

# Requirement Document for Testing of Phptravels Website

Version 9.0

Prepared By:

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

Supervisor:

Eng.Abdelrahman Mohamed Akl

# Content:

1. Master Page(PHPtravels.com)
2. Login Page (Agent & Customer)
3. Signup Page Agent
4. Signup Page Customer
5. Login Page (Admin & Suplier)
6. Admin Dashboard
7. Agent Profile
8. Customer Profile
9. Suplier Dashboard

---

## 1) Master page

### Functional Requirements:

#### Language Selection:

1. Ensure switching between languages (e.g., English) works correctly.
2. Verify if the language change reflects throughout the platform, not just the homepage.

#### Currency Selection:

1. Validate that changing the currency (e.g., USD) affects the displayed prices correctly across the platform.

#### User Type Selection:

1. Check if selecting between “Agents” and “Customer” modifies the user interface or accessible functionalities appropriately.

#### Flight Search Feature:

1. Test “One Way” and “Economy” options work as intended.
2. Verify that clicking “Flying From” and “Destination To” allows selecting locations correctly.
3. Ensure date selection via the calendar is functional and reflects accurately in the input box.
4. Test the “Travelers” field to allow changing the number of travelers and updating the displayed value.
5. Ensure the search button works and triggers the expected search results page.

#### Navigation Tabs:

1. Test that tabs (Flights, Hotels, Tours, Cars, Visa) are clickable and lead to appropriate sections or pages.
2. Ensure visual feedback when a tab is selected (highlighting, border, etc.).

### **Date Selection:**

1. Validate the calendar input format (e.g., 24-03-2025).
2. Ensure date selection works for various future and past dates.

### **Search Functionality**

1. Test the functionality of the search button after providing valid and invalid inputs.
2. Verify that clicking the search button without providing mandatory fields shows appropriate error messages.

### **Responsiveness:**

1. Test the layout on various screen sizes and devices (e.g., desktop, tablet, mobile).

### **Featured Flights Section:**

- Ensure all flight offers are clickable and lead to relevant flight details pages.
- Validate the airline names, routes, and prices are displayed correctly and updated as expected.
- Test the sorting and filtering of flight offers (if applicable).
- Ensure the "From USD" prices are displayed with correct formatting and currency as per user selection.

### **Featured Hotels Section:**

- Validate the hotel offers display correct details (name, location, price per night, rating, etc.).
- Check if clicking on a hotel redirects to the correct details page.
- Test the sorting and filtering of hotel offers (if applicable).
- Ensure ratings are displayed properly (e.g., out of 5 stars).
- Test the price formatting consistency with the selected currency (USD).
- "View More" Button:
  - a. Validate the button redirects to a comprehensive listing of hotels.
  - b. Ensure the text is clickable and visually highlighted when hovered.

### **Popular Tours Section:**

- Validate each tour card displays the correct name, location, and rating.
- Ensure clicking on a tour redirects to the relevant tour details page.
- Test the filtering and sorting options if available for tours.
- Check the "Limited budget?" prompt and verify the "**View More**" button redirects to budget-friendly options.

### **Recommended Transfer Cars Section:**

- Validate each car card displays the correct model, type, price, rating, and location.
- Ensure clicking on the "**Book Now**" button initiates the booking process correctly.
- Test the sorting and filtering options if available for transfer cars.
- Verify the "**View More**" button for discovering more cars is working correctly.

### **Promotional Cards Section (Top):**

- Validate each card displays the correct title, description, and icon.
- Ensure clickable elements (if applicable) respond correctly.

### **Mobile App Promotion Section (Blue Area):**

- Verify the **Play Store** and **App Store** buttons redirect to the correct download links.
- Check if the description text is clear and provides correct information about the app features.

### **Footer Section:**

- Test all links under **About Us, Privacy Policy, Contact Us, etc.** to ensure they redirect to the intended pages.
- Validate the contact details: **Phone number, email, and Contact Us link** are functional.
- Ensure the **Newsletter Signup Form** works correctly:
  - Accepts valid email addresses.
  - Provides appropriate feedback messages for successful or unsuccessful subscription attempts.

## **Non-Functional Requirements:**

### **Performance:**

1. Ensure fast loading times for the homepage and search results.
2. Ensure all images load quickly and correctly.
3. Test the loading time for both Featured Flights and Hotels sections.
4. Validate the smoothness of the transition when navigating between sections.

