



MCCI
BUSINESS SCHOOL

A tradition of excellence

Rapport de stage de BTS SIO

Sujet : Réalisation de développement d'application de tests automatisés dans le domaine des solutions de santé et des services de conseil.

Du 05 juin 2023 au 31 juillet 2023

Stagiaire

**Dimbihasimbola
Stevine Kim
RATOMAHENINA**
kimstevine6@gmail.com
Tél : 5274 0595

Tuteur d'entreprise

**Vanina
ELLAYAH**
Team Manager
v.ellayah@epiconcept.io
Tél : 5491 3044

Date d'envoi du document : 11 Octobre 2023

2. Table des matières

1. Couverture.....	1
2. Table des matières.....	2
3. Remerciements	3
4. Introduction.....	3
5. Présentation de l'entreprise	4
5.1. L'équipe projet.....	4
5.2. Organigramme de l'Équipe	5
5.3. Communication au sein de l'équipe projet.....	6
5.4. Communication hors équipe projet	6
5.5. Ressources fournies et/ou à utiliser	6
5.6. Les outils	6
5.7. Méthodes de travail en projet collaboratif	7
6. Mon Projet	7
6.1. Cahier des charges	7
6.2. Missions du stage	8
6.3. Objectifs du projet	8
6.4. Définir les besoins du client	9
7. Calendrier de réalisation (planification du projet)	9
8. Développement	10
9. Conception	14
10. Outils de Programmation	14
11. Outils de Programmation	15
11.1. Document technique	15
11.2. Manuel d'Utilisation de l'Application d'Automatisation de Tests.....	16
12. Formation des utilisateurs	21
13. Conclusion.....	22
14. Références	23

3. Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude envers l'équipe d'Epiconcept pour m'avoir offert l'opportunité de réaliser mon stage au sein de leur entreprise. Je suis reconnaissant envers toutes les personnes qui m'ont accueilli, soutenu et guidé tout au long de cette expérience enrichissante.

Mes remerciements vont tout particulièrement à mon tuteur, Vanina Ellayah, Team Manager. Son expertise, son soutien et sa bienveillance ont été d'une valeur inestimable tout au long de mon stage. Ses conseils éclairés, sa capacité à résoudre les problèmes et sa passion pour son travail ont contribué à rendre mon expérience de stage des plus gratifiantes.

Enfin, je n'oublie pas de remercier ma famille, mes amis, la MCCI Business School et toutes les personnes qui m'ont soutenu tout au long de ce stage. Leur encouragement et leur soutien indéfectible ont été des moteurs essentiels dans la réalisation de cette expérience.

Ce stage chez Epiconcept a été une véritable opportunité d'apprentissage et de développement professionnel, et je suis reconnaissant envers toutes les personnes qui ont contribué à sa réussite. Leur confiance en moi et leur soutien ont été des éléments clés dans mon parcours.

Je souhaite à l'entreprise Epiconcept une continuité de succès dans ses activités et je suis honoré d'avoir pu faire partie de cette organisation exceptionnelle.

4. Introduction

Ce rapport retrace mon expérience de stage en fin de première année de BTS SIO (Services Informatiques aux Organisations) au sein de l'entreprise Epiconcept. Ce stage constitue une étape cruciale de ma formation, me permettant de mettre en pratique les connaissances théoriques acquises au cours de cette année.

L'objectif de ce stage était de découvrir le fonctionnement concret d'une entreprise spécialisée dans le développement de logiciels SaaS. J'ai eu la chance d'intégrer l'équipe dynamique d'Epiconcept et de participer activement à un projet en lien avec mes domaines d'étude.

Au cours de cette période, j'ai été encadré par des professionnels expérimentés, qui m'ont guidé dans mes missions et m'ont permis de développer mes compétences techniques. J'ai également pu observer et apprendre les méthodes de travail en équipe ainsi que les bonnes pratiques de développement logiciel.

Ce rapport de stage présentera en détail les missions auxquelles j'ai participé, les connaissances acquises et les apports de cette expérience professionnelle pour ma formation et mon développement personnel.

5. Présentation de l'entreprise

Epiconcept est une entreprise innovante spécialisée dans le développement de logiciels et de solutions technologiques pour les organisations. Fondée il y a plus de 25 ans, Epiconcept a su se positionner comme un acteur majeur dans son domaine, offrant des solutions sur mesure et de haute qualité à ses clients.

L'engagement d'Epiconcept envers la sécurité des données est également un élément clé de son activité. L'entreprise met en place des mesures rigoureuses pour garantir la confidentialité et la protection des informations de ses clients, en respectant les normes et les réglementations en vigueur.



