



Rapport des tests fonctionnels



"Application mobile visant à faciliter l'exploration et la découverte du Maroc pour les touristes et les visiteurs."

Réalisé par :

OUGNI Imane

Encadré par :

Mme. Asmaa RETBI

Mme. Asmaa EL KASSIRI M. Zohair ELMOURABIT

Sommaire

1.		Introduction
2.		Les tests fonctionnels et automatisation des tests
8	ì.	Les tests fonctionnels
ł).	Automatisation des tests
3.		Choix des applications à tester
a	ì.	Harmony Library3
ł	٥.	LAMBDATEST3
4.		Scénarios et cas de test
5.		Selenium
6.		Utilisation de Selenium IDE
a	ì.	Les tests
ł	٥.	Les résultats des tests6
C	٥.	Le code Java des tests
C	1.	JUnit
8.		Utilisation Selnium WebDriver
a	ı.	Test 1 : lancer une page LinkedIn
ł	٥.	Test 2 : lancer la page de connexion de LambdaTest
C	٤.	Test 3 : adapter un code de test pour utiliser un fichier CSV
9.		Conclusion

Liste des figures

Figure 1 : test de création du compte	5
Figure 2 : test de connexion	5
Figure 3 : résultat du test de création du compte	6
Figure 4 : résultat du test de création du compte	6
Figure 5 : le code JAVA du test de création du compte	7
Figure 6 : le code JAVA du test de connexion	7
Figure 7 : test du lancement de la page LinkedIn	8
Figure 8 : résultat de test du lancement de la page LinkedIn	9
Figure 9 : test du lancement de la page de connexion de LambdaTest (1)	9
Figure 10 : résultat du test du lancement de la page de connexion de LambdaTest (1)	10
Figure 11 : test du lancement de la page de connexion de LambdaTest (2)	10
Figure 12 : résultat du test du lancement de la page de connexion de LambdaTest (2)	11
Figure 13: test d'utilisation des ficher CSV	11
Figure 14: résultat du test du d'utilisation des ficher CSV	12
Figure 15: les ficher CSV d'entrée (test.csv) et de sortie (result.csv)	12

1. Introduction

Dans le domaine du développement logiciel, les tests fonctionnels et l'automatisation des tests jouent un rôle crucial dans la garantie de la qualité des applications. Les tests fonctionnels permettent de vérifier le bon fonctionnement d'une application du point de vue de l'utilisateur final, tandis que l'automatisation des tests offre la possibilité d'exécuter ces tests de manière rapide et cohérente, sans nécessiter une intervention humaine constante. Dans ce rapport, nous explorerons l'application de ces concepts à deux cas spécifiques

2. Les tests fonctionnels et automatisation des tests

a. Les tests fonctionnels

Les tests fonctionnels sont une méthode de vérification du bon fonctionnement d'une application ou d'un système en examinant ses fonctionnalités du point de vue de l'utilisateur final. Contrairement aux tests unitaires, qui se concentrent sur des parties spécifiques du code, les tests fonctionnels examinent le système dans son ensemble, en simulant les actions et les interactions que les utilisateurs réels effectueraient.

b. Automatisation des tests

L'automatisation des tests consiste à créer des scripts ou des programmes qui exécutent automatiquement des suites de tests sur une application ou un système, sans nécessiter d'intervention humaine. Cela permet de répéter facilement les tests et de les exécuter rapidement et de manière cohérente, ce qui est particulièrement utile pour les applications complexes ou en constante évolution. L'automatisation des tests peut inclure à la fois des tests fonctionnels et des tests unitaires, ainsi que d'autres types de tests comme les tests de performance ou de sécurité.

3. Choix des applications à tester

a. Harmony Library

C'est une application web de gestion de bibliothèque que j'ai développée en utilisant HTML, CSS, JavaScript et le framework Django de Python, avec une base de données SQLite. Cette application permet aux utilisateurs de parcourir et organiser une sélection de livres par catégories. Ils ont la possibilité de créer un compte, ce qui leur permet de louer ou d'acheter des livres disponibles. De plus, l'administrateur peut gérer le stock de livres, ainsi que les utilisateurs et leurs commandes.

b. LAMBDATEST

C'est une plateforme d'exécution de tests basée sur le cloud qui permet aux entreprises d'effectuer des tests manuels et automatisés sur des applications web et mobiles.

