Project Title: Movie Roulette

Test Objectives:

* Functional Testing: Ensure the software correctly assigns movies to each group member.
* Boundary Testing: Check whether the software can handle edge cases (e.g., max number of users, movies, and validate inputs).
* Usability Testing: Ensure the application is easy for our intended audience.
* Performance Testing: Check if the software can handle multiple users and decent-size groups efficiently.
* Security Testing: Identify vulnerabilities in the security including data handling, privacy, and access control.

Test Approach:

* First Test (Valid Data):
  + Objective: Ensure software correctly assigns movies when given the correct input.
  + Test Case: Provide a list of 5 users and 10 movies, then verify each person was assigned the corresponding movie selected for them.
  + Expected Outcome: Each of the 5 users should be assigned exactly one movie.
* Second Test (Boundary Data):
  + Objective: Test the software’s handling based on set boundaries.
  + Test Case:
    - Provide 1 user and 1 movie
    - Provide 10 users and 10 movies
    - Provide 100 users and 100 movies
  + Expected Outcome: The software should be able to handle each case and assign movies without any errors.
* Third Test (Out of Bounds Data):
  + Objective: Make sure the software can handle out-of-bounds data.
  + Test Case:
    - Make sure that only a valid number of users are being assigned a movie and that all remaining movies are unassigned.
    - Invalid entries for movies or users.
  + Expected outcome: All users should be assigned a movie, and the system should prompt the user to re-enter when an invalid entry is put in.

Test Tools:

Test Environment:

* Platform: Rayne’s MacBook (OS)
* Virtual Studio Code and Google Chrome
* Software Version: macOS 14.3.1

Test Criteria:

* Functional Requirements: The system needs to be able to assign or randomly generate movies to users in each group based on user input.
* Non-functional Requirements:
  + Performance: The system needs to assign movies efficiently (Under 5 seconds for 100 users).
  + Security: Sensitive data must be stored and managed correctly

Test Schedule:

* Dates: November 5th, November 7th, November 12th, November 14th November 18th, and November 21st
* Duration: 1-2 hours
* Time Slots: 3:00 PM – 8:00 PM

Test Team:

* Test Lead: Rayne Guinta
  + Testers:
    - Gavin Stierstorfer

Additional Testing Objectives:

* Functional Testing:
  + Check that the UI has easy entry for both users and movies.
  + Ensure the movie assignment is random based on the user’s preferences when the sender chooses to randomly generate a movie.
* Performance Testing:
  + Test the system’s performance when handling the maximum number of users and movies (e.g., 100 users and 100 movies to choose from).
  + Make sure the time taken to send and receive movies is acceptable for our criteria.
* Security Testing:
  + Make sure user data is being stored correctly and no sensitive information is exposed.
* Usability Testing:
  + Check that the UI is user-friendly, easy to use, and intuitive.
  + Confirm that the instructions are clear, and the website provides error-handling alternatives if an incorrect entry is applied.

Test Strategy:

* Test Inputs:
  + Each user and movie will be inputted manually using the UI, and a lost of names for both the users and movies will be used.
* Expected Outputs:
  + For the valid inputs, the system will assign movies to each user.
  + For invalid inputs, the system should handle errors by showing an alert.
* Test Reporting:
  + After each test, the results will be documented, and if there are any issues or unexpected behaviors, they will immediately be logged and addressed for the next iteration.