### **Usability:**

1. Check that the interface is user-friendly and intuitive.
2. Validate text readability and color contrast.
3. Check the responsiveness of both sections on different screen sizes (desktop, mobile, tablet).
4. Validate if the display layout is clear and visually appealing.
5. Validate tooltips or hints appear when hovering over buttons or car cards (if applicable).
6. Check that the layout is responsive across different devices.
7. Ensure the newsletter subscription process is user-friendly and straightforward.

### **Accessibility:**

1. Verify that the page is accessible for users with disabilities (e.g., keyboard navigation, screen reader compatibility).
2. Ensure that all buttons and links are accessible via keyboard navigation.
3. Validate that all text content is readable by screen readers.

### **Visual Consistency:**

1. Check for consistent fonts, button styles, and alignments across all elements.
2. Ensure all cards are styled consistently (fonts, buttons, images, etc.).
3. Test that highlighted items (e.g., selected flights) appear visually distinct.

---

## 2) Login Page(Customer and Agent)

### Functional Requirements:

#### Login Form Inputs:

1. Validate the **Email Address** field accepts only valid email formats (e.g., test@example.com).
2. Validate the **Password** field accepts a variety of characters (letters, numbers, special symbols).
3. Check if the **Remember Me** checkbox retains the user's login credentials if selected.

#### Buttons Functionality:

1. Verify the **Login** button triggers the authentication process and provides relevant messages for:
  - a. Successful login.
  - b. Incorrect email/password.
  - c. Server errors (if applicable).
2. Ensure the **Signup** button redirects to the registration page.
3. Validate the **Reset Password** link redirects to the password recovery page.

### Non-Functional Requirements:

1. **Performance:** Ensure the login process completes within an acceptable timeframe (e.g., under 2 seconds).
2. **Usability:** Check if the form is clearly labeled, accessible, and intuitive to use.
3. **Compatibility:** Test across different browsers and devices.
4. **Accessibility:** Ensure compatibility with screen readers and keyboard navigation

---

## 3) Sign-up for Customer Account

### Functional Requirements:

#### Form Inputs:

1. Validate the **First Name** and **Last Name** fields accept text inputs without special characters or numbers.
2. Ensure the **Select Country** dropdown contains valid country options.
3. Validate the **Phone** field:
  1. Accepts only valid phone numbers.
  2. Provides appropriate format validation (e.g., country code).
4. Validate the **Email Address** field:
  1. Accepts only valid email formats (e.g., name@example.com).
  2. Ensures uniqueness (check if the email is already registered).
5. Validate the **Password** field:
  1. Meets complexity requirements (e.g., length, special characters, numbers, upper/lowercase letters).

### Captcha Verification:

6. Ensure the **hCaptcha** must be successfully completed before submission.

### Terms and Policy Agreement:

7. Ensure the user cannot proceed without checking the agreement box.

### Signup Button:

1. Ensure the **Signup** button triggers the registration process and provides feedback for:
  1. Successful registration.
  2. Error messages for invalid inputs or missing fields.

### Non-Functional Requirements:

1. **Performance:** Ensure registration process completes within acceptable time (e.g., under 3 seconds).
2. **Usability:** Ensure form labels and placeholders are clear and user-friendly.
3. **Compatibility:** Test across different devices and browsers.
4. **Accessibility:** Ensure compatibility with screen readers and keyboard navigation.

---

## 4) Sign-up for Agent Account

### Functional Requirements:

#### Form Inputs:

1.
  1. Validate the **First Name** and **Last Name** fields accept text inputs without special characters or numbers.
  2. Validate the **Select Country** dropdown contains valid country options.
  3. Validate the **Phone** field:
    1. Accepts only valid phone numbers.
    2. Provides appropriate format validation (e.g., country code).
  4. Validate the **Agency City** and **Agency Address** fields:
    1. Accepts valid text inputs.
    2. Limits character length appropriately.
  5. Validate the **Agency Name** field:
    1. Accepts text inputs.
    2. Restricts special characters if needed.
  6. Validate the **Agency License** field:
    1. Accepts text inputs.
    2. Checks for required format or characters (if applicable).
  7. Validate the **Email Address** field:
    1. Accepts only valid email formats.
    2. Ensures uniqueness (checks if the email is already registered).
  8. Validate the **Password** field:

1. Meets complexity requirements (e.g., length, special characters, numbers, upper/lowercase letters).

### **Business Logo Upload:**

9. Allows only image file uploads (e.g., .jpg, .png).
10. Validates the maximum file size (1 MB).
11. Provides proper error messages for unsupported file types or excessive size.

