## G-1 Group

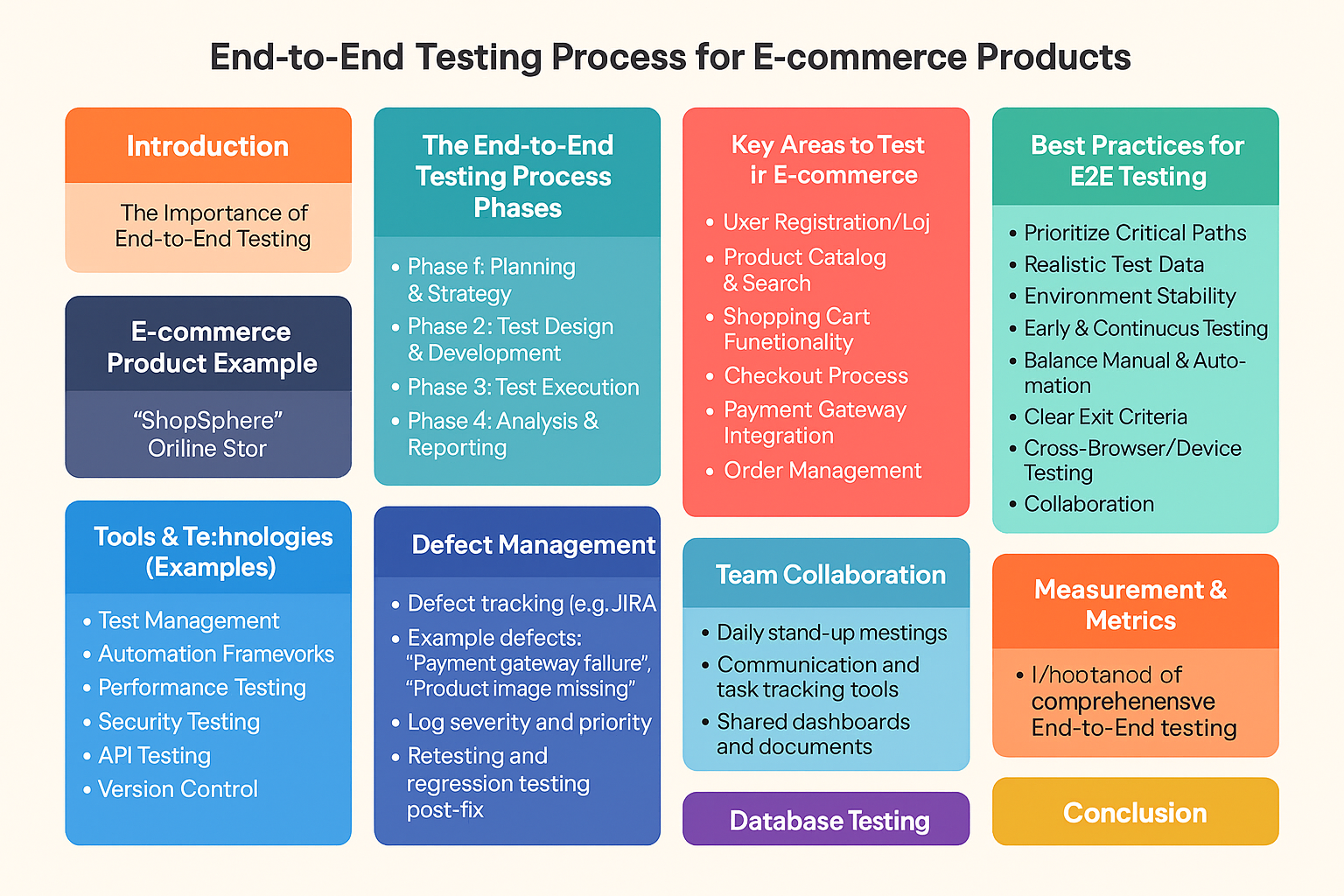
**Date:- 23rd July 2025**

|  |  |
| --- | --- |
| **Group Participants** | **Points** |
| **Abhigyan Maji- 30382** | **1 - 3** |
| **Gourab Das - 30392** | **4 - 5** |
| **Jyotirmay Chowdhury(coordinator) - 30639** | **6** |
| **Nandhu - 30400** | **7 - 8** |
| **Shikha - 31136** | **9 - 13** |

# Topic:- End-to-End Testing Process for E-commerce Products (ShopSphere)

## 1. Introduction: The Importance of End-to-End Testing

End-to-End (E2E) testing is a methodology used to test whether the flow of an application from start to finish is behaving as expected. The purpose of E2E testing is to simulate real user scenarios and ensure that all integrated components of the system work together seamlessly to achieve business goals.



