

**E-COMMERCE**

[CSC217(Fall-2024/MWE-22732)](https://lms.iobm.edu.pk/moodle/course/view.php?id=24728#section-15)

**Project Report:**

**Femina -Website**

# Instructor: Salman Akbar

Prepared by:

**Alisha Nasir (20221-33353)**

**Date Of Submission:**

Contents

[**Femina — Project Report** 1](#_Toc205761783)

[**1.** **Model (Business Model & Goals)** 3](#_Toc205761784)

[**2.** **Mission** 3](#_Toc205761785)

[**3.** **Vision** 3](#_Toc205761786)

[**4.** **Right Technology Stack (recommended)** 3](#_Toc205761787)

[**5.** **Target Audience Analysis** 5](#_Toc205761788)

[**6.** **Competitive Market Research (approach & findings)** 5](#_Toc205761789)

[**7.** **Core Features and Functionality (mapped to your implementation)** 6](#_Toc205761790)

[**8.** **User-Friendly Interface (UX suggestions)** 6](#_Toc205761791)

[**9.** **Planning for Scaling** 7](#_Toc205761792)

[**10.** **Security** 7](#_Toc205761793)

[**11.** **Secure Payment Gateway & Payment Flow** 8](#_Toc205761794)

[**12.** **Integrating Shipping Solutions** 8](#_Toc205761795)

[**13.** **Testing and Quality Assurance** 8](#_Toc205761796)

[**14.** **Market and Customer Acquisition** 9](#_Toc205761797)

[**15.** **Deployment and Launch Strategy** 9](#_Toc205761798)

[**16.** **Budgeting & Costing Summary (rough ranges)** 10](#_Toc205761799)

[**17.** **Deliverables from Current Project (what you already have)** 10](#_Toc205761800)

[**18.** **Short Roadmap — Next 90 Days** 11](#_Toc205761801)

[**19.** **Appendix — Suggested Tech & Tools (quick list)** 11](#_Toc205761802)

1. **Model (Business Model & Goals)**

**Model:** Business-to-Consumer (B2C) multi-seller marketplace focused on women’s clothing — enabling independent sellers/brands to list apparel and reach buyers across Pakistan and other South Asian markets.  
**Primary revenue streams:**

* Commission on each sale (percentage of order value)
* Listing/promotion fees for sellers (featured placement / sponsored products)
* Shipping/fulfillment fees and value-added services (gift wrap, express delivery)  
  **Goals:** Rapidly grow product variety and customer base, provide an easy mobile-first shopping experience, and onboard sellers to scale supply.

1. **Mission**

Make online buying and selling easy for people and small businesses using advanced, secure, and user-friendly technology — enabling sellers to access new customers and enabling customers to discover a curated range of women’s wear.

1. **Vision**

Empower over 100 million people and businesses by expanding choices for customers and extending market reach for sellers across Pakistan and neighboring South Asian countries; be the go-to destination for curated, trustworthy women’s fashion.

1. **Right Technology Stack (recommended)**

This is chosen to match a small-to-medium e-commerce launch that can scale.

**Front-end:**

* HTML5 + CSS3 + Bootstrap 5 (already used in your code) — fast to prototype, responsive UI
* Vanilla JavaScript (for cart, routing) or React/Vue if you want SPA architecture later

**Back-end (recommended for production):**

* Node.js + Express (easy to pair with JS front-end) or Python Flask/Django

**Database:**

* PostgreSQL (relational product/catalog data)
* Redis for session/cache (optional for speed)

**Storage & Media:**

* Cloud object storage for images (AWS S3 / Google Cloud Storage / DigitalOcean Spaces)

**Payments:**

* Stripe or PayPal (for card payments) — regionally, choose a local provider supported in Pakistan (e.g., 2Checkout, PayFast equivalents); integrate a PCI-compliant hosted flow.
* Cash on Delivery (COD) support in checkout (business/merchant side validation & fraud checks)

