|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | PRIORITY | TEST DATA | CASE | STEPS | RESULT | PASS/FAIL | COMMENTS |
| 1-LOGIN | HIGH | username: testuser  password: password | Login is successful with valid credentials. | 1- The user logs in to the system and enters the Login section.  2. Click on the Username container.  3- Enter the user name  4- Click on the Password container.  5- Enter the indicated password  6- Click on the Login button. | The login was successful. | PASS | This case was tested with cypress, with postman, also in manual tests directly in the application. Evidece check. |
| 2-LOGIN | HIGH | username:  password: password | Login is unsuccessful with invalid credentials. Empty user name. | 1- The user logs in to the system and enters the Login section.  2- Enter the indicated password  3- Click on the Login button. | The login does not progress and the message “Login failed” appears on the screen. | PASS | This case was tested with cypress, with postman, also in manual tests directly in the application. Evidece check. |
| 3-LOGIN | HIGH | username: testuser  password: | Validate and verify that the login was not successful with invalid credentials. Empty password. | 1- The user logs in to the system and enters the Login section.  2. Click on the Username container.  3- Enter the user name  4- Click on the Password container.  5- Click on the Login button. | The login does not progress and the message “Login failed” appears on the screen. | PASS | This case was tested with cypress, with postman, also in manual tests directly in the application. Evidece check. |
| 4-LOGIN | HIGH | username:  password: | Login with invalid credentials. Both parameters empty. | 1- The user logs in to the system and enters the Login section.  2- Click on the Login button. | The login does not progress and the message “Login failed” appears on the screen. | PASS | This case was tested with cypress, with postman, also in manual tests directly in the application. Evidece check. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | PRIORITY | TEST DATA | CASE | STEPS | RESULT | PASS/FAIL | COMMENTS |
| 5-DASHBOARD/PAGES | MEDIUM | Any data | Have access to a keyboard. | 1- Press the ENTER key once.  2- Press the TAB key twice. | Redirects correctly | PASS | Manual test |
| 6-DASHBOARD/PAGES | MEDIUM | Any data | Verify the correct navigation through the website using different browsers. | The site works locally, in all browsers and opens without any problem | Redirects correctly | PASS | Manual test |
| 7- WEBSITE | LOW | Any data | Verify website performance, quality, and correctness using Lighthouse for desktop devices (Home page, Login page and Dashborad page). | 1- Open CMD  2- Execute command lighthouse http://localhost:3000 --output html --output-path ./reporte.html before install Lighthouse.  3- Execute | Performance: 93  Accessibility: 91  Best Practices: 100  SEO: 100 | PASS | Analysis performed with application |
| 8-PRODUCT-ADD | HIGH | username: testuser  password: password  id: 1  name: Class Royal  price: 100 | Validate and verify that adding a new Product is successful with valid login and credentials. | 1- Click on the Login request.  2- Enter the username and password indicated in Test Data.  3- Click on the Send button.  4- Click on the "Create a product" request.  5- Check that the method, URL and body are correct. | The response has the status code 201. | PASS | The response is the expected |
| 9-PRODUCT-ADD | HIGH | username: testusersS  password: passwordS  id: 2  name: Pizza  price: 10 | Adding a new Product isn't successful with invalid login and valid credentials. | 1- Add the information of the variables indicated in test data.  2- Click on the button. | The response has the status code 401 Unauthorized, the body shows error information and the new product cannot be created | FAIL | The response is the expected |
| 10- PRODUCT-ADD | LOW | Any data | Adding a new product does not succeed if the ID has symbols. | 1- Click on the " Product" request.  2- Check that the method, URL and body are correct. | The response has the status code 400 Bad Request, the body shows error information and the new product cannot be created | FAIL | The response is the expected |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | PRIORITY | TEST DATA | CASE | STEPS | RESULT | PASS/FAIL | COMMENTS |
| 11- PRODUCT-UPDATE | HIGH | id: 1 | Validate and verify that the product isn't updated successfully, with valid parameter ID and empty fields. | 1- Enter the information of the parameter and varibles indicated in Test Data.  2- Click on the Send button. | The response has the status code 400 Bad Request and the Body is empty. | PASS | The response is the expected |
| 12-PRODUCT-DELETE | HIGH | id: 1 | Verifiy that the product cannot be deleted with invalid ID (is empty). | 1- Enter the information of the parameter indicated in Test Data.  2- Click on the Send button. | The response has the status code 405 Method Not Allowed and the error message is displayed in the Body | PASS | The response is the expected |
| 13-ORDER-CREATE | HIGH | username: testuser  password: password  id: 1  productName: Test  quantity: 10  status: Esto es un test | Adding a new Order is successful | 1- Enter the information of the variables indicated in Test Data.  2- Click on the Send button. | The order is created successfully. The status code, body and response time are correct. | PASS | The response is the expected |
| 13-ORDER-CREATE | CRITICAL | id: 5  quantity: 300  status: OK | Validate and verify that adding a new Order isn't successful without Product Name. | 1- Enter the information of the variables indicated in Test Data.  2- Click on the Send button. | The order is created even without Product name field. | FAIL | Another response was expected, there should be an alert or stop registration. |
| 14-ORDER-CREATE | CRITICAL | id: 5  id: 5 | Create several orders with the same ID | 1- Register an order with an ID by filling in all the fields.  2- Click on the Send button.  3- Register another order with the same ID | The orders are created with the same id, even the same name and others field. | FAIL | The app failed |
| ID | PRIORITY | TEST DATA | CASE | STEPS | RESULT | PASS/FAIL | COMMENTS |
| 15- ORDER-UPDATE | HIGH | id: 1 | Validate and verify that the product isn't updated successfully, with valid parameter of test data | 1- Enter the information of the parameter in Test Data.  2- Click on the Send button. | The response has the status code 400 Bad Request and the Body is empty. | PASS | The response is the expected |
| 15- ORDER-DELETE | HIGH |  | Validate that an order can be deleted successfully | 1- Select the ID number of the order.  2- Click on the Send button.  3- Verify that the order was deleted | The response has the status code 404 | PASS | The response is the expected |
|  |  |  |  |  |  |  |  |

