Testing Strategy for ASAPP QA Automation Challenge

1. Overview

The goal is to ensure that all critical paths in the application are tested effectively across appropriate layers. The focus is on balancing thorough testing with efficient implementation, considering the application's basic nature.

---

2. Testing Layers and Justifications

2.1 Unit Testing

- **Why?** To validate individual functions or methods, particularly in the backend. Examples include user input validation, product inventory management, or login/registration logic.
- **Scope:**
  - Validate user input (e.g., email format, password strength).
  - Ensure backend responses for edge cases, such as duplicate registrations.

2.2 Integration Testing

- **Why?** To ensure seamless interaction between frontend and backend, especially for REST API endpoints.
- **Scope:**
  - Test the API endpoints for login, registration, product retrieval, and cart updates.
  - Validate response status codes, data integrity, and error handling.

2.3 End-to-End (E2E) Testing

- **Why?** To simulate real user flows and ensure the application behaves as expected in a complete environment.
- **Scope:**
  - User registration and login.
  - Adding products to the cart and viewing cart details.
  - Confirming purchase (validation of the pop-up).
  - Logging out successfully.

---

3. Criteria for Initial Suite

3.1 Selection Principles

1. **Critical Paths:** Focus on user registration, login, cart functionality, and logout, as these are essential for the core application flow.
2. **Error Handling:** Include tests for invalid inputs (e.g., wrong credentials, empty cart purchase).
3. **Basic Coverage:** Cover a representative set of positive and negative scenarios.

3.2 Initial Test Cases

Unit Tests

1. Validate input fields for registration (e.g., email, password strength).
2. Ensure the backend returns correct responses for valid/invalid login attempts.

Integration Tests

1. Verify API response for product list retrieval.
2. Check cart API functionality (e.g., add/remove items).

E2E Tests

1. Register a new user and log in.
2. Add one product to the cart and verify the cart updates.
3. Attempt to "purchase" an empty cart and validate the error.
4. Successfully "purchase" items in the cart.
5. Log out and ensure session termination.

---

4. Next Steps for a Real Project

1. **Expand Coverage:** Add tests for edge cases and exploratory scenarios.
2. **Data Management:** Implement mock data or fixtures for predictable test runs.
3. **Performance Testing:** Measure API and UI response times under load.
4. **CI/CD Integration:** Automate test execution on code commits or merges.