Epiconcept, leader en santé publique depuis plus de 25 ans, rassemble une équipe multidisciplinaire d'experts. La société est certifiée ISO-27001 HDS, garantissant la sécurité des données de santé.

Leurs solutions SaaS en e-santé sont utilisées par des milliers de professionnels de la santé, optimisant la gestion des programmes de prévention et de dépistage.

En tant que partenaire de confiance pour la santé publique mondiale, Epiconcept contribue activement à améliorer les programmes de prévention et de dépistage grâce à des solutions technologiques avancées et une compréhension approfondie des enjeux de la santé publique.

5.1. L'équipe projet

L'équipe d'Epiconcept Océan Indien pour le projet actuel est composée de membres hautement qualifiés, chacun jouant un rôle crucial dans la réussite du projet.

Loïc Flachet, Directeur exécutif, supervise l'ensemble du projet, s'assurant du respect des normes de qualité et de l'efficacité des activités.

Vanina, Team Manager, gère quotidiennement l'équipe, garantissant la coordination des ressources et la satisfaction des membres.

Rado et Joshan, Chefs de projet, planifient, organisent et supervisent la gestion globale du projet.

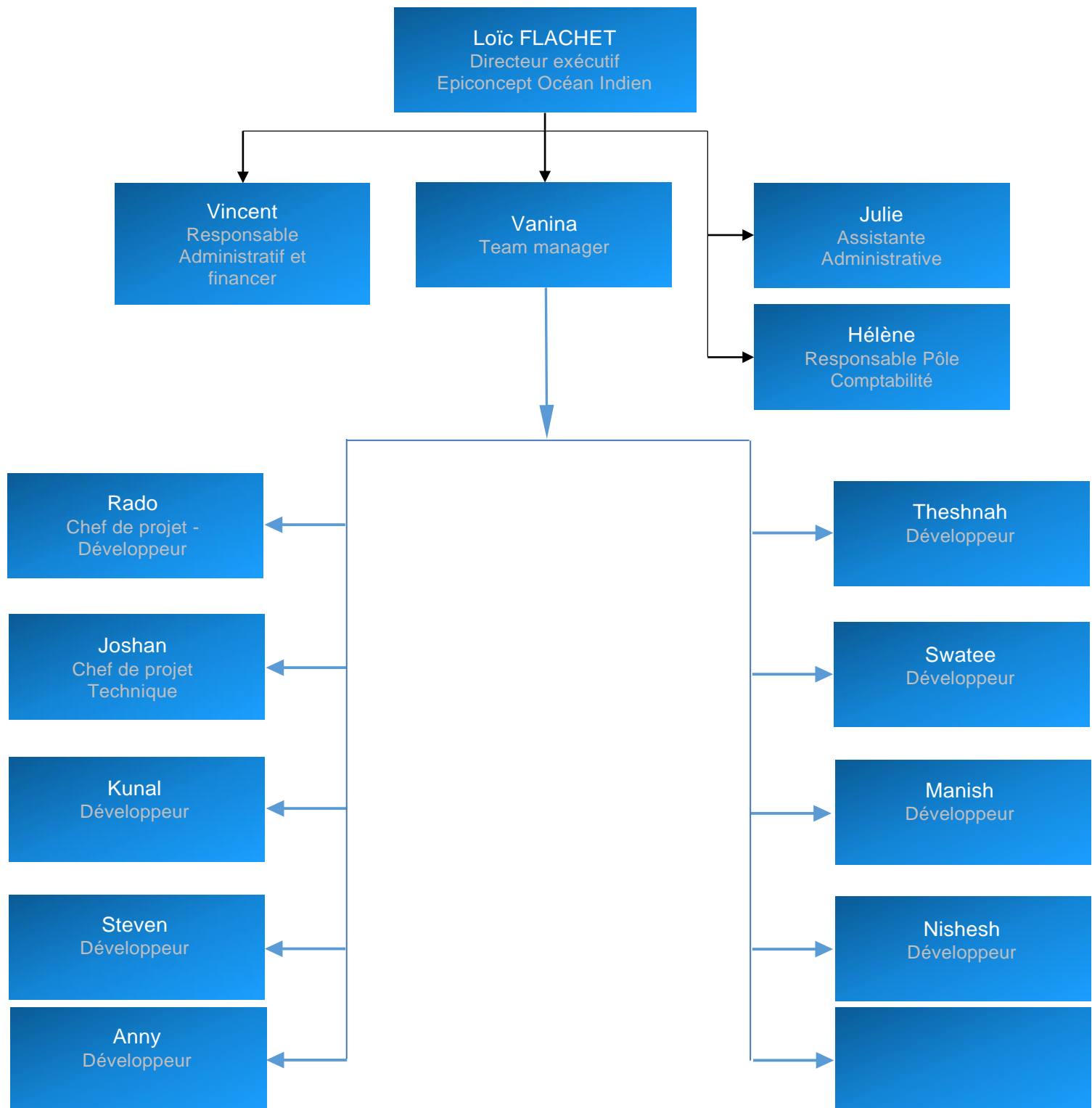
Les développeurs, Kunal, Steven, Anny, Theshnah, Deetesh, Manish et Nishesh, sont chargés de la conception et du développement des solutions logicielles, en étroite collaboration avec les Chefs de projet.

