**This documentation contains answers to the tasks, segregated accordingly in to four parts as listed below. Please go ahead to each section to follow through the answers/instructions.**

1. Write BDD scenarios for the story given.
2. Execute your tests and run exploratory testing in the website in general, create a report with the 5 most critical bugs found.
3. Automate 3 of your BDD scenarios using risk-based approach. You could use any language/tool (ideally C# + Selenium).
4. Automate 1 API test using the tool of your preference.

**A. Write BDD scenarios for the story given.**

Feature: Registration

As a visitor, I want to register on the website so that I can access the platform.

Scenario: A visitor can register with valid password

Given I am on the registration page as a visitor

When I enter valid registration details

And I enter password that meets the minimum requirements

And I submit the registration form

Then a success message is displayed

Scenario: A visitor cannot register with invalid password

Given I am on the registration page as a visitor

When I enter valid registration details

But I enter password that does not meet the minimum requirements

And I submit the registration form

Then an error message is displayed stating the password requirements

Scenario: A visitor can only register once

Given I am on the registration page as a visitor

When I enter valid registration details

And I enter password that meets the minimum requirements

And I submit the registration form

Then a success message is displayed

When I attempt to register again with the same details and password

Then an error message is displayed stating that I am already registered

**B. Execute your tests and run exploratory testing in the website in general, create a report with the 5 most critical bugs found.**

**BUG REPORT -** [**https://buggy.justtestit.org/**](https://buggy.justtestit.org/)

1. **Profile page -> Profile page does not redirect to another page after logging out**

**Severity:** Medium

**Steps to reproduce:**

1. Open <https://buggy.justtestit.org/>
2. On the top navigation bar, on the username field, enter username
3. On the password field, enter password
4. Click ‘Login’ button
5. Click ‘Profile’ link
6. Click ‘Logout’ link
7. Check the page

**Expected result:** It should redirect away from the Profile Page to protect user’s privacy. As to where it will redirect will be based upon requirement.

**Actual result:** The page is still on the Profile page exposing personal information.

**Test Environment:** Windows 11 – Chrome – Wi-Fi

1. **Registration page -> Invalid “Password must have numeric characters” message in the alert box registration form**

**Severity:** Medium

**Steps to reproduce:**

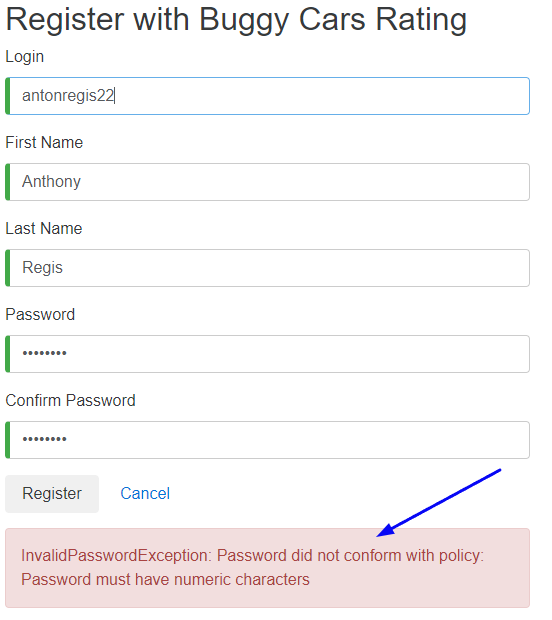
1. Open <https://buggy.justtestit.org/>
2. Click ‘Register’ button on the top right corner
3. Enter valid details on ‘Login’, ‘First Name’ and ‘Last Name’ fields
4. On the ‘Password’ and ‘Confirm Password’ fields, enter “Password”
5. Check alert box message displayed under the form

**Expected result:** The message should say “Password must have symbol characters” since minimum characters, all uppercase and lowercase requirements are met, and numeric character is not a password requirement.

