SDET Assignment

OVERVIEW

The purpose of this assignment is to create the test plan and an indicative automation implementation for some basic Rudderstack flows.

SPECIFICATIONS

The flows can be automated using any JS/TS-supported frameworks (WebdriverIO, Playwright, Cypress). **Extra points for implementation using WebdriverIO (webdriver.io).** Use **CucumberJS** as a test runner with any of the frameworks.

How to Submit the Assignment:

1. Create a GitHub repo
2. Add tests as mentioned in the problem statement
3. Make the repo public and share the link

Problem Statement:

1. Create a basic framework using webdriverIO and CucumberJS**. (or any other framework)**
2. The framework should include an **.env file** to manage credentials for different environments like dev, qa, and production
3. The framework should use a proper structure with all the configuration files (package.json/tsconfig), readme, etc
4. The framework should include a GitHub workflow to execute the tests on a daily schedule.
5. Workspace setup (Manual): Sign up for a free tier account at https://app.rudderstack.com
   a. Use a business email for signup; do not use Gmail. You can create a temp email account at https://temp-mail.org/en/ or a similar provider. Do not sign up for an open-source account; use the Rudder Cloud option.
   b. Create an HTTP source - how-to-add-a-source
   c. Create a Webhook destination - how-to-add-a-destination

d. You can use https://www.requestcatcher.com or similar providers to create a webhook URL and use that as the destination

Automation:

a. Log in to the application and navigate to the connections page.
b. Read and store the data plane URL provided in the top right corner of the page.

c. Copy and store the write key of the HTTP source that you created in step 4b.

d. Use an api call to send an event to the HTTP source using the write key & data plane URL.

Details here - https://www.rudderstack.com/docs/api/http-api/

e. Click on the webhook destination that you created in 4c.
f. Check the events tab of the destination.
g. Read the count of delivered and failed events.