My first goal was to get the list of theoretical questions.

So in a cmd prompt I ran: “docker image pull yondermakers/yonder-devops-tech-assessment:latest” to pull the image.

Then I ran: “docker run –p 30000:8080 yondermakers/yonder-devops-tech-assessment:latest” to start the docker container. Here I bound the port 30000 from the host to port 8080 of the SpringBoot application.

At this point I wanted to see if a simple get request to “/” would work, using Postman.

It worked and I copy-pasted the response body (html) to a text file, which I then saved as YonderQuestions.html so I can open it in Chrome.

My responses to the questions:

1. Queue: used in task scheduling algorithms and in graph traversal algorithms

Graph: used in modeling relationships between data (which would be nodes in this case) and there are algorithms like shortest path that are used for this collection of nodes.

1. The browser makes a request to a Domain Name Server (DNS) and gets back an IP address (almost like looking on a look-up table). The browser can then establish a connection to the web server hosting [www.tss-yonder.com](http://www.tss-yonder.com).

In the past, when I personally needed to check the IP of a certain site, I used a “whois” API/service or nslookup command.

In this case I get back multiple addresses (both IPv4 and IPv6) which means it is hosted on multiple servers.

Addresses: 2606:4700:20::681a:13e

2606:4700:20::ac43:49b1

2606:4700:20::681a:3e

172.67.73.177

104.26.0.62

104.26.1.62

1. TCP – email, browsing

UDP – real-time, low-latency applications like streaming and videocalls

1. Containerize it and deploy it on one of the hosting services providers (Google, AWS etc.)

That is if I do not care about the URL it will have or other details about it (which in this case I don’t)

1. I would add basic auth (which can be added as an option when deploying with one of those hosting service providers previously mentioned.
2. As far as I know, complete asymmetric encryption is too computationally expensive to do for each message, but what could be done is symmetric encryption with keys being distributed encrypted with asynchronous encryption (but in this case its only once per chat session instead of once per each message)
3. Cookies are small pieces of data stored on the user’s device, by the browser. They track and store information about various interactions the user had with the browser but there are also other use cases.

A cookie that my browser saved for tss-yonder.com is “cookieconsent\_status” with value “dismiss” which probably means that I already dismissed the pop-up asking for my consent related to cookies. (how ironic ☺)

1. In C#, with the use of System.Diagnostics, I can use Process childProcess = new Process();

But of course, there is StartInfo that has to be defined and the childProcess.StartInfo should take the value of that StartInfo (which is an object of type ProcessStartInfo that has attributes for FileName and Arguments).

Process states: Running, Ready, Blocked, Terminated, Zombie.(if im not wrong, Zombie only exists on Linux)

1. If by PID is meant Process ID, it can be checked in Task Manager (on Windows).
2. A NoSQL solution for various reasons ☺

Jokes aside, my intuition tells me that a relational database could not cope with huge numbers of writes (if this application would eventually get a lot of users)

So, answer: MongoDB? Cassandra? Elasticsearch?

When it comes to storing passwords: store the hash of the passwords.

Python console application:

I suppose there is no need to do functionality for the first GET (“/”) since it could be 100% done with postman.

When it comes to the rest of the task, the 150 data points should be extracted (from the GET to “/drivers-licenses/list?length=150”).

Then we need functions for listing suspended licenses, valid licenses issued until today’s date and licenses based on category and their count.

All these operations and listings plus the need to output in an excel file remind me of my Data Science project where I used of Pandas (which is what I am about to do here as well).

The operator input is required in the console at runtime.

As a final thing to mention, I’ve had a lot of fun with this task honestly and I congratulate whoever came up with this way of assessment.