**Home Challenge**

Thank you for taking the time to do our Home Challenge. It consists of two parts: First part is about creating a test plan for one of the money management applications. The second part is about writing a set of automated checks for a sample REST API.

Please submit your results by sending us a link to your **GitHub repository** containing:

* A single **markdown file** with the exploratory testing charters.
* A separate folder containing test automation implementation.

Tasks

1. As a **Software Quality Engineer,** you are given the task of performing exploratory testing on one of the Monefy mobile applications. Choose the platform you are more familiar with (iOS or Android) You can download the apps here:

* [iOS](https://apps.apple.com/us/app/monefy-money-manager/id1212024409)
* [Android](https://play.google.com/store/apps/details?id=com.monefy.app.lite)

Monefy is a money management app where you can add your expenses in a dedicated category and track your spendings. You can review your expenses in detailed charts.

Task

Please write down:

* Exploratory testing charters to document your testing. You can use the simple format of: Explore target, with resources, to discover information.
* Findings from your charters. Did everything work as expected? What bugs were discovered?
* Prioritization of those charters - which area of the app or testing would you explore first and why? How much time have you planned to spend for each charter?
* What kind of risks do you need to mitigate for this type of application?

|Note: We want to see how you approach and organize exploratory testing and prioritize your work. **Please do not spend more than 2-3 hours on this task.**

1. As **Software Quality Engineer** you are given the task of testing a RESTful API. Use the Swagger Petstore Sample training tool designed to learn about an API framework in a local environment. Set up your environment as explained in the [readme](https://github.com/swagger-api/swagger-petstore).

Task

* Set up the API playground and run it locally. A list of test cases proposed for automation.
* Please automate proposed test cases.
* A short explanation of the provided solution.

|Note: Make sure your **test automation solution** is **NOT a Postman suite** but a **Framework** built using best **coding/design** practices.

Rules

* Feel free to use whatever frameworks, library, packages you like Use any familiar structure or pattern you are comfortable with.
* Include README about how to use the test suite. Take care about explaining the setup, as we might not have the same library installed.
* Please explain your approach and why you chose the particular tech stack.

Look for all references you need, you can find our technology stack below

Tech at YellowPepper

YellowPepper is completely hosted in the cloud. Our tech stack includes Kotlin, Java, Docker, Vault, Terraform, Spring boot, Kubernetes, GCP, Cypress, nodeJS, loadimpact/k6, etc. These tools and technologies are fully embraced by our test frameworks. Bonus points if you recognize most of them.

Feel free to let us know if you have any questions.