Test summary report

**Test Summary Report Identifier**

GWD-RA-001

**Evaluation**

**Objective:** To evaluate the functionality and data integrity of the Equipment, Exercise, User, and Workout modules in the Gym Workout Database React App project.

**Scope:** System Testing of Create, Update, and Delete functionalities across the mentioned modules.

**Summary**

**Test Period:** 1 November 2023 to 16 November 2023

**Test Environment:** [Description of the test environment, including any relevant software or hardware details]

**Testing Person:** Crystal Sawers

**Summary of activities**

Conducted automated tests using Cypress for the following modules:

* Equipment Creation, Update, and Deletion
* Exercise Creation, Update, and Deletion
* User Creation, Update, and Deletion
* Workout Creation, Update, and Deletion

Executed a total of 24 test cases, including 12 valid tests and 12 invalid tests, primarily focusing on integration and functional testing aspects, and if defects were found, using exploratory testing to identify what the defects were in the app itself.

**Variances**

Discovered a critical defect in the Equipment Creation functionality, where the button was mislabeled, leading to failed creation tests.

Identified inconsistencies in form validation logic across different modules, affecting the reliability of user and workout management features.

The defects found were not initially identified in the test plan, indicating a variance from expected test outcomes.

**Approvals**

**Comprehensiveness assessment**

The application demonstrated critical functional defects that impact its core operations.

The testing process uncovered significant risks in terms of data management and user interaction.

Recommendations for immediate attention to the mislabeled buttons and inconsistent validation logic have been communicated to the development team.

**Summary of results**

1. **Test Cases passed/failed**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test File*** | ***Test Cases Planned*** | ***Test Cases Executed*** | ***Test Cases Passed*** | ***Test Cases Failed*** |
| equipment.cy.js | 6 | 6 | 0 | 6 |
| exercises.cy.js | 6 | 6 | 5 | 1 |
| users.cy.js | 6 | 6 | 5 | 1 |
| workouts.cy.js | 6 | 6 | 5 | 1 |
| **Total** | **24** | **24** | **15** | **9** |

1. **Status and Severity of defects (bugs)**

|  |  |  |  |
| --- | --- | --- | --- |
| Defect ID | Test File | Severity | Status |
| 1 | equipment.cy.js | Critical | Open |
| 2 | exercises.cy.js | High | Open |
| 3 | users.cy.js | High | Open |
| 4 | workouts.cy.js | High | Open |

The Severity and Status columns are filled out based on my findings when using automated testing. ‘Critical’ in this case indicates that at least 2 failed tests have led to problems in the other tests. 'Open' status indicates that the defect has been identified but not yet resolved in the app itself.

1. **Defect Distribution/Density**

|  |  |  |  |
| --- | --- | --- | --- |
| Test File | Total Test Cases | Defects Found | Defect Density (Defects per Test Case) |
| equipment.cy.js | 6 | 6 | 1.00 |
| exercises.cy.js | 6 | 1 | 0.17 |
| users.cy.js | 6 | 1 | 0.17 |
| workouts.cy.js | 6 | 1 | 0.17 |
| Total/Average | 24 | 9 | 0.38 |

Defect Density in the context of my project is calculated as the number of defects found divided by the total number of test cases executed. For example, in 'equipment.cy.js', there were 6 defects for 6 test cases, yielding a defect density of 1.00.