**Login Test Case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | **Test Description** | **Test Values** | **Expected Outcome** |
| 1 | Log into /admin successfully | Username = ‘bokie’  Password = ‘123123’ | Logs into /admin and displays /admin home page |
| 2 | Input wrong username and password when logging in | Username = ‘wrongusername’  Password = ‘wrongpassword’ | Displays error message on login page after attempt to login |

**Log out Test Case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | **Test Description** | **Test Values** | **Expected Outcome** |
| 1 | Log out of /admin successfully | Must be logged into /admin first | Logs out of /admin and redirected to login page |

**Add User Test Case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | **Test Description** | **Test Values** | **Expected Outcome** |
| 1 | Add user to backend through /admin | Username = ‘goodname/10’  Password = ‘123123123D’  2nd Pass = ‘123123123D | New user is added to the system |
| 2 | Add user with bad username | Username = ‘badusername’  Password = ‘123123123D’  2nd Pass = ‘123123123D’ | Error message “Enter a valid username. This value may contain only letters, numbers, and @/./+/-/\_ characters.” pops up after attempt to add new user |
| 3 | Add user with bad password | Username = ‘goodname/10’  Password = ‘badpass’  2nd Pass = ‘badpass’ | Error message “This password is too short. It must contain at least 8 characters.” Pops up after attempt |
| 4 | Add user with bad second password | Username = ‘goodname/10’  Password = ‘123123123D’  2nd Pass = ‘123123123’ | Error message “The two password fields didn't match.” Pops up after attempt |

**Add Blog Test Case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | **Test Description** | **Test Values** | **Expected Outcome** |
| 1 | Add blog to backend through /admin | Title = ‘This is a test blog’  Body = ‘This is a test body’  Category= Django | New blog is added to the system |
| 2 | Add blog with bad title | Title = ‘’  Body = ‘This is a test body’  Category = Django | Error message “This input field is required” pops up after attempt |
| 3 | Add blog with bad body | Title = ‘This is a test blog’  Body = ‘’  Category = Django | Error message “This input field is required” pops up after attempt |
| 4 | Add blog with bad category | Title = ‘This is a test blog’  Body = ‘This is a test body’  Category = ‘’ | Error message “This input field is required” pops up after attempt |

**Add Category Test Case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | **Test Description** | **Test Values** | **Expected Outcome** |
| 1 | Add category to backend through /admin | Name = ‘Test Category’ | New blog is added to the system |
| 2 | Add category with no name | Name = ‘’ | Error message “This field is required.” pops up |

**Project Test Case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | **Test Description** | **Test Values** | **Expected Outcome** |
| 1 | Check if projects show up on /project page |  | Project page is displayed |
| 2 | Test if navigation bar project button leads to project page |  | Project page is displayed |
| 3 | Test if project detail page shows the correct project detail |  | Project is clicked on the project page and leads to project detail page |

**Blog Test Case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Scenario** | **Test Description** | **Test Values** | **Expected Outcome** |
| 1 | Check if blogs show up on /blog page |  | Blog page is displayed |
| 2 | Test if blog detail shows the correct blog detail |  | Blog is clicked on the blog page and leads to blog detail page |
| 3 | Test if valid comment can be added under blog post | Author = ‘Test Name’  Body = ‘Test Comment’ | Page refreshes and comment shows up at the bottom of the blog |
| 4 | Test if comments can go over 80 chars | Author = ‘Test Name’  Body = ‘TestTestTestTestTestTestTestTest  TestTestTestTestTestTestTestTest TestTestTestTestTestTestTestTest  TestTestTestTestTestTestTestTest  TestTestTestTestTestTestTestTest  TestTestTestTestTestTestTestTest TestTestTestTestTestTestTestTest  TestTestTestTestTestTestTestTest’ | Input field should automatically limit the word count when it hits 80 chars, allowing the comment to be submitted |
| 5 | Test if author name can go over 20 chars | Author = ‘TestTestTestTestTestTestTestTest  TestTestTestTestTestTestTestTest’  Body = ‘Test Comment’ | Input field should automatically limit the word count when it hits 20 chars, allowing the comment to be submitted |
| 6 | Test if author name can include special characters | Author = ‘MY%nameis!@#$%^&\*()’  Body = ‘Test Comment’ | No validation is in check so the name is allowed, comment is submitted. |