D'autres membres, dont Vincent, Julie et Hélène, apportent un soutien financier et administratif essentiel.

En tant que stagiaire ingénieur en informatique, sous la supervision de ma tutrice, Vanina, je contribue activement au développement des tests automatisés, proposant des solutions techniques et développant des fonctionnalités spécifiques, alignées sur les objectifs du projet.

5.2. Organigramme de l'Équipe

L'organigramme de l'équipe Epiconcept Océan Indien peut être représenté comme suit :



Cet organigramme reflète la structure hiérarchique de l'équipe d'Epiconcept Océan Indien pour le projet en cours, mettant en évidence les responsabilités et les relations entre les différents membres. Veuillez noter que cet organigramme est une représentation simplifiée et ne tient pas compte de la structure organisationnelle plus large d'Epiconcept Océan Indien.

5.3. Communication au sein de l'équipe projet

La communication au sein de l'équipe d'Epiconcept Océan Indien était principalement basée sur la communication verbale, favorisée par la proximité physique des membres dans le même établissement. Cela a permis des discussions informelles et des réunions spontanées pour résoudre les problèmes. En parallèle, nous avons utilisé des outils en ligne tels que Google Suite pour collaborer sur des documents, des feuilles de calcul et des présentations en temps réel, renforçant la productivité. De plus, Google Chat a facilité les discussions instantanées, assurant une communication efficace même lorsque les membres de l'équipe étaient occupés ou éloignés physiquement.

5.4. Communication hors équipe projet

Pour faciliter les communications internationales au sein du projet d'Epiconcept Océan Indien, nous avons utilisé des outils en ligne comme Google Meet. Cela nous a permis d'organiser des réunions virtuelles pour partager des informations essentielles et collaborer avec des membres basés en France. Par exemple, j'ai reçu une formation sur Passbolt avec Bryan, consultant en support, et participé à une réunion d'initiation aux tests automatisés avec Mariem HAMROUNI, toutes deux basées en France.

5.5. Ressources fournies et/ou à utiliser

Pendant mon stage chez Epiconcept Océan Indien, j'ai utilisé mon propre laptop personnel pour effectuer les tâches et le développement de l'application.

En termes de ressources fournies par l'entreprise, j'ai eu accès au réseau interne, ce qui m'a permis de me connecter aux dossiers contenant les informations du service approvisionnement. De plus, j'ai utilisé des applications et des outils tels que Microsoft Access pour travailler sur l'application existante.

5.6. Les outils

La société Epiconcept Océan Indien utilise une variété d'outils pour soutenir ses activités et optimiser ses opérations. Voici une liste des principaux outils utilisés par l'entreprise :

Google Suite : Epiconcept Océan Indien utilise la suite d'outils G Suite de Google, comprenant Gmail, Google Drive, Google Docs, Google Sheets, Google Slides, Google Calendar, etc. Ces outils permettent la collaboration en temps réel, le partage de documents et la gestion des calendriers.

Jira : Jira est un outil de gestion de projet largement utilisé par Epiconcept Océan Indien. Il permet de suivre les tâches, les problèmes, les demandes de fonctionnalités et les rapports de bogues, facilitant ainsi la coordination et la gestion des projets.

Voozanoo : Voozanoo est une plateforme de développement d'applications web utilisée par Epiconcept Océan Indien. Elle offre des fonctionnalités de développement rapide d'applications (RAD) et permet de créer des applications personnalisées selon les besoins spécifiques de l'entreprise.

Passbolt : Passbolt est un gestionnaire de mots de passe open source utilisé par Epiconcept Océan Indien. Il permet de stocker en toute sécurité les mots de passe et les informations d'identification, facilitant ainsi la gestion et le partage des informations sensibles.

Git et GitHub : Epiconcept Océan Indien utilise Git comme système de contrôle de version pour la gestion du code source de ses projets. GitHub est utilisé comme plateforme de dépôt Git, facilitant la collaboration entre les développeurs et le suivi des modifications du code.

