# TECHNICAL ASSESSMENT

Steps taken:

* I used the command docker - docker run -p 30000:8080 --name yonderapp yondermakers/yonder-devops-tech-assessment:latest to run a container
* I used a browser to access the localhost on port 30000 to read the questions.

Answers:

1. Two data structures that I know are dictionaries and sets. I would use a dictionary when I need a pair of data to be linked together with a unique key, the pair needs to be easily accessible and requires a lot of lookups for specific values. I would use a set when I only need a list of data that can be accessed and managed quickly without the need for a pair data value.
2. The IPv6 address is: - 2606:4700:20::681a:13e (I used the ping command on the site provided to view the IP address or it can also be viewed using the inspector tools available in the browser) The browser looks at the DNS the user provided, it performs a search to the DNS server and finds the IP address associated with that DNS before it can retrieve the website.
3. Two transport protocols are the TCP protocol and the UDP protocol. The TCP protocol is used whenever we need to make sure that the data that is sent is also being received without the loss of data. This protocol is best used in a chatting app like WhatsApp or web pages. The UDP protocol is used when the server does not need to know if the client received the data and when some data loss can be afforded. Two applications that use UDP are video streaming applications and online games.
4. After creating the application using C# and PostgreSQL I will containerize the app using docker then I will push the docker image to the cloud so I can use it later. After this I choose the infrastructure ( server or serverless(auto managed) ) in this example I will use a EC2 instance. I will install the docker dependencies to the EC2 instance and run the container. I will use Inbound rules to map the application port to 80 and then assign an Elastic IP to the EC2 instance so it can be seen from the internet. After this I would use a domain provider to obtain and assign a DNS to the IP address of my application. At this moment the application will run without encryption so I would make use of Nginx to apply a reverse proxy so the application port remains unexposed, I would also use Certbot and Let’s Encrypt to generate and sign a ssl/tsl certificate.
5. One way would be to use a whitelist via the Security Groups service available in AWS. It allows us to assign specific IP addresses that can access the port limiting the access to all other IP addresses. Another way would be to add a kind of authentication using tokens or user credentials inside the application. A silly way that would only temporarily solve the issue would be to just change the port of the application to a rarely used one.
6. To prevent any kind of port listening and conversation spying I would use encryption. To implement encryption, I would use Nginx together with Certbot and Let’s Encrypt to automatically generate and sign a ssl/tsl certificate. This would make all data transmission between users to be encrypted.
7. Cookies are user related data that the browser keeps locally. They are used to store data about how the client interacts with the website. For example the shopping cart of an Ecommerce website can use cookies to remember each client’s basket. A cookie that the page uses is: PHPSESSID with value: 2rbaatmno40mptkc0lmkn3aitm and I assume it is used to store the session ID of the current session.
8. To create a child Thread in C# we need to create a new instance of the Thread class and pass a method to the constructor. The thread will execute the code that the method implements .The possible states of a C# process are unstarted, runnable, running , not runnable and dead.
9. I will investigate the error using the top command to see all processes, then I would use ps aux command to find the process of my application. With the PID I can list the log for that process located at /proc/<pid>/fd/ using the cat command. In my case the application runs inside a docker container so the command docker ps can be used to view all processes inside the container and then check the logs.
10. I would use a RDS service from AWS that is basically a relational database in the cloud. It functions like a normal relational database like PostgreSQL or MySQL or an application like MySQL Manager. This approach offers high accessibility and fast data access. For the password storing I would use the same database with the mention that I would encrypt the passwords using hash encryption.

* I used the endpoint /drivers-licenses/list to see the list of objects
* When coding the Python application I first defined a function to make a get request at the endpoint and to return the objects called get\_licenses.
* I defined another function that uses the get\_licenses function to iterate through the data and check if the “suspendat” value is true. If it is, at each iteration it print the entire item that meets the condition.
* For the valid licenses I created another function that also iterares through the data, cashes the value described by “dataDeExpirare” and converts it to a datetime format using the datetime library. A check is also made each iteration to only print the item that has datetime value greater than the current date and also has not been suspended.
* The approach I used to find the license number by category is by simply creating a counter variable for each category, loop through the list of objects and increment the counter variable each time a category matching the increment variable is found. After the loop is completed I print each counter variable along with the category it belongs to.
* For the input ID operations I used generic IDs to determine which operation should the script run inside the main function.
* For exporting data to a Microsoft Exel compatible format I decided to use the csv format since it is compatible with Exel and is also easily implemented with python.
* I adjusted each function to also generate the csv file in addition to printing the data in the console.