SDET Assessment

Hello,

Congratulations for making it through to the next stage of your application. There are two key aspects to being an awesome SDET.

1. You have a good testing mindset
2. You have solid programming experience.

To progress to the next stage of the application, please complete these two tasks:

1) Due to time constraints, you only have time to complete automation testing of 10 scenario's for this new website. https://jupiter.cloud.planittesting.com/#/

Determine 10 test cases which reflects the priorities of this website

Note: This is Planit's training sample website and no data will be saved or stored

2)Using the programming language of your choice, please choose 2 of following programming challenges and develop a solution for them. During your interview, we will go through your solution.

Please provide a github repo with your answers to both tasks.

Thank you, and I'm looking forward to meeting you in person.

Cheers,
Cuong Pham
Director - Automation

Challenge 1:

Write a solution that prints the numbers from 1 to 100. But for multiples of three print "Planit" instead of the number and for the multiples of five print "Testing". For numbers which are multiples of both three and five print "PlanitTesting".

For example:

Output: 1 2 Planit 4 Testing Planit 7 8 Planit Testing 11 Planit 13 14 PlanitTesting …etc.

Challenge 2:

Write a solution to find the character that has the highest number of occurrences within a certain string, ignoring case. If there is more than one character with equal highest occurrences, return the character that appeared first within the string.

For example:

Input: "Character"

Output: c

Challenge 3:

Write a solution that reverses a string and replaces any spaces with hyphens (-)

For example:

Input: "reverse order"

Output: "redro-esrever"

Challenge 4:

Build a program that can convert temperatures from and to Celsius/Fahrenheit/Kelvin. No need to create a UI, a main function that receives parameters is enough.

For example:

Input: 32, “C”, “K”

Output: “305.15 K”

Challenge 5:

Build a food ordering system where the user inputs the food name into the food order and then the order gets passed to the appropriate restaurant on a list and the restaurant prints the receipt on stdout. No need to create a UI, a main function that receives parameters is enough.

For example:
Input: “pepperoni pizza”
Output: "Awesome pizza place, pepperoni pizza, $20”

For example:
Input: “burger”
Output: “wild burger joint, burger, $15”

Challenge 6:

Given a list of people (each person has attributes like name, DOB, nationality) and in this world two people can’t have the exact same name then a user can:
* Obtain a list of duplicates from the original list
* Alter the original list and remove duplicates
* Calculate the average age
* Find all the people with age less than N
* Obtain a list of unique countries