For an e-commerce product, E2E testing is crucial because:

* It validates the entire customer journey, from browsing to purchase and post-purchase activities.
* It identifies issues across integrated systems (e.g., payment gateways, shipping APIs, inventory management).
* It ensures a smooth, reliable, and secure user experience, directly impacting customer satisfaction and revenue.

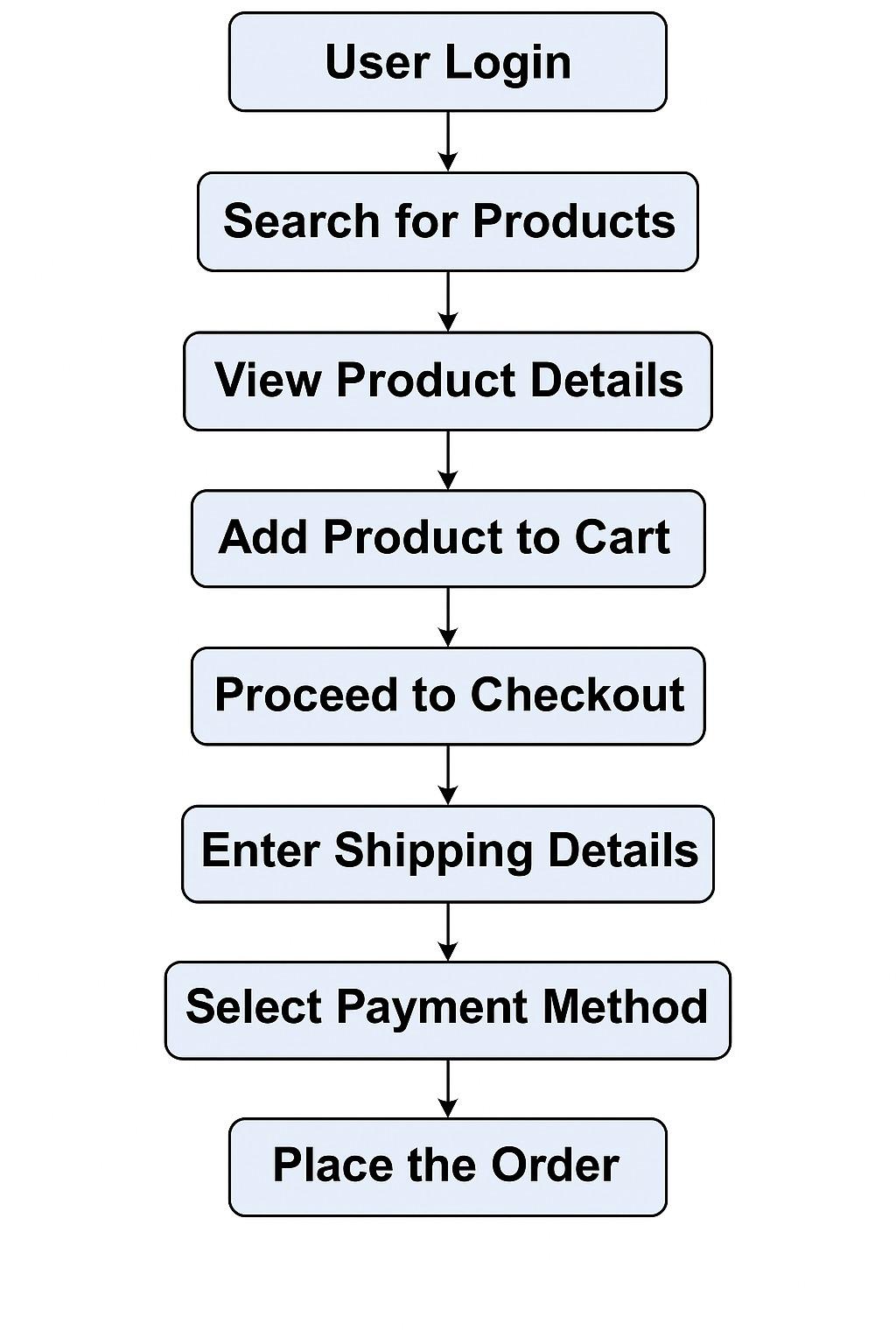
## 2. E-commerce Product Example: "ShopSphere" Online Store

