**Case Study: PHP Travels – Customer Registration & Flight Search**

**Application Under Test (AUT):** <https://www.phptravels.net/>

**1. Test Strategy**

**Framework Approach**

* **Cucumber BDD Model:** Write test cases in Gherkin syntax for better collaboration with stakeholders.
* **Page Object Model (POM):** Maintain reusable and modular page classes.
* **TestNG Integration:** XML suite execution with parallel/multi-browser support.
* **Data Driven Testing:** Input values read from **Excel (Apache POI)** or **Cucumber Examples/Table**.
* **Modular Framework:** Separate modules for Login, Registration, Flight Search, Booking, Payment.

**Other Strategy**

* **Cross-Browser Execution:** Chrome, Firefox, Edge.
* **Coding Standards:** Use consistent naming, proper logging, exception handling.
* **Screenshots:** Capture at each validation step (pass/fail).
* **Reports:** Generate Extent Reports + TestNG Reports with screenshots.
* **Property Reader Utility:** All locators maintained in .properties file.

**2. Case Study 1: Customer Registration Business Flow**

**Business Flow Objective**

Verify that new customers can successfully **register an account** and handle error validations for invalid inputs.

**Activity 1 – Main Flow (Valid Data Registration)**

1. Navigate to [**https://www.phptravels.net/home**](https://www.phptravels.net/home).
2. Click **My Account → Sign Up**.
3. On Sign Up Page:
   * Enter *First Name* and *Last Name*.
   * Enter *Mobile Number*.
   * Enter *Email Address*.
   * Enter *Password* and *Confirm Password*.
4. Click **Sign Up**.
5. Verify account is created successfully and confirmation is shown.

**Automation Notes:**

* All locators read from property file.
* All test data read from Excel file.
* Exception handling with custom error messages.
* Generate console report + Extent report at each step.

**Activity 2 – Error Flow (Invalid Data Scenarios)**

1. Enter *First Name* and *Last Name* only → Click Sign Up → Validate alert: **“Fill these out”**.
2. Enter *invalid email* (missing @ or domain) → Validate alert message.
3. Enter *weak password* (no alphanumeric/special characters) → Validate password error.
4. Enter *Phone Number >10 digits* → Validate error message.

**3. Case Study 2: Flight Search Navigation**

**Business Flow Objective**

Verify that customers can **search and book flights** using different filters, trip types, and payment methods.

**Activity 1 – Main Flow (Flight Search & Booking)**

1. **Homepage / Dashboard**
   * Navigate to PHP Travels homepage.
   * Select **Flights** tab.
2. **Select Flight Type**
   * Choose from: *One Way / Round Trip / Multi City*.
3. **Enter Travel Details**
   * *From (Departure Airport)* – select from auto-suggestion.
   * *To (Destination Airport)*.
   * *Departure Date* (and Return Date for round trip).
   * *Travel Class* (Economy / Business / First Class).
   * *Number of Passengers* (Adults, Children, Infants).
4. **Advanced Filters (Optional)**
   * Non-stop flights.
   * Preferred Airlines.
   * Price Range.
5. **Click Search**
   * System fetches flights via APIs (Amadeus, Sabre, Travelport, etc.).
6. **Search Results Page**
   * Verify: Airline Name, Logo, Departure/Arrival, Duration, Stops, Price.
   * Verify *Book Now* and *Details* buttons are visible.
7. **Select a Flight**
   * Click **Book Now**.
   * If *Round Trip* → Select both departure and return flights.
8. **Passenger Information**
   * Enter *Name, Gender, DOB, Passport Info (for international)*.
   * Enter *Contact details* (email, phone).
9. **Payment Page**
   * Select Payment Method (PayPal, Stripe, Razorpay, etc.).
   * Enter details → Confirm booking.
10. **Booking Confirmation**

* Verify success page with PNR/Booking ID.
* Verify confirmation email/SMS received.

**Activity 2 – Error Flow (Invalid Data Scenarios)**

1. Leave *Departure/Arrival city blank* → Error message.
2. Select invalid *date ranges* (e.g., past date) → Validation error.
3. Exceed max passengers (e.g., 10 adults) → Error message.
4. Invalid payment credentials → Payment declined message.
5. Network/API failure simulation → Display error handling message.

**4. Deliverables**

* **Test Plan:** Scope, objectives, environment setup, resources.
* **Test Scenarios & Test Cases:** For Registration + Flight Search.
* **Automation Scripts:** Selenium + TestNG + Cucumber + POM.
* **Test Data:** Excel sheets for registration & flight booking.
* **Reports:** Extent Reports + Execution logs + Screenshots.
* **Defect Reports:** Logged in Jira (with severity & priority).

**5. Conclusion**

* The **PHP Travels Registration** module ensures proper handling of valid and invalid customer signups.
* The **Flight Search and Booking** module validates end-to-end flow from search to confirmation.
* By using **BDD + POM + TestNG + Data Driven Testing**, the solution will be:
  + **Readable** for stakeholders.
  + **Reusable** for regression testing.
  + **Scalable** to include additional modules like Hotels, Cars, Tours, etc.