**Automated Test Cases:**

1. **Roulette Wheel Spin:**
    - **Description:** Automated test to simulate the spinning of the roulette wheel and validate its behaviour.
    - **Actions:**
        - Open the roulette game interface.
        - Initiate the spin action programmatically.
        - Capture the outcome of the wheel spin.
    - **Expected Result:** The roulette wheel spins smoothly and stops at a random position, with the outcome accurately captured for further validation.
2. **Bet Placement:**
    - **Description:** Automated test to systematically place bets on different bet types and amounts.
    - **Actions:**
        - Navigate to the bet placement section of the game.
        - Programmatically place bets on various bet types (colour, number) with different amounts.
        - Verify that bets are successfully placed and reflected in the game interface.
    - **Expected Result:** Bets are accurately placed according to specified parameters, and the game interface displays the placed bets correctly.
3. **Winning Bets and Payout Calculation:**
    - **Description:** Automated test to validate winning bets and calculate payouts accurately.
    - **Actions:**
        - Place bets on winning outcomes programmatically.
        - Simulate the roulette wheel spin to determine the winning number/colour.
        - Verify that winning bets are correctly identified and payouts are calculated accurately.
    - **Expected Result:** Winning bets are accurately identified, and payouts are calculated correctly based on the game outcome.
4. **Stability under Load:**
    - **Description:** Automated test to simulate multiple concurrent users placing bets to assess game stability.
    - **Actions:**

        - Simulate multiple virtual users accessing the game simultaneously.
        - Programmatically perform actions such as placing bets and spinning the wheel for each user.
        - Monitor for any performance degradation or system crashes.
    - **Expected Result:** The game remains stable under load conditions, with no performance degradation or system crashes observed.

5. **Consistency of Outcomes:**
    - **Description:** Automated test to verify the consistency of game outcomes across multiple sessions.
    - **Actions:**
        - Programmatically initiate multiple game sessions with the same initial conditions.
        - Capture and compare game outcomes (winning numbers/colours) across sessions.
    - **Expected Result:** Game outcomes remain consistent across multiple sessions, ensuring reliability and fairness.
6. **Response Time:**
    - **Description:** Automated test to measure the response time for various game actions.
    - **Actions:**
        - Programmatically initiate actions such as placing bets and spinning the wheel.
        - Measure the time taken for each action to complete.
    - **Expected Result:** Response times for game actions meet specific criteria ensuring smooth gameplay experience.

Reasons why the above are manual tests:

Automation ensures that test cases are executed consistently and repeatedly, reducing the likelihood of human error. Automated tests can be run more efficiently, saving time and effort over manual testing. Automating critical functionalities like bet placement and outcome validation ensures thorough coverage and accuracy. Automation makes it easier to test under various conditions (load, response time), allowing for a more comprehensive performance assessment.