Technical Test #4

Question #1 requirement

1. Develop a function `f(n)` that will be looping from 1 to n
2. During the looping, when the counter is divided by 3, print "fizz"
3. During the looping, when the counter is divided by 5, print "buzz"
4. During the looping, when the counter is divided both 3 and 5, print "fizzbuzz"
5. else print out number
6. In the end of the program, run `f(15)`
7. Keep the program until the demo session

Question #2 requirement

You are building an API Service for a Random Hash Machine(RHM) application.

1. Develop an endpoint #1, where it will be returning a random hash string using SHA-256 algorithmn. In every hash generation, it will always return a unique hash string and not repeatable. Endpoint #1 shall only return after 1 seconds.
2. Develop an endpoint #2, where it will be keep requesting a hash string from endpoint #1, and endpoint #2 shall only return a success response body when last 1 character of the hash is a number and it is an odd number.
3. Write a load test to test endpoint #2, with 1 request per second.
4. Optional: Please investigate if there is anyway to increase the performance of the RHM if possible.

```
Example of SHA-256 hash:
a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae4
The last 4 character are `4`. This is an even number. Does not Pass.
Example of SHA-256 hash:
a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27aeB
The last 4 character are `B`. This is a alpabent. Does not Pass.
Example of SHA-256 hash:
a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27a33
The last 4 character are `3`. This is a number and odd number. Pass!
```

Note

1. You may use any framework that you are comfortable with(Please inform the interviewer your preference before starting), unless your interviewer specify.
2. Connect it to the git for version control before your start working on the project. Every feature/commit should be submit separately since beginning
3. Prepare a short documentation in the readme.md at the root folder, please include a screenshot of the app as well if there is.
4. Please do consider if any potential loophole and ensure that the application is durable.
5. If you need any clarification, please contact your interviewer.

6. Share git repository as public to the interviewer once it is ready.