Functional Requirements

These define what the system must do, mapped from the user stories in the product backlog.

1. Account Creation – Allow customers to create an account using email and password to track orders.
2. Login – Enable customers to log in securely with email and password.
3. Browse Products – Allow browsing by category for easy navigation.
4. Add to Cart – Permit customers to add products to a shopping cart for multi-item purchases.
5. View Cart – Display shopping cart contents for review before checkout.
6. Checkout – Support multiple payment methods (credit card, PayPal, COD).
7. Order Confirmation – Send an email upon successful purchase.
8. Admin Product Management – Allow admins to add, edit, or remove products from the catalog.
9. Admin Sales Reports – Enable admins to view sales reports for business performance analysis.
10. Order Tracking – Allow customers to track order status and estimated arrival.

Non-Functional Requirements

These define *how the system performs*, ensuring quality and alignment with agile sprint goals.

1. Security – Encrypt sensitive data (passwords, payments); secure login to prevent unauthorized access.
2. Performance – Page load time under 2 seconds; handle up to 100 concurrent users.
3. Usability – Intuitive, mobile-responsive UI; accessible per *WCAG 2.1* guidelines.
4. Reliability – System uptime *99.9%*; confirmation emails delivered within 1 minute.
5. Scalability – Support thousands of products/orders; modular architecture for future sprints.
6. Maintainability – Clean architecture, documented codebase, role-specific documentation.

Stakeholder Analysis

Stakeholders identified from user stories, team roles, and project goals.

Stakeholders & Interests

* Customers – Need easy account management, browsing, shopping, and tracking.
* Admins – Need product management and analytics tools.
* Development Team – Developers, tester, Scrum Master; need clear tasks and tools.
* Project Owner/Manager – Monitors delivery, backlog, progress tracking.
* External Parties – Payment providers, email services for integrations.

Analysis Matrix

| **Stakeholder** | **Needs** | **Risks** | **Mitigation** |
| --- | --- | --- | --- |
| Customers | Secure login, easy browsing | Data breaches | Encryption, testing |
| Admins | Product management tools | Inaccurate reports | Validation in reports |
| Dev Team | Clear tasks, deadlines | Delays in sprints | Kanban/Gantt tracking |
| Project Owner | Progress visibility | Scope creep | Prioritized backlog |
| External | Reliable API integrations | Downtime | Redundant providers |

Requirement Gathering Techniques

Agile-aligned methods ensured requirements were *stakeholder-driven*:

1. Brainstorming – Team sessions to generate user stories in “As a … I want … so that …” format.
2. Interviews – Conducted with customers and admins (e.g., payment options, report views).
3. User Story Mapping – Prioritized stories by points (1–8) and priority (High/Medium/Low).
4. Prototyping – Early wireframes for forms (e.g., registration) during sprint planning.
5. Observation – Studied similar e-commerce sites for browsing/checkout flow inspiration.