This is our main product and the landing page https://csgoempire.com/roulette
How would you approach the testing of roulette? Describe the tests that you would do and the things you'd pay attention to.

Add your solutions to a repo and share the link to it with us below.

1. Functional testing
   For fuctional testing, I identified several functional areas:
   a. Betting options
      i. Verify that when a user places a bet on orange, black, or dice, the corresponding coins are correctly wagered. Ensure that the payout for winning bets is accurate, double for black or orange coin and up to a x14 multiplier for dice bets. Verify that users who lose their bets also lose the corresponding amount of money. This testcase should be also verify for three users and more than three to check the scalability of the betting system.
      ii. Verify that the maximum bet amount is 100.000 coins and the minimum bet amount is 0.01. Verify that user cannot bet more than 100.000 coins and less than 0.01 coins.
      iii. Verify that user cannot bet on more than one option.
      iv. Place bets one by one and verify that after betting is finished, the user's account balance is updated in real-time after each win or loss.
      v. Test that user is no longer able to place bet when the spinning started and is in progress.
      vi. Verify that the user is not able to bet if they have no coins left, and when attempting to bet, they are instructed that they need to deposit.
   b. Roulette buttons
      i. Test the functionality of buttons by clicking them in random orders. Verify that the betting amount computation is accurate after each button click. Ensure that no unexpected outputs occur after clicking buttons intended to modify the bet amount.
      ii. Verify Clear button functionality. Verify that after clicking of the clear buttons the bet amount is set to default value 0.00.
      iii. Verify that the Max button sets the bet amount as the maximum coins that the user has in their account.
      iv. Verify that he input number is always converted to a two-decimal number format. For example, if the user types '2', it will be converted to '2.00'.
      v. Verify special characters behavior:
         - For a bet amount that contains a number and a special character (such as !@#$%^&*()_+`-=[]{};'/|), only the integer part of the filled-in number will be displayed when the user finishes typing. For example, '47/2' is converted to '47.00'. This case applies only if the number is the first part of the input. If the special character comes first and the number follows it, the displayed bet amount is 'NaN'.

        - For a bet amount that is a floating point number and contains either a comma or a dot, when the user finishes typing, the displayed amount always contains just a dot. For example, '15,20' is converted to '15.20'.
        - For a bet amount that consists of just one or more special characters input (such as !@#$%^&*()_+`-=[]{};'/|,.) in the bet amount field, the displayed amount will be 'NaN'. This case includes both comma and dot.
    vi. Verify strings behavior:
        - For a bet amount that consists solely of a string input, the displayed amount will be 'NaN'.
        - For a bet amount that contains both a number and a string, only the integer part of the filled-in number will be displayed when the user finishes typing. This case applies only if the number is the first part of the input. For example, '34ab' is converted to '34'. This case applies only if the number is the first part of the input. If the string comes first and the number follows it, the displayed bet amount is 'NaN'.
    vii. Verify that an error is displayed when the user attempts to bet a negative value.
    viii. Verify that when the user inputs a bet amount with leading zeros, all leading zeros are condensed into a single zero, and the following digits become the decimal part of the number. For example, the input '00031' is converted to '0.31'.

c. Help system
    i. After clicking the Help button, in the Home tab, verify the correctness of the help topics available (Trade Punishments, Daily Coins, XP & Bonus Cases) both grammatically and in accordance with the documentation.
    ii. Test the Search bar to ensure it returns proper results containing the needed information.
    iii. Test the search bar with a query that does not match any help topics and ensure that no results are found. Additionally, verify that a proper message is displayed to let the user know that no results were found for this query.
    iv. Verify that the 'Start a conversation' button functions correctly and that the bot forwards the conversation to the help team when necessary.
    v. Verify that the user is not allowed to send a message that exceeds the character limit set for the message bar.
    vi. After clicking the Help button, in the Help tab, verify the correctness of the help topics available (Account, Deposits, Withdrawals etc.) both grammatically and in accordance with the documentation.
    vii. Check that in the Messages tab of the Help section, the user is able to see the previous conversations that they had with the help team.

d. Chat rooms
    i. Verify that user is able to switch between all options of chat rooms that are available and to send messages anf execute the other actions that are available for the chat rooms. (Send a tip, Chat rules, Manage Muted Users, Close chat)

        ii. Verify that user is able to send messages from message bar, and the three dots button allows the user to Send a tip to other player, access the Chat rules, Manage muted user and Close chat.
        iii. Verify that users cannot send messages using prohibited words, characters or content that should be blocked by the chat filter.
        iv. Verify that the chat will not incorrectly block messages that do not contain obscene or inappropriate words.
        v. Verify that the user is not allowed to send a message that exceeds the character limit set for the message bar.
        vi. Verify that users are not allowed to send a message in the chat room unless they have placed a total bet of 250.00 coins or have a deposit of at least 1.00 coins.
        vii. Verify that the user is not incorrectly prevented from sending messages despite meeting the bet and deposit requirements.
2. Compatibility Testing:
    a. Mobile
        i. Test the website on different mobile operating systems such a Android, iOS, MIUI and versions
        ii. Verify that the website is responsive and functions correctly on various screen sizes, including smartphones with smaller screens.
        iii. Test the website on different mobile browsers such as Chrome, Safari, Firefox, Opera to ensure proper functionality.
        iv. Verify that various touch interactions as tapping and swipping work as expected on mobile devices.
        v. Verify that after the user uses the 'Request Desktop Site' option on mobile, the browser works properly and displays the desktop version of the website without functionality, usability, or layout issues
    b. Desktop
        i. Test the website in different operating systems such as macOS, Windows, Linux and versions of these.
        ii. Verify that the website is rendered properly on various desktop screen resolutions.
        iii. Test the website on different browsers such as Chrome, Safari, Firefox, Edge, Opera to ensure functionality.
        iv. Verify that the website is allows both mouse interaction and as well keyboard interaction.
        v. Verify that by attempting to navigate the website by tab key, the elements are focused in a logical order.
    c. Scalling
        i. Verify that the webpage is rendered properly on screens with various resolutions such as 1920x1080, 1366x768, 1024x768.
        ii. Verify that the website's content remains readable and usable at different browser zoom levels.
        iii. Verify that the website is still usable when the browser window is resized or maximized or minimized.

        iv. Verify that the website is readable and usable with various scaling settings on the device or operating system.

3. Localization Testing
    a. Test the page with different language settings to ensure that all text and content are properly translated and displayed. In this case, assistance from a translator or native speaker is required.
4. Security Testing
    a. Verify the pattern in user statistics, so if a user deviates from the pattern, their logs can be checked to detect fraud.
    b. Verify that error logs are properly configured to detect and respond to security incidents.
    c. Verify the behavior of the webpage when security headers are missing such as CSP or X-Content-Type-Options.
    d. Verify access control by checking that users are not able to access unauthorized pages such as the admin dashboard or other administrative pages by trying links like www.csgoempire.com/admin or similar variations.