4. Scénarios et cas de test

Dans cette partie j'ai expliqué les exemples des scénarios et cas de test que j'ai tester en utilisant selenium

ID Scenario	Description	ID Cas de test	Description	Étapes	Données	Réponse attendu
Sc1	Vérifier la fonctionnalité de la création du compte	Cas 1.1	Création d'un compte avec email valides	-Accéder à l'application -Cliquer sur « créer un compte » -Enter les données -Cliquer sur « create account »	-Email : ougniimane@gmail.com -mot de passe : ammouna321	Redirection vers la page de connexion
		Cas 1.2	Création d'un compte avec email invalides		-Email : <u>ougniimane</u> -mot de passe : ammouna321	Message d'erreur « écrivez votre Gmail sous la forme 'abcd@gmail.com »
		Cas 1.3	Création d'un compte avec email déjà utilisé		-Email : ougniimane@gmail.com -mot de passe : ammouna321	Message d'erreur indiquant que l'email est déjà utilisé et qu'il faut essayer un autre email.
Sc 2	Vérifier la fonctionnalité de la connexion	Cas 2.1	Connexion avec des informations valides	-Accéder à l'application -Cliquer sur « log in»	-Email: ougniimane@gmail.com -mot de passe: ammouna321	Redirection vers la page d'accueil « home »
		Cas 2.2	Connexion avec email invalide	-entrer l'email et le mot de passe -cliquer sur « se connecter »	-Email: email_invalide@gmail.com -mot de passe: ammouna321	Message d'erreur indiquant que l'adresse e-mail n'est pas reconnue.
		Cas 2.3	Connexion avec password invalide		-Email : ougniimane@gmail.com -mot de passe : ammouna	Message d'erreur indiquant que le mot de passe est incorrect
		Cas 2.4	Connexion avec email et password invalides		-Email : <u>email_invalide</u> @gmail.com -mot de passe : ammouna	Message d'erreur indiquant que les informations sont incorrectes

5. Selenium

Selenium est en effet un framework open-source utilisé pour automatiser les tests d'applications web, avec ses composants principaux que sont Selenium WebDriver et Selenium IDE. Il offre une gamme de fonctionnalités qui répondent aux besoins variés des testeurs et des développeurs, des solutions simples pour des tests de base à des options plus avancées pour des tests complexes.

J'ai mis en œuvre les deux frameworks dans les exemples suivants, utilisant le langage Java pour les scripts de test avec Selenium IDE et le langage Python pour les scripts de test avec Selenium WebDriver.

6. Utilisation de Selenium IDE

C'est une extension de navigateur qui permet d'enregistrer et de rejouer des interactions utilisateur sur un navigateur web. Il offre une interface graphique conviviale pour créer des scripts de test sans nécessiter de compétences de programmation avancées. Cependant, il est limité en termes de fonctionnalités avancées et de personnalisation.

a. Les tests

Le premier scénario concernant la création du compte.

