Behaviour based testing strategy and main description of stages

1. Home screen
Test 1 : To load the home screen successfully

2. Product details
Test 2 : To select a product by keyword “Sleeve” and scroll to see it
Test 3: To View the details so we can be able to go to the cart

3. Cart & cart summary
Test 4 : After adding the product go and click to see the cart summary
Test 5: Once confirmed the purchase the system shall bring to a account creation

4. Account creation & sign in
Test 6: Entering the Name, email and hit the button to proceed to next screen

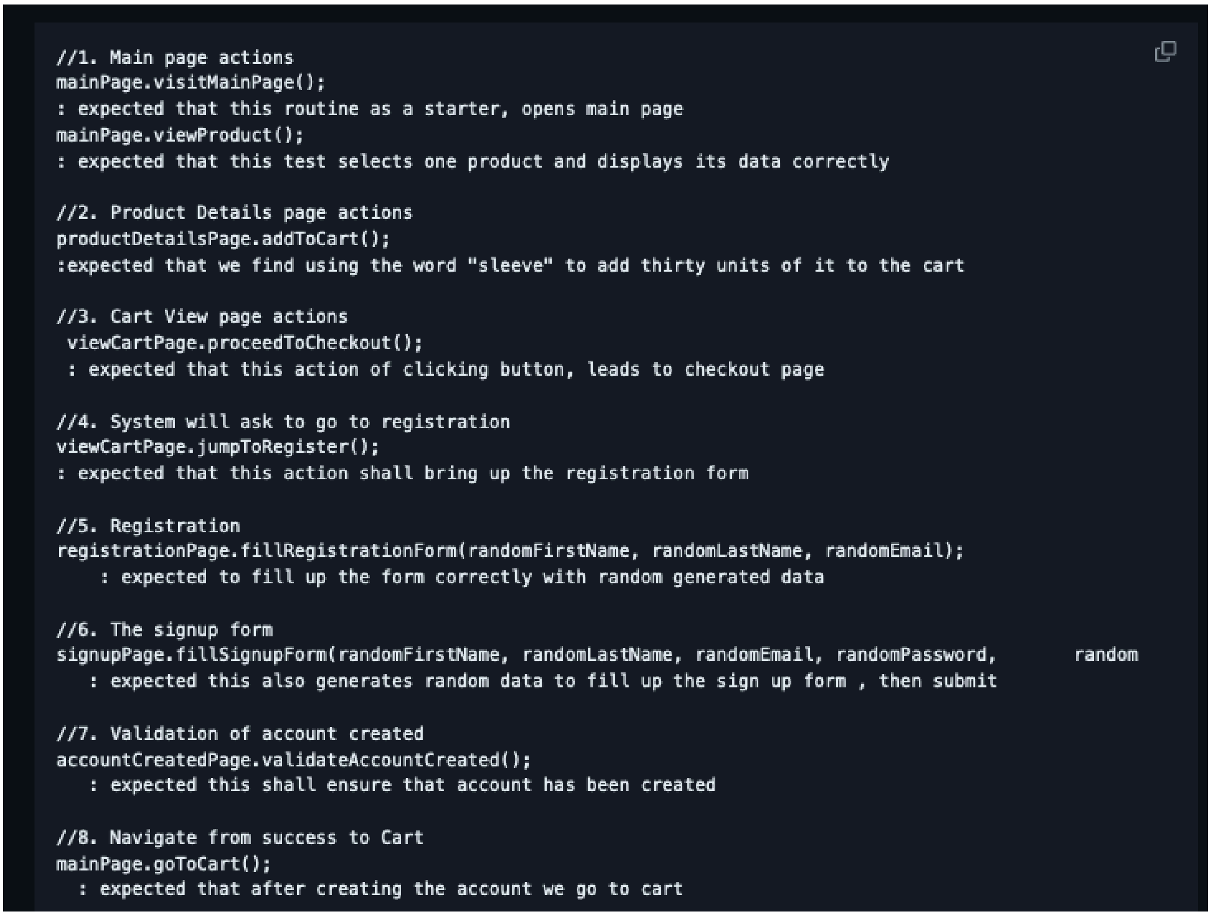
5. Address
Test 7: Entering the address information with random data and saving the form

6. PaymentUser
Test 8: Entering the Payment details shall use random data to complete the form
Test 9: Once the data is entered we go to order confirmation page

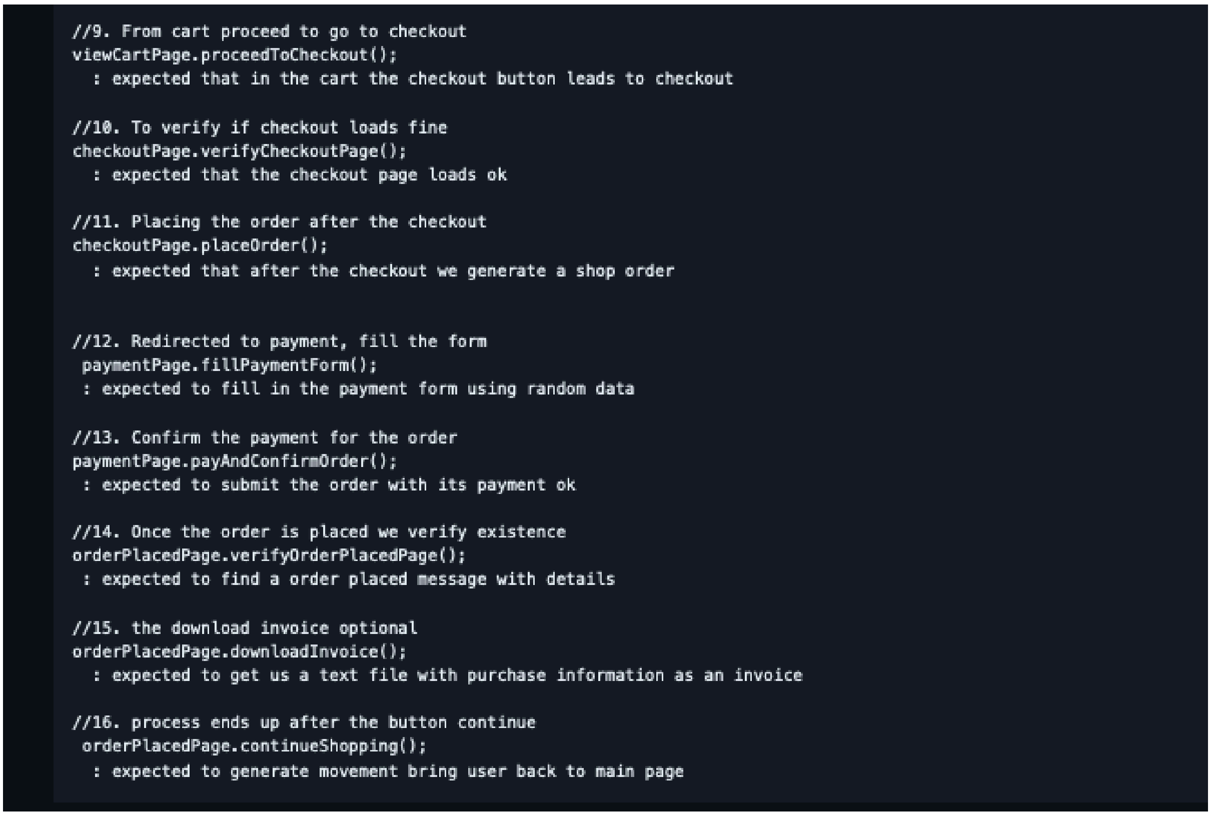
7. Order confirmation
Test 10: To download the invoice and clicking continue button we go back to the home page.

˝

POM based Automated Tests Report



```
//1. Main page actions
mainPage.visitMainPage();
: expected that this routine as a starter, opens main page
mainPage.viewProduct();
: expected that this test selects one product and displays its data correctly

//2. Product Details page actions
productDetailsPage.addToCart();
:expected that we find using the word "sleeve" to add thirty units of it to the cart

//3. Cart View page actions
 viewCartPage.proceedToCheckout();
 : expected that this action of clicking button, leads to checkout page

//4. System will ask to go to registration
viewCartPage.jumpToRegister();
: expected that this action shall bring up the registration form

//5. Registration
registrationPage.fillRegistrationForm(randomFirstName, randomLastName, randomEmail);
    : expected to fill up the form correctly with random generated data

//6. The signup form
signupPage.fillSignupForm(randomFirstName, randomLastName, randomEmail, randomPassword,       random
   : expected this also generates random data to fill up the sign up form , then submit

//7. Validation of account created
accountCreatedPage.validateAccountCreated();
   : expected this shall ensure that account has been created

//8. Navigate from success to Cart
mainPage.goToCart();
  : expected that after creating the account we go to cart
```



```
//9. From cart proceed to go to checkout
viewCartPage.proceedToCheckout();
  : expected that in the cart the checkout button leads to checkout

//10. To verify if checkout loads fine
checkoutPage.verifyCheckoutPage();
  : expected that the checkout page loads ok

//11. Placing the order after the checkout
checkoutPage.placeOrder();
  : expected that after the checkout we generate a shop order


//12. Redirected to payment, fill the form
 paymentPage.fillPaymentForm();
 : expected to fill in the payment form using random data

//13. Confirm the payment for the order
paymentPage.payAndConfirmOrder();
 : expected to submit the order with its payment ok

//14. Once the order is placed we verify existence
orderPlacedPage.verifyOrderPlacedPage();
 : expected to find a order placed message with details

//15. the download invoice optional
orderPlacedPage.downloadInvoice();
  : expected to get us a text file with purchase information as an invoice

//16. process ends up after the button continue
 orderPlacedPage.continueShopping();
  : expected to generate movement bring user back to main page
```

Code Routines

```
//1. Main page actions
mainPage.visitMainPage();
```
: expected that this routine as a starter, opens main page
```
mainPage.viewProduct();
```
: expected that this test selects one product and displays its data correctly

```
//2. Product Details page actions
productDetailsPage.addToCart();
```
:expected that we find using the word "sleeve" to add thirty units of it to the cart

```
//3. Cart View page actions
 viewCartPage.proceedToCheckout();
```
 : expected that this action of clicking button, leads to checkout page

```
//4. System will ask to go to registration
viewCartPage.jumpToRegister();
```
: expected that this action shall bring up the registration form

```
//5. Registration
registrationPage.fillRegistrationForm(randomFirstName, randomLastName, randomEmail);
```
: expected to fill up the form correctly with random generated data

```
//6. The signup form
signupPage.fillSignupForm(randomFirstName, randomLastName, randomEmail,
randomPassword,randomAddress, randomState,           randomCity, randomZipcode,
randomMobileNumber);
```
: expected this also generates random data to fill up the sign up form , then submit

```
//7. Validation of account created
accountCreatedPage.validateAccountCreated();
```
: expected this shall ensure that account has been created

```
//8. Navigate from success to Cart
mainPage.goToCart();
```
: expected that after creating the account we go to cart

```
//9. From cart proceed to go to checkout
viewCartPage.proceedToCheckout();
```
: expected that in the cart the checkout button leads to checkout

```
//10. To verify if checkout loads fine
checkoutPage.verifyCheckoutPage();
```
: expected that the checkout page loads ok

```
//11. Placing the order after the checkout
checkoutPage.placeOrder();
```

  : expected that after the checkout we generate a shop order

```
//12. Redirected to payment, fill the form
 paymentPage.fillPaymentForm();
```

 : expected to fill in the payment form using random data

```
//13. Confirm the payment for the order
paymentPage.payAndConfirmOrder();
```

 : expected to submit the order with its payment ok

```
//14. Once the order is placed we verify existence
orderPlacedPage.verifyOrderPlacedPage();
```

 : expected to find a order placed message with details

```
//15. the download invoice optional
orderPlacedPage.downloadInvoice();
```

  : expected to get us a text file with purchase information as an invoice

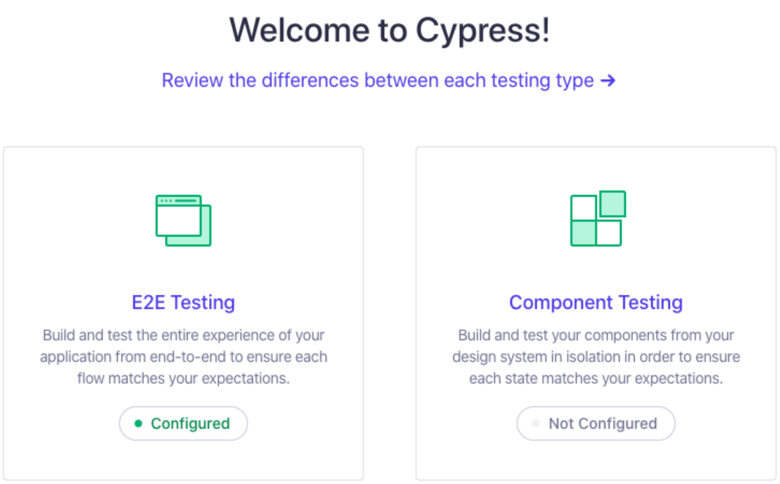
```
//16. process ends up after the button continue
 orderPlacedPage.continueShopping();
```

  : expected to generate movement bring user back to main page

Usage Instructions

0. Install Git for Windows , Linux or Mac
1. open a Terminal or CMD
2. git clone https://github.com/devmansworld/cartshopfinal.git
3. within the folder, type ‘cd cartshopv7’ to navigate to test folders
4. within the folder, type ‘npx cypress open’

5. wait a few seconds and then choose E2E



Welcome to Cypress!

Review the differences between each testing type →

E2E Testing

Build and test the entire experience of your application from end-to-end to ensure each flow matches your expectations.

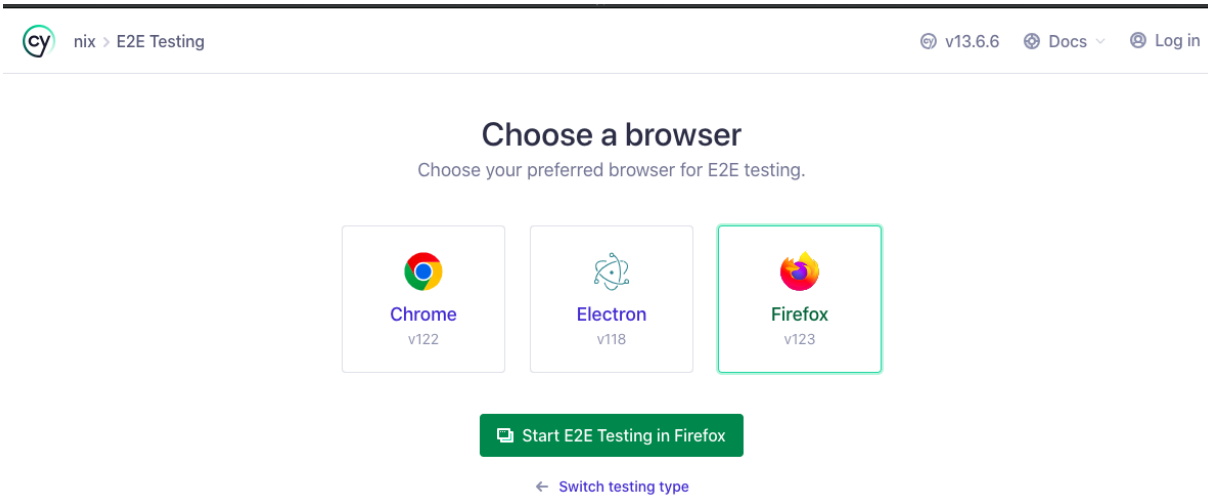
● Configured

Component Testing

Build and test your components from your design system in isolation in order to ensure each state matches your expectations.

○ Not Configured

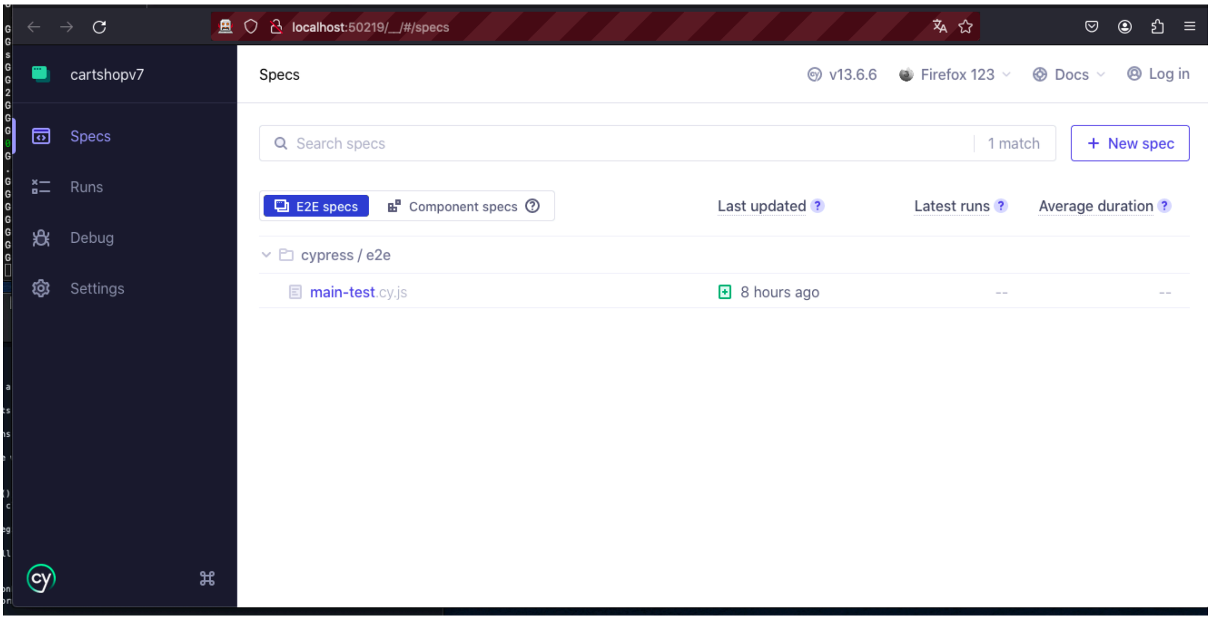
6. Choose best option Firefox browsera and hit “Start E2E testing in Firefox”



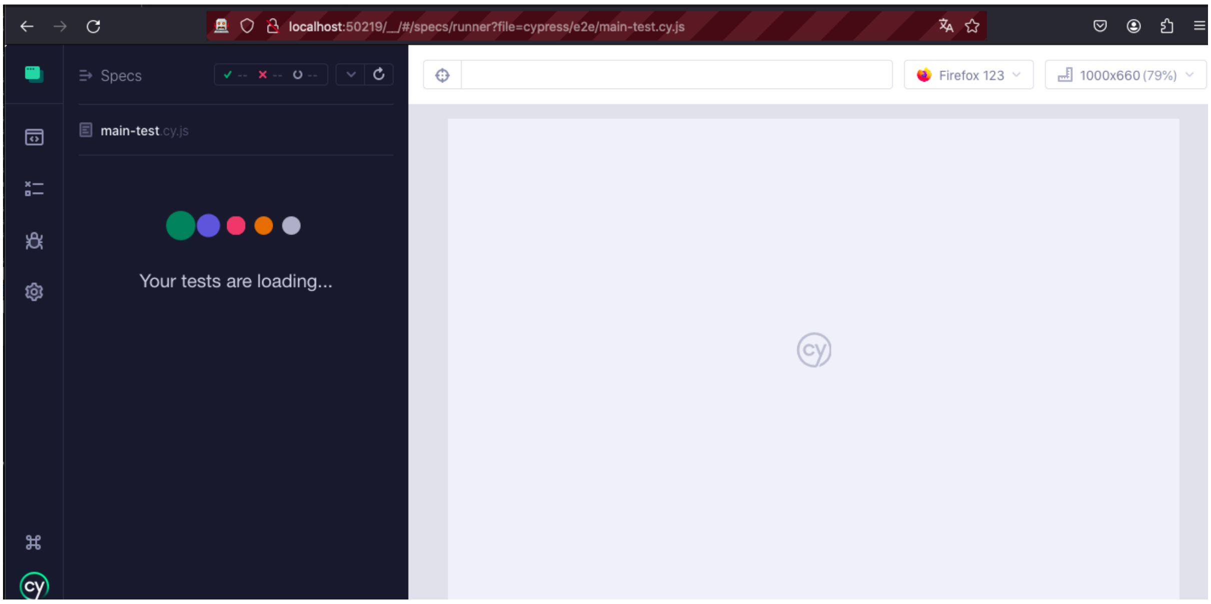
Choose a browser

Choose your preferred browser for E2E testing.

7 . After clicking “main-spec.js”

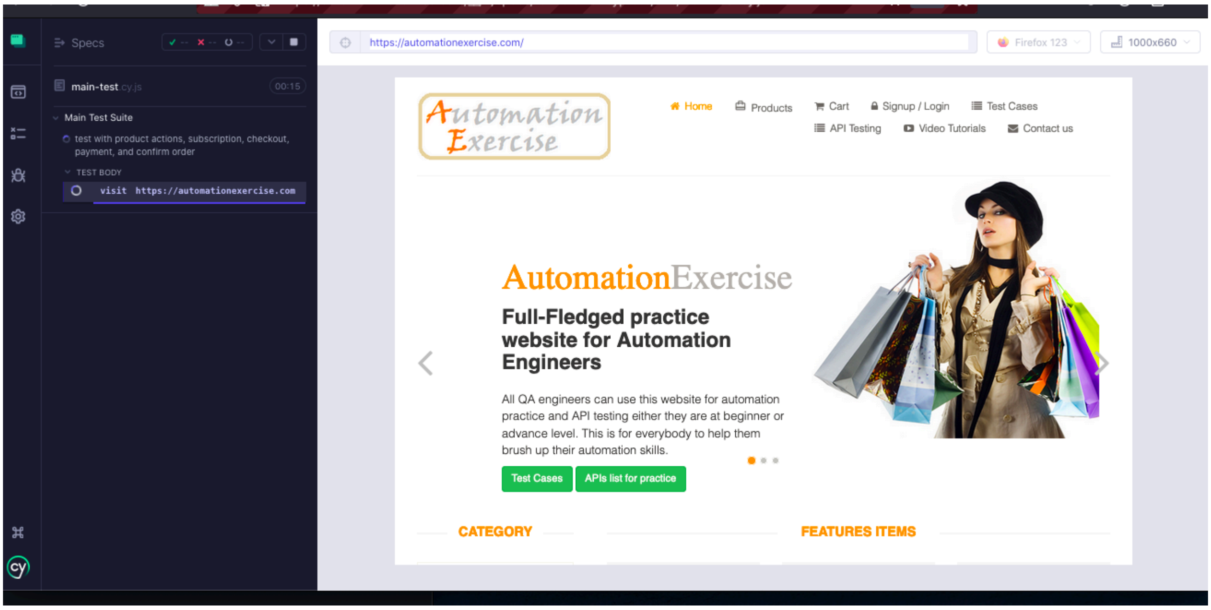


8. the program will show a loading page


localhost:50219/__/#/specs/runner?file=cypress/e2e/main-test.cy.js
Specs
✓ --
✕ --
main-test.cy.js
Firefox 123
1000x660 (79%)

Your tests are loading...

9. Finally the tests should be running



test with product actions, subscription, checkout, payment, and confirm order

```
visit https://automationexercise.com
```

https://automationexercise.com/

AutomationExercise

Full-Fledged practice website for Automation Engineers

All QA engineers can use this website for automation practice and API testing either they are at beginner or advance level. This is for everybody to help them brush up their automation skills.

Useful commands:

1. npx install cypress : To install Cypress environment via Node
2. npx cypress open: To run the tests within the browser
3. npx cypress run: shall run the same tests headlessly for more speed
4. npx install faker : in case you need to reinstall the library to add simulated data