VSCode : VSCode (Visual Studio Code) est un éditeur de code source léger et puissant utilisé par Epiconcept Océan Indien. Il offre des fonctionnalités avancées d'édition, de débogage et d'intégration avec d'autres outils, ce qui en fait un choix populaire parmi les développeurs.

MySQL : MySQL est un système de gestion de base de données relationnelle utilisé par Epiconcept Océan Indien pour stocker et gérer les données des applications et des projets.

Selenium : Selenium est un ensemble d'outils d'automatisation des tests utilisé par Epiconcept Océan Indien pour tester les applications web. Il permet de créer et d'exécuter des scripts de tests automatisés, garantissant ainsi la qualité et la fiabilité des logiciels développés.

Seleniumbase : Seleniumbase est une extension de Selenium qui offre des fonctionnalités supplémentaires pour les tests automatisés. Epiconcept Océan Indien utilise cet outil pour renforcer ses capacités de test et améliorer l'efficacité des tests automatisés.

5.7. Méthodes de travail en projet collaboratif

Epiconcept Océan Indien a adopté la méthode **Agile Scrum** pour la gestion de ses projets de développement de logiciels. Voici les principaux aspects de cette approche :

- *Équipe Scrum* : Une équipe auto-organisée et interfonctionnelle est responsable de la planification, de la réalisation et de la livraison des fonctionnalités.
- *Sprint* : Le travail est organisé en itérations appelées "Sprints", chacune ayant une durée fixe de deux à quatre semaines.
- *Planification du Sprint* : Au début de chaque Sprint, l'équipe sélectionne les éléments du backlog à réaliser, les décompose en tâches plus petites et estime le temps nécessaire.
- *Daily Scrum* : Chaque jour, l'équipe se réunit pour une réunion de Daily Scrum afin de discuter des progrès, des défis et de coordonner les activités.
- *Revue de Sprint* : À la fin de chaque Sprint, l'équipe présente les fonctionnalités développées au Product Owner et aux parties prenantes pour recueillir des commentaires.
- *Rétrospective de Sprint* : Après la revue de Sprint, l'équipe évalue les performances du Sprint précédent et identifie des opportunités d'amélioration.

L'utilisation de cette méthode Agile favorise la collaboration, l'adaptation aux changements et la livraison itérative de résultats de haute qualité. Elle encourage la transparence, la communication et l'engagement de toutes les parties prenantes, contribuant ainsi au succès des projets d'Epiconcept Océan Indien.

6. Mon Projet

6.1. Cahier des charges

Objectifs du Projet : Automatiser les tests pour améliorer la qualité des applications web, optimiser les processus et réduire les erreurs.

Périmètre du Projet : Inclut les applications Sboxlo et NeaonatLabo

Spécifications Techniques : Utilisation de Selenium avec Python, données de test, scénarios définis.

Structuration des Cas de Tests : Modèle pour la structuration des cas de tests.

Environnement Technique : Navigateurs, systèmes d'exploitation, configuration matérielle.

Intégration avec les Outils de Gestion de Projet : GitHub pour la gestion du code, JIRA pour le suivi des tâches.

Plan de Validation et de Reporting : Critères de succès, scénarios de validation, processus de rapport.

Collaboration et Communication : Réunions hebdomadaires de suivi.

6.2. Missions du stage

Pendant la période de mon stage au sein de la société Epiconcept Océan Indien, j'ai été chargé de diverses missions qui ont contribué à mon apprentissage et à ma contribution aux activités de l'entreprise. Les principales missions qui m'ont été confiées sont les suivantes :

1. **Accueil et Présentation** : Mon stage chez Epiconcept Océan Indien a débuté par un accueil chaleureux et une séance de présentation des activités de l'entreprise.
2. **Formation Agile Scrum** : J'ai suivi une formation sur la méthodologie Agile Scrum pour collaborer efficacement au sein de l'équipe.
3. **Utilisation d'outils** : J'ai été formé à l'utilisation de GitHub pour la gestion du code source et de JIRA pour la gestion de projets.
4. **Objectif du stage** : Mon objectif principal était de développer des tests automatisés pour les applications web de l'entreprise.
5. **Rédaction des cas de tests** : J'ai collaboré avec l'équipe QA pour rédiger des cas de tests clairs et précis.
6. **Développement des tests automatisés** : J'ai acquis des compétences en Python et en Selenium pour développer les tests.
7. **Suivi quotidien et hebdomadaire** : J'ai participé aux rituels Scrum quotidiens et eu des points d'avancement hebdomadaires avec mon Team Manager.
8. **Génération de rapports** : J'ai généré des rapports détaillés pour documenter les résultats de mes tests automatisés.

