**Test Plan for Flipkart**

1. ***Identifier***

TP-FK-WEB-2025-V1.0

1. ***Introduction***

The test plan outlines the strategy, scope, objectives, resources, and schedule for testing the Flipkart e-commerce website. The goal is to ensure the website's functionality, performance, security, and usability meet the requirements and provide a seamless experience for users.

Objectives:

* Verify core functionalities like user registration, product search, and checkout process.
* Ensure seamless integration with payment gateways.
* Test the website's performance under various load conditions.
* Validate the website's security features to protect user data.
* Test compatibility across devices, browsers, and platforms.
* Ensure a positive user experience with intuitive navigation and design.

Scope:

In-Scope:

* + User authentication (registration, login, and password recovery).
  + Product browsing and search functionality.
  + Shopping cart management.
  + Order placement, cancellation, and tracking.
  + Payment methods (credit/debit cards, UPI, wallets, cash on delivery, etc.).
  + Offers and discounts functionality.
  + Customer support features (chat, FAQs, help desk).
  + Backend API testing.

Out-of-Scope:

* + Testing of third-party payment gateway servers.
  + Non-functional attributes of third-party services.

1. ***Test Items***

**User Account Management**

1. User registration (email/mobile-based).
2. Login functionality (email, mobile, OTP).
3. Forgot password and password reset functionality.

**Search and Navigation**

1. Product search (keywords, categories, and filters).
2. Navigation through categories and subcategories.
3. Search suggestions and auto-complete functionality.
4. Recently viewed items and product recommendations.

**Product Management**

1. Product listing (display, sorting, and filtering).
2. Product details page (images, descriptions, specifications, pricing, and reviews).
3. Stock availability and out-of-stock notifications.

**Shopping Cart**

1. Adding/removing items to/from the cart.
2. Updating the quantity of items.
3. Display of total cost, including taxes and discounts.
4. Persistent cart functionality across sessions.

**Payments**

1. Integration with payment gateways.
2. Payment method selection (credit/debit cards, UPI, wallets, COD, EMI options).
3. Successful and failed payment scenarios.
4. Refund processing.
5. Applying promo codes, coupons, and gift cards.

**Orders and Tracking**

1. Order placement confirmation.
2. Order status updates (processing, shipped, delivered).
3. Order tracking with delivery partner details.
4. Invoice generation and download.

**Returns, Cancellations, and Refunds**

1. Requesting returns or replacements.
2. Order cancellation functionality.
3. Refund process and timelines.

**Offers and Discounts**

1. Display and application of discounts on eligible products.
2. Functionality of promo codes and coupons.
3. Bank offers and EMI discounts.

**Notifications**

1. Email and SMS notifications (order updates, promotional offers).
2. Alert messages for critical actions (e.g., payment failures).

**Customer Support**

1. Chat support functionality.
2. Help center and FAQs.
3. Raising and tracking support tickets.

**Security**

1. Secure login with session management.
2. Data encryption for sensitive information.
3. Handling of invalid or malicious inputs.
4. Prevention of unauthorized access to user accounts.

**Performance**

1. Page load times across various devices and networks.
2. Website performance under concurrent user load.
3. Checkout process response times.

**Usability**

1. Intuitive navigation and design.
2. Accessibility for users with disabilities (e.g., screen readers).
3. Readability of error messages.

**Compatibility**

1. Cross-browser compatibility (Chrome, Firefox, Safari, Edge).
2. Device compatibility (smartphones, tablets, desktops).
3. Operating system compatibility (Windows, macOS).

**Backend and API**

1. Data retrieval for product listings and details.
2. API responses for payment gateway integration.
3. Authentication and session validation via APIs.
4. ***Features to Be Tested***

**Core Functionalities**

1. User registration, login, and logout processes.
2. Password recovery and account management.
3. Search functionality with filters and sorting.
4. Product browsing, details display, and recommendations.

**Shopping Cart and Checkout**

1. Adding/removing products to/from the cart.
2. Persistent shopping cart across sessions.
3. Checkout process, including address selection and payment methods.
4. Application of promo codes, discounts, and gift cards.

**Payments**

1. Payment gateway integration (credit/debit cards, UPI, wallets, etc.).
2. Handling of successful, failed, and canceled payments.
3. Refund and reversal processes for failed payments.

