

✓ passed

⚙ unknown

Execution summary

35 scenarios

21 scenarios

Duration

5m 27.214s

Implementation

cucumber-jvm - 6.7.0

Runtime

Java HotSpot(TM) 64-Bit Server VM - 11.0.9+7-LTS

OS

Windows 10

CPU

amd64

Some text or @tags

Q

ⓘ You can use either plain text for the search or [cucumber tag expressions](#) to filter the output.

Filter by scenario status: ☒ passed ☐ unknown

✓ file:///C:/Users/mondelop/Desktop/Java%20Selenium/WestPac4/sample-cucumber-testng/src/test/resources/features/LoginTest.feature

Feature: Login Test

Background:

- ✓ **Given** Launch Application
- @tests @logins @smoketest

Scenario Outline: Login to application using valid credentials

- ✓ **Given** User enters <username> and <password>
- ✓ **When** User clicks on submit button
- ✓ **Then** user is successfully logged in

Examples:

	username	password
✓	paolomondelo1	Paolomondelo89@

@tests @logins @smoketest

Scenario Outline: Login to application using invalid credentials

- ✓ **Given** User enters <username> and <password>
- ✓ **When** User clicks on submit button
- ✓ **Then** User is denied from logging in

Examples:

	username	password
✓	fakeusername	fakepassword

✓ file:///C:/Users/mondelop/Desktop/Java%20Selenium/WestPac4/sample-cucumber-testing/src/test/resources/features/OverallRating.feature

Feature: Overall Rating Test

Background:

✓ **Given** Launch Application

@tests @overallratings @smoketest

Scenario Outline: Validate that the user can visit the Overall Rating Page

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmspassword>
- ✓ **When** the user clicks the Overall Rating Page Button
- ✓ **Then** the user is redirected to Overall Rating Page

Examples:

	loginname	firstname	lastname	password	confirmspassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

@tests @overallratings @smoketest

Scenario Outline: Validate that the user can vote and comment

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmspassword>
- ✓ **When** the user clicks the Overall Rating Page Button
- ✓ **Then** the user is redirected to Overall Rating Page
- ✓ **And** the user chooses a model Diablo
- ✓ **And** populates the comment box: <comments> and submits the vote
- ✓ **And** validates that the vote is successful

Examples:

	loginname	firstname	lastname	password	confirmspassword	comments
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Articuno

@tests @overallratings @smoketest

Scenario Outline: Validate that the user can vote and comment and Author should be <firstname> <lastname> for Model :rally037

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmspassword>
- ✓ **When** the user clicks the Overall Rating Page Button

- ✓ **Then** the user is redirected to Overall Rating Page
- ✓ **And** the user chooses a model Lancia
- ✓ **And** populates the comment box: <comments> and submits the vote
- ✓ **And** validates that the vote is successful
- ✓ **And** validate if the Author column is populated
- ✓ **And** validate if rally037 comment section is populated with <comments>

Examples:

	loginname	firstname	lastname	password	confirmpassword	comments
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Zapdos

@tests @overallratings @smoketest

Scenario Outline: Validate that the pagination for overall rating page is working

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Overall Rating Page Button
- ✓ **Then** the user is redirected to Overall Rating Page
- ✓ **And** the page pagination is validated and is working properly on the overall rating page

Examples:

	loginname	firstname	lastname	password	confirmpassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

✓ ✓ file:///C:/Users/mondelp/Desktop/Java%20Selenium/WestPac4/sample-cucumber-testng/src/test/resources/features/PopularMakeTest.feature

Feature: Popular Make Test

Background:

- ✓ **Given** Launch Application

@tests @popularmake @smoketest

Scenario Outline: Validate that the user can visit the Popular Make Page

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page

Examples:

	loginname	firstname	lastname	password	confirmpassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

@tests @popularmake @smoketest

Scenario Outline: Validate that the user can vote and comment

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page
- ✓ **And** the user chooses a model Diablo
- ✓ **And** populates the comment box: <comments> and submits the vote
- ✓ **And** validates that the vote is successful

Examples:

	loginname	firstname	lastname	password	confirmpassword	comments
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Articuno

@tests @popularmake @smoketest

Scenario Outline: Validate that the user can vote and comment and Author should be <firstname> <lastname> for Model :Reventon

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page
- ✓ **And** the user chooses a model Reventon
- ✓ **And** populates the comment box: <comments> and submits the vote
- ✓ **And** validates that the vote is successful
- ✓ **And** validate if the Author column is populated
- ✓ **And** validate if the reventon comment section is populated with <comments>

Examples:

	loginname	firstname	lastname	password	confirmpassword	comments
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Zapdos