**Actual result:** The message appears as “InvalidPasswordException: Password did not conform with policy: Password must have numeric characters” as seen on screenshot. The “InvalidPasswordException:” on the message is not user friendly.

**Test Environment:** Windows 11 – Chrome – Wi-Fi

**Screenshot:**

****

1. **Registration page -> Not user friendly alert message is returned when entered password is under 6 characters**

**Severity:** Medium

**Steps to reproduce:**

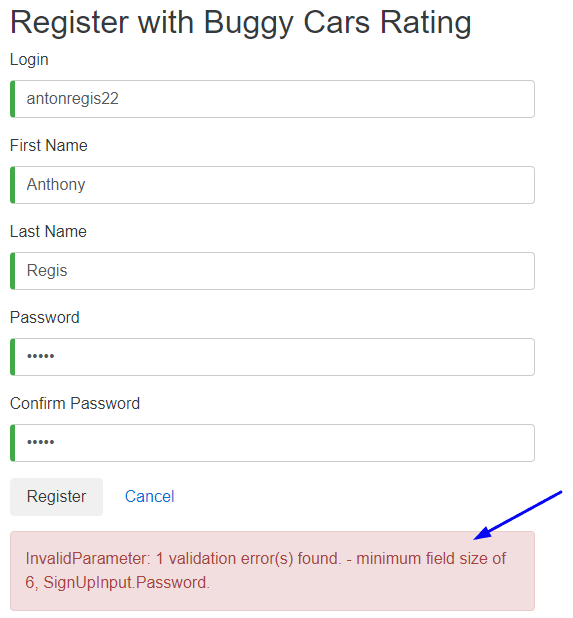
1. Open <https://buggy.justtestit.org/>
2. Click ‘Register’ button on the top right corner
3. Enter valid details on ‘Login’, ‘First Name’ and ‘Last Name’ fields
4. Enter 5 alphanumeric characters on ‘Password’ field
5. Enter the same 5 alphanumeric characters on ‘Confirm Password’ field
6. Click ‘Register’ button below the registration form
7. Check alert box message displayed under the form

**Expected result:** The message should say “Password must have 8 characters minimum”.

**Actual result:** The message shows “InvalidParameter: 1 validation error(s) found. - minimum field size of 6, SignUpInput.Password.” as seen on screenshot below.

**Test Environment:** Windows 11 – Chrome – Wi-Fi

**Screenshot:**



1. **Profile page -> Gender dropdown cannot be changed after Female is selected**

**Severity:** Medium

**Steps to reproduce:**

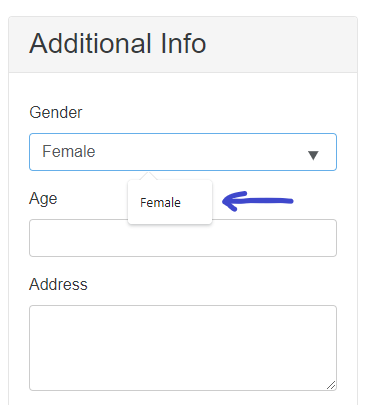
1. Open <https://buggy.justtestit.org/>
2. On the top navigation bar, on the username field, enter username
3. On the password field, enter password
4. Click ‘Login’ button
5. Click ‘Profile’ link
6. On the ‘Additional Info’ block, click ‘Gender’ dropdown
7. Select ‘Female’
8. Check ‘Gender’ dropdown again
9. Check dropdown options

**Expected result:** Both “Male” and “Female” should show as dropdown options.

**Actual result:** Only “Female” appears on the dropdown as seen on screenshot below.

**Test Environment:** Windows 11 – Chrome – Wi-Fi

**Screenshot:**

****

1. **Login -> Username and Password fields on the top of the page is not properly labelled**