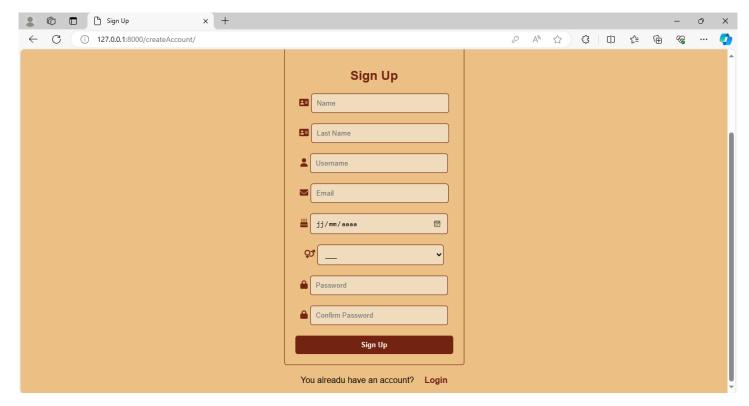


Figure 1 : test de création du compte

Le deuxième scénario est de la connexion

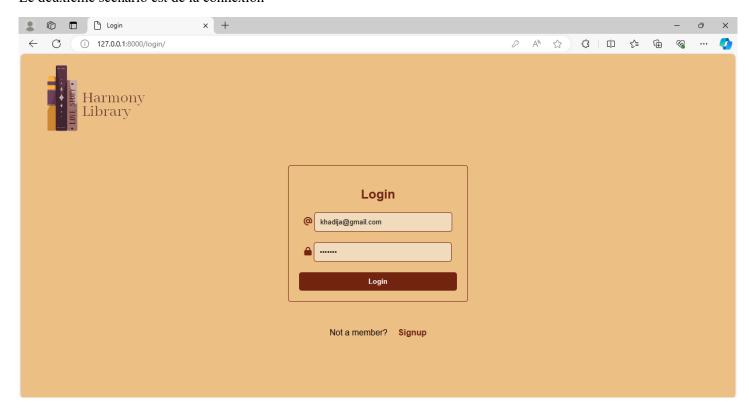


Figure 2: test de connexion

b. Les résultats des tests

Au cours du test, Selenium IDE vérifie chaque étape s'elle est passé avec succès ou échec

