

# Test Summary Report

## Project Information

**Project Name:** PHPTravels Web Application – end to end manual testing

**Application URL:** <https://www.phptravels.net/>

**Test Type:** Manual Functional Testing

**Prepared By:** Smit Karanjia

**Version:** 1.0

**Reviewed By:** Self-reviewed

---

## 1. Objective

The objective of this testing project was to evaluate the functional correctness, user-facing functional behavior, and end-to-end business flows of the PHPTravels web application. This report summarizes the test execution results, key defects identified, risk areas, and overall quality assessment of the application based on manual functional testing.

---

## 2. Scope of Testing

### In Scope

The following modules and functionalities were covered as part of this testing cycle:

- User Sign Up
- User Login
- Forgot Password
- User Logout
- Flight Search
- Flight Search Results
- Filters and Modify Search functionality
- Flight Booking Flow (Guest and Logged-in User)
- Payment Gateway Redirection
- Booking Confirmation and Invoice

- Booking Cancellation Request
- Dashboard
- My Bookings
- User Profile Management
- Header Navigation
- Footer Navigation

## Out of Scope

The following testing activities were not performed in this test cycle:

- Hotel booking module
  - Tours booking module
  - Cars booking module
  - Visa module
  - Agent login and agent-specific functionalities
  - Supplier and vendor functionalities
  - Admin panel and backend operations
  - Backend logic and database validation
  - Performance, load, and stress testing
  - Security and penetration testing
  - Payment gateway transaction processing (only redirection tested)
  - Email delivery validation where environment or configuration limitations exist
- 

## 3. Test Approach

Manual functional testing was conducted based on defined requirements, test scenarios, and test cases. Testing focused on validating core user workflows and expected system behavior across the application.

### Test Design and Execution Strategy

Aspect	Description
Basis of Testing	Business requirements, UI behavior, and functional workflows
Test Design Techniques	Requirement-based testing, positive & negative testing
Execution Method	Manual execution with Pass / Fail / Not Executed status
Traceability	RTM used to ensure full requirement coverage

Defect Handling	Defects logged for failed test cases with severity and priority
-----------------	---

---

## 4. Test Results Summary

### Overall Test Execution Status

All planned test scenarios and associated test cases were executed for the defined scope. Test execution status was tracked at the test case level.

Execution Status	Description
Pass	Functionality worked as expected
Fail	Functionality deviated from expected result and defect was raised
Not Executed	Test case could not be executed due to dependency or limitation

### Module-wise Test Execution Summary

Module	Total Test Cases	Passed	Failed	Not Executed
Authentication	60	32	10	18
Flights	75	54	10	11
Booking	77	65	6	6
Payment	21	16	3	2
User Account(Dashboard & Profile)	67	63	3	1
UI & Navigation (Header, Footer & Home page)	76	75	1	0
<b>Total</b>	<b>376</b>	<b>305</b>	<b>33</b>	<b>38</b>

---

## 5. Key Findings and Issues

The following table summarizes some of the key defects identified during testing:

Findin g ID	Related Bug IDs	Module	Key Issue Description	Severit y	Priorit y	Status
KF_01	BUG_004, BUG_007	Signup	Critical validation and compliance gaps in Signup flow including weak password enforcement and missing Terms & Policy acceptance.	High	P1	Open
KF_02	BUG_011, BUG_012, BUG_013, BUG_014	Flight Search	Core flight search allows invalid inputs and violates passenger business rules, leading to unreliable search behavior.	Medium	P2	Open
KF_03	BUG_018, BUG_019, BUG_020	Booking	Booking flow integrity issues including invalid traveler input, session timer inconsistency, and unauthorized page access via direct URLs.	High	P1	Open
KF_04	BUG_021, BUG_022, BUG_023	Payment / Invoice	Payment control weaknesses such as missing countdown timer and unrestricted access to invoice pages	High	P1	Open

			without successful payment.			
KF_05	BUG_024, BUG_025	Booking Confirmation / Invoice	Unauthorized access to booking confirmation and invoice pages without valid booking reference, posing data exposure risk.	High	P1	Open
KF_06	BUG_026	Booking Cancellation	Booking lifecycle state is not updated immediately after cancellation request submission, causing user confusion.	High	P1	Open
KF_07	BUG_027	User Account (Dashboard)	Wallet balance does not reflect selected currency, resulting in incorrect financial information display.	High	P1	Open
KF_08	BUG_028	User Profile	Lack of consistent validation for profile updates allows invalid user data to be saved.	High	P1	Open

(Note: Multiple failed test cases are linked to the same underlying defects.)

## 6. Defect Summary

### Defect Summary by Severity

Severity	Description	Count
Critical	Blocks core business flow	0
High	Major functionality impacted	7
Medium	Partial functionality impacted	1

Low	UI / cosmetic issues	0
<b>Total</b>		8

---

## 7. Test Coverage

Functional testing achieved complete coverage of all in-scope requirements and modules as mapped in the Requirements Traceability Matrix (RTM). Core business flows such as flight search, booking, payment redirection, and confirmation were thoroughly validated.

---

## 8. Risk Assessment

Risk	Impact	Mitigation
Payment flow issues	Revenue loss and booking failures	Fix critical defects and re-test payment flow
Booking validation issues	Incorrect bookings or user confusion	Improve input validation and error handling
UI navigation issues	Poor user experience	Fix broken links and UI inconsistencies

---

## 9. Conclusion

Based on the current test execution and defect analysis, the PHPTravels application is functionally test-complete for the defined scope. However, the application is recommended for release only after resolving all High severity defects.

---

## 10. Next Steps

- Defect fixing by development team
- Regression testing post-fix
- Optional expansion to automation and API testing in future cycles