, Spring Boot backend, and MySQL database.
This ensured that APIs were functioning correctly, data flow was seamless, and the frontend was accurately displaying dynamic data received from the backend.
Key Focus Areas:

- Testing RESTful API endpoints for product and order management.
- Verifying database transactions for order placement and inventory updates.
- Checking cross-origin requests between frontend (React) and backend (Spring Boot).
- Ensuring token-based authentication (JWT) was correctly implemented and validated.

This phase confirmed the stability of the complete system and verified that data consistency and integrity were maintained throughout all user operations.

8.3 User Acceptance Testing (UAT)

To validate usability and functionality from the end-user's perspective, User Acceptance Testing was carried out with feedback from a few local sellers and sample customers.
Participants were asked to perform real-world actions such as registering, uploading products, browsing, and placing orders.
Feedback Collected:

- Sellers appreciated the simplicity of product management tools.
- Customers found the UI intuitive and the checkout process smooth.
- Minor suggestions included clearer order status messages and improved product search filters.

These insights helped refine the interface and improve the platform's user experience before deployment.

8.4 Performance Testing

Performance testing aimed to evaluate how the system behaved under different loads and stress levels. Tools such as Postman and Apache JMeter were used to simulate concurrent user access.

Testing Metrics Included:

- Response Time: Ensuring API responses remained below 300ms under moderate load.
- Throughput: Checking how many simultaneous transactions could be processed without degradation.
- Scalability: Verifying that multiple sellers and customers could interact with the system simultaneously.
- Database Optimization: Ensuring query efficiency with proper indexing and caching.

The performance tests confirmed that the application could handle expected user loads efficiently, making it reliable for real-world deployment.

8.5 Security and Validation Testing

Security testing was also conducted to verify safe authentication, protection from SQL injection, and secure data transfer. Spring Security with JWT was validated for access control, ensuring that only authorized users could access sensitive endpoints.

Overall, these comprehensive testing efforts ensured that ApnaDukaan met both functional and non-functional requirements effectively.

---

9. Future Enhancements

Although ApnaDukaan is fully functional and meets its current objectives, there is strong potential to enhance its features and scalability to serve a wider audience. Future updates and technological advancements can make it more efficient, user-friendly, and market-competitive.

9.1 AI-Based Product Recommendations

Machine Learning algorithms can be introduced to analyze customer behavior, purchase patterns, and browsing history. This data can be used to provide personalized product suggestions, improving customer engagement and sales conversion rates.

9.2 Mobile Application Development

Since a significant portion of online shopping is done via smartphones, a mobile version of ApnaDukaan for Android and iOS will make it more accessible.

Using React Native or Flutter, the same codebase can be used to deliver cross-platform compatibility while maintaining performance and design consistency.

9.3 Real-Time Communication

Integrating a real-time chat system between customers and sellers will enable quick resolution of queries regarding product details, stock availability, and delivery times.

This feature can be built using WebSocket or Firebase Realtime Database for instant communication.

9.4 Delivery Tracking Integration

By connecting with third-party logistics APIs (like Delhivery, Shiprocket, or India Post), real-time order tracking can be made available.

Customers will be able to see the current status of their orders—packed, shipped, or delivered—enhancing transparency and trust.

9.5 Loyalty and Discount Programs

Implementing a points-based loyalty system can encourage repeat purchases.

Customers could earn rewards on each order, which can later be redeemed for discounts.

Sellers can also promote seasonal discounts or referral programs to attract more buyers.

9.6 Admin Analytics Dashboard

A data-driven admin dashboard will be introduced to visualize sales performance, active users, revenue, and seller engagement.

Using libraries like Chart.js or Recharts, graphical analytics will make monitoring easier and support better decision-making.