**Orders and Tracking**

1. Order placement and confirmation.
2. Status updates (e.g., shipped, delivered).
3. Order tracking with estimated delivery times.
4. Invoice generation.

**Post-Order Actions**

1. Order cancellation, returns, and replacements.
2. Refund process and timelines.
3. Customer support for order-related issues.

**Offers and Promotions**

1. Display and application of discounts.
2. Functionality of promo codes and bank offers.

**Notifications**

1. Email, SMS, and push notifications for order updates.
2. Alerts for promotions and critical actions.

**Security**

1. Secure login and data encryption.
2. Prevention of unauthorized access and session hijacking.
3. Handling invalid or malicious inputs.

**Performance**

1. Website and app load times under various conditions.
2. Scalability under peak traffic.

**Usability and Compatibility**

1. Cross-browser compatibility.
2. Device and OS compatibility.
3. Accessibility for users with disabilities.
4. ***Features Not to Be Tested***

**Third-Party Services**

1. Functionalities of third-party payment gateway servers (e.g., bank systems, wallets).
2. SMS/email delivery systems managed by third-party providers.

**Non-Critical Features**

1. Internal administrative tools for managing inventory or offers.
2. Analytics dashboards used internally.

**Non-Functional Attributes**

1. Detailed performance of third-party APIs beyond response validations.
2. Code-level security implementations unless exposed via vulnerabilities.

**Future Enhancements**

1. Features planned for future releases or updates.
2. New promotional campaigns or seasonal updates yet to be launched.

**Outdated Features**

1. Features or pages deprecated or planned for removal.
2. ***Approach***

**Testing Methodology**

**Manual Testing**

1. Perform functional and usability testing manually.
2. Execute test cases to verify core functionalities like registration, search, cart management, and checkout.
3. Validate UI/UX to ensure a user-friendly experience.

**Automated Testing**

1. Use automation tools like Selenium for repetitive tasks (e.g., regression testing, smoke testing).
2. Automate scenarios like login/logout, search functionality, and checkout flows to save time and increase accuracy.

**API Testing**

1. Validate backend APIs using tools like Postman or REST Assured.
2. Ensure APIs for login, search, cart updates, payment processing, and order tracking return correct data and handle edge cases.

**Performance Testing**

1. Use tools like JMeter to simulate high traffic and measure response times.
2. Conduct load and stress testing to identify bottlenecks.

**Security Testing**

1. Perform penetration testing using tools like Burp Suite or OWASP ZAP.
2. Identify vulnerabilities like SQL injection, XSS, or CSRF.
3. Test secure handling of user data and payment information.

**Compatibility Testing**

1. Use Browser Stack or Sauce Labs for cross-browser and cross-device compatibility.
2. Test on a variety of browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile).

**Usability Testing**

1. Evaluate the user interface and workflows.
2. Gather feedback from real users or stakeholders to improve accessibility and user satisfaction.

**Testing Levels**

**Unit Testing**

1. Verify individual modules (e.g., product search, cart updates) by developers before integration.

**Integration Testing**

1. Validate interactions between modules like search results integration with the cart or payment gateways with checkout.

**System Testing**

1. Test the complete website end-to-end to ensure all components work as expected.

**Regression Testing**

1. Re-execute test cases after bug fixes or feature updates to ensure existing functionality is unaffected.

**Smoke Testing**

1. Perform quick checks after every build to verify critical functionalities.

**Test Data Preparation**

1. **User Data**: Dummy user accounts for different roles (new user, returning user, etc.).
2. **Product Data**: Mock data for product listings, including stock levels, discounts, and categories.
3. **Payment Data**: Test credit card numbers, UPI IDs, and wallet credentials.
4. **Order Data**: Sample orders with varying statuses (pending, shipped, delivered).

**Test Execution Process**

1. **Test Case Prioritization**: Execute high-priority cases first (e.g., login, checkout).
2. **Defect Reporting**: Log issues in a defect-tracking tool like JIRA, with detailed steps and screenshots.
3. **Defect Retesting**: Verify fixes and close bugs after successful validation.
4. **Regression Testing**: Run regression suites after every major change or release.
5. ***Items Pass and Fail Criteria***

**Pass Criteria**

**Functionality**:

1. All functionalities (e.g., login, search, checkout) work as per the requirements.
2. No high or critical severity bugs are present in the tested item.
3. The system behaves as expected under normal and boundary conditions.