@tests @popularmake @smoketest

Scenario Outline: Validate that the Model Column is present

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page
- ✓ **And** Model column should be present

Examples:

	loginname	firstname	lastname	password	confirmpassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

@tests @popularmake @smoketest

Scenario Outline: Validate that the Model Column is populated

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page
- ✓ **And** Model column should be populated

Examples:

	loginname	firstname	lastname	password	confirmpassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

@tests @popularmake @smoketest

Scenario Outline: Validate that the Rank Column is present

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page
- ✓ **And** Rank column should be present

Examples:

	loginname	firstname	lastname	password	confirmpassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

@tests @popularmake @smoketest

Scenario Outline: Validate that the Rank Value is populated

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page
- ✓ **And** Rank value should be populated

Examples:

	loginname	firstname	lastname	password	confirmpassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

@tests @popularmake @smoketest

Scenario Outline: Validate that the page pagination is working

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmspassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page
- ✓ **And** the page pagination is validated and is working properly

Examples:

	loginname	firstname	lastname	password	confirmspassword
✓	paolovvalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

✓ file:///C:/Users/mondelp/Desktop/Java%20Selenium/WestPac4/sample-cucumber-testng/src/test/resources/features/PopularModelTest.feature

Feature: Popular Model Test

Background:

- ✓ **Given** Launch Application

@tests @popularmodel @smoketest

Scenario Outline: Validate that the user can visit the Popular Model Page

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmspassword>
- ✓ **When** the user clicks the Popular Model Button
- ✓ **Then** the user is redirected to the Popular Model Page

Examples:

	loginname	firstname	lastname	password	confirmspassword
✓	paolovvalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

@tests @popularmodel @smoketest

Scenario Outline: Validate that the user can vote and comment while inside the Popular Model Page

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmspassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page

- ✓ **And** the user chooses a model Diabolo
- ✓ **And** populates the comment box: <comments> and submits the vote
- ✓ **And** validates that the vote is successful

Examples:

	loginname	firstname	lastname	password	confirmpassword	comments
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Bulbasaur

@tests @popularmodel @smoketest

Scenario Outline: Validate that the Model Column is present while inside the Popular Model Page

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page
- ✓ **And** Model column should be present

Examples:

	loginname	firstname	lastname	password	confirmpassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

@tests @popularmodel @smoketest

Scenario Outline: Validate that the Model Column is populated while inside the Popular Model Page

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page
- ✓ **And** Model column should be populated

Examples:

	loginname	firstname	lastname	password	confirmpassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

@tests @popularmodel @smoketest

Scenario Outline: Validate that the Rank Column is present while inside the Popular Model Page

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page

- ✓ **And** Rank column should be present

Examples:

	loginname	firstname	lastname	password	confirmpassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

@tests @popularmodel @smoketest

Scenario Outline: Validate that the number of votes is populated while inside the Popular Model Page

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the Popular Make Button
- ✓ **Then** the user is redirected to the Popular Make Page
- ✓ **And** number of votes should be displayed

Examples:

	loginname	firstname	lastname	password	confirmpassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

✓ ✓ file:///C:/Users/mondelp/Desktop/Java%20Selenium/WestPac4/sample-cucumber-testng/src/test/resources/features/Profile.feature

Feature: Profile Test

Background:

- ✓ **Given** Launch Application

@tests @profile @smoketest

Scenario Outline: Validate that the user can successfully change new password

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the profile link
- ✓ **And** the user is redirected to the profile page
- ✓ **And** the user changes the profile first name <firstnameppchange> and profile last name <lastnameppchange>
- ✓ **And** the user changes the profile age <age> and address <address> and phone <phone>
- ✓ **And** the user types the current password <confirmpassword> and new password <newpassword> and confirm new password <confirmnewpassword> by saving it
- ✓ **Then** a message that The profile has been saved successful should be displayed after

saving

Examples:

	loginname	firstname	lastname	password	confirmpassword	firstnameppchange	l
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Professor	C

@tests @profile @smoketest

Scenario Outline: Validate that the user can visit the Profile Page

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the profile link
- ✓ **And** the user is redirected to the profile page
- ✓ **Then** the user will be able to view the profile page

Examples:

	loginname	firstname	lastname	password	confirmpassword
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123

@tests @profile @smoketest

Scenario Outline: Validate that the user can successfully change first name and last name

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the profile link
- ✓ **And** the user is redirected to the profile page
- ✓ **And** the user changes the profile first name <firstnameppchange> and profile last name <lastnameppchange>
- ✓ **Then** a message that The profile has been saved successful should be displayed after saving