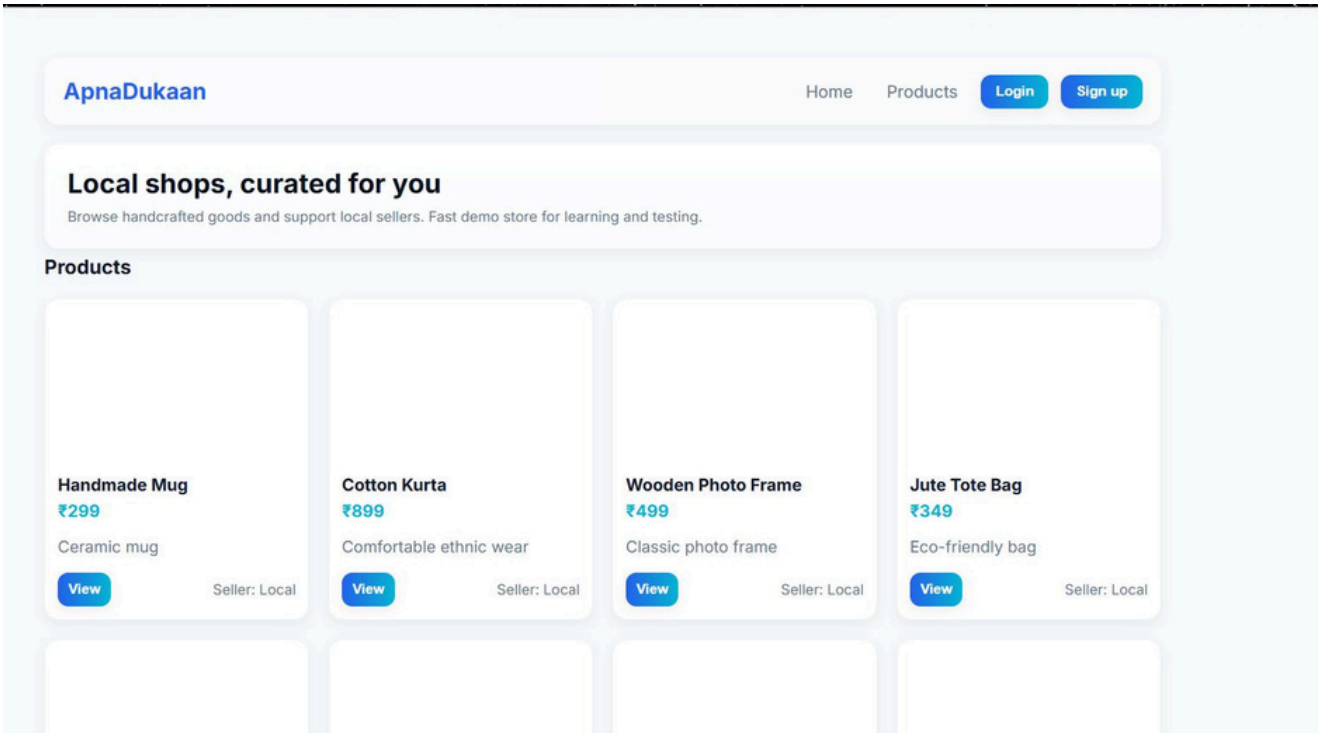
9.7 Cloud Deployment and CI/CD

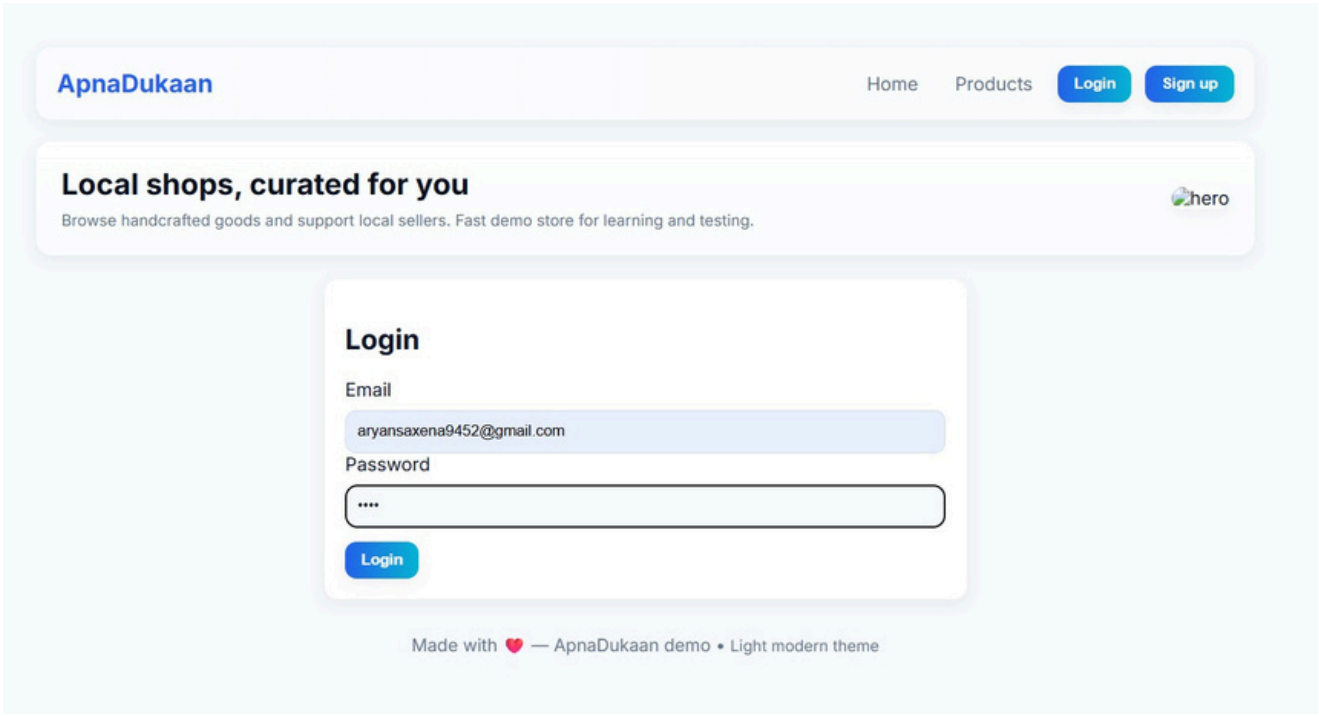
To support large-scale operations, deploying ApnaDukaan on cloud platforms like AWS, Azure, or Google Cloud can ensure better uptime, performance, and scalability.

