**TESTING VALIDATION AQUATHON\_APP**

**Category of the Testing**

1. **MT\_CRUD\_PARTICIPANT**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Expect Output** |
| 1 | Test Add Participant button renders correctly | Able to add Participant button visible on the page |
| 2 | Test Delete Participant button renders correctly | Able to delete Participant button visible on the page |
| 3 | Test Add Participant button redirects to the correct URL | Able to redirect to /races/{slug}/participants/create |
| 4 | Test Delete Participant button triggers participant deletion | Able to participant should be successfully deleted and page updated |
| 5 | Test confirmation dialog appears when attempting to delete a participant | Able to confirmation dialog appear before deletion |
| 6 | Test form validation for invalid participant details (e.g., empty name, invalid age) | Error messages appear for invalid input fields |
| 7 | Test participant creation form submission | After form submission, participant can be created, and the page also update |
| 8 | Test edit participant functionality | Editing a participant able to update the participant's details |
| 9 | Test that user receives confirmation after successfully creating a participant | A confirmation message appears after successful creation |
| 10 | Test that clicking the cancel button in the participant creation form navigates back without saving | Can clicking cancel and navigate back without saving |
| 11 | Test that participant creation form prevents submission when required fields are empty | Form did not submit when required fields are missing |
| 12 | Test that participant list updates automatically when a new participant is added | List have updated without manual page refresh after adding a participant |
| 13 | Test multi-user scenario where two users edit the same participant | Proper conflict resolution or overwrite didn't have protection in place |

1. **MT\_CRUD\_RACE**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Expect Output** |
| 1 | Test Add Race button renders correctly | Able to add Race button should be visible |
| 2 | Test Delete Race button removes the race and shows toast message | Able to race should be deleted, and a success message appear |
| 3 | Test deleting a race navigates to the /races page | Able to after race deletion, redirect to /races |
| 4 | Test Add Race button redirects to /races/create | Able to redirect to /races/create |
| 5 | Test form validation for invalid race details (e.g., missing race name or date) | Error messages appear for missing or invalid inputs |
| 6 | Test race creation form submission | After race creation form submission, the race appears in the list |
| 7 | Test edit race functionality | Editing a race able to update the race details in the list |
| 8 | Test that user receives confirmation after successfully creating a race | A confirmation message appears after successful race creation |
| 9 | Test that clicking the cancel button in the race creation form navigates back without saving | Can clicking cancel and navigate back without saving |
| 10 | Test that race creation form prevents submission when required fields are empty | Form did not submit when required fields are missing |
| 11 | Test that deleting a race removes all dependencies from the database (e.g., participants) | All associated records removed from the database |
| 12 | Test that race creation form prevents duplicates | Form hasn’t prevent creating races with identical names/dates |
| 13 | Test error handling when API is down during race creation | App gracefully handle API errors with clear messages |

1. **MT\_PERFORMANCE**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Expect Output** |
| 1 | Test that all buttons and form fields are accessible via keyboard (tab navigation) | All interactive elements able to be focusable via keyboard |
| 2 | Test participant details are correctly updated in the database after editing | Edited details reflected in the database and UI |
| 3 | Test server-side performance when loading a large list of races | Server handle large data efficiently without crashing |
| 4 | Test API response time for participant creation/deletion | API response quick with seconds |
| 5 | Test front-end performance for rendering races on slow network connections | UI stills be responsive, with loading indicators |
| 6 | Test loading indicators while waiting for participants to load | Loading spinners or skeleton loaders appear while data is loading |
| 7 | Test that sensitive participant data is not visible in the network logs (e.g., during API calls) | Sensitive data did not be logged or visible in network traffic |
| 8 | Test that race start/end times are validated correctly | Race start/race times is work perfectly |
| 9 | Test confirmation modal animations are smooth and not laggy | Animations smooth without performance issues |
| 10 | Test app behavior when offline or on a poor network connection | App show offline messages or cached content when disconnected |

1. **MT\_RESPONSE**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Expect Output** |
| 1 | Test loading time for race details page | Race details load in an acceptable time 2 seconds |
| 2 | Test that race details page renders correct race information | Correct race information has shown based on the race ID |
| 3 | Test that participant deletion removes participant from the list | Participant has no longer appear in the list after deletion |
| 4 | Test participant and race count is updated after adding/deleting participants or races | Correct count appears after adding/deleting entities |
| 5 | Test that filtering participants by a certain criterion works correctly (if available) | Filtering shows the correct participants based on the selected filter |
| 6 | Test filtering races by specific criteria (e.g., date, location) | Filtering shows correct races based on selected criteria |
| 7 | Test race sorting (e.g., sort races by date or alphabetical) | Races haven't able reorder correctly based on the selected sort of option |
| 8 | Test that race deletion removes all associated participants | All participants in a race removed when the race is deleted |
| 9 | Test that invalid URLs (e.g., /races/invalid id) show a 404-error page | A custom 404 page has appeared for invalid race URLs |
| 10 | Test that error messages are clear and user-friendly | Error messages should be descriptive and easy to understand |
| 11 | Test memory usage when many races/participants are loaded | Memory usage remains stable and not cause crashes |
| 12 | Test that participants are sorted alphabetically by default | Participants didn’t sort by name in ascending order |
| 13 | Test that participant records cannot be duplicated | System prevents creating duplicate participants through detect the bib of participant |

1. **MT\_SECURITY**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Expect Output** |
| 1 | Test form validation to ensure malicious input (e.g., HTML/JS injection) is sanitized | Form inputs reject any HTML/JS injection attempts |
| 2 | Test CSRF (Cross-Site Request Forgery) protection on form submissions | All forms reject CSRF attacks |

1. **MT\_STYLE**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Expect Output** |
| 1 | Test that Adds Race button is styled correctly (e.g., correct color, dimensions) | Add Race button have appropriate styles (rounded, background color, etc.) |
| 2 | Test that Deletes Participant button is styled correctly (e.g., correct color, icon alignment) | Delete button visually aligned and colored appropriately |
| 3 | Test load time when rendering participant list for many participants | Page load in an acceptable time |
| 4 | Test empty participant list rendering (when no participants are present) | A message like "No participants found" appear |
| 5 | Test race list rendering when no races are available | A message like "No races available" appear |
| 6 | Test that clicking on a participant navigates to the participant details page | Redirect to /races/{slug}/participants/ {participant Id} |
| 7 | Test that success and error toasts (notifications) appear in the correct location on the screen | Toasts appear in the correct area of the screen at bottom-right |
| 8 | Test that forms fields have proper labels for screen readers | All form fields should have accessible labels (e.g., using aria-labels) |
| 9 | Test responsive behavior of the add and delete buttons on different screen sizes | Buttons have render correctly on mobile, tablet, and desktop views |
| 10 | Test that all forms are mobile-responsive and adjust properly on small screens | Forms resize and remain functional on mobile devices |
| 11 | Test that focuses indicators are clearly visible when tabbing through form fields | Focus visible on all interactive elements |
| 12 | Test placeholder text and label clarity for all form inputs | Placeholder text provide clear guidance to users |
| 13 | Test that race dates are formatted correctly across all parts of the UI | Dates display in a consistent and correct format |
| 14 | Test that icons (e.g., for delete, edit) are visible and function correctly on all screen sizes | Icons have properly aligned and responsive on all devices |
| 15 | Test font sizes and readability on all screen sizes | Font sizes responsive and readable on both mobile/desktop fine |
| 16 | Test that deletes confirmation dialogs are styled and centered correctly | Confirmation dialogs clearly visible and centered |
| 17 | Test max length limits for text inputs in participant and race forms | Fields should enforce max length limits |