**Severity:** Medium

**Steps to reproduce:**

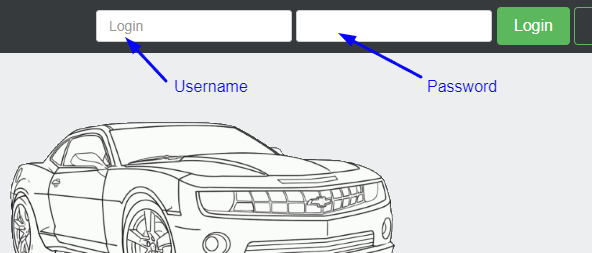
1. Open <https://buggy.justtestit.org/>
2. Check the two fields on top of the page on the left side of Login button

**Expected result:** Each of these two fields should be properly labelled within its own field. The field on the left should be “Username”, and “Password” on the right with grey font color as seen on Screenshot 2.

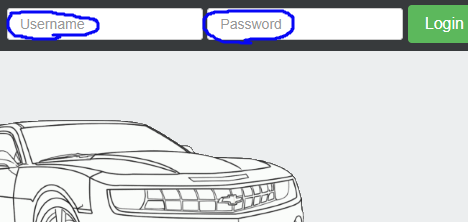
**Actual result:** Field on the left is labelled as “Login” and the field next to it has no label as seen on Screenshot 1.

**Test Environment:** Windows 11 – Chrome – Wi-Fi

**Screenshot 1:**

****

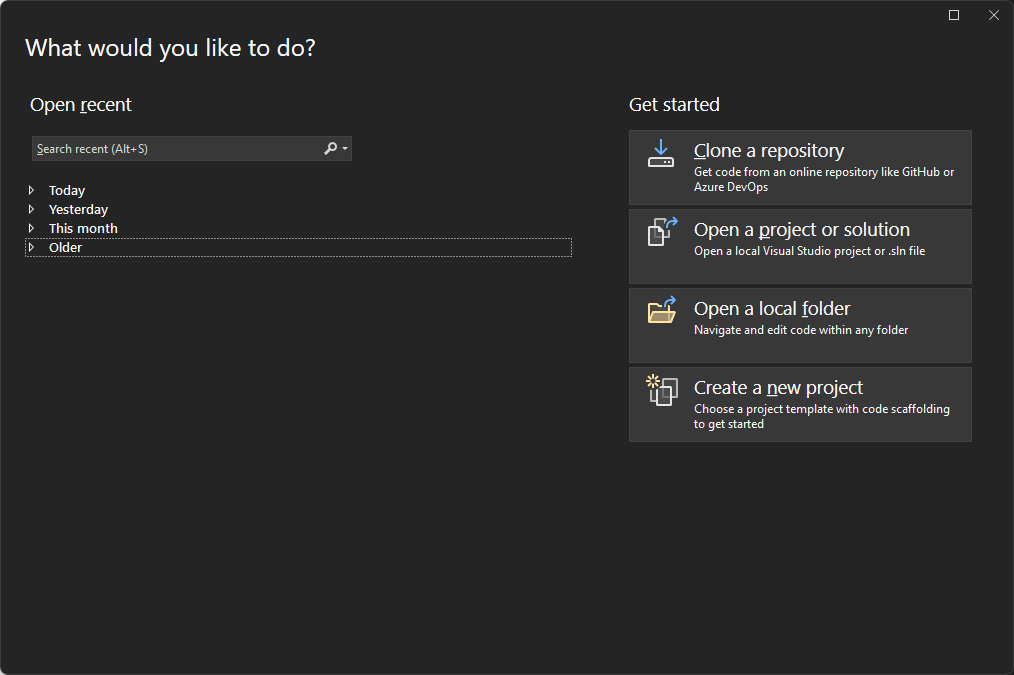
**Screenshot 2:**

****

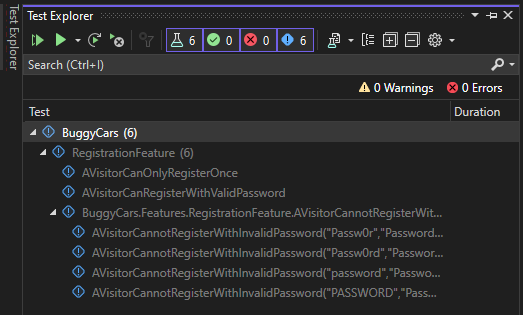
1. **Automate 3 of your BDD scenarios using risk-based approach. You could use any language/tool (ideally C# + Selenium).**