To illustrate the E2E testing process, let's consider a hypothetical e-commerce product called "ShopSphere."

**ShopSphere Features:**

* **User Management:** Registration, Login, Profile Management (address, order history).
* **Product Catalog:** Browse categories, search products, view product details (images, descriptions, pricing, reviews).
* **Shopping Cart:** Add/remove items, update quantities, apply coupons.
* **Checkout:** Shipping address selection, shipping method, payment options (Credit Card, UPI, Net Banking), order summary.
* **Order Management:** Order confirmation, order history, order status tracking.
* **Integrations:** Payment Gateway (e.g., Stripe, Stripe), Shipping Carrier API.

## 3. Flowchart Diagram of ShopSphere



## 4. The End-to-End Testing Process Phases

The E2E testing process can be broken down into several key phases:

### Phase 1: Planning & Strategy

This foundational phase defines the scope and approach for testing.

* **Define Scope:**
  + Identify all critical business flows and user journeys to be tested (e.g., "Guest user purchases a product," "Registered user updates profile and buys multiple items").
  + Determine the boundaries of the testing (which systems/integrations are in scope).
  + Agree on the definition of "done" for E2E testing.
* **Identify Test Scenarios:**
  + Based on business requirements and user stories, brainstorm high-level scenarios.
  + Consider both happy paths (successful flows) and unhappy paths (error conditions, edge cases).
  + *Example Scenarios for ShopSphere:*
    - Successful purchase by a registered user.
    - Guest user attempts to purchase without registration.
    - Adding out-of-stock item to cart.
    - Applying an invalid coupon code.
    - Payment failure and retry.
* **Choose Tools & Environment:**
  + Select appropriate testing tools (manual, automation, performance, security).
  + Define the test environment setup (e.g., staging, pre-production, mirroring production data).

### Phase 2: Test Design & Development

This phase involves creating detailed test artifacts.

* **Create Detailed Test Cases:**
  + Translate high-level scenarios into step-by-step test cases.
  + Each test case should include: Test Case ID, Title, Pre-conditions, Steps, Expected Results, Post-conditions.
  + *Example Test Case (ShopSphere - Partial):*
    - **ID:** E2E\_PURCHASE\_001
    - **Title:** Registered User Completes Standard Purchase
    - **Pre-conditions:** User "testuser@example.com" registered and logged in. Product "Laptop X" is in stock.
    - **Steps:**
      1. Navigate to Product "Laptop X" page.
      2. Click "Add to Cart."
      3. Click "Proceed to Checkout."
      4. Select the existing shipping address.
      5. Select "Standard Shipping."
      6. Choose "Credit Card" as payment method.
      7. Enter valid credit card details.
      8. Click "Place Order."
    - **Expected Results:** Order confirmation page displayed with correct order details. Order status is "Processing" in user's order history.
* **Prepare Test Data:**
  + Generate or procedure realistic test data for users, products, addresses, payment methods, etc.
  + Ensure data covers various permutations (e.g., different product types, price ranges, user roles).
* **Develop Automation Scripts (if applicable):**
  + For repetitive and critical paths, develop automated test scripts using chosen frameworks (e.g., Selenium, Playwright).
  + Focus on stability and maintainability of scripts.

### Phase 3: Test Execution

This is where the actual testing takes place.