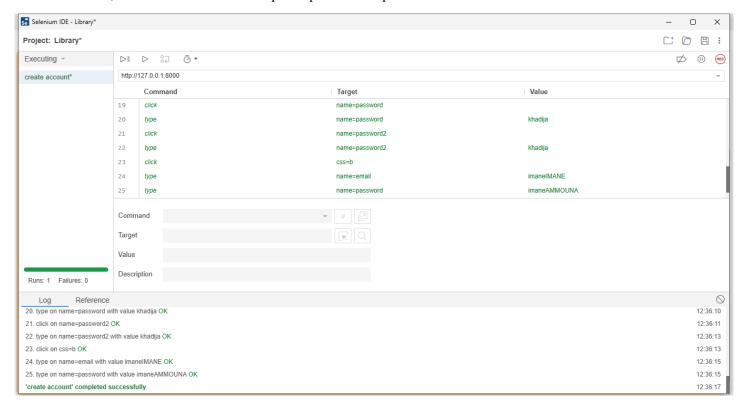


Figure 3 : résultat du test de création du compte

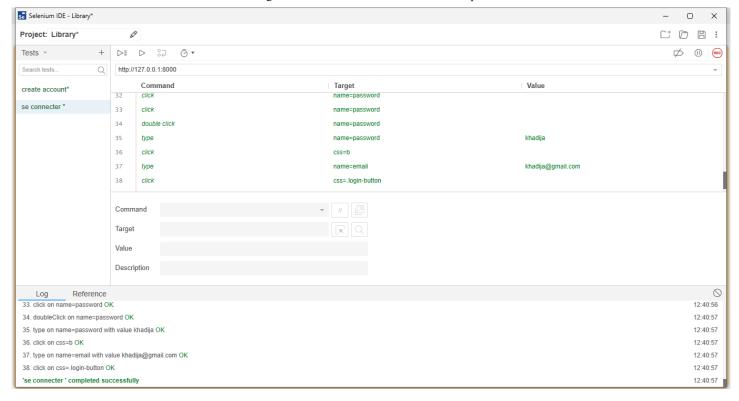


Figure 4 : résultat du test de création du compte

c. Le code Java des tests

Après que les tests sont effectués, le code est généré avec plusieurs langage de programmation

```
≺ File Edit Selection View Go Run
                                                                                                                                                                                                                                                                                                                                                                                                                       Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
                   J CreateaccountTest.iava X
                   C: > Users > IMANE > Downloads > J CreateaccountTest.java
                                           @Test
 ၀ဍ
                                               driver.get("http://127.0.0.1:8000/");
                                             driver.manage().window().setSize(new Dimension(1382, 744));
driver.findElement(By.cssSelector("li:nth-child(2) b")).click();
driver.findElement(By.name("nom")).click();
driver.findElement(By.name("nom")).sendKeys("khadija");
driver.findElement(By.name("prenom")).sendKeys("khadija");
driver.findElement(By.name("prenom")).sendKeys("khadija");
driver.findElement(By.name("username")).sendKeys("khadija");
driver.findElement(By.name("username")).sendKeys("khadija");
driver.findElement(By.name("email")).sendKeys("khadija@gmail.com");
driver.findElement(By.name("date_de_naissance")).click();
driver.findElement(By.name("date_de_naissance")).sendKeys("0020-12-05");
driver.findElement(By.name("date_de_naissance")).sendKeys("0020-12-05");
driver.findElement(By.name("date_de_naissance")).sendKeys("0200-12-05");
driver.findElement(By.name("date_de_naissance")).sendKeys("0200-12-05");
driver.findElement(By.name("date_de_naissance")).sendKeys("0200-12-05");
driver.findElement(By.name("date_de_naissance")).sendKeys("0200-12-05");
driver.findElement(By.name("genre")).click();
{
                                                driver.manage().window().setSize(new Dimension(1382, 744));
æ
                                                     WebElement dropdown = driver.findElement(By.name("genre"));
dropdown.findElement(By.xpath("//option[. = 'Femelle']")).click();
                                               driver.findElement(By.name("password")).click();
driver.findElement(By.name("password")).sendKeys("khadija");
driver.findElement(By.name("password2")).click();
driver.findElement(By.name("password2")).sendKeys("khadija");
(8)
                                                 driver.findElement(By.cssSelector("b")).click();
£
                                                 driver.findElement(By.name("email")).sendKeys("imaneIMANE");
           TRestricted Mode ⊗ 0 ♠ 0 ₩ 0
                                                                                                                                                                                                                                                                                                                                                                                                        Ln 1, Col 1 Spaces: 2 UTF-8 LF Java
```

Figure 5 : le code JAVA du test de création du compte

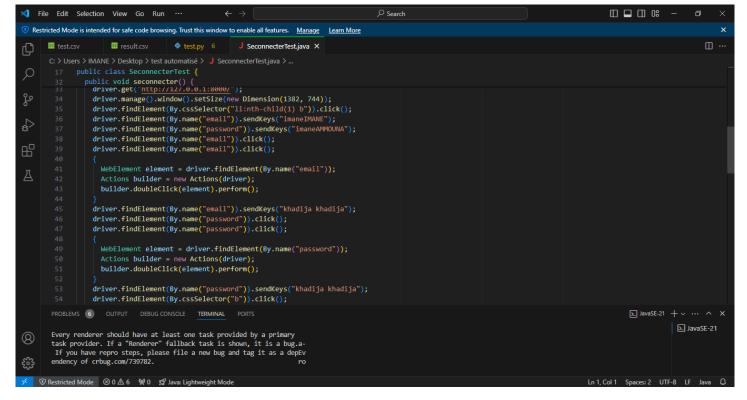


Figure 6: le code JAVA du test de connexion

d. JUnit

JUnit est un framework open-source de test unitaire pour le langage de programmation Java. Il permet aux développeurs de créer et d'exécuter des tests automatisés pour vérifier le bon fonctionnement des différentes parties de leurs applications Java. JUnit offre une structure claire et cohérente pour écrire des tests, avec des annotations permettant de définir les

méthodes de test, les méthodes d'initialisation et de nettoyage, ainsi que des assertions pour vérifier les résultats attendus des tests. Il est largement utilisé dans le développement logiciel Java pour garantir la qualité et la fiabilité des applications.

8. Utilisation Selnium WebDriver

C'est une bibliothèque de programmation qui offre une approche programmable pour automatiser les tests web. Les développeurs utilisent des langages de programmation tels que Java, Python, C#, etc., pour écrire des scripts de test qui contrôlent le navigateur et effectuent des actions spécifiques. WebDriver offre une plus grande flexibilité, une personnalisation avancée et une puissance accrue pour la création de tests automatisés complexes.