**Performance**:

1. Pages load within the acceptable time limit (e.g., <3 seconds for key pages like Home, Product Search, Checkout).
2. System maintains stability under expected load conditions.

**Compatibility**:

1. The application displays correctly and functions seamlessly on all supported browsers and devices.

**Usability**:

1. The interface is intuitive and meets usability standards (e.g., WCAG compliance for accessibility).
2. Mobile responsiveness and navigation are error-free.

**Security**:

1. No vulnerabilities (e.g., SQL injection, XSS, CSRF) are detected.
2. User data is encrypted and handled securely.

**Regression Testing**:

1. All previously tested functionality remains intact after bug fixes or feature updates.

**Fail Criteria**

**Functionality**:

1. Any feature (e.g., Add to Cart, Payment Gateway) does not perform as expected or fails under defined conditions.
2. High or critical severity bugs are found and remain unresolved.

**Performance**:

1. Page load times exceed acceptable limits under normal or high load conditions.
2. The application crashes or becomes unresponsive during performance testing.

**Compatibility**:

1. Inconsistent UI/UX or broken functionality on supported browsers or devices.
2. Features fail on specific screen sizes or resolutions.

**Usability**:

1. Key usability issues are identified, such as poor navigation, inaccessible content, or broken responsiveness.

**Security**:

1. Vulnerabilities are detected that could compromise user data or system integrity.
2. HTTPS or other encryption standards are not enforced where required.

**Regression Testing**:

1. Existing features break after updates or modifications.
2. ***Suspension Criteria***

**Critical System Failures**

1. The application becomes unstable, preventing further testing (e.g., frequent crashes, server downtime).
2. Core functionalities (e.g., login, product search, checkout) are non-operational and block further test case execution.

**Environment Issues**

1. The test environment is unavailable or malfunctioning, such as:
2. Database connectivity issues.
3. Network outages or slow response times.
4. Configuration or deployment errors in the test setup.

**Incomplete Test Data**

1. Required test data (e.g., user accounts, product catalogs, payment gateway credentials) is missing, incorrect, or corrupted.
2. Dependency on external systems (e.g., third-party APIs, payment gateways) fails or is unavailable.

**High Number of Critical Bugs**

1. A high number of critical or blocker-level defects are identified, making further testing inefficient or redundant until they are resolved.

**Scope Changes**

1. Significant changes in requirements or scope during the testing phase that render previously prepared test cases invalid.

**Resource Constraints**

Unavailability of key testing resources, such as:

* + Testing tools.
  + Human resources (e.g., QA engineers).
  + Time constraints or delays in approvals.

1. ***Test Deliverables***
2. Test case
3. Bug report
4. User Guide
5. Release Notes
6. Defect report
7. Test execution report
8. Performance testing results
9. Security audit report
10. ***Testing Task***
11. Test Planning Tasks
12. Test Case Development
13. Test Environment Setup
14. Test Execution
15. Defect Management
16. Reporting and Communication
17. Test Closure Activities
18. ***Environmental Needs***

**Operating Systems**

1. **Desktop**: Windows, macOS, Linux.
2. **Mobile**: Android, iOS.

**Browsers**

Latest versions of:

* + - Chrome
    - Firefox
    - Microsoft Edge
    - Opera

**Testing Tools**

* **Automation**: Selenium, Appium.
* **Performance**: JMeter, LoadRunner.
* **Security**: Burp Suite, OWASP ZAP.
* **API Testing**: Postman, REST Assured.
* **Defect Tracking**: JIRA, Bugzilla.
* **Compatibility Testing**: Browser Stack, Sauce Labs.

**Database**

MySQL, MongoDB, or any database used in production for testing data.

1. ***Schedule***

* **Test Planning**: Week 1
* **Test Case Design**: Weeks 2–3
* **Test Execution**: Weeks 4–6
* **Defect Fix Verification**: Week 7
* **Final Report Submission**: Week 8

1. ***Risk & Contingencies***

* Environmental Risks
* Resource Risks
* Data Risks
* Functional Risks
* Tool and Technology Risks
* Time Risks
* External Risks
* Security Risks

1. ***Approvals***

**Test Lead**:

**QA Manager**:

**Project Manager**: