

# Effective AngularJS Web Testing using Protractor

Deakin University

Faculty of Science, Engineering and Build Environment

Assignment 2

Tharini Hasara Manawadu De Silva – 217336638

## Executive Summery

To deliver a high quality software product to customer it is essential to perform Software testing which is an essential feature of Software development life cycle. In this paper it is discussed about the analysis and evaluation of selected testing tool against selected web page and the way the testing is done. This analysis basically focused on AngularJS based test automation tool, Protractor and software testing methods that used to function on a web application, Lands' End which is an AngularJS based web application.

In this report it gives a brief description about the testing tool and chosen web application including their features and behaviours. Then it discusses the analysis of the testing approaches and evaluation of those. Then describes about all the test cases and finally it demonstrates all the findings of the process. This report basically focuses on how Protractor performs on Angular based web applications testing and how it operates functional and unit testing on a selected web application. Recommendations discussed include:

- Increasing effectiveness and efficiency of test automation tools.
- Novel web application development including JavaScript base technologies.
- Expanding AngularJS based test automation.
- Detailed optimization in testing methods.

This report has several limitations. Some of these includes:

This testing process basically conducted upon a web application. All the testing methods and tools are discussed in this report are fix upon according to that web application. This web application is an AngularJS based web application and the testing tool and testing methods are chosen prior to the web application.

## Introduction

Unlike traditional web sites, the modern web development strongly concentrates on high performance, efficiency, effectiveness, user-friendliness and reusable components. Client side web development technologies reduces the number of server calls and save the bandwidth. So it is identified as rapid delivery of functionalities in a web application. AngularJS can be figured out as the best technology for achieve the previously cited sophisticated web application attributes. The easiness in the development procedure, AngularJS technology has become a bias in web development field. AngularJS basically use for Single Page Application Development because of the clarity in templating, animating, routing, validating and data binding. AngularJS is used to build rich web applications by accessing AngularJS Directives which extends HTML properties (McLaughlin 2015).

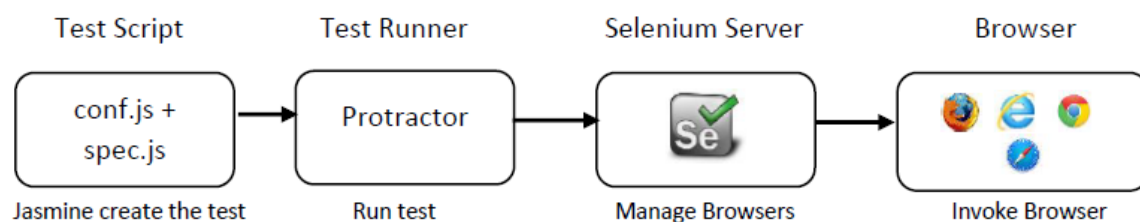
The decisive focus in software development life cycle is handing over a perfect and quality software product. To maintain the quality, efficiency, effectiveness and integrity of a software product software quality testing act a major role before deliver the final product to the customer. It is mandatory to check whether the all the functions in the web application perform as expected in order to deliver a best quality software product (Kuhn et al. 2009).

Perform a manual testing on a huge scale software applications is costly and time consuming. So it is helpful to function on large number of test cases in a short time and verify whether the functionalities work as expected by automating the testing process. As a result use of automated software testing

tool becomes more important in software development life cycle. AngularJS directives like ng-controller, ng-repeater, ng-model, ng-if etc. are specifically used when developing an AngularJS based web application. Traditional test automation tools cannot be used to test these AngularJS based web applications because of the AngularJS specific directives. Traditional automation tools cannot locate this AngularJS specific directives. So angular base web application testing can be done only using particular test automation tool which can locate AngularJS directives. So the testing automation tool Protractor can be taken as a test automation tool which locate AngularJS directives (Chaplin 2015).

Protractor is a testing framework that perform on end-to-end test automating specifically for AngularJS based web applications. Protractor runs on top of Selenium 2.0/WebDriver and it act as an abstraction layer of Selenium. It is a node.js framework. Protractor inherits all the attributes, behaviours and properties from Selenium sever and then adds commands, functions and services which specific to AngularJS web automation testing. Protractor associates with Jasmine, Mocha and Cucumber to render an integrated software solution (Fat et al. 2016). Jasmine framework is used to create test cases and Jasmine accommodates model, reporting tool and syntax to continue with the test process (Zhou et al. 2008).

Protractor avoids waiting and sleeping time and increase the efficiency and effectiveness of test automating process. It functions on both headless and real browsers. Protractor perform on both functional and unit testing which addresses in this report. The following diagram indicates how protractor works (<http://www.protractortest.org/#/>).



**Figure 1: Architecture of the Protractor tool**

To install Protractor and other relevant materials refer <http://www.protractortest.org/#/> link.

In this report it discussed about the chosen web application including its functions components and features. After that it briefly discussed about analysis of the testing approaches and evaluation of these testing approaches. Then provided all the test cases that have used in the analysis and description of each and every test case. Then it demonstrated the testing process and documented the all findings.

## Chosen Program

The chosen web application Lands' End is an e-commerce web application which allows users to purchase products online. The most important thing about this web application is that application is an AngularJS based web application and it contains AngularJS specific directives like ng-repeat, ng-controller, and ng-scope ng-submit etc. This web application have several features that are available for clients like creating a user account, sign in to the system, search for a particular product, adding filters to locate the best selected items and then continue on purchasing.

In account creation function, it allows users to create a user accounts prior to purchase by giving First Name, Email and Password. In here users have to confirm a reCAPTCHA before creating the account to make sure the user is not a machine. Sign in function allows users to sign in to the system by giving pre created email and password. As a guest user or as a registered user, it could be searched for a particular item by giving the product name. Users can filter the products according to the brand, size range, categories etc. Also users can directly find sale items. Sorting items functions allows users to sort products according to recommended, newest, cheapest etc. order. In purchasing function, users can purchase items as a registered user or as a guest user. In purchasing function user can select an item to purchase and then have to add to cart. After that give the shipping details like name and shipping address. After successfully completing shipping details user can purchase items by giving credit/debit card details or PayPal details. The chosen web application, Lands' End have these different functionalities and features.

## Analysis of testing approaches

Nowadays, the trend is online shopping. It cannot find anyone who does not do online shopping. Online retail can identified as a business that succeeds only on their online customers. Online shopping experience save time of the online customers and it can easily access the products any time at any location. Maintaining a good and stable online web site is the core of the success of the business. With the expansion on online shopping experience, there are many blooming and existing online retail shops. So as an online retailer it should give a reliable and stable shopping experience to the customer from its beginning. Otherwise customer may leave the site. So testing on the web application is essential. The better online experience creates better business (Childers et al. 2001).

Here are the specifications that test throughout this web application.

1. Create an account
2. Sign In process
3. Search for a particular item
4. Add filtering on products
5. Sort products
6. Add products to cart
7. Proceed on payment by giving shipping details

### Create an account

Creating a user account in an online retails shop is important and it gives a best user experience.

Test taken place

1. Validate for mandatory fields
2. Validate email address
3. Validate Password

In here this test approach validate by,

Giving error messages for incomplete and invalid inputs and successfully create accounts for valid inputs.

### Sign In process

Check whether user name and password before sign in to the account is mandatory to avoid malicious attacks to the web site.

Test taken place

1. Check for valid and in valid email address.
2. Check for valid and invalid password.

In here this test approach validate by,

Giving error messages for incomplete and invalid inputs and successfully Sign In for valid email and password.

### Search for a particular item

For the success of an e-commerce site search algorithm is crucial point, because developers always couldn't succeed on delivering products that users really want to see in the catalogue.

Test taken place

1. Giving an anonymous word as the product name.
2. Give a product name which having little spelling mistake.
3. Give a valid product name.

In here this test approach validate by,

1. Giving an error message for anonymous word as the product name.
2. Adjust the spelling mistake and deliver search result which the product closely match with the entered name.
3. Give a proper search result according to the searched product.

### Add filtering on products

Users are allowed to filter their products by adding some filters. This system have some filters like:

1. Filtering a product according the brand.  
Basic filter inputs: Lands' End, Lands' End Canvas
2. Filtering a product according categories.  
Basic filter inputs: Women's, Men's, Girl's, Boy's etc.
3. Filtering a product according the size range.  
Basic filter inputs: Regular, small, Plus etc.
4. Filtering a Sale items.
5. Sorting items according to recommended, top rated, newest, cheapest etc.

In here test approach validate by observing the output. Check whether the filter is applied by seeing the filter span.

## Add products to cart

After finding the products by clicking on the product then the adding items to the cart should be available for the user.

Test taken place

1. Check for mandatory options before adding to cart
2. Add different quantities
3. Continue shopping
4. Proceed to checkout
5. Remove items
6. Validate checkout details.

In here test approaches validate by giving error messages for mandatory and invalid inputs and observing the resulted page details.

Referred (Beizer 1995) and (Howden 1980) on writing tests against specifications.

## Test Cases

### Test cases for create a user account.

Account creation page require some inputs from clients like:

- First Name, Email, Password and reCAPTCHA identified as mandatory fields.
- Create Account button in the bottom of the form.
- Password must contains 8 to 20 characters and it should contain an at least one number and one letter.

#### Test Case 01

Test case Identifier	TC-01
Name	Check mandatory fields
Description	Check whether all the mandatory fields are can be identified by the system by not filling all the fields in the form.
Preconditions	<ol style="list-style-type: none"><li>1. Do not fill any field in the form</li><li>2. Click on Create Account button</li></ol>
Inputs	None
Expected Output	It should display an error message in red colour and do not allow users to create an account.
Observed Output	Displayed an error message in red colour and not allow to create an account.
Pass/Fail	Pass

Creating a user without giving values to the system creates malicious users. So it should check all the mandatory fields are filled by the user. So it can avoid creating malicious users.

## Test Case 02

Test case Identifier	TC-02
Name	Check mandatory fields
Description	Check whether all the mandatory fields are can be identified by the system by filling all the fields in the form.
Preconditions	<ol style="list-style-type: none"> <li>1. Fill all field in the form by giving valid inputs</li> <li>2. Click on Create Account button</li> </ol>
Inputs	First Name : Modata Email : <a href="mailto:modata@gmail.com">modata@gmail.com</a> Password : abcdefgh1234
Expected Output	It should display an error message in red colour to confirm as a real person by activating the reCAPTCHA.
Observed Output	Displayed an error message in red colour to confirm as a real person, picture selected window pop-ups and not allow to create an account.
Pass/Fail	Pass

Creating a user without confirming the reCAPTCHA can create users by machines. So it should check the reCAPTCHA to make sure the user is a real human.

## Test Case 03

Test case Identifier	TC-03
Name	Validate the email
Description	<ol style="list-style-type: none"> <li>1. Check whether can give an email address without '@' character.</li> <li>2. Check whether it is possible to give 'at' word to represent '@'.</li> <li>3. Check whether it is possible to give an email address without dot.</li> <li>4. Check whether it can give a random string instead of filling real email.</li> <li>5. Check whether can give an invalid domain</li> </ol>
Preconditions	<ol style="list-style-type: none"> <li>1. Give an invalid email address</li> <li>2. Click on Create Account button</li> </ol>
Inputs	<ol style="list-style-type: none"> <li>1. modatagmail.com</li> <li>2. modataatgmail.com</li> <li>3. modata@gmailcom</li> <li>4. <a href="mailto:abcdefghijklhkvvdvv@gmail.com">abcdefghijklhkvvdvv@gmail.com</a></li> <li>5. modata@gmail.adcfghjkl</li> </ol>
Expected Output	It should display an error message in red colour by asking give a valid email.
Observed Output	Displayed an error message in red colour with the error message.
Pass/Fail	Pass

Creating a user by giving invalid email corrupts the information of the users. So it should check the email that user enters is a valid one and then can avoid incorrect data in the database.

#### Test Case 04

Test case Identifier	TC-04
Name	Validate the email
Description	Check for valid email addresses
Preconditions	1. Give a valid email address 2. Click on Create Account button
Inputs	1. <a href="mailto:modata@gmail.com">modata@gmail.com</a>
Expected Output	It should not give any kind of validation error message.
Observed Output	Not get any validation error message
Pass/Fail	Pass

Creating a user by giving valid email address make easier the identified the particular user and maintain valid data in the user database.

#### Test Case 05

Test case Identifier	TC-05
Name	Validate the password
Description	Check for password limitations by giving password which is less than minimum characters. (less than 8)
Preconditions	1. Give a password less than 8 characters with at least one number and one letter. 2. Click on Create Account button
Inputs	Abcd5
Expected Output	It should display an error message in red colour saying as give a valid password
Observed Output	Got an error message in red colour as “give a valid password”.
Pass/Fail	Pass

It is essential to secure the system by encouraging users to give a strong password with 8 to 20 characters with at least one number and one letter. Since the password is strong can avoid the malicious attacks.



#### Test Case 06

Test case Identifier	TC-06
Name	Validate the password
Description	Check for password limitations by giving password which is more than minimum characters. (more than 20)
Preconditions	<ol style="list-style-type: none"> <li>1. Give a password more than 20 characters with at least one number and one letter.</li> <li>2. Click on Create Account button</li> </ol>
Inputs	Abcd5cdijnmasdfgghjkkloyreoiuytrew56789
Expected Output	It should display an error message in red colour saying as give a valid password
Observed Output	Got an error message in red colour as “give a valid password”.
Pass/Fail	Pass

It is essential to secure the system by encouraging users to give a strong password with 8 to 20 characters with at least one number and one letter. Since the password is strong can avoid the malicious attacks.

#### Test Case 07

Test case Identifier	TC-07
Name	Validate the password
Description	Check for password by giving only numbers within the password limitation.
Preconditions	<ol style="list-style-type: none"> <li>1. Give a password which only contains numbers within the password limitation.</li> <li>2. Click on Create Account button</li> </ol>
Inputs	123456789012
Expected Output	It should display an error message in red colour saying as give a valid password
Observed Output	Got an error message in red colour as “give a valid password”.
Pass/Fail	Pass

It is essential to secure the system by encouraging users to give a strong password with 8 to 20 characters with at least one number and one letter. Since the password is strong can avoid the malicious attacks.

#### Test Case 08

Test case Identifier	TC-08
Name	Validate the password
Description	Check for password by giving only letters within the password limitation.
	<ol style="list-style-type: none"> <li>1. Give a password which only contains letters within the password limitation.</li> <li>2. Click on Create Account button</li> </ol>
Inputs	abcdertyufghjyh
Expected Output	It should display an error message in red colour saying as give a valid password
Observed Output	Got an error message in red colour as “give a valid password”.
Pass/Fail	Pass

It is essential to secure the system by encouraging users to give a strong password with 8 to 20 characters with at least one number and one letter. Since the password is strong can avoid the malicious attacks.

#### Test Case 09

Test case Identifier	TC-09
Name	Validate the password
Description	Check for password by giving valid password within the password limitation and containing least one number and one letter
Preconditions	<ol style="list-style-type: none"> <li>1. Give a valid password.</li> <li>2. Click on Create Account button</li> </ol>
Inputs	abcdefgh1234
Expected Output	It should not give any kind of validation error message.
Observed Output	Not get any validation error message.
Pass/Fail	Pass

It is essential to secure the system by encouraging users to give a strong password with 8 to 20 characters with at least one number and one letter. Since the password is strong can avoid the malicious attacks.

### Test cases for Sign In.

Sign In page require some inputs from clients like previously created account details (Email and Password).

#### Test Case 01

Test case Identifier	TC-01
Name	Validate user sign In
Description	Check for user sign in by giving invalid email and correct password.
Preconditions	<ol style="list-style-type: none"><li>1. Give an invalid email</li><li>2. Give the correct password</li><li>3. Click on Sign In button</li></ol>
Inputs	Email : <a href="mailto:modatath12345th@gmail.com">modatath12345th@gmail.com</a> Password : abcdefgh1234
Expected Output	It should display a validation error message in red colour.
Observed Output	Displayed a validation error message in red colour.
Pass/Fail	Pass

Invalid email and valid password should not allow user to sign in, both email and password should be matched before sign in to the system to avoid malicious attacks.

#### Test Case 02

Test case Identifier	TC-02
Name	Validate user sign In
Description	Check for user sign in by keep the email text box empty and correct password.
Preconditions	<ol style="list-style-type: none"><li>1. Keep the email text box empty</li><li>2. Give the correct password</li><li>3. Click on Sign In button</li></ol>
Inputs	Email : Password : abcdefgh1234
Expected Output	It should display a validation error message in red colour.
Observed Output	Displayed a validation error message in red colour.
Pass/Fail	Pass

Valid password with blank email should not allow user to sign in, both email and password should be matched before sign in to the system to avoid malicious attacks.

#### Test Case 03

Test case Identifier	TC-03
Name	Validate user sign In
Description	Check for user sign in by giving correct email and incorrect password.
Preconditions	<ol style="list-style-type: none"> <li>1. Give a correct email address</li> <li>2. Give the incorrect password</li> <li>3. Click on Sign In button</li> </ol>
Inputs	Email : modata@gmail.com Password : abcdefgh1234XXY
Expected Output	It should display a validation error message in red colour.
Observed Output	Displayed a validation error message in red colour.
Pass/Fail	Pass

Invalid password and valid email should not allow user to sign in, both email and password should be matched before sign in to the system to avoid malicious attacks.

#### Test Case 04

Test case Identifier	TC-03
Name	Validate user sign In
Description	Check for user sign in by giving correct email and keep the password field empty.
Preconditions	<ol style="list-style-type: none"> <li>1. Give a correct email address</li> <li>2. Keep password text box empty</li> <li>3. Click on Sign In button</li> </ol>
Inputs	Email : modata@gmail.com Password :
Expected Output	It should display a validation error message in red colour.
Observed Output	Displayed a validation error message in red colour.
Pass/Fail	Pass

Valid email with blank password should not allow user to sign in, both email and password should be matched before sign in to the system to avoid malicious attacks.

#### Test Case 05

Test case Identifier	TC-05
Name	Validate user sign In
Description	Check for user sign in by giving valid email and valid password.
Preconditions	<ol style="list-style-type: none"> <li>1. Give a valid email address</li> <li>2. Give a valid password</li> <li>3. Click on Sign In button</li> </ol>
Inputs	Email : modata@gmail.com Password : abcdefgh1234
Expected Output	It should successfully Sing in the user to the system.
Observed Output	Successfully sign in to the system.
Pass/Fail	Pass

Valid email address and password make sure the user is a legal user and it avoids unauthorised accesses to the system.

#### Test Case 06

Test case Identifier	TC-06
Name	Show user password
Description	Check whether the show password link is working.
Preconditions	Click on "show password" link
Inputs	None
Expected Output	It should display the password with letters and numbers.
Observed Output	Displayed the password with letters and numbers correctly.
Pass/Fail	Pass

If a user can see the password after entering the password it avoids entering incorrect password and reduce the waiting period.

#### Test Case 07

Test case Identifier	TC-07
Name	Create account for new users
Description	Check for create new user account link is working.
Preconditions	Click on "create one" link
Inputs	None
Expected Output	It should navigate to create an account page.
Observed Output	Navigated to create an account page.
Pass/Fail	Pass

If an unregistered user directly navigate to the Sign In page and could not sign in, then it is useful to have a direct link for navigate to the account creation page.

### Test cases for Search box.

Search box allow users to directly find a particular product. This require an input like name the particular product that search for.

#### Test Case 01

Test case Identifier	TC-01
Name	Validate search box
Description	Check for search box by giving an anonymous word as a product name.
Preconditions	1. Give anonymous word as a product name 2. Click on Search icon
Inputs	abcdefghijkl
Expected Output	It should navigated to an error page with containing the error message for incorrect search.
Observed Output	Navigated to an error page with containing the error message for incorrect search.
Pass/Fail	Pass

Searching a product by name is essential for an e-commerce web site. Anonymous word in a search bar should give an error message to avoid displaying wrong details to the user.

#### Test Case 02

Test case Identifier	TC-02
Name	Validate search box
Description	Check for search box by giving product name with some incorrect spellings.
Preconditions	1. Give a product name with incorrect spellings 2. Click on Search icon
Inputs	toruser
Expected Output	It should navigated to a page which displays trousers and show a message for adjusting the search request name.
Observed Output	Navigated to trouser page and got a message as "We didn't find any matches for "toruser". We adjusted your search to "trouser".".
Pass/Fail	Pass

Searching a product by name is essential for an e-commerce web site and it should correctly find the similar search name because users may do some mistakes when entering the product name in the search bar. This reduces the user's searching time.

### Test Case 03

Test case Identifier	TC-03
Name	Validate search box
Description	Check for search box by giving a correct product name.
Preconditions	1. Give correct product name 2. Click on Search icon
Inputs	blouse
Expected Output	It should navigated to a page which displays all blouses.
Observed Output	Navigated to a page which containing blouses.
Pass/Fail	Pass

Searching a product by name is essential for an e-commerce web site because users can easily navigate to the items the wish to purchase.

### Test cases for Applying Filters

Users are allowed to filter their products by adding some filters. This system have some filters like:

1. Filtering a product according the brand.  
Basic filter inputs: Lands' End, Lands' End Canvas
2. Filtering a product according categories.  
Basic filter inputs: Women's, Men's, Girl's, Boy's etc.
3. Filtering a product according the size range.  
Basic filter inputs: Regular, small, Plus etc.
4. Filtering a Sale items.
5. Sorting items according to recommended, top rated, newest, cheapest etc.

### Test Case 01

Test case Identifier	TC-01
Name	Validate product filters
Description	Check for Brand filter by selecting particular brand name.
Preconditions	Select Lands' End brand name in the check box
Inputs	none
Expected Output	It should show only Lands' End branded clothes.
Observed Output	Displayed only Lands' End branded clothes.
Pass/Fail	Pass

Filtering products according to the brand allows users to select favourite brands by minimising searching time. So it is essential to check filtering by brand is working.

#### Test Case 02

Test case Identifier	TC-01
Name	Validate product filters
Description	Check for categories filter by selecting a particular category.
Preconditions	Click on Girl's link as category
Inputs	none
Expected Output	It should only display Girl's clothes.
Observed Output	Displayed only Girl's clothes.
Pass/Fail	Pass

Filtering products according to the categories allows users to select items according to their favourite category by minimizing searching time. So it is essential to check filtering by categories is working properly.

#### Test Case 03

Test case Identifier	TC-03
Name	Validate product filters
Description	Check for Brand filter by selecting particular size range.
Preconditions	Select regular button as size range.
Inputs	none
Expected Output	It should show only regular sized clothes.
Observed Output	Displayed only regular sized clothes.
Pass/Fail	Pass

Filtering products according to the size allows users to select well fitted items by minimizing searching time. So it is essential to check filtering by size range is working properly.

#### Test Case 04

Test case Identifier	TC-04
Name	Validate product filters
Description	Check for Sale filter.
Preconditions	Click on Sale link.
Inputs	none
Expected Output	It should display only Sale items.
Observed Output	Displayed only Sale items.
Pass/Fail	Pass

It should check whether the user can find Sale items by selecting sale option then they can easily locate the sale items.



#### Test Case 05

Test case Identifier	TC-04
Name	Validate product sort
Description	Check for sorting by clicking on a particular sorting way.
Preconditions	Click on top rated link.
Inputs	none
Expected Output	It should sort the items according to the rate.
Observed Output	Displayed only items according to the rate.
Pass/Fail	Pass

If the items are sorted according to some criteria then users can easily find the newest, rated and cheapest items in a single click. It minimizes their time.

#### Test cases for Add items to cart.

##### Test Case 01

Test case Identifier	TC-01
Name	Add item to cart
Description	Check whether all the mandatory options can be identified by the system by not selecting any option before Add to bag.
Preconditions	1. Select an item 2. Click on Add to bag
Inputs	none
Expected Output	It should display mandatory options in colour and should not allow to add to bag.
Observed Output	Displayed an error message as "Please select a size" in red colour, Size buttons turns into red and do not allow to add to cart.
Pass/Fail	Could not obtained

It should check the all mandatory fields are filled before adding items to cart. It avoids users in satisfaction after delivering the product. So selecting correct item with correct options is mandatory before checkout.

## Test Case 02

Test case Identifier	TC-02
Name	Add item to cart
Description	Check whether all the mandatory options can be identified by the system by not selecting any option before Add to bag.
Preconditions	<ol style="list-style-type: none"> <li>1. Select an item</li> <li>2. Select size as 10</li> <li>3. Click on Add to bag</li> </ol>
Inputs	none
Expected Output	It should display a pop up window which contains the details of the product including total amount and Continue shopping and Proceed to checkout buttons.
Observed Output	Pop up window which contains the details of the product including total amount and Continue shopping and Proceed to checkout buttons.
Pass/Fail	Could not obtained

It should check the all mandatory fields are filled before adding items to cart. It avoids users in satisfaction after delivering the product. So selecting correct size is mandatory before checkout.

## Test Case 03

Test case Identifier	TC-03
Name	Add item to cart
Description	Check whether it can select more than one item and then Add to bag
Preconditions	<ol style="list-style-type: none"> <li>1. Select an item</li> <li>2. Select size as 10</li> <li>3. Select quantity</li> <li>4. Click on Add to bag</li> </ol>
Inputs	Quantity: 2
Expected Output	It should display a pop up window which contains the details of the product including quantity and correct total amount and Continue shopping and Proceed to checkout buttons.
Observed Output	Pop up window which contains the details of the product including quantity and correct total amount and Continue shopping and Proceed to checkout buttons.
Pass/Fail	Could not obtained

Users usually purchase more than one quantity from one item. So it is essential to check whether users can purchase more than one quantity from one product.

#### Test Case 04

Test case Identifier	TC-04
Name	Add item to cart
Description	Check whether the user can continue shopping after adding item to the bag.
Preconditions	<ol style="list-style-type: none"> <li>1. Select an item</li> <li>2. Select size as 10</li> <li>3. Click on Add to bag</li> <li>4. Click on Continue shopping button on pop up window</li> </ol>
Inputs	None
Expected Output	It should add the current selected item to the bag and navigate to the previous page.
Observed Output	Added the selected product to the shopping cart and navigate to the previous page to continue shopping.
Pass/Fail	Could not obtained

User should be able to continue shopping after adding a one item. So it is essential to make sure that user can continuing shopping after adding one item to cart.

#### Test Case 05

Test case Identifier	TC-05
Name	Proceed to checkout
Description	Check whether the user can proceed to check out the items after adding items to the cart.
Preconditions	<ol style="list-style-type: none"> <li>1. Select an item</li> <li>2. Select size as 10</li> <li>3. Click on Add to bag</li> <li>4. Click on Proceed to check out button on pop up window</li> </ol>
Inputs	None
Expected Output	It should display the selected items with the quantity and calculated total amount of the items.
Observed Output	Display the all selected items and total amount of the items.
Pass/Fail	Could not obtained

After adding items to the shopping cart it should check whether the user can proceed the purchasing process.

#### Test Case 06

Test case Identifier	TC-06
Name	Remove items from the cart
Description	Check whether the user can remove items after adding items to the cart.
Preconditions	<ol style="list-style-type: none"> <li>1. Select 2 or 3 item</li> <li>2. Click on Add to bag</li> <li>3. Click on Proceed to check out button on pop up window</li> <li>4. Remove item in cart detail page</li> </ol>
Inputs	None
Expected Output	It should remove the item and total amount should be updated accordingly.
Observed Output	Remove the item and total amount should be updated accordingly.
Pass/Fail	Could not obtained

After adding items to the shopping cart it should check whether the user remove items before the purchasing process.

#### Test Case 07

Test case Identifier	TC-07
Name	Validate the checkout details
Description	Check whether all the mandatory fields are can be identified by the system by not filling all the fields in the payment form.
Preconditions	<ol style="list-style-type: none"> <li>1. Select an item</li> <li>2. Select size as 10</li> <li>3. Click on Add to bag</li> <li>4. Click on Proceed to checkout button on pop up window</li> <li>5. Click continue checkout cart page</li> <li>6. Do not fill any field in the form</li> <li>7. Click on continue</li> </ol>
Inputs	None
Expected Output	It should display an error message in red colour, change the colour in to red in mandatory fields and do not allow user to continue.
Observed Output	Show an error message in red colour, change the colour in to red in mandatory fields and do not allow to continue.
Pass/Fail	Could not obtained

Continue the purchasing process without filling shipping information should be checked. It is essential to give mandatory shipping information to avoid uncertainties in purchasing process.

## Test Case 08

Test case Identifier	TC-08
Name	Validate the checkout details
Description	Check whether validate the shipping address.
Preconditions	<ol style="list-style-type: none"> <li>1. Select an item</li> <li>2. Select size as 10</li> <li>3. Click on Add to bag</li> <li>4. Click on Proceed to checkout button on pop up window</li> <li>5. Click continue checkout cart page</li> <li>6. Fill all the mandatory fields other than the address by giving valid details and give anonymous address as address.</li> <li>7. Click on continue</li> </ol>
Inputs	First Name: Modata Last Name: Kaplan Address: 5, sclnmvdfklvmdfd City: Burwood Postal Code: 3125
Expected Output	It should display an error message about saying the incorrect address.
Observed Output	Navigated to the card details page successfully.
Pass/Fail	Could not obtained /Test case Failed when perform manually

It is essential to check whether the shipping address is correct with corresponding to the city. So it should validate the shipping address and should recognize the malicious address before continuing.

## Test Case 09

Test case Identifier	TC-09
Name	Validate the checkout details
Description	Check whether all the mandatory fields are can be identified by the system by filling all the fields in the payment form.
Preconditions	<ol style="list-style-type: none"> <li>8. Select an item</li> <li>9. Select size as 10</li> <li>10. Click on Add to bag</li> <li>11. Click on Proceed to checkout button on pop up window</li> <li>12. Click continue checkout cart page</li> <li>13. Fill all the mandatory fields by giving valid details</li> <li>14. Click on continue</li> </ol>
Inputs	First Name: Modata Last Name: Kaplan Address: 5, Burwood road, City: Burwood Postal Code: 3125
Expected Output	It should successfully navigate to card details page.
Observed Output	Navigated to the card details page successfully.
Pass/Fail	Could not obtained

Giving valid shipping details increase the responsiveness of the web application and can avoid uncertainties during the shipping process. So it is essential to check whether the shipping details are valid.

Referred to (Dick & Faivre 1993) on creating test cases.

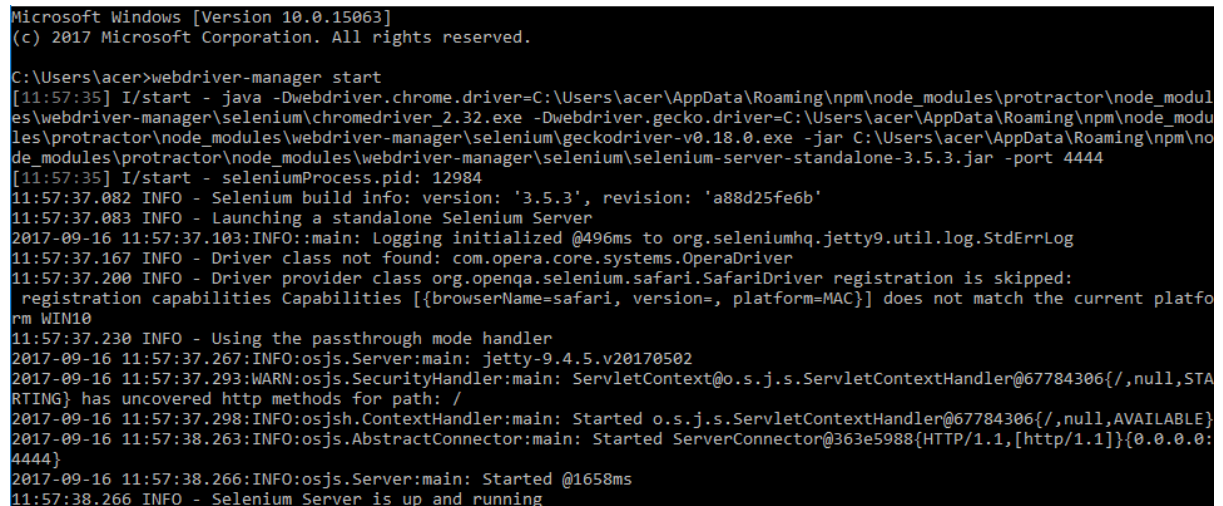
## Test Demonstration

Before perform the test cases first have to install protractor and other required tool. For installations, refer <http://www.protractortest.org/#/> link.

After installations first of all have to run the web driver manager. To run the web driver manager type the command as,

```
webdriver-manager start
```

Then the result shown as below.



```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\acer>webdriver-manager start
[11:57:35] I/start - java -Dwebdriver.chrome.driver=C:\Users\acer\AppData\Roaming\npm\node_modules\protractor\node_modules\webdriver-manager\selenium\chromedriver_2.32.exe -Dwebdriver.gecko.driver=C:\Users\acer\AppData\Roaming\npm\node_modules\protractor\node_modules\webdriver-manager\selenium\geckodriver-v0.18.0.exe -jar C:\Users\acer\AppData\Roaming\npm\node_modules\protractor\node_modules\webdriver-manager\selenium\selenium-server-standalone-3.5.3.jar -port 4444
[11:57:35] I/start - seleniumProcess.pid: 12984
11:57:37.082 INFO - Selenium build info: version: '3.5.3', revision: 'a88d25fe6b'
11:57:37.083 INFO - Launching a standalone Selenium Server
2017-09-16 11:57:37.103:INFO::main: Logging initialized @496ms to org.seleniumhq.jetty9.util.log.StdErrLog
11:57:37.167 INFO - Driver class not found: com.opera.core.systems.OperaDriver
11:57:37.200 INFO - Driver provider class org.openqa.selenium.safari.SafariDriver registration is skipped:
  registration capabilities Capabilities [{browserName=safari, version=, platform=MAC}] does not match the current platform WIN10
11:57:37.230 INFO - Using the passthrough mode handler
2017-09-16 11:57:37.267:INFO:osjs.Server:main: jetty-9.4.5.v20170502
2017-09-16 11:57:37.293:WARN:osjs.SecurityHandler:main: ServletContext@o.s.j.s.ServletContextHandler@67784306[/,null,STARTING] has uncovered http methods for path: /
2017-09-16 11:57:37.298:INFO:osjs.ContextHandler:main: Started o.s.j.s.ServletContextHandler@67784306[/,null,AVAILABLE]
2017-09-16 11:57:38.263:INFO:osjs.AbstractConnector:main: Started ServerConnector@363e5988{HTTP/1.1,[http/1.1]}{0.0.0.0:4444}
2017-09-16 11:57:38.266:INFO:osjs.Server:main: Started @1658ms
11:57:38.266 INFO - Selenium Server is up and running
```

Then locate to the folder that contains all codes from another command line and then to run the Protractor test cases give the command as,

```
protractor config.js
```

Find the link to locate the source file of the config.js file:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/config.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/config.js?at=master&fileviewer=file-view-default)

## Create user account

### Check for mandatory fields by giving blank data

In config.js file change the specs as **spec-mandatory** before start the test.

```
exports.config = {  
  framework: 'jasmine',  
  seleniumAddress: 'http://localhost:4444/wd/hub',  
  specs: ['spec-mandatory.js']  
}
```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-mandatory.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-mandatory.js?at=master&fileviewer=file-view-default)

Check whether all the mandatory fields are can be identified by the system by not filling any field in the form and then click on Create account button, then display an error message in red colour. Test case pass. 1 spec, 0 failures.

First Name

Email


[Privacy Policy & Contact Info](#)

Password

[Show password](#)

\* Your password must be a minimum of 8 characters and a maximum of 20 characters, and contain at least one number and one letter.

☐ I'm not a robot

  
reCAPTCHA  
[Privacy](#) - [Terms](#)

⚠ Please confirm you are a real person using the ReCAPTCHA above.

CREATE ACCOUNT

```

E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[15:21:25] I/launcher - Running 1 instances of WebDriver
[15:21:25] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[15:21:58] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:21:58] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:21:58] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:21:58] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
All the mandatory fields should be filled.

1 spec, 0 failures
Finished in 32.394 seconds
[15:22:02] I/launcher - 0 instance(s) of WebDriver still running
[15:22:02] I/launcher - chrome #01 passed
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>

```

### Check for mandatory fields by giving valid data

In config.js file change the specs as **spec-create.js** before start the test.

```

exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec-create.js']
}

```

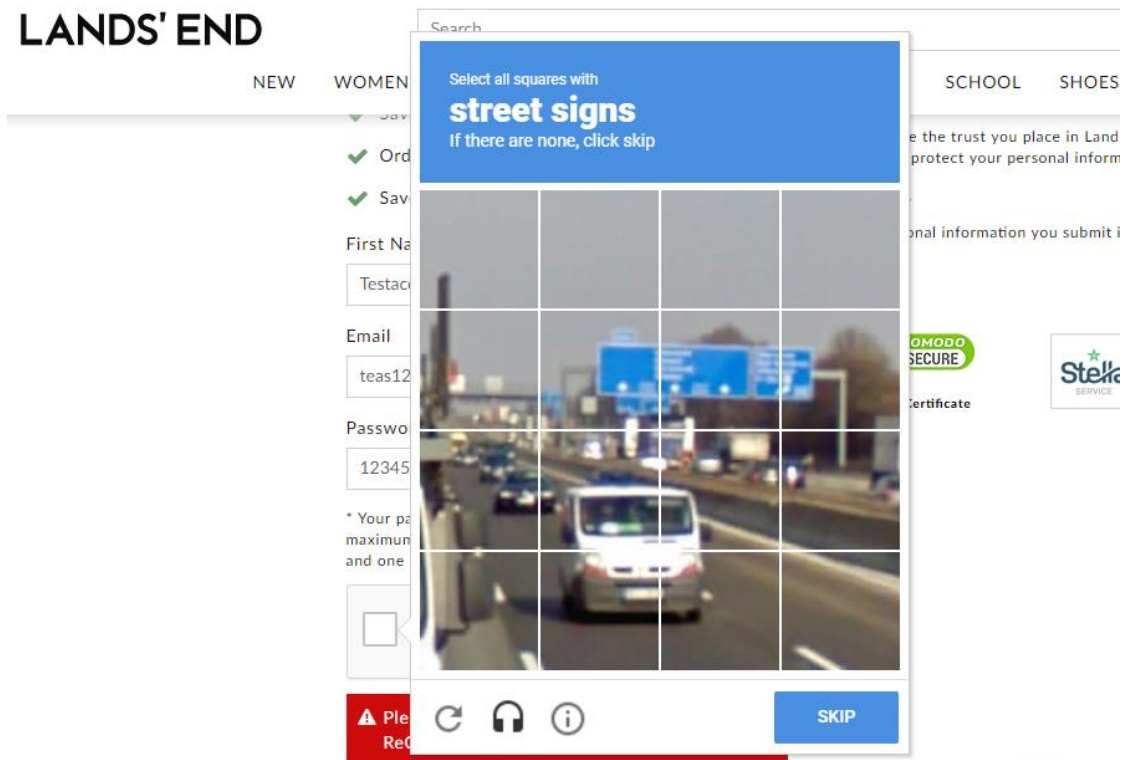
Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-create.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-create.js?at=master&fileviewer=file-view-default)

Check whether all the mandatory fields are can be identified by the system by filling all the fields in the form and then click on Create account button, then display an error message in red colour to confirm as a real person by activating the reCAPTCHA. Test case pass. 1 spec, 0 failures.



# LANDS' END



The test case result got from the command line as follows.

```
E:\EDUCATION\Trim2_yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[15:19:59] I/launcher - Running 1 instances of WebDriver
[15:19:59] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[15:20:30] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:20:30] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:20:30] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:20:30] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Should perform on reCAPTCHA before create an account.

1 spec, 0 failures
Finished in 32.091 seconds
[15:20:36] I/launcher - 0 instance(s) of WebDriver still running
[15:20:36] I/launcher - chrome #01 passed
E:\EDUCATION\Trim2_yr1\SIT707 - Software Quality And Testing\Assignment2\codes>
```

## Validate email address

In config.js file change the specs as **spec-email.js** before start the test.

```
exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec-email.js']
}
```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-email.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-email.js?at=master&fileviewer=file-view-default)

In above source code it validate the email address when creating a user account. Here it uses six functions for different email patterns. It inputs 5 invalid and 1 valid email addresses as follows.

#### Invalid addresses

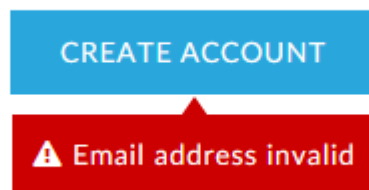
1. modatagmail.com
2. modataatgmail.com
3. abcdefghihkvvdvv@gmail.com
4. modata@gmail.abcfghjk
5. modata@gmailcom

#### Valid address

1. [modata@gmail.com](mailto:modata@gmail.com)

Test case pass. 6 spec, 0 failures.

For invalid email addresses it gives an error message in red colour as follows.



The results obtained for the above 6 scenarios as follows.

```
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[15:36:47] I/launcher - Running 1 instances of WebDriver
[15:36:47] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[15:37:13] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:13] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:13] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:14] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Invalid email address modatagmail.com
[15:37:26] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:26] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:26] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:26] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Invalid email address amodataatgmail.com.
[15:37:37] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:37] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:37] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:37] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Invalid email address abcdefghihkvvdvv@gmail.com.
[15:37:54] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:54] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:54] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:37:54] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Invalid email address modata@gmail.abcfghjkl.
```

```

[15:38:09] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:38:09] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:38:09] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:38:10] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Invalid email address modata@gmail.com.
[15:38:18] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:38:18] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:38:18] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[15:38:18] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Valid email address modata@gmail.com.

6 specs, 0 failures
Finished in 92.003 seconds

[15:38:23] I/launcher - 0 instance(s) of WebDriver still running
[15:38:23] I/launcher - chrome #01 passed

E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>

```

## Validate password

In config.js file change the specs as **spec-password.js** before start the test.

```

exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec-password.js']
}

```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-showpassword.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-showpassword.js?at=master&fileviewer=file-view-default)

In above source code it validate the email password when creating a user account. Here it uses five functions for different password patterns. It inputs 4 invalid and 1 valid passwords as follows.

### Invalid Passwords

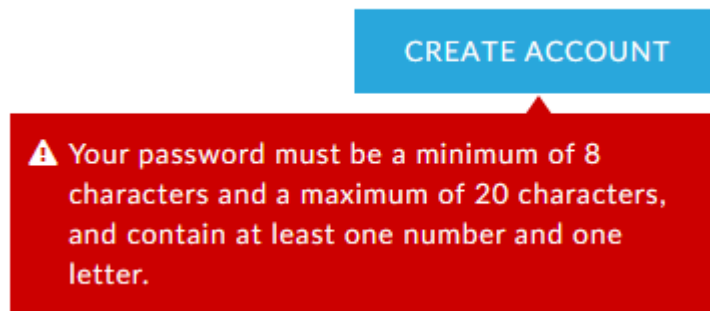
1. Password less than 8 characters: Abcd5
2. Password more than 20 characters: Abcd5cdijnmasdfgghjkkloyreoiuytrew56789
3. Password without a letter: 123456789012
4. Password without a number: abcdertyufghjyh

### Valid address

1. Strong password 8-20 character with containing at least one letter and one number: abcdefgh1234

Test case pass. 5 spec, 0 failures.

For invalid password it gives an error message in red colour as follows.



The results obtained for the above 5 scenarios as follows.

```
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[16:03:55] I/launcher - Running 1 instances of WebDriver
[16:03:55] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[16:04:22] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:22] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:22] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:22] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Invalid email address Abcd5.
[16:04:35] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:35] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:35] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:35] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Invalid Password Abcd5cdijnmasdfghjkkloyreoiuytrew56789.
[16:04:46] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:46] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:47] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:47] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Invalid password 123456789012.
[16:04:53] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:54] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:54] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:04:54] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Invalid password abcdertyufghjyh.
[16:05:03] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:05:03] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:05:03] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[16:05:03] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Valid password abcdefgh1234.

5 specs, 0 failures
Finished in 78.371 seconds

[16:05:17] I/launcher - 0 instance(s) of WebDriver still running
[16:05:17] I/launcher - chrome #01 passed

E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>
```

## Sign In process

Validate Password by giving invalid email and valid password and Validate Password by giving blank email and blank password.

In config.js file change the specs as **spec-signIn-1.js** before start the test.

```
exports.config = {  
  framework: 'jasmine',  
  seleniumAddress: 'http://localhost:4444/wd/hub',  
  specs: ['spec-signIn-1.js']  
}
```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-signIn-1.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-signIn-1.js?at=master&fileviewer=file-view-default)

Both test cases are written in the same spec file.

First it perform on invalid email modatath12345th@gmail.com and password abcdefgh1234 and then click on Sign In button. After that blank email and valid password. Test Case pass. 2 spec, 0 failures.

Invalid email modatath12345th@gmail.com and password abcdefgh1234 figure is follows.

### **SIGN IN**

Email

modatath12345th@gmail.com

Password

[Show password](#)



I'm not a robot



reCAPTCHA  
[Privacy](#) - [Terms](#)

[Cannot access your account?](#)

**SIGN IN**

**⚠ We did not recognize that email and/or password. Please try again or [click here to manage or reset your password](#).**

Don't have a Lands' End account? [Create one.](#)

Blank email and password abcdefgh1234 figure is follows.

## SIGN IN

Email

Password

[Show password](#)

[Cannot access your account?](#)

SIGN IN

⚠ We did not recognize that email and/or password. Please try again or [click here to manage or reset your password.](#)

Test case results for both test cases as follows.

```
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[11:33:50] I/launcher - Running 1 instances of WebDriver
[11:33:50] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[11:34:16] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:16] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:16] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:16] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Could not Sign in, check on credentials.
[11:34:38] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:38] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:38] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:38] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Could not Sign in, check on credentials.
.
2 specs, 0 failures
Finished in 69.199 seconds
[11:35:02] I/launcher - 0 instance(s) of WebDriver still running
[11:35:02] I/launcher - chrome #01 passed
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>
```

Validate Password by giving valid email and invalid password and validate Password by giving valid email and blank password.

In config.js file change the specs as **spec-signIn-2.js** before start the test.

```
exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec-signIn-2.js']
}
```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-signIn-2.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-signIn-2.js?at=master&fileviewer=file-view-default)

Both test cases are written in the same spec file.

First it perform on valid email modata@gmail.com and invalid password abcdefgh1234XXY and then click on Sign In button. After that blank email and valid password. Test Case pass. 2 spec, 0 failures.

Valid email modata@gmail.com and invalid password abcdefgh1234XXY figure is follows.

## SIGN IN

Email

modata@gmail.com

Password

[Show password](#)

[Cannot access your account?](#)

SIGN IN

**⚠** We did not recognize that email and/or password. Please try again or [click here to manage or reset your password.](#)

Don't have a Lands' End account? [Create one.](#)

Resulted valid email modata@gmail.com and blank password figure is follows.

## SIGN IN

Email

modata@gmail.com

Password

[Show password](#)

[Cannot access your account?](#)

SIGN IN

**⚠** We did not recognize that email and/or password. Please try again or [click here to manage or reset your password.](#)

Don't have a Lands' End account? [Create one.](#)

The resulted command line figure as follows.

```
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[11:33:50] I/launcher - Running 1 instances of WebDriver
[11:33:50] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[11:34:16] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:16] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:16] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:16] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Could not Sign in, check on credentials.
[11:34:38] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:38] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:38] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:34:38] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Could not Sign in, check on credentials.
.
2 specs, 0 failures
Finished in 69.199 seconds
[11:35:02] I/launcher - 0 instance(s) of WebDriver still running
[11:35:02] I/launcher - chrome #01 passed
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>
```



Validate Password by giving valid email and valid password.

In config.js file change the specs as **spec-signIn-correct.js** before start the test.

```
exports.config = {  
  framework: 'jasmine',  
  seleniumAddress: 'http://localhost:4444/wd/hub',  
  specs: ['spec-signIn-correct.js']  
}
```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-signIn-correct.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-signIn-correct.js?at=master&fileviewer=file-view-default)

Give the valid email [modata@gmail.com](mailto:modata@gmail.com) and password abcdefgh1234 and then click on Sign In button.  
Test Case pass. 1 spec, 0 failures.

## SIGN IN

Email

modata@gmail.com

Password

[Show password](#)

.....

[Cannot access your account?](#)

SIGN IN

Don't have a Lands' End account? [Create one.](#)

The resulted command line figure as follows.

```
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[10:58:17] I/launcher - Running 1 instances of WebDriver
[10:58:17] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[10:58:45] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[10:58:46] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[10:58:46] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[10:58:46] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Successfully Signed In to the account.

1 spec, 0 failures
Finished in 39.88 seconds

[10:59:01] I/launcher - 0 instance(s) of WebDriver still running
[10:59:01] I/launcher - chrome #01 passed
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>
```

### Show Password

In config.js file change the specs as **spec-showpassword.js** before start the test.

```
exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec-showpassword.js']
}
```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-signIn-correct.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-signIn-correct.js?at=master&fileviewer=file-view-default)

Click on Show password link and then display the password for user to avoid uncertainties. Test Case pass. 1 spec, 0 failures.

## SIGN IN

Email

modata@gmail.com

Password

[Hide password](#)

abcdefgh1234|

[Cannot access your account?](#)

SIGN IN

Don't have a Lands' End account? [Create one.](#)

The resulted command line figure as follows.

```
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[11:13:00] I/launcher - Running 1 instances of WebDriver
[11:13:00] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[11:13:23] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:13:24] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:13:24] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[11:13:24] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Successfully displayed the password.
.
1 spec, 0 failures
Finished in 33.38 seconds
[11:13:37] I/launcher - 0 instance(s) of WebDriver still running
[11:13:37] I/launcher - chrome #01 passed
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>
```

### Create account for new users

In config.js file change the specs as **spec-createone.js** before start the test.

```
exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec-createone.js']
}
```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-createone.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-createone.js?at=master&fileviewer=file-view-default)

If new user navigate to Sign In page and couldn't sign in then can create a new account by clicking on create one link and then it navigate to create account page. Test Case pass. 1 spec, 0 failures.

Don't have a Lands' End account? [Create one.](#)

The resulted command line,

```
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[12:14:17] I/launcher - Running 1 instances of WebDriver
[12:14:17] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[12:14:46] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[12:14:46] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[12:14:46] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
[12:14:46] W/element - more than one element found for locator By(css selector, .user-log-in-menu) - the first result will be used
Successfully navigated to create account page.

1 spec, 0 failures
Finished in 36.359 seconds
[12:14:59] I/launcher - 0 instance(s) of WebDriver still running
[12:14:59] I/launcher - chrome #01 passed
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>
```

## Search box

In config.js file change the specs as **spec-search.js** before start the test.

```
exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec-search.js']
}
```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-search.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-search.js?at=master&fileviewer=file-view-default)

All test cases for validating search box are in the same source file. First in validate the search box by giving anonymous product name as “asdffgghjkl”, then it gives a product name with wrong spellings as “truoser”. After that it gives correct search name as trouser. Test case pass 1 spec, 0 failures.

Search results for anonymous product name as follows.

SORRY, WE DIDN'T FIND ANY RESULTS FOR "asdffgghjkl".

- Check for typos, misspellings or incorrect item numbers.
- Try alternating search terms.
- Utilize single words.
- If you'd like assistance, feel free to contact us via [live chat](#).

Search results for wrong spellings product name as follows.

We didn't find any matches for "truoser". We adjusted your search to "trouser".

SORT: Recommended Newest **Top Rated** Price Low To High Price High To Low

126 ITEMS



QUICK VIEW



QUICK VIEW



QUICK VIEW



QUICK VIEW

Search result for valid product name as follows.

SORT: **Recommended** Newest Top Rated Price Low To High Price High To Low

126 ITEMS



QUICK VIEW



Men's Comfort Waist Pleat Year'rounder Wool Trousers



QUICK VIEW



Women's Mid Rise Wide Leg Chino Pants



QUICK VIEW



Men's Comfort Waist Year'rounder Wool Trousers



QUICK VIEW



Men's Traditional Fit Year'rounder Wool Trousers

The test results for above the use cases is follows.

```
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[14:04:23] I/launcher - Running 1 instances of WebDriver
[14:04:23] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[14:04:47] W/element - more than one element found for locator By(css selector, [ng-model="headerSearchCtrl.userInput"]) - the first result will be used
[14:04:48] W/element - more than one element found for locator By(css selector, [ng-model="headerSearchCtrl.userInput"]) - the first result will be used
[14:04:48] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[14:04:48] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[14:04:48] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
Can not find item for anonymous product name.
[14:05:07] W/element - more than one element found for locator By(css selector, [ng-model="headerSearchCtrl.userInput"]) - the first result will be used
[14:05:07] W/element - more than one element found for locator By(css selector, [ng-model="headerSearchCtrl.userInput"]) - the first result will be used
[14:05:07] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[14:05:07] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[14:05:07] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
Search for truoser adjusted your search to trouser.
[14:05:25] W/element - more than one element found for locator By(css selector, [ng-model="headerSearchCtrl.userInput"]) - the first result will be used
[14:05:25] W/element - more than one element found for locator By(css selector, [ng-model="headerSearchCtrl.userInput"]) - the first result will be used
[14:05:25] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[14:05:25] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[14:05:25] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
Search for trouser completed.
```

## Applying Filters

In config.js file change the specs as **spec-filters.js** before start the test

```
exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec-filters.js']
}
```

Find the source code from:



[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-filters.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-filters.js?at=master&fileviewer=file-view-default)

All test cases for applying filters and sorting are written in the same spec file. First it apply Lands' End filter as brand then Girl's filter as categories, regular filter as size range. After that sort according to top rated. Test case pass 1 spec, 0 failures.

After applying filters figure as follows.

## GIRLS

Search Results: blouse / Girls / **Tops**

| **Regular**  **Lands' End** 

---

### BRANDS

---

☒ Lands' End

---

### CATEGORIES

---

Tank Tops

Shirts & Blouses

---

### SIZE RANGE

---

---

### SALE

---

☐ Sale

The resulted command line.

```
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[13:02:47] I/launcher - Running 1 instances of WebDriver
[13:02:47] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[13:03:09] W/element - more than one element found for locator By(css selector, [ng-model="headerSearchCtrl.userInput"]) - the first result will be used
[13:03:09] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[13:03:09] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[13:03:10] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
Correctly applied Lands' End filter from brands.
Correctly applied Girl's filter from Caterhories.
Correctly applied regular filter from size range.

1 spec, 0 failures
Finished in 32.862 seconds
[13:03:22] I/launcher - 0 instance(s) of WebDriver still running
[13:03:22] I/launcher - chrome #01 passed
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>
```

## Locate sale items

In config.js file change the specs as **spec-sale.js** before start the test.


```
exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec-sale.js']
}
```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-sale.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-sale.js?at=master&fileviewer=file-view-default)

Locate the sale items by clicking on Sale check box. Test case pass. 1 spec, 0 failures.

## SALE

Search Results: blouse **Sale** 

## CATEGORIES

Women's

Girls

```
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[13:12:18] I/launcher - Running 1 instances of WebDriver
[13:12:18] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[13:13:01] W/element - more than one element found for locator By(css selector, [ng-model="headerSearchCtrl.userInput"]) - the first result will be used
[13:13:01] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[13:13:01] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[13:13:01] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
Correctly located sale items.

1 spec, 0 failures
Finished in 52.792 seconds
[13:13:14] I/launcher - 0 instance(s) of WebDriver still running
[13:13:14] I/launcher - chrome #01 passed
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>
```



## Sort Items

In config.js file change the specs as **spec-sort.js** before start the test.

```
exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec-sort.js']
}
```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-sort.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-sort.js?at=master&fileviewer=file-view-default)

Sort items according to top rated items to make easier the selecting process to user. 1 spec, 0 failures.

**SORT:**   Recommended   Newest   **Top Rated**   Price Low To High   Price High To Low   **232 ITEMS**

The test result for sorting as below.

```
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>protractor config.js
[13:23:27] I/launcher - Running 1 instances of WebDriver
[13:23:27] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
[13:23:50] W/element - more than one element found for locator By(css selector, [ng-model="headerSearchCtrl.userInput"]) - the first result will be used
[13:23:51] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[13:23:51] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
[13:23:51] W/element - more than one element found for locator By(css selector, [aria-label="Search"]) - the first result will be used
Correctly sorted according to top rated.
.
1 spec, 0 failures
Finished in 36.445 seconds
[13:24:06] I/launcher - 0 instance(s) of WebDriver still running
[13:24:06] I/launcher - chrome #01 passed
E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>
```

## Add items to cart

### Check for mandatory fields

In config.js file change the specs as **spec-addtocart.js** before start the test.

```
exports.config = {
  framework: 'jasmine',
  seleniumAddress: 'http://localhost:4444/wd/hub',
  specs: ['spec-addtocart.js']
}
```

Find the source code from:

[https://bitbucket.org/tmanawadu/testing\\_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-addtocart.js?at=master&fileviewer=file-view-default](https://bitbucket.org/tmanawadu/testing_protractor/src/6f9cd39df676f079f1b1e7504fabbf11cdf909be/spec-addtocart.js?at=master&fileviewer=file-view-default)

Check whether all the mandatory fields are can be identified by the system by not filling any field in the form and then click on Add to bag, then display an error message in red colour. Recognized that cannot perform add to bag without selecting the size.

The following screenshot should be the resulted page, but it does not allow machine to click “Add to Bad” button and could not find the button in the page source also. Gave an error and end up the test case.



The test result failed and could not perform other test cases.

```
1 spec, 1 failure
Finished in 71.952 seconds

[21:40:22] I/launcher - 0 instance(s) of WebDriver still running
[21:40:22] I/launcher - chrome #01 failed 1 test(s)
[21:40:22] I/launcher - overall: 1 failed spec(s)
[21:40:22] E/launcher - Process exited with error code 1

E:\EDUCATION\Trim2 yr1\SIT707 - Software Quality And Testing\Assignment2\codes>
```

This is an unexpected error and add this feature as future work, since the time is limited.

## Conclusion

The procedure of carry into effect a software program with intension to find out software issues, bugs and errors define as software testing. This effort confirms that software product is on the threshold of deliver to customer. To deliver a good quality software product to customer, software testing act a major role in software development life cycle. In this paper it is discussed about testing approaches and how execute these tests in a selected web application using test automation tool Protractor. Then analysis the testing approaches and evaluate these testing approaches. All test cases are wrote prior to the AngularJS based web application, Lands' End which is an e-commerce web application. In this report include online purchasing of a product until giving shipping details. Tests are executed according to the written test cases and all most all the test cases are passes until add items to bag. After that got an unexpected error and could not be obtained the test results for other test cases. It can be considered as a future work. As another future work, test cases can be increased up to finalizing the purchasing process and also wish to continue to adding products as an admin to the system. And also jasmine default time intervals are not sufficient for some test cases. It also can take as a future work on this project. As a conclusion, identified Protractor as the best automation tool to test AngularJS based end-to-end testing. Test automation on AngularJS web applications using protractor is a dominant step that has taken in software development industry.

## References

Beizer, B 1995, *Black-box testing: techniques for functional testing of software and systems*, John Wiley & Sons, Inc.

Chaplin, T 2015, *AngularJS Test-driven Development*, Packt Publishing Ltd.

Childers, TL, Carr, CL, Peck, J & Carson, S 2001, 'Hedonic and utilitarian motivations for online retail shopping behavior', *Journal of retailing*, vol. 77, no. 4, pp. 511-35.

Dick, J & Faivre, A 1993, 'Automating the generation and sequencing of test cases from model-based specifications', in *FME'93: Industrial-Strength Formal Methods*, pp. 268-84.

Fat, N, Vujovic, M, Papp, I & Novak, S 2016, 'Comparison of AngularJS framework testing tools', in *Zooming Innovation in Consumer Electronics International Conference (ZINC), 2016*, pp. 76-9.

Howden, WE 1980, 'Functional program testing', *IEEE Transactions on Software Engineering*, no. 2, pp. 162-9.

<http://www.protractortest.org/#/> *Protractor.js*.

Kuhn, R, Kacker, R, Lei, Y & Hunter, J 2009, 'Combinatorial software testing', *Computer*, vol. 42, no. 8.

McLaughlin, L 2015, 'Endicia Interactive Style Guides for Future Web Development', Worcester Polytechnic Institute.

Zhou, Z, Wang, L, Cui, Z, Chen, X & Zhao, J 2008, 'Jasmine: A tool for model-driven runtime verification with UML behavioral models', in *High Assurance Systems Engineering Symposium, 2008. HASE 2008. 11th IEEE*, pp. 487-90.