**Delivery & Shipping:**

* Integrate with local shipping partners (call their API for rates and tracking)
* Webhooks for delivery status

**Hosting & DevOps:**

* Cloud (AWS/GCP/Azure/DigitalOcean) — start with a single VM or managed app (Elastic Beanstalk, App Engine, DigitalOcean App Platform), and use containers (Docker) for portability
* CI/CD: GitHub Actions / GitLab CI for automated testing and deployments
* CDN (Cloudflare / AWS CloudFront) for static assets and images

**Monitoring & Logs:**

* Sentry for frontend & backend error monitoring
* Prometheus + Grafana or hosted monitoring for metrics

1. **Target Audience Analysis**

**Primary:** Women aged 18–45, fashion-conscious, online shoppers using mobile and desktop.  
**Secondary:** Gift buyers (family/partners), small boutique owners searching for wholesale/resell items.  
**Location:** Urban & suburban Pakistan initially; expand to South Asia.  
**Buyer behavior insights:** Price-sensitive but value trust, reviews, and easy returns. Mobile-first UX matters.

1. **Competitive Market Research (approach & findings)**

**Approach:**

* Identify direct competitors (local e-commerce clothing marketplaces & boutiques).
* Benchmark product assortment, shipping costs, return policies, UI patterns, payment options, seller onboarding process, and promotions.

**Actionable findings:**

* Fast, free (or cheap) shipping and clear returns increase conversions.
* Clear size charts and product images reduce returns.
* Localized payments and COD are crucial for markets with low card penetration.

1. **Core Features and Functionality (mapped to your implementation)**

**Already implemented (in your single-file app):**

* Responsive Navbar, Hero & Trending section
* Categories/Filters (category & price range)
* Products grid and product detail pages (with size & qty)
* Add-to-cart / cart with qty update & remove (localStorage)
* Checkout page with COD and mock card payments
* Order confirmation (stored to localStorage)

**To implement next (production-ready):**

* User accounts & authentication (email/password, OTP mobile login)
* Seller dashboards (product management, orders)
* Inventory management & stock checks on add to cart
* Real payment gateway integration & PCI compliance
* Order management & shipping integrations
* Reviews & ratings, wishlists, search & product recommendations

1. **User-Friendly Interface (UX suggestions)**

* Keep big product imagery, consistent product-card heights, readable fonts. (Bootstrap helps.)
* Add size guide modal on product page.
* Use a cart drawer for quicker checkout flow on mobile.
* Clear CTA colors for “Add to cart” / “Checkout”; show microcopy for shipping times.
* Add inline form validation on checkout and friendly error messages.

1. **Planning for Scaling**

**Short-term (MVP):** Single backend instance, managed database, CDN for static assets.  
**Medium-term:** Autoscaling app servers behind a load balancer, read-replicas for DB, caching layer (Redis).  
**Long-term:** Microservices for catalog, payments, orders; sophisticated search (Elasticsearch) and recommendation system; multi-region deployment.

**Capacity planning tips:**

* Monitor peak concurrent users; benchmark cart & checkout requests; ensure DB connection pooling and caching for heavy reads (product catalogs).

1. **Security**

* Use HTTPS everywhere (TLS) — mandatory.
* Sanitize all user inputs (server-side) and validate on client side.
* Protect sessions (secure, HttpOnly cookies) and implement rate limiting and account lockouts.
* Implement CSRF protection for state-changing requests.
* Store only minimal sensitive data; never store raw card details on your servers. Use PCI-compliant payment providers.
* Regular vulnerability scans & keep dependencies up to date.

1. **Secure Payment Gateway & Payment Flow**

**Recommendations:**

* Use a hosted/redirect payment flow so sensitive card data is handled by the payment provider (Stripe Checkout, PayPal Checkout, or a local PSP).
* Implement server-side webhooks for payment confirmation and order fulfillment.
* For COD: add address/phone verification and risk checks (order limits, manual review for high-risk orders).

