**Problem Statement**

ShopEase is a scalable e-commerce platform designed to provide a seamless online shopping experience. As the application grows in complexity—integrating modules like user authentication, product listings, payment gateways, and customer reviews—ensuring software quality through effective testing becomes crucial.

Given the **complex project structure**, **limited budget**, and **tight delivery deadlines**, ShopEase requires a **balanced testing strategy** that ensures software reliability, functional correctness, and performance without overextending time or resources.

**Recommended Testing Strategy for ShopEase**

**1. Adopt a Hybrid Testing Approach**

Combine multiple testing types to achieve depth and breadth in coverage:

* **Unit Testing (Automated)**
  + Tools: JUnit (Java), Jest (JavaScript)
  + Scope: Individual components such as login, cart, product APIs
  + Frequency: Continuous integration level
* **Integration Testing (Automated + Manual)**
  + Tools: Postman, REST Assured
  + Focus on API interactions, database connectivity, and third-party service integrations (e.g., payment gateway)
* **Functional/UI Testing (Automated for core flows)**
  + Tools: Selenium, Cypress
  + Target key flows: login/signup, checkout, product search
  + Manual testing for low-priority UI or cosmetic features
* **Regression Testing (Automated)**
  + Automate test suites for high-priority and frequently used features
  + Schedule regression tests during each sprint
* **Exploratory Testing (Manual)**
  + Performed by testers to uncover edge-case bugs not covered by automated scripts
* **Performance Testing (Limited Scope due to budget)**
  + Tools: JMeter, Lighthouse
  + Focus on load time of product pages, checkout process, and response time of APIs

**2. Leverage Agile & Shift-Left Testing**

* Integrate testing into early development stages (“Shift Left”)
* Use **Test-Driven Development (TDD)** or **Behavior-Driven Development (BDD)** for key modules
* Incorporate QA involvement in sprint planning and retrospectives

**3. Efficient Resource Utilization**

* Use **Cloud-based testing platforms** (e.g., BrowserStack, Sauce Labs) to reduce infrastructure costs
* Prioritize test cases using **risk-based testing** to focus on critical features
* Reuse test scripts for regression and smoke testing
* Outsource low-priority manual testing tasks if necessary

**Conclusion**

The recommended testing strategy blends **automated and manual testing**, leverages **agile methodologies**, and optimizes cost and time. This approach ensures ShopEase can maintain high software quality, meet release deadlines, and operate within budget constraints.