6.3. Objectifs du projet

Le projet chez Epiconcept Océan Indien vise à développer des tests d'automatisation pour deux applications web spécifiques, https://sboxiophp7.preprod.voozanoo.net/formation_kra et <https://neonatlabo-alpha.preprod.voozanoo.net/sigl/>. Les principaux objectifs sont les suivants :

Assurer la qualité du logiciel : Les tests automatisés permettent de détecter et corriger rapidement les erreurs, améliorant ainsi la qualité globale du produit. Cela garantit que les utilisateurs disposent d'applications de haute qualité.

Optimiser l'efficacité du processus de test : L'automatisation des tâches répétitives permet de gagner du temps et d'accélérer le processus de test. Cela libère les ressources pour se concentrer sur des activités à plus forte valeur ajoutée.

Renforcer la stabilité des applications : Les tests automatisés minimisent le risque d'erreurs lors des mises à jour ou des modifications des applications, garantissant ainsi leur stabilité et une meilleure expérience utilisateur.

Faciliter la collaboration entre les équipes : Les tests automatisés encouragent la collaboration entre les développeurs, les testeurs et les chefs de projet en fournissant des résultats clairs et reproductibles.

Accélérer la rétroaction des tests : Les tests automatisés permettent aux développeurs de vérifier rapidement les modifications, accélérant ainsi la détection et la correction d'éventuelles erreurs.

En somme, ce projet vise à garantir la qualité, la stabilité et la fiabilité des applications web mentionnées.

6.4. Définir les besoins du client

La définition des besoins du client est cruciale pour le développement des tests d'automatisation chez Epiconcept Océan Indien. Voici les principaux besoins du client :

Validation fonctionnelle : Vérifier que les fonctionnalités clés des applications web fonctionnent conformément aux spécifications, y compris la navigation, les formulaires, la gestion des données et la génération de rapports.

Compatibilité multiplateforme : S'assurer que les applications web sont compatibles avec différents navigateurs, systèmes d'exploitation et appareils pour garantir une expérience utilisateur uniforme.

Sécurité et confidentialité : Effectuer des tests de sécurité pour protéger les données et garantir la confidentialité des informations contre les vulnérabilités et les attaques potentielles.

Performances et temps de réponse : Évaluer les performances et les temps de réponse des applications sous différentes charges pour une expérience utilisateur fluide.

Fiabilité et robustesse : Identifier les erreurs, bogues et scénarios d'échec potentiels pour garantir la stabilité des applications.

Facilité d'utilisation : Assurer l'ergonomie, la navigation intuitive et l'accessibilité des fonctionnalités pour une expérience conviviale.

Documentation et rapports de test : Fournir une documentation claire et des rapports détaillés pour suivre les progrès, les résultats et les actions correctives du projet.

7. Calendrier de réalisation (planification du projet)

N°	Action	Resp.	Date prévue	Statut
1	Compléter la formation Git/GitHub	Joshua RANLAUL Kim Ratomahenina	12/06/2023	Accomplie
2	Git & GitHub : création de Branch et Pull Request à faire	v.ellayah@epiconcept.io Kim Ratomahenina	20/06/2023	Accomplie
3	Première tâche : AUTO-34 - Lancer le navigateur et accéder à l'URL de l'application	Kim Ratomahenina	27/06/2023	Accomplie le 27/06/2023
4	Tâche : AUTO-36 - Saisir les login/MDP et se connecter à l'application	Kim Ratomahenina	04/07/2023	Accomplie le 06/07/2023
5	Tâche : AUTO-37 - Implémenter le test de connexion sur une autre application	Kim Ratomahenina	11/07/2023	Accomplie le 10/07/2023
6	Tâche : AUTO-38 - [e2e] [NEONATLABO] Tester la redirection vers la bonne page	Kim Ratomahenina	24/07/2023	Accomplie le 21/07/2023

8. Développement

Analyse des tâches continues effectuées lors de mon stage chez Epiconcept Océan Indien :

- *Tâche AUTO-34 - Lancer le navigateur et accéder à l'URL de l'application :*

Pendant mon stage chez Epiconcept Océan Indien, ma première tâche était de développer une fonctionnalité permettant de lancer un navigateur et d'accéder à une URL d'application. J'ai utilisé la bibliothèque Selenium et le module webdriver de Selenium pour cette tâche. Après avoir configuré le navigateur Chrome avec webdriver.Chrome(), j'ai chargé l'URL spécifiée à l'aide de la méthode get(). Malgré quelques difficultés liées à l'installation de pytest, j'ai réussi cette tâche. Cette expérience m'a permis de maîtriser Selenium et le module webdriver, tout en comprenant les défis des tests automatisés et de la configuration des environnements de test.