a. Test 1 : lancer une page LinkedIn

L'objectif de ce test est de lancer une page LinkedIn

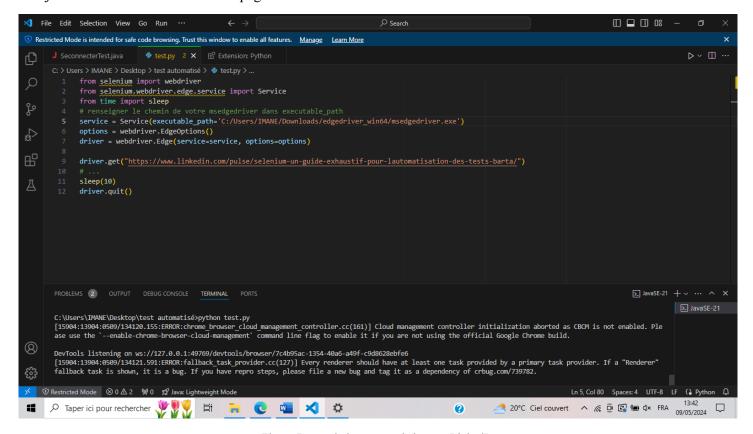


Figure 7 : test du lancement de la page LinkedIn

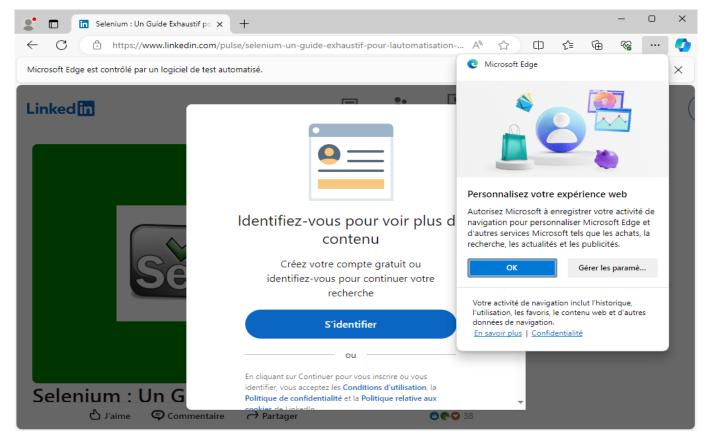


Figure 8 : résultat de test du lancement de la page LinkedIn

b. Test 2 : lancer la page de connexion de LambdaTest

L'objectif du deuxième test est de lancer la page de connexion de LanbdaTest avec des différentes données

* Test avec des données pour un compte qui n'existe pas

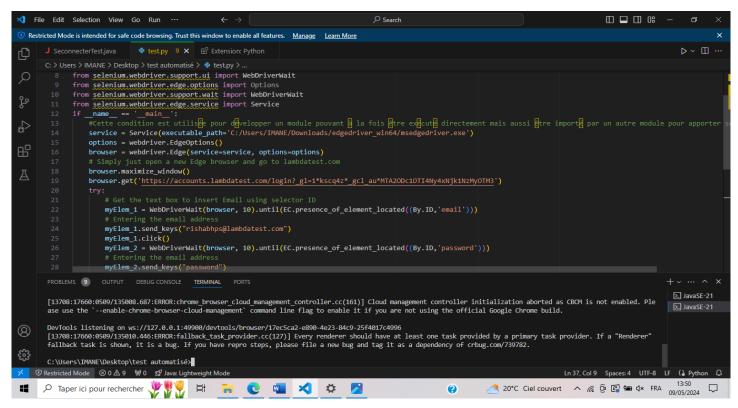


Figure 9 : test du lancement de la page de connexion de LambdaTest (1)

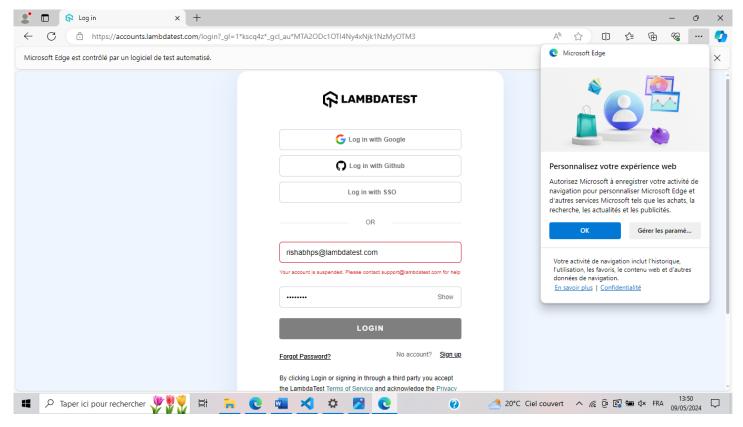


Figure 10 : résultat du test du lancement de la page de connexion de LambdaTest (1)