* **Environment Setup:**
  + Ensure the test environment is stable, configured correctly, and contains the necessary test data.
  + Verify all integrated systems are accessible and functional.
* **Execute Test Cases:**
  + **Manual Testing:** Execute complex, exploratory, or visual test cases manually.
  + **Automated Testing:** Run automated test suites.
  + Prioritize test execution based on criticality and risk.
* **Defect Reporting & Tracking:**
  + Log any deviations from expected behavior as defects.
  + Provide clear steps to reproduce, actual results, expected results, screenshots, and relevant logs.
  + Track defects through their lifecycle (New, Open, In Progress, Fixed, Re-tested, Closed).

### Phase 4: Analysis & Reporting

Evaluating the testing effort and communicating results.

* **Test Results Analysis:**
  + Analyze pass/fail rates for test cases.
  + Review defect trends and severity.
  + Identify areas of the application with high defect density.
* **Reporting:**
  + Generate comprehensive test reports summarizing test coverage, execution status, and defect metrics.
  + Provide a Go/No-Go recommendation for release based on defined exit criteria.
* **Regression Testing:**
  + After bug fixes or new feature development, perform regression testing to ensure existing functionality remains intact and no new bugs have been introduced.

### Phase 5: Release & Post-Release

The final steps before and after deployment.

* **Sign-off:**
  + Obtain formal sign-off from stakeholders (Product Owners, Business Analysts) indicating readiness for release.
* **Monitoring:**
  + Post-release, monitor the production environment for any issues that might have slipped through testing.
  + Gather user feedback for continuous improvement.

## 5. Key Areas to Test in E-commerce (ShopSphere Example)

Specific functional and non-functional areas crucial for E2E testing in an e-commerce context:

* **User Registration/Login:**
  + New user registration (valid/invalid data, existing email).
  + Login/Logout (valid/invalid credentials).
  + Password reset/forgot password flow.
  + Social media logins (if applicable).
* **Product Catalog & Search:**
  + Product browsing by category, sub-category.
  + Search functionality (exact match, partial match, no results, special characters).
  + Filtering and sorting options.
  + Product detail page (images, description, price, availability, reviews, related products).
* **Shopping Cart Functionality:**
  + Adding/removing single/multiple items.
  + Updating item quantities.
  + Applying valid/invalid coupon codes.
  + Emptying the cart.
  + Cart persistence across sessions.
* **Checkout Process:**
  + Guest checkout vs. Registered user checkout.
  + Shipping address entry/selection.
  + Shipping method selection (standard, express, pick-up).
  + Order summary accuracy (items, prices, shipping, tax, total).
  + Error handling for missing required fields.
* **Payment Gateway Integration:**
  + Successful payment with various methods (Credit Card, Debit Card, UPI, Net Banking).
  + Payment failures (insufficient funds, invalid card, network issues).
  + Transaction status updates.
  + Refund/cancellation process.
* **Order Management:**
  + Order confirmation email/SMS receipt.
  + Viewing order history and details in user profile.
  + Order status updates (processing, shipped, delivered, cancelled).
* **User Profile Management:**
  + Updating personal information, addresses.
  + Viewing past orders, wishlists.
* **Security Testing:**
  + Authentication (session management, brute force prevention).
  + Authorization (user roles and permissions).
  + Data encryption (especially payment details).
  + Protection against common vulnerabilities (OWASP Top 10).
* **Performance Testing:**
  + Website load times under normal and peak traffic.
  + Responsiveness of key transactions (search, add to cart, checkout).
  + Scalability under increasing user load.
* **Responsiveness:**
  + Layout and functionality across various devices (desktop, tablet, mobile) and browsers.
* **Integrations:**
  + Verify data flow and functionality with third-party APIs (e.g., shipping tracking, inventory updates).