oi-automated-tests / Sboxlo / TEST.py

epi-kra AUTO 34 - Lancer le navigateur et accéder à l'URL de l'application

0745ac0 · last month History

Code Blame 16 lines (11 loc) · 366 Bytes

Raw Copy Download Edit View

```
1 from selenium import webdriver
2 from vars import url
3
4 # Configuration du navigateur Chrome
5 driver = webdriver.Chrome()
6
7 # Accès à l'URL
8 driver.get(url)
9
10 # Attendez indéfiniment pour éviter la fermeture automatique du navigateur
11 while True:
12     pass
13
14
15 # Si vous souhaitez fermer le navigateur à un moment ultérieur, utilisez la méthode quit() :
16 # driver.quit()
```

oi-automated-tests / Sboxlo / vars.py

epi-kra AUTO 34 - Lancer le navigateur et accéder à l'URL de l'application

0745ac0 · last month History

Code Blame 2 lines (2 loc) · 101 Bytes

Raw Copy Download Edit View

```
1 # Test Lancer navigateur et ouvrir url
2 url = "https://sboxiophp7.preprod.voozanoo.net/formation_kra"
```

- *Tâche AUTO-36 - Saisir les login/MDP et se connecter à l'application :*

J'ai automatisé la saisie des identifiants de connexion (login et mot de passe) et la connexion à l'application en utilisant SeleniumBase, une extension de Selenium. J'ai également stocké les identifiants dans un fichier vars.py. Pour simplifier, j'ai employé la méthode open de SeleniumBase pour accéder à l'URL de l'application en une seule instruction, puis j'ai utilisé les méthodes type et press_enter pour remplir les champs de connexion. J'ai résolu le défi de spécifier les identifiants en définissant les variables d'environnement USERNAME et PASSWORD. Cette tâche m'a permis de développer une approche modulaire pour la saisie des identifiants et de comprendre l'importance des variables d'environnement dans la configuration des tests automatisés.

oi-automated-tests / Sboxlo / TEST.py



epi-kra Définition de username et password dans vars

3a2e242 · 3 weeks ago

History

Code Blame 31 lines (24 loc) · 990 Bytes

Raw



```
1 import pytest
2 from seleniumbase import BaseCase
3 from selenium.webdriver.common.keys import Keys
4 from vars import url, username, password
5
6 class MyTestClass(BaseCase):
7     def test_login(self):
8         # Accès à l'URL
9         self.open(url)
10
11         # Attendre 3 secondes (pour permettre la saisie des identifiants)
12         self.sleep(3)
13
14         # Remplir automatiquement les champs de connexion
15         self.type('[name="login"]', username)
16         self.type('[name="password"]', password)
17
18         # Appuyer sur la touche Entrée pour soumettre le formulaire
19         password_input = self.wait_for_element('[name="password"]')
20         password_input.send_keys(Keys.ENTER)
21
22         # Attendre que l'utilisateur appuie sur Entrée avant de fermer le navigateur
23         input("Appuyez sur Entrée pour fermer le navigateur...")
24
25     def test_other(self):
26         # Autres tests
27         pass
28
29 if __name__ == "__main__":
30     # Lancer les tests avec pytest
31     pytest.main(["-s", "TEST.py"])
```

oi-automated-tests / Sboxlo / vars.py



epi-kra Définition de username et password dans vars

3a2e242 · 3 weeks ago

History

Code Blame 6 lines (5 loc) · 163 Bytes

Raw



```
1 # Test Lancer navigateur et ouvrir url
2 url = "https://sboxiophp7.preprod.voozanoo.net/formation_kra"
3
4 # Identifiants de connexion
5 username = "XXX"
6 password = "XXX"
```

- *Tâche AUTO-37 - Implémenter le test de connexion sur une autre application :*

J'ai automatisé le test de connexion pour l'application "NeonatLabo". Pour ce faire, j'ai créé un fichier "login_logic.py" où j'ai défini une classe "LoginLogic" avec une méthode "def_login()" pour gérer la logique de connexion. Cette méthode utilisait des variables importées depuis "vars.py" pour chaque application, telles que l'URL, le nom d'utilisateur et le mot de passe.