### **Captcha Verification:**

12. Ensure the **hCaptcha** must be successfully completed before submission.

### **Terms and Policy Agreement:**

13. Ensure the user cannot proceed without checking the agreement box.

### **Signup Button:**

14. Ensure the **Signup** button triggers the registration process and provides feedback for:
  1. Successful registration.
  2. Error messages for invalid inputs or missing fields.

### **Non-Functional Requirements:**

15. **Performance:** Ensure registration process completes within acceptable time (e.g., under 3 seconds).
16. **Usability:** Ensure form labels and placeholders are clear and user-friendly.
17. **Compatibility:** Test across different devices and browsers.
18. **Accessibility:** Ensure compatibility with screen readers and keyboard navigation.

---

## **5) Login page (Admin and Suplier)**

### **Functional Requirements:**

#### **Form Inputs:**

1. Validate the **Email** field:
  1. Accepts only valid email formats (e.g., user@example.com).
  2. Shows error messages for incorrect formats or empty inputs.
2. Validate the **Password** field:
  1. Accepts text inputs.
  2. Ensures the password input is masked (hidden characters).

#### **Language Selection:**

3. Validate the **Language Dropdown:**
  1. Displays all available language options.

2. Changes the interface language accordingly.

### **Remember Me Checkbox:**

4. Verify that selecting this option retains user login information on future visits.
5. Validate that deselecting it clears the saved login information.

### **Forgot Password Link:**

1. Ensure clicking the link redirects to the **Password Recovery Page**.
2. Validate that the **Password Recovery Page** provides instructions for resetting the password.

### **Login Button:**

1. Ensure clicking the button triggers the login process.
2. Validate feedback for:
  1. Successful login (redirect to the user dashboard or homepage).
  2. Incorrect email or password (error message is shown).

### **Non-Functional Requirements:**

1. **Performance:** Ensure the login process completes within acceptable time (e.g., under 2 seconds).
2. **Usability:** Ensure labels and placeholders are clear and user-friendly.
3. **Compatibility:** Test across different devices and browsers.
4. **Accessibility:** Ensure compatibility with screen readers and keyboard navigation

---

## **6) Admin Dashboard**

### **Functional Requirements:**

#### **Alerts Section:**

1. Display the correct number of alerts (e.g., red circle with number count).
2. Clicking on Alerts should redirect to the alerts page or dropdown.

#### **Bookings Section:**

1. Verify the dropdown reveals different booking categories (e.g., Hotels, Flights, Tours, Cars).
2. Clicking on each booking type should redirect to the appropriate page.

#### **Settings Section (Dropdown):**

1. Validate each of the following:
  1. **General Settings:** Modify and save settings successfully.
  2. **Users Roles:** Create, edit, delete, and view user roles.
  3. **Payment Gateways:** Add, edit, delete, and configure payment gateways.
  4. **Currencies:** Add, edit, delete, and set default currency.
  5. **Email Settings:** Test SMTP configurations and notification templates.
  6. **Locations:** Add, edit, delete, and manage location settings.
  7. **Languages:** Add, edit, delete, and switch between languages.



8. **Countries:** Add, edit, delete, and manage country lists.

### **Modules Section:**

1. Expand the **Modules** dropdown.
2. Enable/Disable specific modules.
3. Verify module settings and configurations.

### **Markups Section:**

1. Expand the **Markups** dropdown.
2. Add, edit, delete, and view markups.
3. Verify that changes reflect in the final pricing calculations.

### **Users Section:**

1. View all users.
2. Add new users.
3. Edit and delete existing users.

### **Flights, Hotels, Tours, Cars Sections:**

1. View listings under each section.
2. Add, edit, delete entries under each category.
3. Test search and filtering functionalities.

### **Users Section (Dropdown):**

#### **All Users:**

- View all users with filtering and sorting options.
- Search for users by name, email, or role.

#### **Admin, Supplier, Employee, Customer, Agent:**

- Create new user profiles for each category.
- Edit user profiles, including details like name, email, password, role, etc.
- Delete or deactivate users and verify they cannot log in.
- Assign and modify roles and permissions correctly.

### **CMS (Content Management System):**

#### **CMS Add Page:**

- Create new pages with titles, descriptions, and content.
- Save, edit, delete, and preview pages.

#### **CMS Pages:**

- View all existing pages with filtering and sorting.
- Edit and delete individual pages.

#### **CMS Menu:**

- Create and edit menus.
- Assign pages to menus and verify navigation works as intended.

## **Blogs Section:**

### **Blogs Add Page:**

- Create new blog posts with title, content, category, tags, and images.
- Save, edit, delete, and preview blog posts.