* Test avec des données d'un compte qui existe déjà

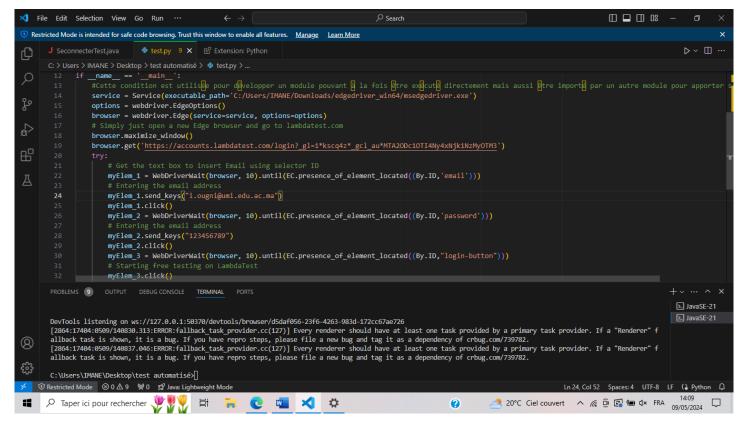


Figure 11 : test du lancement de la page de connexion de LambdaTest (2)

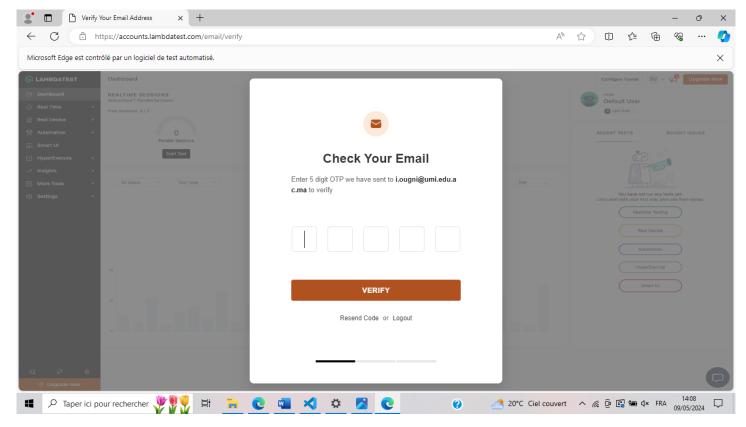


Figure 12 : résultat du test du lancement de la page de connexion de LambdaTest (2)

c. Test 3 : adapter un code de test pour utiliser un fichier CSV

L'objectif de ce dernier test est d'adapter le code pour utiliser un fichier CSV un fichier CSV avec des valeurs différentes pour l'email et le mot de passe (Num_cas_test; mail; pwd), et de reporter les résultats dans un autre fichier CSV (Num_cas_test;OK) où 1 signifiera OK et 0 KO.

```
≺ File Edit Selection View Go Run ···
                                                                                                                                                                                                                                     🕠 Restricted Mode is intended for safe code browsing. Trust this window to enable all features. <u>Manage</u> <u>Learn More</u>
                                  ■ result.csv
          C: > Users > IMANE > Desktop > test_automatisé > � test.py > ՙ main 49   aer run_test(email, passwora):
                                 browser.close()
                    def main():
    input_file = 'c:/Users/IMANE/Downloads/test.csv'
    output_file = 'c:/Users/IMANE/Downloads/result.csv'
₽
                           with open(input_file, mode='r') as infile, open(output_file, mode='w', newline='') as outfile:
    reader = csv.DictReader(infile)
    writer = csv.DictWriter(outfile, fieldnames=['Num_cas_test', 'OK'])
                                 writer.writeheader()
                                 for idx, row in enumerate(reader, start=1):
                                       email = row.get('email')
password = row.get('password')
num_cas_test = str(idx)  # Utilise l'index de ligne comme numero de cas de test
                                       result = run_test(email, password)
                                             writer.writerow({'Num_cas_test': num_cas_test, 'OK': '1'})
                                             writer.writerow({'Num_cas_test': num_cas_test, 'OK': '0'})
                                            DEBUG CONSOLE TERMINAL PORTS

    JavaSE-21 + ∨ ··· ∧ X

           PROBLEMS 6
                                                                                                                                                                                                                                                              ≥ JavaSE-21
           Every renderer should have at least one task provided by a primary task provider. If a "Renderer" fallback task is shown, it is a bug.a-
If you have repro steps, please file a new bug and tag it as a depEv endency of crbug.com/739782.
(2)
             stricted Mode 🛇 0 \Lambda 6 💖 0
```