Ensuite, j'ai importé la classe "LoginLogic" dans les fichiers "TEST.py" de chaque application, "Sboxlo" et "NeonatLabo". En créant des instances de cette classe, j'ai pu effectuer la connexion spécifique à chaque application.

Cette approche a permis de gérer les particularités de chaque application de manière modulaire grâce à la programmation orientée objet (OOP) de Python. Malgré les différences entre les applications, le code restait organisé et structuré, ce qui facilitait les futures mises à jour et modifications.

Cette tâche a renforcé ma compréhension de la programmation orientée objet et m'a montré l'importance de gérer les spécificités des applications dans les tests automatisés.

oi-automated-tests / NeonatLabo

TEST.py

vars.py

Sboxlo

TEST.py

vars.py

base

login.py

vars_login.py

README.md

oi-automated-tests / NeonatLabo / TEST.py

...



epi-kra Implémenter le test de connexion sur une autre application : NeonatLabo

d54eb74 · 2 weeks ago

History

Code

Blame

15 lines (13 loc) · 414 Bytes

Raw



```
1 import sys
2 import os
3 sys.path.insert(0, os.path.abspath(os.path.join(os.path.dirname(__file__), "..")))
4 import pytest
5 from seleniumbase import BaseCase
6 from base.login import Login as NeonatLogin
7
8 class NeonatLaboTestClass(BaseCase):
9     def test_login(self):
10         login = NeonatLogin(self)
11         login.login()
12
13 if __name__ == "__main__":
14     # Lancer les tests avec pytest
15     pytest.main(["-s", "TEST.py"])
```

oi-automated-tests / NeonatLabo / vars.py

...



epi-kra Implémenter le test de connexion sur une autre application : NeonatLabo

d54eb74 · 2 weeks ago

History

Code

Blame

2 lines (2 loc) · 99 Bytes

Raw



```
1 # Test Lancer navigateur et ouvrir url
2 url = "https://neonatlabo-alpha.preprod.voozanoo.net/sigl/"
```

oi-automated-tests / Sboxlo / TEST.py

...



epi-kra Implémenter le test de connexion sur une autre application : NeonatLabo

d54eb74 · 2 weeks ago

History

Code

Blame

15 lines (13 loc) · 406 Bytes

Raw



```
1 import sys
2 import os
3 sys.path.insert(0, os.path.abspath(os.path.join(os.path.dirname(__file__), "..")))
4 import pytest
5 from seleniumbase import BaseCase
6 from base.login import Login as SboxLogin
7
8 class SboxIoTestClass(BaseCase):
9     def test_login(self):
10         login = SboxLogin(self)
11         login.login()
12
13 if __name__ == "__main__":
14     # Lancer les tests avec pytest
15     pytest.main(["-s", "TEST.py"])
```

oi-automated-tests / Sboxlo / vars.py

...



epi-kra Implémenter le test de connexion sur une autre application : NeonatLabo

d54eb74 · 2 weeks ago

History

Code

Blame

2 lines (2 loc) · 101 Bytes

Raw



```
1 # Test Lancer navigateur et ouvrir url
2 url = "https://sboxiophp7.preprod.voozanoo.net/formation_kra"
```

[oi-automated-tests / base / login.py](#)

epi-kra Implémenter le test de connexion sur une autre application : NeonatLabo

d54eb74 · 2 weeks ago

History

Code Blame 25 lines (19 loc) · 908 Bytes

Raw

```
1 from selenium.webdriver.common.keys import Keys
2 from vars import url
3 from base.vars_login import username, password
4
5 class Login:
6     def __init__(self, base_case):
7         self.base_case = base_case
8
9     def login(self):
10        # Accès à l'URL
11        self.base_case.open(url)
12
13        # Attendre 3 secondes (pour permettre la saisie des identifiants)
14        self.base_case.sleep(3)
15
16        # Remplir automatiquement les champs de connexion
17        self.base_case.type('[name="login"]', username)
18        self.base_case.type('[name="password"]', password)
19
20        # Appuyer sur la touche Entrée pour soumettre le formulaire
21        password_input = self.base_case.wait_for_element('[name="password"]')
22        password_input.send_keys(Keys.ENTER)
23
24        # Attendre que l'utilisateur appuie sur Entrée avant de fermer le navigateur
25        input("Appuyez sur Entrée pour fermer le navigateur...")
```