### **Blogs Pages:**

- View all blogs with filtering and sorting options.
- Edit and delete individual blogs.

### **Blogs Category:**

- Add, edit, delete, and view blog categories.
- Ensure categories are correctly assigned to blog posts.

## **Newsletter Section:**

- Expand **Newsletter** dropdown.
- Add new newsletters and manage subscribers.
- Send newsletters and verify delivery success.
- View reports of opened emails, clicked links, etc.

---

# **7) Agent Profile**

## **1. Profile Information Page (My Profile)**

### **Functional Requirements:**

- Users should be able to **view and edit their profile information**, including:
  - First Name
  - Last Name
  - Email (Read-only)
  - Password (Hidden)
  - Country (Selectable with flag & code)
  - Phone number
  - State
  - City
  - Address 1
  - Address 2
- There should be a **"Update Profile" button** to save changes.
- Error messages should be displayed if required fields are left empty or contain invalid data.

### **Non-Functional Requirements:**

- Data should be **saved and reflected immediately** without requiring a page refresh.
- Error handling should be **clear and user-friendly**.
- Profile data should be stored **securely**, especially passwords.

## **2. Bookings Page (My Bookings)**

## Functional Requirements:

- Users should be able to **view their bookings in a table format**, including:
  - Client Name
  - Booking ID
  - PNR (Passenger Name Record)
  - Payment Status (Filterable: All, Paid, Unpaid, etc.)
  - Booking Status (Filterable: All, Confirmed, Pending, etc.)
  - Type (Filterable: All, Hotel, Flight, etc.)
  - Date (Filterable: Search by Date)
  - Price (Filterable: Search by Price)
- Pagination should be present for navigating multiple entries.
- A search bar should be provided to **filter bookings** by name, booking ID, or PNR.
- No bookings message should be displayed when there are no records.

## Non-Functional Requirements:

- Data retrieval should be **fast and efficient**, even with a large number of bookings.
- The table should be **responsive and accessible** on all devices.

## 3. Navigation Bar

### Functional Requirements:

- The navigation bar should allow the user to **switch between different sections**, including:
  - Dashboard
  - My Bookings
  - My Profile
- Users should be able to **change language and currency** using the respective dropdowns.
- Users should be able to **view their profile name** and log out if needed.

### Non-Functional Requirements:

- The navigation bar should be **consistent and accessible** across all pages.
- It should be **responsive and work properly on all devices**.

---

## 8) Customer Profile

### Profile Information:

- User must enter:
  - First Name
  - Last Name
  - Email (e.g., agent@phptravels.com)
  - Phone Number (e.g., +1 1204567)
  - Country, State, City
  - Address 1 and Address 2
- User should be able to **update their profile**.

### Agency Profile:

- User must provide:
  - Agency Name (e.g., Paragon Travels)
  - Agency License (e.g., 123456)
  - Agency City (e.g., New Jar)
  - Agency Address (e.g., 64 Street Next bridge Miggle Fornia)
- Option to **upload and replace agency logo**.

### Transactions:

- User can view:
  - Transaction ID
  - Date
  - Amount
  - Currency
  - Payment Method
  - Status
  - Details of transactions

### Markups Management:

- User can **configure markup percentages** for different travel services:
  - Hotels: 10%
  - Flights: 5%
  - Tours: 2%
  - Cars: 8%

### Bookings Management:

- Search and filter options:
  - By Client Name, Booking ID, PNR, Date, Price
  - By Booking Type and Payment Type
- Displayed data includes:
  - Markup Price
  - Net Profit
- The table currently shows no data.

### General Navigation:

- Includes tabs for:
  - Dashboard
  - Bookings
  - Markups
  - Deposit
  - Agency Profile / My Profile

---

## 9) Suplier Dashboard

### Main Sections:

1. **Dashboard:** Main overview page for the user.

2. **Bookings:** Section for managing bookings.
3. **Hotels:**
  - Dropdown option: Hotels (Manage or view hotel-related data)
4. **Tours:**
  - Dropdown option: Tours (Manage or view tour-related data)
5. **Cars:**
  - Dropdown option: Cars (Manage or view car-related data)
6. **Reports:**
  - Possibly has sub-options to generate or view different reports.

### **User Profile Section (Bottom Menu):**

- **Dashboard:** Takes the user back to the main dashboard.
- **Profile:** Access or edit user profile.
- **Logout:** Log out of the platform.
- **User Type:** Shown as "**Demo Supplier**".