This automated test is demonstrated with the use of SpecFlow along with Selenium and C#. Please follow instructions below to download and run the project. Make sure to use Windows 10 or 11, updated version of Chrome browser and any updated flavor of Visual Studio 2022.

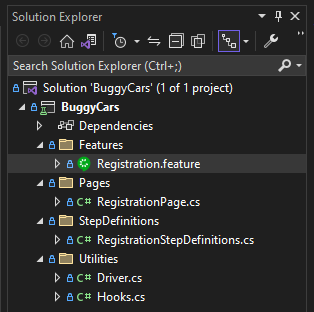
1. Open <https://github.com/antonregis/buggycarsregistrationpage>
2. Click on the green ‘Code’ dropdown button
3. Click ‘Download ZIP’. This file should download as buggycarsregistrationpage-main.zip.
4. Locate the downloaded zip file using Windows Explorer and right click on it
5. Click ‘Extract All’ to extract project files
6. Choose a destination folder of choice
7. Click ‘Extract’ button
8. Remember where the extracted files are located
9. Now load Visual Studio 2022. This should load the ‘Start Window’ as seen on screenshot below. If it is not showing, navigate to Visual Studio’s menu bar, click ‘File’, then ‘Start Window’.



1. Click ‘Open a project or solution’
2. Browse and locate the extracted folder
3. Navigate to Specflow folder
4. Click ‘BuggyCars.sln’
5. Click ‘Open’ button
6. On the menu bar, click on ‘Test’
7. Click on ‘Test Explorer’
8. On the ‘Test Explorer’ panel, expand ‘BuggyCars’ including the sub tests under it



1. On the upper left corner of ‘Test Explorer’ panel, click  button. This should start running all the test.
2. To view the BDD scenarios (SpecFlow feature file), navigate to the Solution Explorer panel, usually on the right side of the window. If it is not there, load it from the menu bar, click View>Solution Explorer.

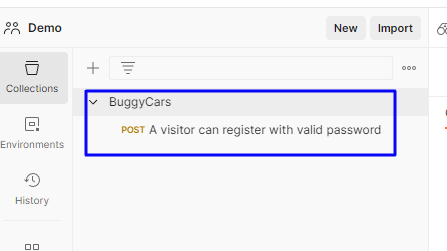


1. Double click on Registration.feature file to view the BDD scenarios.
2. **Automate 1 API test using the tool of your preference.**

The API test is executed using Postman. The chosen scenario for this automation is “A visitor can register with valid password”.

To keep it simple, the web based version of Postman on Chrome browser is used for this test with the assumption that steps 1 to 8 of the previous section (C.) were already performed. That said, please follow the steps below to proceed.

1. Open postman.com on Chrome browser
2. Click ‘Sign In with Google’
3. On the upper left corner of the screen, click ‘Workspaces’
4. Click ‘Create Workspace’
5. On the Name field, enter “Demo”
6. Click ‘Create Workspace and Team’
7. Click ‘Go to Workspace’
8. On the upper left corner of the screen, click ‘Import’
9. Click ‘files’
10. Navigate to the extracted folder (from section C, step 8) and look for ‘Postman’ folder
11. Double click ‘Postman’ to open the folder
12. Click ‘BuggyCars.postman\_collection.json’ file
13. Click ‘Open’ then the ‘BuggyCars’ appears on the Collection panel as seen on the screenshot below



1. Right click on ‘BuggyCars’
2. Click ‘Run collection’
3. On the right side of screen, click ‘Run BuggCars’ button
4. The test runs and a report will appear as seen on screenshot

