***Day 8 – Mini Project Automation***

**Web app**

**Casa Antonio** appears to be a travel and accommodation portal focused on home stays. The website provides users with a comprehensive platform for travel information, home stay options, community engagement through blogs, and customer support.

**Key Features**

**User Authentication**

* Secure login system with credential validation
* User profile management
* Session management with logout functionality

**Navigation Structure**

* Home/Index page serving as the main landing page
* Clear navigation to various sections including:
  + Login/Authentication
  + User Profile
  + Blog
  + Home Stay listings
  + About section
  + Contact page

**Blog Section**

* Travel blog with article previews and "Read More" functionality
* Popular topics section
* Newsletter subscription feature allowing users to receive updates

**Home Stay Features**

* Dedicated page for browsing home stay destinations
* Review system allowing users to:
  + View existing reviews
  + Hide reviews when desired
  + Add their own reviews

**User Engagement**

* Contact form for user inquiries and support
* Profile management for registered users

**User Experience**

The website is structured with clear navigation paths and consistent page titling:

* Casa Antonio (Index page)
* Login
* User Profile
* Travel Blog - Casa Antonio
* Home Stay Destinations
* About Us
* Contact Us

**Target Audience**

The website appears to target travelers looking for home stay accommodations, travel advice through the blog section, and a platform to share their experiences through reviews.

The testing suite demonstrates a comprehensive web application with user authentication, content browsing, and interactive features for a travel-focused audience.

test('Verify user can hide reviews of home stay page', async ({ indexPage, loginPage, homeStayPage }) => {

        await indexPage.verifyUserIsOnIndexPage();

        await indexPage.navigateToLogin();

        await loginPage.verifyUserIsOnLoginPage();

        await loginPage.login('admin', 'password');

        await indexPage.verifyUserIsLoggedInSuccessfully();

        await indexPage.userClickonHomeStay();

        await homeStayPage.verifyUserIsOnHomeStayPage();

        await homeStayPage.userCanHideReview();

    })

    test('Verify user can add a review for home stay page', async ({ indexPage, loginPage, homeStayPage }) => {

        await indexPage.verifyUserIsOnIndexPage();

        await indexPage.navigateToLogin();

        await loginPage.verifyUserIsOnLoginPage();

        await loginPage.login('admin', 'password');

        await indexPage.verifyUserIsLoggedInSuccessfully();

        await indexPage.userClickonHomeStay();

        await homeStayPage.verifyUserIsOnHomeStayPage();

        await homeStayPage.userCanAddReview('This is a test review.');

        await homeStayPage.verifyReviewisAddedSuccessfully('This is a test review.');

    })

<div id="review-modal" class="modal">

        <div class="modal-content">

            <span class="close-button">&times;</span>

            <h2 id="modal-title"></h2>

            <form id="review-form">

                <label>Rating:

                    <select id="rating" required>

                        <option value="">Select...</option>

                        <option value="1">1</option>

                        <option value="2">2</option>

                        <option value="3">3</option>

                        <option value="4">4</option>

                        <option value="5">5</option>

                    </select>

                </label>

                <label>Comment:

                    <textarea id="comment" rows="3" required></textarea>

                </label>

                <button type="submit" class="btn">Submit Review</button>

            </form>

        </div>

  // Form submission

  form.addEventListener('submit', e => {

    e.preventDefault();

    saveReview(currentIndex, +ratingInput.value, commentInput.value);

    closeModal();

    updateDisplay(currentIndex);

    // Refresh reviews pane if open

    const reviewsDiv = list.children[currentIndex].querySelector('.reviews-list');

    if (reviewsDiv.style.display === 'block') renderReviews(currentIndex);

  });

  // Open & Close Modal

  function openModal(index) {

    currentIndex = index;

    modalTitle.textContent = `Review: ${destinations[index].name}`;

    ratingInput.value = '';

    commentInput.value = '';

    modal.style.display = 'block';

  }

  function closeModal() {

    modal.style.display = 'none';

  }