Figure 13: test d'utilisation des ficher CSV

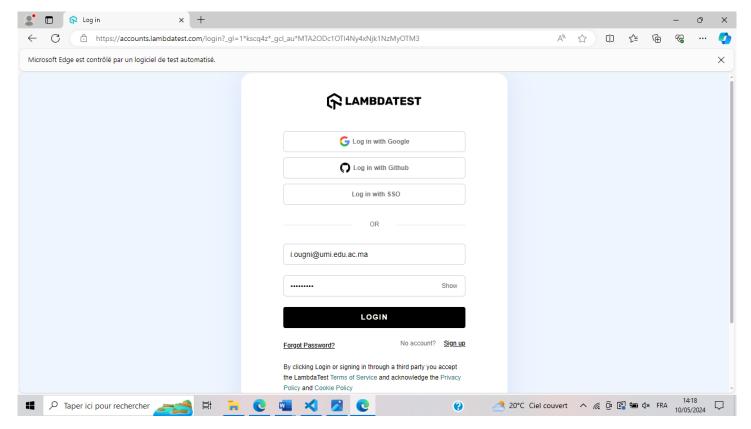


Figure 14: résultat du test du d'utilisation des ficher CSV

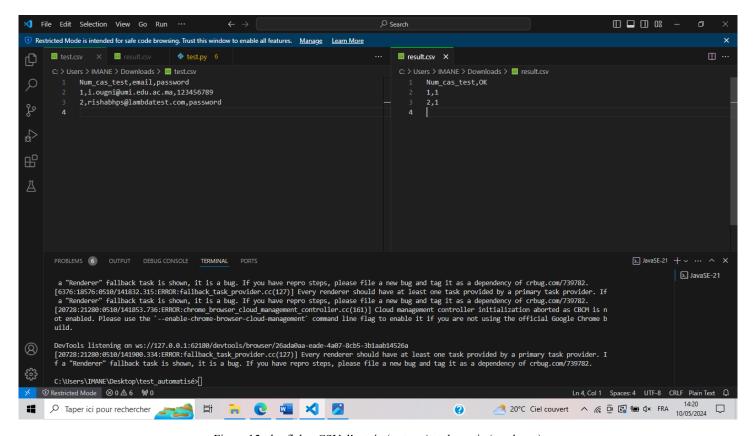


Figure 15 : les ficher CSV d'entrée (test.csv) et de sortie (result.csv)

9. Conclusion

En conclusion, les tests fonctionnels et l'automatisation des tests sont des éléments essentiels dans le processus de développement logiciel, permettant d'assurer la fiabilité, la performance et la sécurité des applications. En utilisant des outils tels que Selenium IDE qui est plus adapté aux utilisateurs novices ou aux tests simples, tandis que Selenium WebDriver est mieux adapté pour les tests automatisés avancés et complexes, nécessitant une personnalisation et une programmation plus approfondies.

En somme, en intégrant les tests fonctionnels et l'automatisation des tests dans leur processus de développement, les équipes logicielles peuvent aspirer à des produits plus fiables, performants et sécurisés, répondant ainsi aux attentes croissantes des utilisateurs dans un environnement technologique en constante évolution.