**LEGEND:**

**CRITICAL**

Impact: Affects the basic functioning or availability of the site.

Examples: Site does not load, payment errors, data loss, critical security flaws.

Priority: Urgent; must be resolved immediately.

**HIGH**

Impact: Affects important functionality, user experience or causes visible failures.

Examples: Errors in the purchase flow, problems in core forms, excessive load times.

Priority: High; must be resolved soon to avoid impact on users.

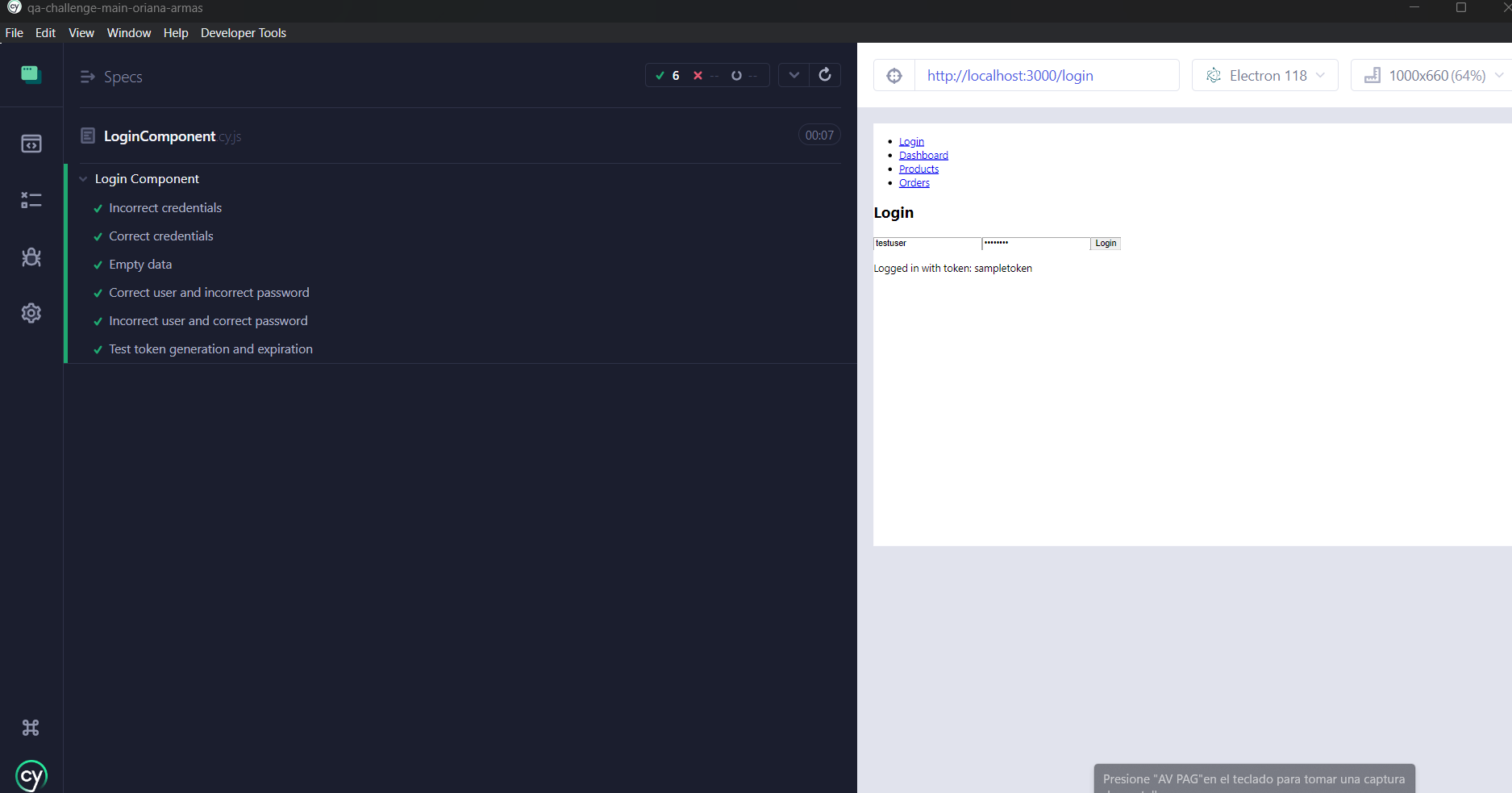
**LOW**

Impact: Minor bugs; do not affect functionality or main navigation.

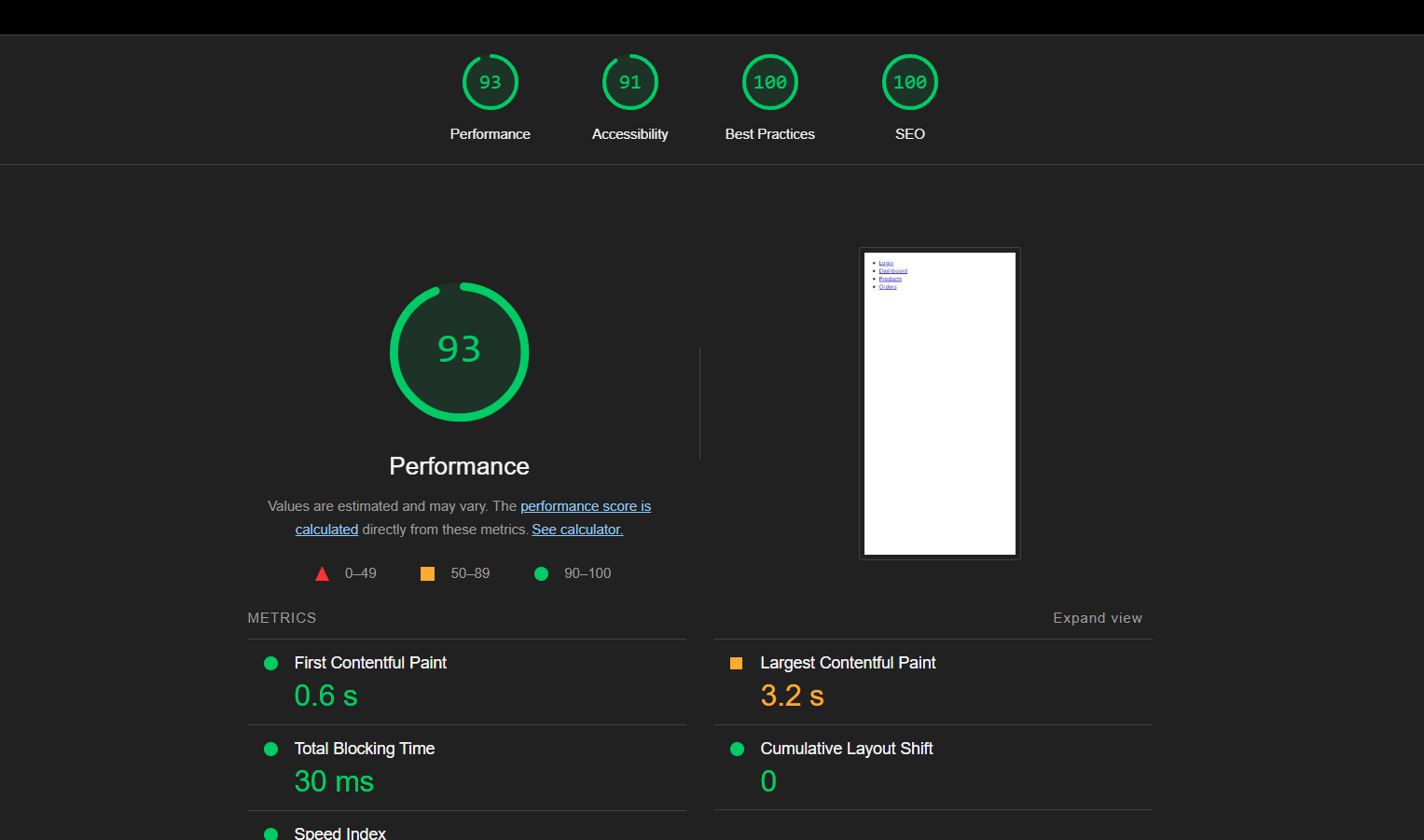
Examples: Minor visual problems, typographical errors, slight design lags.

Priority: Low; may be fixed in future versions or as available.

EVIDENCE **CASE #01 CYPRESS**



ACCESITIBILY



ANALYSIS Lighthouse

