TEST PLAN: TOOLSHOP QA PROJECT

Introduction

The objective of this test plan is to define the strategy for verifying the functional and non-functional requirements of the Toolshop e-commerce application. This project employs a multi-tier testing approach including manual UI testing, a consolidated Python automation suite, and backend API validation.

Test Scope

- **Manual Testing:** Execute 25 manual test cases to validate user facing features such as Search, Cart functionality, and User Registration.
- **Automation Testing**: Utilize a single Python script containing 10 automated test cases to perform smoke testing and ensure core navigation stability.
- **API Testing**: Execute 8 Postman requests to validate backend endpoints for status codes and JSON structure integrity.
- **UI/UX Requirements**: Verify the implementation of **hover effects** on the Sign In, Sign Up, and Login buttons.

Test Strategy

- **Automated Suite**: A single Python file is designed to run 10 distinct test scenarios to verify baseline application stability.
- **Manual Exploratory Testing**: Testers will perform exploratory sessions to identify UI issues, such as feedback messages during user registration.
- **Defect Management**: All identified issues are to be tracked and managed within the Jira TQP project backlog.

Acceptance Criteria

- All 8 API tests must return a **200 OK** status to pass backend validation.
- The Python automation script must execute 10/10 test cases without failure.
- Primary navigation buttons (Sign In, Sign Up, and Login) must demonstrate a visible **hover effect**.
- A minimum of 7 defects must be documented in Jira with full steps to reproduce, expected results, and actual results.

Risk Assessment & Mitigation

Risk ID	Risk Description	Impact	Mitigation Strategy
R-01	**Environment Downtime:** The Toolshop practice site may experience server outages.	High	Utilize Postman "Mock" collections or saved JSON responses to continue API validation.
R-02	**Selector Volatility:** Frequent UI updates on the platform may break Playwright automation locators.	Medium	Use resilient CSS/Data-Test selectors and implement "Auto-wait" logic in Python scripts.
R-03	**Data Persistence:** Using JavaScript for storage (non-API) may lead to data loss on browser refresh.	Medium	Design test cases to verify state immediately after execution without relying on long-term sessions.
R-04	**Scope Overrun:** Exceeding the 25 manual test cases may delay the final report delivery.	Low	Prioritize testing based on the RTM to ensure core features are tested first.

Test Deliverables

The following artifacts will be provided upon completion of the testing cycle:

- **Test Plan Document**: Outlining the strategy for Manual, Automated, and API testing.
- **Manual Test Suite**: A document containing 25 test cases with execution results.
- **Automation Suite**: One Python script containing 10 automated smoke tests.
- **API Collection**: A Postman collection featuring 8 validated endpoints.
- **Defect Log**: A Jira backlog containing 7 logged defects.