Project Test Plan and Test Management
For PHPtravel Website

Version 9.0

Prepared By:

Ziad Gamal
Omar Talaat
Ibrahim Samir
Rodina Ahmed
Rewan Abdelazim
Basmala Abouzaid

Supervisor:

Eng.Abdelrahman Mohamed Akl

Table of Context:

1) Project Overview

❖ Introduction

PHPtravels website is an e-commerce application that designed for the online travel business, providing comprehensive tools for hotel bookings, and flight reservations. This test plan describes the testing approach and overall framework that will drive the testing of the PHPtravels Version 9.0, and to define the strategy, scope, objectives, schedule, and resources required to test the PHPTravels website.

❖ Test Objective

1. **Evaluating Work Products:**
   Assess and review all relevant work products, including user guides, to ensure they meet the specified quality standards and align with the project's objectives.
2. **Causing Failures and Finding Defects:**
   Intentionally test the system to cause failures and uncover defects in the PHPTravels website.
3. **Reducing the Risk Level of Inadequate Software Quality:**
   Identify potential risks associated with software quality by performing comprehensive testing, thus enhancing reliability of the website.
4. **Verifying Core Functionalities:**
   Verify the core functionalities of the PHPTravels website, ensuring that essential features such as booking, and user authentication work correctly and efficiently.
5. **Performance, and Usability Aspects:**
   performance testing to check responsiveness and stability under different loads, and usability testing to ensure an intuitive and user-friendly interface.
6. **Identifying and Reporting Defects for Resolution:**
   Document identified defects with detailed descriptions, reproducibility steps, and severity levels to facilitate effective resolution by the development team.

❖ Reference Material

The following reference materials were used in the development of the Master Test Plan for PHPtravels website.

- International Software Testing Qualification Board (ISTQB) – Standard glossary of terms used in Software Testing Version 4.0.1 (September 15, 2024)
- IEEE 829-2008 – IEEE Standard for Software and System Test Documentation
- IEEE Standard 1012-2012: System and Software Verification and Validation
- ISO/IEC 16085-2006: System and Software Engineering – Lifecycle Processes – Risk Management
- ISO/IEC/IEEE 29119-1:2013 Software and Systems Engineering – Software Testing Part 1: Concepts and Definitions
- ISO/IEC/IEEE 29119-2:2013 Software and Systems Engineering – Software Testing Part 2: Test Processes.

2) Test Context

❖ Test Scope

1) Functional Testing

- User Registration & Login

  Validate user sign-up with valid and invalid credentials
  Verify login functionality with valid and invalid credentials.
  Check password reset and email verification.

- Booking System

  Test search functionality for flights, hotels, and tours.
  Verify booking flow (selecting a service, entering details, and confirming)
  Check for confirmation emails after booking.

- User Dashboard

  Verify dashboard navigation and accessibility.
  Check past bookings, upcoming trips, and cancellations.
  Validate user profile updates (name, email, phone, etc)

- Admin Panel

  Verify admin login and dashboard features.
  Check the ability to manage users, bookings, and payments
  Validate reports and analytics functionality.

2) UI/UX Testing

Check responsiveness on different devices (mobile, tablet, desktop)
Verify consistency in fonts, buttons, and color schemes.
Ensure proper navigation flow across pages.

3) Compatibility Testing

Check functionality across different browsers (Chrome, Firefox, Safari, Edge.
Verify behavior on different operating systems (Windows, macOS, Android, iOS).

4) API Testing

Verify API responses for search, booking.
Check response time and data validation.

❖ Software Risks

1. Technical Risks

- Performance issues causing slow page loads and delayed responses.
- Compatibility issues with different browsers and devices.
- Payment failures due to third-party integrations.

2. Management Risks

- Changes in project requirements affecting timelines.
- Limited resources such as developers, testers, or infrastructure.
- Delays in release schedules impacting business operations.

3. Operational Risks

- System downtime due to server failures.
- Incorrect booking details affecting users' travel plans.
- User errors leading to payment failures or incorrect reservations

---

3) Test Plan and Test Strategy

1) Test Design Approach

- **Equivalence Partitioning**: Categorize different booking scenarios such as valid and invalid user inputs.
- **Boundary Value Analysis**: Test edge cases such as booking limits per hotel, maximum/minimum stay durations.
- **Decision Table Testing**: Evaluate different booking combinations, including special offers and discounts.
- **State Transition Testing**: Verify system behavior during transitions such as pending, confirmed, or canceled bookings.
- **Error Guessing**: Identify common failure points such as double bookings, payment failures, and incorrect charges.

2) Defect Management

To ensure efficient tracking and resolution of defects, we implement a structured **Defect Management Process** using **JIRA**.