Integrating Continuous Integration/Continuous Deployment (CI/CD) pipelines will automate testing and deployment, reducing downtime and manual errors.

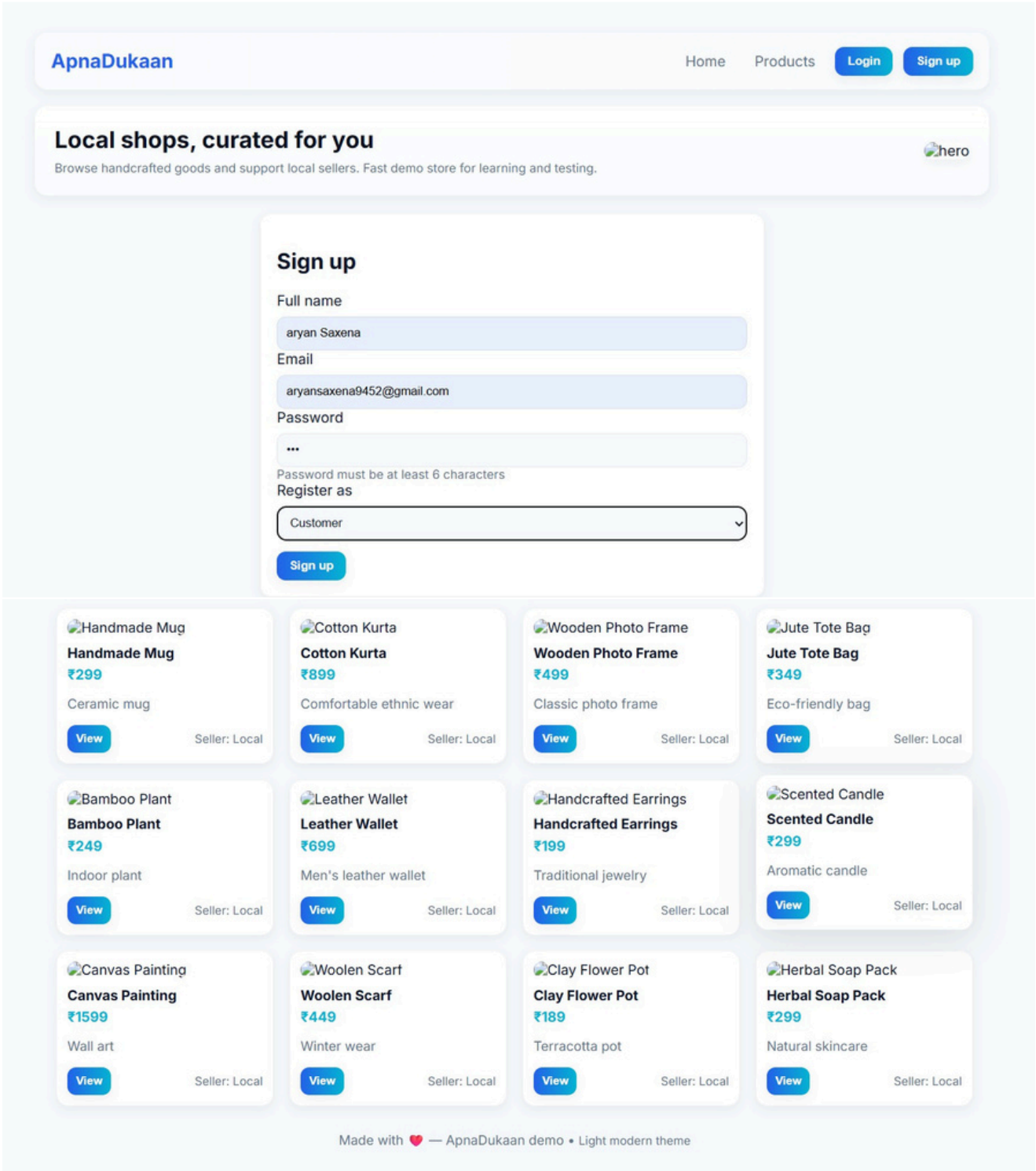
9.8 Multilingual and Multi-Currency Support

