

## Automating Booking Process on PHPTravels

### Task 1: Navigate to the PHPTravels Website

Write a test that:

- Write a test that navigates to the PHPTravels website (<https://www.phptravels.net/>)

### Task 2: Search for Hotels

Create a new test file named **hotelSearch.spec.js**.

Write a test that:

- Navigates to the PHPTravels website.
- Search for hotels in a specific location (e.g., "USA") with a specific check-in and check-out date.
- Asserts that the search results are displayed correctly.

### Task 3: Select and book a hotel

Create a new test file named **hotelBooking.spec.js**.

Write a test that:

- Navigates to the PHPTravels website.
- Searches for hotels in a specific location.
- Selects a hotel from the search results.
- Initiates the booking process and completes the booking.
- Asserts that the booking is successful.

### Task 4: User Authentication

Create a new test file named **userAuthentication.spec.js**.

Write a test that:

- Navigates to the PHPTravels website.
- Log in with valid credentials.
- Verifies that the user is successfully logged in.

### Task 5: Handle Timeouts

Create a new test file named **timeoutHandling.spec.js**.

Write a test that intentionally waits for a delayed element to appear on the page and handles the timeout gracefully.

**Submission:**

- Project file name should be "PAA\_Cypress\_UnitName\_YourName"
- Submit the Cypress project (excluding node\_modules) into ["https://github.com/SELISEdigitalplatforms/l2-selenium-testing-library/"](https://github.com/SELISEdigitalplatforms/l2-selenium-testing-library/) repo.
- Include a README file with instructions on how to run the tests.

Note: Try to explore various Cypress commands such as `cy.get()`, `cy.click()`, `cy.type()`, and assertions like `should()` and `expect()`. Emphasize the importance of waiting for elements using Cypress commands to handle asynchronous behavior on the website.