A. Defect Lifecycle:

1. **Defect Identification:** Testers identify defects during manual and automated testing.
2. **Defect Logging:** Defects are logged in JIRA with details such as:
    - Issue Summary
    - Steps to Reproduce
    - Expected vs. Actual Results
    - Severity & Priority
    - Screenshots or Logs (if applicable)
3. **Defect Assignment:** Assigned to the relevant developer for fixing.
4. **Defect Fixing:** Developer analyzes and fixes the issue.
5. **Retesting:** Tester verifies the fix using Selenium scripts and manual testing.
6. **Defect Closure:** If the issue is resolved, it is marked as **Closed** in JIRA; otherwise, it is **Reopened**.

B. Defect Reporting & Tracking:

- **JIRA Dashboard:** Used to track defect trends and severity distribution.
- **Defect Reports:** Weekly reports generated to monitor defect resolution progress.
- **Priority Levels:**
    - **Critical:** Blocks major functionalities (e.g., payment failure)
    - **High:** Affects key features but has a workaround
    - **Medium:** Minor functionality issues
    - **Low:** UI/UX inconsistencies

C. Key Performance Metrics:

- **Defect Density:** Number of defects per module
- **Defect Fix Rate:** Percentage of fixed defects compared to logged defects
- **Defect Retest Time:** Time taken to verify defect fixes
- **Defect Closure Rate:** Ratio of closed defects vs. open defects

3) Test Data Requirement

- Use realistic hotel booking data, including different room types and pricing.
- Ensure test cases cover peak and off-peak seasons.
- Utilize automated tools for test data generation if required.

4) Test Environment Requirement

A. Hardware Requirements:

- **Operating System:** Windows 11
- **Processor:** Intel Core i5 / i7 (or equivalent)
- **RAM:** Minimum 8GB (Recommended: 16GB)
- **Storage:** At least 50GB free space for logs, reports, and automation scripts

B. Software Requirements:

- **Testing Tools:** Selenium WebDriver for automation testing
- **Bug Tracking System:** JIRA for defect management
- **Browsers for Testing:**
  - Google Chrome (Latest Version)
  - Mozilla Firefox (Latest Version)
  - Microsoft Edge (Latest Version)
- **Database:** MySQL (as used by PHPTRAVELS)
- **Programming Language for Test Automation:** Java (used with Selenium)

C. Network Requirements:

- Stable internet connection (minimum 10 Mbps)
- Access to the staging and production environments for testing
- Proper firewall settings to allow communication with test servers

D. Test Execution Environment:

- **Manual Testing:** Conducted on Windows 11 with real browsers
- **Automation Testing:** Executed using Selenium WebDriver with test scripts written in Java/Python
- **Test Case Management:** Managed through JIRA

5) Test Tools

1. Automation & Functional Testing
    - **Selenium –** for testing web applications across different browsers.
    - **TestNG** – Works well with Selenium for structured test execution and reporting.
2. API Testing
    - **Postman –** Helps in sending requests (GET, POST, PUT, DELETE) and checking responses.
3. Performance Testing
    - **JMeter** – To test website load and performance.
4. Reports
    - **Excel**- for Test cases and bug report
5. Task management
    - **Trello –** for organizing work in a visual way.

6) Automation Plan

This plan aims to detect the website's reliability, functionality, and performance through automated testing. Automated testing tests the efficiency and accuracy by running tests faster, eliminating human errors, and allowing test reuse and comparison of actual outcomes with predicted outcomes. It also supports continuous integration, scalability, and detailed reporting for analysis.

1. Scope of Automation Testing
    - User Registration
    - User Login
    - Search for Flights, Hotels, Tours, Transfer Cars, and Visa
    - Booking Process (Flights, Hotels, Tours, Visa, Transfer Cars)
    - User Profile Management
    - Logout Functionality
    - Performance Testing
2. Test Script Design
    - Automation scripts will be developed using Selenium with Java.

3. Test Execution

- The test scripts will be executed regularly to verify the stability and performance of the website.
- Test Execution Frequency: Weekly

4. Conclusion

- This automation plan provides a road-map for the quality of the Phptravels website through automated testing. Continuously testing the platform's robustness, reliability, and user-friendliness throughout the testing process.

---

4) Test Management

Test Management is the planning, estimating, monitoring, and controlling of test activities, typically carried out by the Test Manager Team. For **Phptravels**, test management and administrative responsibilities for test execution will be performed by the tester team.

❖ Administration

- Responsible for overall test management and strategy.
- Plans, monitors, and controls all testing activities.
- Prepares test plans, estimates resources, schedules testing, and tracks progress.
- Oversees defect management and ensures testing objectives are met.

❖ Schedules

The tester team are the responsible for all this test activity.

Activity	Start Date	Deadline
Test Plan	3/14/2025	3/21/2025
Define requirements	3/14/2025	3/21/2025
Prioritizing Test Case For Automation	3/7/2025	3/21/2025
Environment Setup	3/22/2025	3/24/2025
Test Script and Execution	3/24/2025	4/11/2025
Test Analysis and Reporting	4/12/2025	4/25/2025