10.Images


ApnaDukaan
Home
Products
Login
Sign up
Local shops, curated for you
Browse handcrafted goods and support local sellers. Fast demo store for learning and testing.
Products
Handmade Mug
₹299
Ceramic mug
View
Seller: Local
Cotton Kurta
₹899
Comfortable ethnic wear
View
Seller: Local
Wooden Photo Frame
₹499
Classic photo frame
View
Seller: Local
Jute Tote Bag
₹349
Eco-friendly bag
View
Seller: Local


ApnaDukaan
Home
Products
Login
Sign up
Local shops, curated for you
Browse handcrafted goods and support local sellers. Fast demo store for learning and testing.
hero
Login
Email
aryansaxena9452@gmail.com
Password
Login
Made with — ApnaDukaan demo • Light modern theme

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.



11. Conclusion

ApnaDukaan represents a meaningful step toward digital empowerment of local sellers and inclusive e-commerce growth. By focusing on simplicity, security, and scalability, it bridges the existing digital divide between small shopkeepers and the online marketplace. The project successfully demonstrates how modern technologies like React.js, Spring Boot, and MySQL can be combined to build a real-world, community-oriented solution. The decoupled architecture ensures maintainability, and the robust backend ensures that transactions are secure and reliable. From a technical perspective, the project showcases strong skills in frontend development, RESTful API design, and database integration. From a social perspective, it reflects an initiative to empower small businesses, encourage entrepreneurship, and contribute to the Digital India vision. As technology continues to evolve, ApnaDukaan has the potential to expand into a full-fledged digital commerce ecosystem that supports thousands of sellers, integrates smart analytics, and offers a truly local yet global shopping experience. By combining innovation, usability, and community focus, ApnaDukaan stands as a scalable, impactful, and future-ready platform that embodies the true spirit of modern Indian e-commerce.

12.References

1. MongoDB Official Documentation – https://www.mongodb.com/docs

2. Express.js Official Documentation – https://expressjs.com

3. ReactJS Official Documentation – https://react.dev 4. Node.js Official

Documentation – https://nodejs.org 5. Tailwind CSS Framework

Documentation – https://tailwindcss.com/docs 6. Stack Overflow

Developer Community – https://stackoverflow.com

Bibliography

● “Full Stack Web Developmentwith React andNode.js” – Shama Hoque, Packt Publishing (2022)

● “Web Development with MongoDB and Node.js” – Brad Dayley, Addison-Wesley Professional

● “Software Engineering” – Ian Sommerville, Pearson Education

● “Clean Code: A Handbook of Agile Software Craftsmanship” – Robert C. Martin

● “Designing Data-Intensive Applications” – Martin Kleppman