

Test PLAN

*Product Name: Hotel Booking Website
(PHPTravels Demo)*

Prepared by: Syeda Tamanna Sheme
Date: 20 June 2025

Table of Contents

- 1. Overview**
 - 2. Scope**
 - 3. Inclusions**
 - 4. Test Environments**
 - 5. Exclusions**
 - 6. Test Strategy**
 - 7. Defect Reporting Procedure**
 - 8. Roles/Responsibilities**
 - 9. Test Schedule**
 - 10. Test Deliverables**
 - 11. Pricing**
 - 12. Entry and Exit Criteria**
 - 13. Suspension and Resumption Criteria**
 - 14. Tools**
 - 15. Risks and Mitigations**
 - 16. Approvals**
-

1. Overview

This Test Plan outlines the comprehensive testing methodology that will be applied to validate the Hotel Booking Website built on the PHPTravels demo environment. It is intended to ensure that the application meets all functional expectations and offers smooth, error-free user experience across all supported platforms. This document details the scope of testing, the approach taken, test environments used, key roles and responsibilities, timeline for execution, and the overall strategy for manual QA. The goal is to identify issues before release, improve the system's robustness, and ensure the product aligns with the end-user's needs.

2. Scope

The testing will focus on validating the frontend functionalities of the PHPTravels demo hotel booking platform. The primary emphasis will be on user journeys including hotel search, registration, booking, and payment simulation. This also includes testing responsiveness across different devices and evaluating how well the system handles expected user flows. The test will cover both common and edge-case scenarios to ensure the application performs reliably.

3. Inclusions

The following modules and functionalities are considered in-scope for testing:

- User Registration and Account Creation
- Login and Logout Workflow
- Password Recovery (Forgot Password)
- Hotel Search by City, Date, Guest Count
- Hotel Details View with Room Options
- Booking Process with Calendar and Guest Input
- Simulated Payment Gateway Handling
- Booking Confirmation and Invoice Generation
- User Profile Access and Editing
- Dashboard Features (Booking History, Cancellations)
- Navigation Usability and Responsive UI Checks

4. Test Environments

The tests will be executed across a range of desktop and mobile operating systems to ensure browser compatibility and responsive behavior:

- Windows 10: Google Chrome (latest), Mozilla Firefox
- macOS: Safari Browser
- Android OS: Chrome Mobile Browser

- iOS Devices: Safari Mobile Browser

Testing will also include various screen resolutions to assess mobile-first design implementations.

5. Exclusions

The following areas are excluded from the current testing scope:

- Administrative or Backend Panel functionalities
- Real-time payment integrations (Only simulated)
- Performance/Load/Stress Testing
- Accessibility Testing for WCAG compliance

6. Test Strategy

The overall strategy will involve phased testing through a combination of traditional test design techniques and exploratory testing. Initially, we will develop structured test scenarios and cases using Boundary Value Analysis, Equivalence Class Partitioning, Use Case Testing, and Error Guessing. Smoke Testing will be used to validate the basic functionality before moving to deeper Functional Testing, UI Testing, and Regression Testing.

We will also execute End-to-End Scenarios to simulate real-world usage patterns. Each test cycle will be followed by detailed reporting and retesting to confirm defect fixes.

7. Defect Reporting Procedure

All deviations from expected results will be documented and reported as defects with proper severity and priority classification. Every bug will be accompanied by reproduction steps, actual vs expected results, screenshots, and relevant environment details. Daily status reports summarizing new and open bugs will be shared with stakeholders. Usability concerns and observations will be captured separately.

8. Roles and Responsibilities

- **Test Manager:** Oversees the entire testing life cycle, handles escalations, and ensures alignment with project goals.
- **Test Lead:** Manages test planning, resource allocation, and documentation; ensures adherence to the strategy and coordinates daily testing efforts.
- **Test Engineers:** Execute test cases, log defects, retest fixes, document results, and contribute to exploratory testing efforts.

9. Test Schedule

- Test Plan Development: 20-06-2025 to 22-06-2025
- Test Case Design: 23-06-2025 to 25-06-2025
- Execution of Test Cases: 26-06-2025 to 28-06-2025
- Compilation of Reports and Summary Submission: 30-06-2025

10. Test Deliverables

- Test Plan Document outlining scope, strategy, and schedule
- Detailed Functional Test Cases
- Defect Log with status tracking
- Daily Status Reports
- Final Test Summary and Recommendations

11. Pricing

This project is conducted purely for educational and academic purposes. No commercial pricing or billing is associated.

12. Entry and Exit Criteria

- **Entry Criteria:**
 - Requirements are clearly defined and reviewed
 - Test environment setup is complete and accessible
 - Test Plan document is approved
- **Exit Criteria:**
 - All planned test cases are executed
 - Major and critical defects are resolved
 - Test Summary Report is approved and signed off

13. Suspension and Resumption Criteria

Testing may be suspended if critical issues block core workflows or the environment becomes unavailable. Testing will resume only after the issues are addressed or alternate solutions are provided. Stakeholders will be informed of delays and resumption timelines.

14. Tools

- Microsoft Excel / Google Sheets for maintaining test cases
- Snipping Tool for capturing screenshots of defects
- Google Drive or Jira for defect tracking and collaboration

- Google Docs for reporting and shared documentation

15. Risks and Mitigations

- Site Downtime: Schedule tests in flexible windows or switch to another task while waiting.
- Limited Access/Credentials: Request demo login or contact PHPTravels support.
- Delays in Test Execution: Allocate parallel testers or extend testing period.
- Unstable Builds: Postpone regression testing until stable version is released.

16. Approvals

All formal documentation including the Test Plan, Test Cases, Bug Reports, and Final Summary must be approved by the project supervisor or QA lead. Testing phases cannot advance without required approvals to ensure accountability and completeness.