

Test Plan: Manual Testing for automationexercise.com

1. Homepage & Navigation

Test Case	Description	Expected Result
TC-001	Load the homepage	Page loads with banners, products, navigation
TC-002	Navigate through menu items (Products, Cart, Signup/Login)	Each menu redirects to the correct page
TC-003	Try navigating while offline	Proper browser error message
TC-004	Click on "View Product" of an item	Redirects to product detail page
TC-005	Open site in mobile view	Site is responsive and menus are accessible
TC-006	Change language settings (if applicable)	Texts are translated or handled gracefully

2. Signup / Login

Test Case	Description	Expected Result
TC-007	Submit empty login form	Error message or validation
TC-008	Login with valid credentials	User is logged in successfully
TC-009	Login with wrong credentials	Proper error message shown
TC-010	Signup with an existing email	Error: Email already exists
TC-011	Signup with invalid email format	Error: Invalid email format
TC-012	Password field accepts only blank spaces	Error or trimmed input
TC-013	Copy/paste password into confirm field	Accepts and matches
TC-014	Session persists after refresh	User stays logged in
TC-015	Click "Logout" and try to go back using browser	Should not return to logged-in state

3. Product Page / Cart / Checkout

Test Case	Description	Expected Result
TC-016	Add product to cart	Cart updates with item
TC-017	Add the same item multiple times	Quantity updates properly
TC-018	Go to cart and remove item	Item is removed successfully
TC-019	Increase quantity in cart	Price updates accordingly
TC-020	Add product with quantity = 0 or negative	Should be blocked
TC-021	Try adding to cart without being logged in	Allowed or redirected to login
TC-022	Proceed to checkout without filling all fields	Should show validation messages
TC-023	Attempt checkout with an expired card or invalid input	Proper error/handling
TC-024	Apply coupon code (if feature exists)	Validated or error shown

4. Forms & Inputs (Contact Us, Subscription, etc.)

Test Case	Description	Expected Result
TC-025	Submit "Contact Us" form with valid data	Confirmation shown
TC-026	Submit empty "Contact Us" form	Error: Required fields
TC-027	Enter invalid characters in input fields	Validation or sanitization
TC-028	Submit email subscription with invalid email	Error message shown
TC-029	Enter script tags in input (<code><script>alert(1)</script></code>)	Escaped or rejected (XSS test)

5. User Account & Profile (Post-login)

Test Case	Description	Expected Result
TC-030	Update profile info	Changes saved successfully
TC-031	Change password	Password updated and logout enforced
TC-032	View order history	Correct data displayed
TC-033	Delete account	User is logged out and account removed
TC-034	Try accessing user pages after account deletion	Access denied or redirected

6. Exploratory & Corner Cases

Scenario	Notes
Exploratory-01	Try using site with JavaScript disabled
Exploratory-02	Use browser back/forward after form submission
Exploratory-03	Try submitting forms with extremely long inputs (e.g., 500+ chars)
Exploratory-04	Access URLs directly without login (e.g., /checkout)
Exploratory-05	Open multiple tabs and interact (e.g., cart in one, checkout in another)
Exploratory-06	Test behavior under slow network or throttling
Exploratory-07	Try uploading unsupported file types (if file upload exists)

Test Plan: Automation Testing for automationexercise.com

Scope:

Test the website for basic functionality, UI consistency, API test cases (User and Products), e2e flows (combination of UI and API) — including the shopping workflow, api calls, and forms.

1. UI - Login & Navigation

Test Case	Description	Expected Result
UI-001	Login with valid credentials	User successfully logged in
UI-002	Login with wrong credentials	UI should be display an error
UI-003	Header Navigation	Each menu redirects to the correct page

2. API tests - User & Products

Test Case	Description	Expected Result
APIUSER001	Get user account details	Status 200 or success response
APIUSER002	Verify Login with user created	POST method should return that the user exists
APIUSER003	Update User Account	PUT method should return that the user was updated
APIUSER004	Delete User Account	Account deleted
APIPROD001	Validate Products List	Product list validation fields (id, name, category)
APIPROD002	Search dress and validate total products	Search product and validate the total of products
APIPROD003	Search top and validate total products	Search product and validate the total of products
APIPROD004	Search t shirts and validate total products	Search product and validate the total of products
APIPROD005	Search jeans and validate total products	Search product and validate the total of products

3. E2E flows / API calls / UI interactions

Test Case	Description	Expected Result
E2E-001	Login with a new user and validate products	Products list validation flow
E2E-002	Login with a new user and search a product	Search a product flow
E2E-003	Login with a new user and buy a product	Buy a product flow

Notes:

- Design and implement an end-to-end test suite using a tool like Cypress to automate UI and API flows
- Website: <https://automationexercise.com>
- Page Object Model (cart, checkout, homePage, login, payment, products, etc)

- Use `fixtures` for test data
- Project Structure (e2e -> api, flows, ui)
- Clear Structure: Separate UI and API Tests
- Custom Commands (`commands.js`)
- Stable Selectors (avoid fragile CSS/text selectors)
- Use before and after hooks
- Assertions (URL, visibility, text, etc)
- Use `cy.viewport()` (iphone and tablet)

Things to improve:

- Use `cy.intercept()` to mock or spy on requests
- More negative test cases and combination of data (404, empty forms, etc)

Run results:

Spec	Tests	Passing	Failing	Pending	Skipped
✓ api/products.cy.js	00:02	5	5	-	-
✓ api/user.cy.js	00:01	4	4	-	-
✓ ui/login.cy.js	00:21	3	3	-	-
✓ flows/flows.cy.js	00:26	3	3	-	-
✓ All specs passed!	00:52	15	15	-	-

How can we run tests locally?

I create some scripts in package.json file:

```
"scripts": {
  "cypress:ui": "cypress run --spec cypress/e2e/ui/**/*.js",
  "cypress:api": "cypress run --spec cypress/e2e/api/**/*.js",
  "cypress:e2e": "cypress run --spec cypress/e2e/flows/**/*.js",
```

```
"cypress:chrome": "cypress run --browser chrome --spec  
cypress/e2e/**/*.cy.js"  
}
```

Command `npm run 'script'` → **`npm run cypress:ui`**