Background Questions:

1. What are the questions you would ask the team, and context you would want to know, before devising a test plan?

A: I would really like to know if there is a documentation of requirements of the product/app/game that you want to implement or any specific details, game plan from dev or what to expect of the app from Project Manager or Product Owner, maybe some specific functionality that was not implemented yet, but you guys plan to implement it in the near future (next sprint)
Most importantly I would really love to know customer expectation for whom this game build for.

If there is not requirements I will use Quality Control Standards, as well my experience with UI - HTML5 and CSS3 standards to determine - user quality and expectations.

2. What do you believe are the right testing technologies for this assignment and why? (What technologies would you *not* recommend for this game and why?)

A: I personally have would use SeleniumWebDriver, with any language (personally prefer Python for simplicity while working in the teams) on Cucumber for test cases for this particular product, TestRail and with extensions for JIRA for IDEA, Git for code sharing and Jenkins for CI to run the builds while they on the stage environment for the, API using Postman or Charles Proxy if needed manual for security, SQL for database user creation validation if there would be.

3. Describe your test plan. What parts of the test plan are you choosing to execute and submit to ClickTime and why?

A: Very often while working on the application we don’t have time to cover all 100% functionality of the application.

So, first thing I would do is to prioritize the the test plan around main functionality in order to catch as much, error as possible, accent on P0 and P1(Blockers of course) like login, user creation, functionality that uses most often, while those that can be fixed later P3-P4 like UI design I would leave for later, as polishing,

Test Plan - that will cover most of the functionality of the application, repetitive I would put on Automation for the Regression and Acceptance, to run for every sprint, while writing framework that can be reused in the future in case of changes, by giving management - time expectation of how long it will take QA to test it.

Also while working with application good to have history from bug triage from QA, where very often bugs that were introduced in the past can reappear as a result of code change in the future, based on that you can create the Test Plan - as well as test cases that will cover most of the in the same area, advanced test cases that will cover not just functionality but will cover if else cases, as well as error handling.

Test Plan

Category selection - section:

- Verify that user can open the webpage - https://clicktime-wordgame-exercise.vercel.app/ with 200 code - *(can be done in the Postman)*

- Verify that user - can enter valid credentials in “Username” field

- Verify that user - can *choose/select/click* categories from drop-list *(example “TV Shows”)* - and able to observe - active “Load Game” button - upon choosing the category. (gray > blue)

- Verify that user is able to start - the game starts after user - pressed “Load Game”

- Verify that user - can change the selection from the “drop-list - Categories from “TV Shows” back to Select Category” and Load button - become inactive (blue > gray)

“Welcome to the WordGame test! - section:

- Verify that user - is able to press all valid characters, and finish the game “FRIENDS” - validate that user see’s - “You have missed 0 guess(es), you have 6 attempt(s) left. Congratulations you just won” - text

- Verify that user are able to press all 27 characters, and they become inactive upon pressing

- Verify that user - after pressing 6 wrong letters/char - 27 characters become inactive(white > gray)