**Integration steps**

1. Create merchant account with chosen PSP.
2. Implement server endpoint to create payment intent/order.
3. Client redirects/opens PSP checkout; on success, PSP sends webhook to your backend to confirm order and update status.
4. For COD, create order record with payment\_method: cod and status: pending\_payment until delivery.
5. **Integrating Shipping Solutions**

* Connect to local couriers’ APIs for real-time rates and tracking (enumerate partners during implementation).
* Offer simple shipping rules: free shipping over X, flat small order shipping, or per weight/zone.
* Provide tracking link in user order history & order confirmation email/SMS.

1. **Testing and Quality Assurance**

**Types of testing**

* Unit tests for back-end logic (orders, payments).
* Integration tests for payment & shipping flows (including mock providers).
* End-to-end tests for critical user journeys: browse → add to cart → checkout (Cypress, Playwright).
* Accessibility testing (axe) and responsive testing on key device dimensions.

**Acceptance criteria:**

* Checkout success (card and COD) must result in order creation and persistent order id.
* Cart persistence across page reload / mobile.
* Stock decrement on successful payment and prevention of over-selling.

1. **Market and Customer Acquisition**

**Channels:**

* Social media (Instagram / Facebook) with influencer partnerships and targeted ads.
* Google Shopping / SEO & product feeds.
* Email marketing (welcome discounts, abandoned cart reminders).
* Promotions: season sales, free shipping thresholds.

**KPIs:**

* Conversion rate, average order value (AOV), customer acquisition cost (CAC), repeat purchase rate.

1. **Deployment and Launch Strategy**

**MVP launch:**

* Soft launch with limited inventory and selected sellers; monitor orders and feedback.
* Use staged rollout: dev → staging → production with CI/CD (GitHub Actions).
* Analytics set up (Google Analytics / GA4 + events) and error monitoring (Sentry).

**Go-to-market:**

* Launch announcement + promo codes, social campaigns with product images (use the site hero and trending imagery).

1. **Budgeting & Costing Summary (rough ranges)**

These are ballpark monthly / upfront estimates and should be validated against local vendor pricing.

**Initial (MVP):**

* Development time: 3–8 weeks (1–2 developers)
* Hosting & infra (cloud VM + managed DB + CDN): $50–300 / month (small scale)
* Payments & gateway fees: per-transaction (2–5% + fixed fee)
* Marketing initial budget: $200–1,500 (ads/influencers)

**Scaling (6–12 months):**

* Increased infra: $300–2,000 / month depending on traffic
* Additional engineering for seller dashboard, logistics: $5k–20k (one-time)

Record real quotes when selecting cloud provider, PSP, shipping partners and adjust the budget.

1. **Deliverables from Current Project (what you already have)**

* Single-file Bootstrap demo with: hero, categories, product grid, product details, add-to-cart, cart management (localStorage), checkout (mock card + COD).
* Client-side routing and small product catalogue (5 sample products).

1. **Short Roadmap — Next 90 Days**
2. Add server backend (Node/Express) + PostgreSQL — persist products, users, orders.
3. Integrate real payment gateway (hosted flow) and implement webhooks.
4. Implement seller onboarding & basic dashboard for product management.
5. Add email/SMS notifications (order confirmations) and analytics.
6. Run end-to-end tests and security review.
7. Soft launch; monitor and iterate based on metrics.
8. **Appendix — Suggested Tech & Tools (quick list)**

* Frontend: Bootstrap 5, optional React for future SPA
* Backend: Node.js + Express or Django
* Database: PostgreSQL; caching with Redis
* Payments: Stripe/PayPal or local PSP with hosted checkout
* Hosting: DigitalOcean / AWS EC2 + RDS or managed app platform
* CDN: Cloudflare or CloudFront
* CI/CD: GitHub Actions
* Monitoring: Sentry, Grafana/Prometheus