**Automation Rehovot Tech Lead Assignment**

**Task 1 - API:**

Programming languages: C#, Python, Java

**Task Description**  
You are given the following APIs which are fake online Rest APIs for testing (the APIs are working).  
Your task is to implement and validate those APIs.

**Requirements**  
1. Create a new project.  
2. Implement the APIs calls for testing it.  
3. Create tests which will validate the APIs (**1 positive and 1 negative tests for each given API**) - please validate both the **status code** and the **response body**.  
  
**Assumptions**  
1. You are free to use any package or dependency that you want.  
2. You can use any programming language from the following: **C#, Java or Python**.  
3. You are free to implement the APIs calls in any way that you think of.  
4. Once completing the task, please **zip** it, upload it to **your google drive and send the public shared link of the zipped project to the following mails:**[tzahiYlahav@gmail.com](mailto:tzahiYlahav@gmail.com)[tzahi.lahav@qualitestgroup.com](mailto:tzahi.lahav@qualitestgroup.com) **The APIs:**  
1.   
**Route  
/api**/users  
  
**Method**  
GET  
  
**Type**  
JSON  
  
**Full route**  
https://reqres.in/api/users

**Description**  
Get list of users

**Response Example**  
{  
   "page":1,  
   "per\_page":6,  
   "total":12,  
   "total\_pages":2,  
   "data":[  
      {  
         "id":1,  
         "email":"george.bluth@reqres.in",  
         "first\_name":"George",  
         "last\_name":"Bluth",  
         "avatar":"https://reqres.in/img/faces/1-image.jpg"  
      },  
      ...  
      ]   
}  
  
2.   
**Route**  
/api/users/{id}

**Method**  
GET  
  
**Type**  
JSON  
  
**Full route**  
https://reqres.in/api/users/2

**Description**  
Get a single user data

**Response Example**  
{  
    "data": {  
        "id": 2,  
        "email": "janet.weaver@reqres.in",  
        "first\_name": "Janet",  
        "last\_name": "Weaver",  
        "avatar": "https://reqres.in/img/faces/2-image.jpg"  
    },  
    "support": {  
        "url": "https://reqres.in/#support-heading",  
        "text": "To keep ReqRes free, contributions towards server costs are appreciated!"  
    }  
}

3.  
**Route**  
/api/users  
  
**Method**  
POST  
  
**Type**  
JSON  
  
**Full route**  
https://reqres.in/api/users  
  
**Request Body Example**  
{  
    "name": "morpheus",  
    "job": "leader"  
}

**Description**  
Create new user

**Response Example**  
{  
    "name": "morpheus",  
    "job": "leader",  
    "id": "563",  
    "createdAt": "2024-04-17T05:36:40.461Z"  
}  
  
4.  
**Route**  
/api/users/{id}  
  
**Method**  
PUT  
  
**Type**  
JSON  
  
**Full route**  
https://reqres.in/api/users/2  
  
**Request Body Example**  
{  
    "name": "morpheus",  
    "job": "zion resident"  
}

**Description**  
Update a user record

**Response Example**  
{  
    "name": "morpheus",  
    "job": "zion resident",  
    "updatedAt": "2024-04-17T05:38:37.770Z"  
}

5.  
**Route**  
/api/users/{id}  
  
**Method**  
DELETE  
  
**Type**  
JSON  
  
**Full route**  
https://reqres.in/api/users/2

**Description**  
Delete a user record

**Task 2 - UI:**

Programming languages: C#, Python, Java

**Task Description**  
The following site is a simple online login page for testing purpose <https://practicetestautomation.com/practice-test-login/>

Your task is to implement an automated tests to validate whether the form works correctly.  
  
**Requirements**  
1. Create a new project.  
2. Use **Selenium WebDriver** in your project to perform the actions in the login page.  
3. Create 3 test cases to validate the login page (**The cases are shown both here and in the site itself**).  
  
**Assumptions**  
1. Before jumping to the code, you can play with the site to see how it works.  
2. Please use **Selenium WebDriver** for testing the site.  
3. There is no need to validate the site in multi browsers - it is enough to use **only 1 browser**.  
4. You are free to use any additional design pattern or methods that you want or need.  
5. You can use any programming language from the following: **C#, Java or Python.**  
6. Once completing the task, please **zip** it, upload it to **your google drive and send the public shared link of the zipped project to the following mails:**[tzahiYlahav@gmail.com](mailto:tzahiYlahav@gmail.com)[tzahi.lahav@qualitestgroup.com](mailto:tzahi.lahav@qualitestgroup.com) **Required Test Cases (Shown also in the site itself):**  
**Test case 1: Positive Login test**  
1.Open page  
2.Type username student into Username field  
3.Type password Password123 into Password field  
4.Push Submit button  
5.Verify new page URL contains practicetestautomation.com/logged-in-successfully/  
6.Verify new page contains expected text ('Congratulations' or 'successfully logged in')  
7.Verify button Log out is displayed on the new page  
  
**Test case 2: Negative username test**  
1.Open page  
2.Type username incorrectUser into Username field  
3.Type password Password123 into Password field  
4.Push Submit button  
5.Verify error message is displayed  
6.Verify error message text is Your username is invalid!  
  
**Test case 3: Negative password test**  
1.Open page  
2.Type username student into Username field  
3.Type password incorrectPassword into Password field  
4.Push Submit button  
5.Verify error message is displayed  
6.Verify error message text is Your password is invalid!

**Task 3 – OOP Concepts:**

Programming languages: C#, Python, Java

**Task description**  
You are given the following animals, their specific abilities and a set of rules:  
Goose - Eating, Diving, Screaming  
Duck - Diving, Screaming  
Dog - Eating, Screaming  
Fish - Diving

**Given Rules**  
1. Duck and Goose Diving in an identical manner.  
2. All the other animals abilities are not identical.  
  
Your task is to create a class hierarchy that supports the implementation of the mentioned animals and their abilities and minimizes code duplication.  
  
**Requirements**  
1. Create a new project.  
2. Create classes/interfaces/relations or any other design to describe the system that mentioned above.  
3. Add a **comment** to the project and explain **how your design minimized the code duplication**.

**Assumptions**  
1. You can use any programming language from the following: **C#, Java or Python.**  
2. You **do not need** to implement the animals abilities methods logic - you can do whatever you want in the method itself (the logic is irrelevant).  
3. Once completing the task, please **zip** it, upload it to **your google drive and send the public shared link of the zipped project to the following mails:**[tzahiYlahav@gmail.com](mailto:tzahiYlahav@gmail.com)[tzahi.lahav@qualitestgroup.com](mailto:tzahi.lahav@qualitestgroup.com)

**Good Luck!!!**