## 6. Tools & Technologies (Examples)

* **Test Management:** Jira, TestRail, Azure DevOps
* **Automation Frameworks:** Selenium, Playwright, Cypress (Web UI), Appium (Mobile)
* **Performance Testing:** JMeter, LoadRunner, K6
* **Security Testing:** OWASP ZAP, Burp Suite
* **API Testing:** Postman, SoapUI
* **Version Control:** Git

## 7. Best Practices for E2E Testing

* **Prioritize Critical Paths:** Focus testing efforts on the most important user journeys and business functionalities.
* **Realistic Test Data:** Use data that closely mimics production data to uncover real-world issues.
* **Environment Stability:** Ensure a dedicated and stable test environment that mirrors production as closely as possible.
* **Early & Continuous Testing:** Integrate E2E testing early in the development lifecycle and perform it continuously.
* **Balance Manual & Automation:** Automate repetitive and stable test cases, but retain manual testing for exploratory and complex scenarios.
* **Clear Exit Criteria:** Define specific conditions that must be met for E2E testing to be considered complete.
* **Cross-Browser/Device Testing:** Test across different browsers, devices, and operating systems relevant to your target audience.
* **Collaboration:** Foster strong collaboration between QA, Development, and Business teams.

## 8. Test Case Execution Summary

This section provides a summary of key E2E test cases and their execution status. This table would typically be populated dynamically from a test management tool, but here's a static example for illustration.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Title** | **Feature Area** | **Execution Date** | **Status** | **Tester** | **Notes** |
| E2E\_PURCHASE\_001 | Registered User Completes Standard Purchase | Checkout | 2025-07-27 | PASS | QA Team | All steps successful. |
| E2E\_CART\_003 | Add Out-of-Stock Item to Cart | Shopping Cart | 2025-07-27 | FAIL | QA Team | Error message not displayed as per spec. (Defect ID: #1234) |
| E2E\_LOGIN\_002 | Login with Invalid Credentials | User Management | 2025-07-27 | PASS | QA Team | Correct error message displayed. |
| E2E\_PAYMENT\_005 | Payment Failure - Insufficient Funds | Payment Gateway | 2025-07-28 | PASS | QA Team | User redirected to payment retry page. |
| E2E\_SEARCH\_001 | Product Search - Exact Match | Product Catalog | 2025-07-28 | PASS | QA Team | Correct product displayed in results. |
| E2E\_PROFILE\_001 | Update Shipping Address | User Profile | 2025-07-28 | PASS | QA Team | Address updated successfully in profile. |
| E2E\_CHECKOUT\_007 | Apply Invalid Coupon Code | Checkout | 2025-07-28 | FAIL | QA Team | System allowed invalid coupon to be applied, no error. (Defect ID: #1235) |

# 9. Defect Management

Defect management involves identifying, logging, tracking, and resolving bugs during the testing lifecycle.  
  
 In the context of e-commerce:  
 - Use JIRA or Bugzilla for defect tracking  
 - Example defects: “Payment gateway failure”, “Product image missing”  
 - Log severity and priority  
 - Perform retesting and regression testing post-fix.

# 10. Database Testing

Database testing ensures the integrity, accuracy, and security of data stored by the application.  
  
 For an e-commerce platform:  
 - Validate product listings, cart contents, and orders in the database  
 - Ensure user login, signup, and order history are saved correctly  
 - Use SQL queries to verify data storage and retrieval

# 11. Team Collaboration

Effective collaboration is essential among developers, testers, and business stakeholders to ensure smooth testing and delivery.  
  
 - Daily stand-up meetings (Agile)  
 - Use tools like Slack, Microsoft Teams, and JIRA for communication and task tracking  
 - Shared dashboards and documents improve coordination

# 

# 12. Measurement & Metrics

Metrics help evaluate the effectiveness of the testing process and identify areas for improvement.  
  
 Common metrics used in e-commerce testing:  
 - Test Case Coverage = (Executed/Total Test Cases) × 100%  
 - Defect Density = Total defects / size of module  
 - Defect Leakage = Defects found post-release / Total defects  
 - Test Execution Rate = Passed / Executed test cases

## 13. Conclusion

End-to-End testing is very important for delivering a good-quality e-commerce product. It checks how everything works together—just like a real user would use the site. This helps make sure the platform (like "ShopSphere") runs smoothly, is safe to use, and gives customers a great shopping experience, which helps the business succeed.