**Fortinet – Home exercise:**

**Tasks:**

1. Implement the system we designed in the first interview(system requirements are attached in another file). Out of the 3 reputation clients, implement only Bamboo and Avocado.
   1. Documentation:
      1. Avocado docs: [http://34.134.220.86:8080/docs#/](http://34.134.220.86:8080/docs" \l "/)
      2. Bamboo docs: [http://34.134.220.86:8081/docs#/](http://34.134.220.86:8081/docs" \l "/)
2. Create a webserver. The webserver will have one REST API, `classify\_file`. The API’s input and output should be identical to the system input and output implemented in section 1.
   1. Hashes(SHA1) that exists in Avocado and Bamboo(should be used for testing purposes):
      1. d9bbb4e4900ff03b0486fac32768170249dad82d
      2. cf60fa60d2f461dddfdfcebf16368e6b539cd9ba
      3. 77a8060d1629183e457fdbc2f34143a5070bdd47
3. Install docker on your pc. Create a **dockerfile** for the service (which installs and runs the service).
4. Implement **tests** for the service*.* Use your preferred testing framework. Assume that a file’s classification can change from time to time (For example, for a specific hash, Bamboo returns 90 today but next month it might return 80).
5. Bonus: implement a **cache**. In order to reduce the amount of requests for each reputation client
   1. We are planning to have this service “scaled-out” (deployed simultaneously on multiple machines). Take it into account while implementing the cache.

**Notes:**

1. Please use your **best practices -** Conventions, documentation, function & variable names, etc.
2. **Quality over quantity** – don’t rush, we prefer high quality code over finishing all the tasks above.
3. You can use the internet freely.
4. Use any programming language you’d like.

Good luck!