ABCCommerce – Project Requirement Document (PRD)

1. Project Overview

1.1 Project Name

ABCCommerce

1.2 Project Description

ABCCommerce is a full-featured e-commerce platform designed to support online product sales with secure authentication, role-based administration, order management, payments, analytics, and scalable infrastructure. The system is built with modern web technologies and follows industry best practices for security, performance, and maintainability.

1.3 Project Goals

- Provide a seamless shopping experience for customers
- Enable administrators to manage products, orders, users, and content efficiently
- Ensure secure authentication, authorization, and payment processing
- Support scalability and future feature expansion
- Provide analytics and reporting for business insights

1.4 Target Users

- **Customers**: Browse products, place orders, track deliveries
- **Admins**: Manage products, orders, users, analytics
- **Support Staff (Optional)**: Handle refunds, customer inquiries

---

2. Scope

2.1 In Scope

- Web-based e-commerce application
- User authentication & authorization
- Product catalog & inventory management
- Shopping cart & checkout
- Payment integration
- Order lifecycle management
- Admin dashboard
- Analytics & reporting

2.2 Out of Scope (Phase 1)

- Native mobile applications
- Marketplace (multi-vendor)
- AI-based recommendations (future phase)

---

3. System Architecture

3.1 High-Level Architecture

- **Frontend**: Next.js (React)
- **Backend**: Next.js API routes / Node.js
- **Database**: Supabase (PostgreSQL)
- **Authentication**: Supabase Auth
- **ORM**: Prisma or Drizzle
- **Payments**: Stripe
- **Hosting**: Vercel (frontend) + Supabase
- **Storage**: Supabase Storage (product images)

3.2 Architecture Diagram (Conceptual)

```
Client (Browser)
   ↓
Next.js Frontend
   ↓
Next.js Backend API
   ↓
Supabase (Auth + DB + Storage)
   ↓
Stripe Payment Gateway
```

---

4. Functional Requirements

4.1 Authentication & Authorization

4.1.1 User Registration

- Email & password signup
- Email verification
- Password reset

4.1.2 User Login

- Secure login with session handling
- Token-based authentication

4.1.3 Roles

- **Admin**: Full system access
- **Customer**: Shopping & order access

4.2 User Management (Admin)

- Create, update, delete users
- Assign roles
- View user order history

4.3 Product Management

4.3.1 Product

- Fields:
- Product ID
- Name
- Description
- Price
- Category
- Stock quantity
- Images
- Status (active/inactive)

4.3.2 Admin Features

- Create / edit / delete products
- Upload product images
- Manage inventory

4.3.3 Customer Features

- Browse products
- Search & filter products
- View product details

4.4 Shopping Cart

- Add/remove products

- Update quantity
- Persistent cart (logged-in users)
- Price calculation (subtotal, tax, total)

4.5 Checkout & Payment

4.5.1 Checkout Flow

1. Review cart
2. Enter shipping details
3. Payment
4. Order confirmation

4.5.2 Payment

- Stripe integration
- Secure payment intent handling
- Support for cards

4.6 Order Management

4.6.1 Order Status

- Pending
- Paid
- Shipped
- Completed
- Cancelled

4.6.2 Customer

- View order history
- Track order status

4.6.3 Admin

- View all orders
- Update order status
- Process refunds

4.7 Analytics & Reporting

4.7.1 Metrics

- Total sales

- Orders per day/week/month
- Revenue per product
- Active users

4.7.2 Dashboard

- Visual charts
- Export reports (CSV)

---

5. Non-Functional Requirements

5.1 Performance

- Page load < 2 seconds
- API response < 500ms (average)

5.2 Security

- HTTPS everywhere
- Secure password hashing
- Role-based access control
- SQL injection & XSS protection

5.3 Scalability

- Horizontal scaling via serverless
- Optimized database indexes

5.4 Availability

- 99.9% uptime target

---

6. Database Design (High Level)

6.1 Tables

- users
- products
- categories
- carts
- cart_items
- orders
- order_items
- payments
- analytics_events

7. API Requirements

7.1 Auth APIs

- POST /auth/signup
- POST /auth/login
- POST /auth/logout

7.2 Product APIs

- GET /products
- POST /products (admin)
- PUT /products/:id (admin)
- DELETE /products/:id (admin)

7.3 Order APIs

- POST /orders
- GET /orders
- PUT /orders/:id/status (admin)

8. UI/UX Requirements

8.1 Design Principles

- Clean, modern UI
- Mobile-responsive
- Accessible (WCAG 2.1)

8.2 Key Pages

- Home
- Product Listing
- Product Detail
- Cart
- Checkout
- Order Confirmation
- Admin Dashboard

9. Deployment & Environment

9.1 Environments

- Development

- Staging
- Production

9.2 CI/CD

- GitHub Actions
- Automated testing
- Automatic deployment

---

10. Risks & Mitigation

Risk	Impact	Mitigation
Payment failures	High	Stripe retries & webhooks
Security breach	High	Regular audits
Performance issues	Medium	Caching & monitoring

---

11. Future Enhancements

- Mobile app
- AI product recommendations
- Multi-vendor support
- Loyalty & rewards system

---

12. Acceptance Criteria

- Users can register, login, and place orders
- Admin can manage products and orders
- Payments are processed securely
- System meets performance benchmarks

---

13. Approval

Role	Name	Signature	Date
Product Owner			
Developer			
Client			