GIG QA Analyst - TECHNICAL CHALLENGE

The concept behind this exercise is to gauge the candidate's skills and knowledge in areas related to basic testing and automation.
In general, a candidate is expected to be well versed in the following areas which are prerequisites to completing this challenge:

- Any other IDE of your choice;
- Knowledge of Git. For this challenge, kindly use: o https://github.com
- Good knowledge Javascript (and/or Java / Python / C#);
- Testing frameworks such as Cypress (and/or / Selenium / Playwright);
- Restful API

You will need to install the required testing framework packages and / or extensions.

The choice of the application's target framework and output type is up to you. Just make sure to verify compatibility for any packages you might be using and how you intend on running the tests.

It is important that your code is easy to understand and follow. Where needed, concise comments may be used. Proper use and understanding of classes, methods, etc. Finally, the solution must be built and run successfully.

All code, screenshots, etc must be pushed to a repository of your choice on GitHub. Share a project and url to the repository (by email) a day before the deadline with QA manager's email - darko.pavlovic@gig.com . Candidates are given a maximum of 1 week to complete and submit the challenge.

GOOD LUCK!

1. Start by building a solution for the foundation of which is to retrieve user data via REST API from a given source. User data must be stored as objects. For this challenge, you will be using https://gorest.co.in to retrieve user data.

2. Expanding on 1, write the following tests:

TEST 1: Retrieve a list of users and verify that at least one user is being returned. The test fails when a response is not returned, or the list contains zero users.

TEST 2: Retrieve a list of users and verify that there is at a minimum one user whose name starts with the letter C. The test fails either when a response is not returned, or no usernames match the set criteria.

TEST 3: Retrieve a list of users and display it in the console. The test fails when a response is not returned, or the list contains zero users.

3. Using the same solution, write a test that uses a framework of your choice to test a scenario in which a user adds an item to a shopping cart and successfully checks out. Each step of the test must be asserted. For instance, when logging in, ensure that the correct page has indeed loaded. The test result is considered a PASS only if all the steps execute successfully.

TEST 4:
- Navigate to site https://www.saucedemo.com
- Log in as standard_user using password secret_sauce
- Buy an item of your choosing.
- Complete the checkout process.
- Close the browser.

4. Candidates can be (optionally) asked to demonstrate and run their tests.