Examples:

	loginname	firstname	lastname	password	confirmpassword	firstnameppchange	l
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Professor	C

@tests @profile @smoketest

Scenario Outline: Validate that the user can successfully change Gender, Age, Address, and Phone

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the profile link

- ✓ **And** the user is redirected to the profile page
- ✓ **And** the user changes the profile first name <firstnameppchange> and profile last name <lastnameppchange>
- ✓ **And** the user changes the profile age <age> and address <address> and phone <phone>
- ✓ **Then** a message that The profile has been saved successful should be displayed after saving

Examples:

	loginname	firstname	lastname	password	confirmpassword	firstnameppchange	l
✓	paolovallidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Professor	C

@tests @profile @smoketest

Scenario Outline: Validate that app should not accept any updates with age less than 0 or age greater than 96 and that are whole numbers

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmpassword>
- ✓ **When** the user clicks the profile link
- ✓ **And** the user is redirected to the profile page
- ✓ **And** the user changes the profile first name <firstnameppchange> and profile last name <lastnameppchange>
- ✓ **And** the user changes the profile age <age> and address <address> and phone <phone>
- ✓ **And** the user types the current password <password> and new password <newpassword> and confirm new password <confirmnewpassword> by saving it
- ✓ **Then** the app should not accept values greater than 95 or less than 0

Examples:

	loginname	firstname	lastname	password	confirmpassword	firstnameppchange	l
✓	paolovallidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Professor	C
✓	paolovallidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Professor	C
✓	paolovallidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Professor	C
✓	paolovallidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Professor	C

@tests @profile @smoketest

Scenario Outline: Validate that app should not accept any updates with age less than 0 or age greater than 96 and that contains a decimal point

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user

enters confirms password: <confirmspassword>

- ✓ **When** the user clicks the profile link
- ✓ **And** the user is redirected to the profile page
- ✓ **And** the user changes the profile first name <firstnameppchange> and profile last name <lastnameppchange>
- ✓ **And** the user changes the profile age <age> and address <address> and phone <phone>
- ✓ **And** the user types the current password <password> and new password <newpassword> and confirm new password <confirmnewpassword> by saving it
- ✓ **Then** the app should not accept any values with decimal points or non ascii char

Examples:

	loginname	firstname	lastname	password	confirmspassword	firstnameppchange	l
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Professor	C
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Professor	C
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Professor	C
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	Professor	C

@tests @profile @smoketest

Scenario Outline: Validate that app should throw an error message stating that the new password used must have a combination of alphanumeric and must have lowercase and must have a nonascii character

- ✓ **Given** a user creates a new user with login name: <loginname> and fist name: <firstname> and last name: <lastname> and password name: <password> and user enters confirms password: <confirmspassword>
- ✓ **When** the user clicks the profile link
- ✓ **And** the user is redirected to the profile page
- ✓ **And** the user types the current password <password> and new password <newpassword> and confirm new password <confirmnewpassword> by saving it
- ✓ **Then** an error message should be displayed after saving

Examples:

	loginname	firstname	lastname	password	confirmspassword	newpassword	co
✓	paolovalidacnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123	paolomondelo1234	

✓ file:///C:/Users/mondelp/Desktop/Java%20Selenium/WestPac4/sample-cucumber-testng/src/test/resources/features/RegistrationTest.feature

Feature: Registration Test

Background:

- ✓ **Given** Launch Application

@tests @registration @smoketest

Scenario Outline: Validate that the user cannot register using invalid password format

- ✓ **Given** a user goes to the Register section
- ✓ **And** user enters login name: <loginname> and fist name: <firstname>
- ✓ **And** user enters last name: <lastname> and password name: <password>
- ✓ **And** user enters confirms password: <confirmspassword>
- ✓ **When** the user submits the request
- ✓ **Then** system should not allow the user to register and error message InvalidPasswordException should be displayed

Examples:

	loginname	firstname	lastname	password	confirmspassword
✓	invalidpaolo	Ash	Ketchum	password	password

@tests @registration @smoketest

Scenario Outline: Validate that the user can register using valid password format

- ✓ **Given** a user goes to the Register section
- ✓ **And** user enters login name: <loginname> and fist name: <firstname>
- ✓ **And** user enters last name: <lastname> and password name: <password>
- ✓ **And** user enters confirms password: <confirmspassword>
- ✓ **When** the user submits the request
- ✓ **Then** system should allow the user to register successfully

Examples:

	loginname	firstname	lastname	password	confirmspassword
✓	paolomondelovalidacctnt	Ash	Ketchum	Paolomondelo#123	Paolomondelo#123