1. Verify that the user can select a valid job type from the dropdown menu.
2. Check if the user can enter a valid subject with a minimum length of 4 characters and a maximum of 100 characters.
3. Ensure that the user can input a valid vehicle registration number according to the specified pattern.
4. Validate that the user can enter a customer name with only letters and spaces, adhering to the length constraints.
5. Test if the user can input a valid email address that matches the required pattern.
6. Confirm that the user can enter a valid Kenyan phone number for both the customer and contact person.
7. Verify that the user can select a valid vehicle make from the dropdown menu.
8. Check if the user can input a vehicle model with a minimum length of 2 characters and a maximum of 50 characters.
9. Ensure that the user can enter a vehicle color with the specified length constraints.
10. Validate that the user can input a vehicle location that meets the length requirements.
11. Test if the user can enter a valid loan amount that is a nonnegative number.
12. Attempt to submit the form without selecting a job type and verify that an error message is displayed.
13. Enter a subject with fewer than 4 characters and check for an appropriate validation error.
14. Input an invalid vehicle registration number and ensure the form does not submit and displays an error.
15. Try entering a customer name that contains numbers or special characters and check for validation failure.
16. Enter an invalid email format and verify that the form does not accept it.
17. Input a phone number that does not match the required Kenyan formats and check for an error message.
18. Select a vehicle make but leave the model field empty and verify if the form prompts for completion.
19. Enter a vehicle color that is too short (fewer than 2 characters) and check for an error.
20. Input a vehicle location that exceeds the maximum character limit and ensure validation catches it.
21. Attempt to enter a negative loan amount and verify that an error is displayed.
22. Test the form submission with valid data but simulate a network failure to see how the user is informed of the error.
23. Check if the form can handle rapid changes in the job type selection and ensure that the dependent fields update accordingly without errors.
24. Simulate a user trying to copy and paste invalid data into the fields and verify that the validation mechanisms prevent submission.
25. Test the form's behavior when a user tries to submit it with all fields filled with valid data but without the required documents uploaded.
26. Verify the responsiveness of the form by testing on different devices and screen sizes to ensure all fields are accessible and usable.
27. Check how the form behaves when a user tries to navigate away from the page with unsaved changes and if a warning is displayed.
28. Test the form with a screen reader to ensure that all labels and error messages are correctly announced to visually impaired users.
29. Simulate a user who uses browser autofill to fill in the form and verify that the autofilled data meets the validation criteria.
30. Test the form with various languages and character sets to ensure that it correctly handles international characters in name and address fields.
31. Evaluate the user experience when navigating through the form using only the keyboard, ensuring all fields are accessible and manageable.

Ticket Button:

1. Verify that clicking the link navigates the user to the "New Ticket" page.
2. Ensure that the link is visually distinct and easily identifiable as clickable.
3. Check that the link is accessible via keyboard navigation (e.g., tabbing).
4. Confirm that the link is functional on various devices (desktop, tablet, mobile).
5. Validate that the link opens in the same tab as expected.
6. Test what happens if the link is clicked when the user is not logged in (should redirect to a login page or show an error).
7. Verify that the link does not work if the user has JavaScript disabled.
8. Check the behavior of the link when the URL is malformed (e.g., href=/newticket should be href="/newticket").
9. Ensure that clicking the link does not result in an error page (404 or otherwise).
10. Confirm that the link does not navigate if the user has a popup blocker enabled.
11. Simulate a scenario where the user has a slow internet connection and verify that clicking the link shows a loading indicator.
12. Create a test case where the user tries to access the link multiple times rapidly and check for any performance issues or unexpected behavior.
13. Test the link with various screen readers to ensure that it is properly announced and understandable.
14. Check how the link behaves when the user is on a page with JavaScript errors (should it still be clickable?).
15. Validate that the link is still accessible and functional when the browser window is resized or in different orientations (landscape/portrait).

New Ticket:

1. Verify that a user can successfully select a job type from the dropdown and that the selection is reflected in the form.
2. Check that the user can enter a valid vehicle registration number and that it adheres to the specified pattern.
3. Ensure that the user can input a valid email address and that it is accepted by the form.
4. Test that the user can enter a valid customer phone number format and that it is accepted by the form.
5. Validate that the user can select a vehicle make from the dropdown and that the selection is saved correctly.
6. Confirm that the user can input a loan amount that is a positive number and that it is accepted by the form.
7. Check that all required fields display the correct validation messages when left empty upon form submission.
8. Attempt to submit the form with an empty job type selection and verify that an error message is displayed.
9. Input an invalid vehicle registration number (e.g., "XYZ123") and check that the form rejects the input.
10. Enter an invalid email format (e.g., "user@com") and ensure that the form does not accept it.
11. Input a customer phone number that does not match the required pattern (e.g., "123456789") and verify that the form shows an error.
12. Try selecting a vehicle make but then modify it to an invalid value (e.g., "INVALID\_MAKE") and check if the form provides feedback.
13. Enter a negative loan amount and check that the form does not accept it.
14. Leave the customer name field blank and verify that an appropriate error message is displayed upon submission.
15. Test the form's behavior when a user rapidly changes the job type selection multiple times to see if it handles quick interactions without errors.
16. Simulate a user copying and pasting a very long string into the subject field to check if the form correctly enforces the maxlength constraint.
17. Verify that the tooltips provide the correct guidance when hovering over the input fields, especially for complex patterns like phone numbers and registration numbers.
18. Test the form's responsiveness by resizing the browser window and ensuring that all elements remain accessible and usable.
19. Check if the form correctly handles autofill from the browser when the user has previously filled similar forms, ensuring that the correct data is populated.
20. Simulate a scenario where a user starts filling the form but then navigates away and returns later, ensuring that previously entered data is retained.
21. Validate the form's behavior when JavaScript is disabled, ensuring that it still provides meaningful feedback or error messages where applicable.