**3**. **Requirements Gathering**

3.1 Functional Requirements

**1. User Management**

* User Registration
* Login/Logout
* Password Recovery
* Profile Update
* Manage Addresses

**2. Product Management**

* View Product Listings
* View Product Details
* Search Functionality
* Filters & Sorting
* Category-Based Product Listing
* Add to Wishlist
* Add to Cart

**3. Cart & Checkout Management**

* Update Cart Items
* Remove Items from Cart
* Checkout Process
* Select Payment Method
* Order Summary
* Order Confirmation & Email Notification
* Order History

**4. Payment & Transactions**

* Multiple Payment Methods (Credit Card, PayPal, Cash on Delivery)
* Payment Confirmation
* Failed Payment Handling
* Payment Notification Emails

**5. User Experience & Navigation**

* Responsive Design
* Multi-language Support
* User Notifications
* Easy Navigation

**6. Communication & Support**

* Contact Us Form
* FAQ Page
* Product Reviews & Ratings
* Newsletter Subscription

**7. Security & Session Management**

* User Data Encryption
* Session Timeout & Unauthorized Access Prevention
* Two-Factor Authentication (if supported)
* Protection against SQL Injection & XSS attacks

3.2 Nonfunctional Requirements

Automation Exercise is a website designed for practicing automation testing. Below are the key nonfunctional requirements tailored for its functionality and purpose:

Performance Requirements

- The website should load within 2 seconds under normal conditions.

- It should support at least 10,000 concurrent users without affecting performance.

- API responses should be processed within 500 milliseconds.

Scalability Requirements

- The system should support horizontal scaling, allowing more servers to be added as traffic increases.

- The database should be optimized to handle large amounts of test data efficiently.

Reliability and Availability

- The website should have 99.9% uptime to ensure continuous availability.

- Automatic backup mechanisms should be in place to prevent data loss, with backups stored every 24 hours.

Security Requirements

- User passwords should be hashed using SHA-256 and stored securely.

- All data transmissions should use HTTPS with TLS 1.3 encryption.

- Role-based access control (RBAC) should be implemented to limit permissions for different user types.

- Protection against SQL injection, XSS, and CSRF attacks should be enforced.

Usability Requirements

- The website should have an intuitive user interface with a simple and clear navigation structure.

- Forms and automation scenarios should be easily accessible within 3 clicks.

- The website should support mobile-friendly and responsive design.

Maintainability and Supportability

- The system should follow a modular design to allow easy updates and modifications.

- Logs and error tracking should be implemented to detect and resolve issues in real-time.

- The test scenarios should be updated monthly to align with the latest automation frameworks.

Interoperability

- The website should be compatible with automation testing tools like Selenium, Cypress, and Playwright.

- APIs should follow RESTful architecture for easy integration with external testing tools.

- It should support different browsers (Chrome, Firefox, Edge, Safari) for cross-browser testing.

Compliance and Legal Requirements

- The website should comply with GDPR to protect user data.

- Regular security audits should be conducted every 6 months.

Efficiency and Energy Consumption

- The website should optimize server resource usage to minimize energy consumption.

- Test data should be managed efficiently to avoid excessive database load.

Testability

- The website should be structured in a way that supports automated UI and API testing.

- Logs should be generated for every test execution to assist in debugging.