[oi-automated-tests / base / vars_login.py](#)

epi-kra Implémenter le test de connexion sur une autre application : NeonatLabo

d54eb74 · 2 weeks ago

History

Code Blame 3 lines (3 loc) · 61 Bytes

Raw

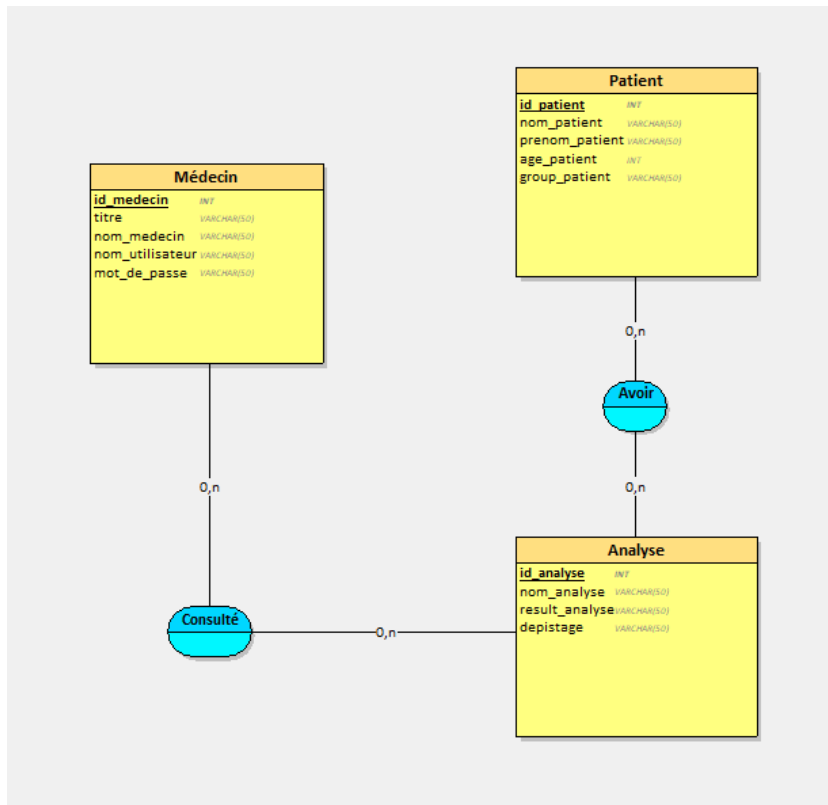
```
1 # Identifiants de connexion
2 username = "XXX"
3 password = "XXX"
```

- *Tâche AUTO-38 - [e2e] [NEONATLABO] Tester la redirection vers la bonne page :*

J'ai automatisé le test de redirection sur l'application NeonatLabo, vérifiant si un bouton spécifique était cliqué et si un texte spécifique était présent sur la nouvelle page. J'ai surmonté des difficultés, notamment la gestion du code source du bouton. J'ai utilisé SeleniumBase pour cette tâche et amélioré la génération de rapports avec "pytest-html", fournissant des rapports détaillés des résultats pour chaque étape. Ces activités ont renforcé mes compétences en automatisation de tests web et en résolution de problèmes liés à l'interface utilisateur sur des applications complexes.

9. Conception

Un Modèle Conceptuel de Données (MCD) est une représentation graphique essentielle en conception de bases de données, décrivant la structure logique et les relations entre les entités clés d'un système d'information. Il se compose d'entités (représentant des objets), d'attributs (définissant leurs propriétés) et de relations (indiquant comment les entités interagissent). Dans le contexte du projet de Test d'Automatisation de l'application web, le MCD offre une vue simple de la structure sous-jacente de l'application, permettant de mieux comprendre l'organisation et l'utilisation des données.



10. Outils de Programmation

Pendant mon stage enrichissant au sein de l'entreprise Epiconcept Océan Indien, j'ai eu l'opportunité d'explorer une gamme d'outils de programmation et d'interfaces indispensables qui ont été déterminants dans le développement et l'exécution des tests automatisés pour les applications web. Voici une présentation des principaux outils que j'ai exploités :

1. Framework de Test Automatisé - Selenium :

Le cadre de travail open-source Selenium s'est avéré être la pierre angulaire de notre démarche d'automatisation de tests. En s'appuyant sur le modèle WebDriver, j'ai pu interagir de manière programmatique avec les composants des pages web, reproduisant ainsi les actions d'un utilisateur réel. Cette flexibilité a permis la mise en œuvre de divers scénarios de test, garantissant la couverture exhaustive des fonctionnalités des applications.



2. Langage de Programmation - Python :

Python a été mon langage de programmation de prédilection pour la création des scripts de tests automatisés. La lisibilité de sa syntaxe et la richesse de sa bibliothèque standard ont grandement facilité la conception des scénarios de test complexes. En intégrant des modules tels que pytest, j'ai pu structurer et gérer efficacement mes cas de test, favorisant ainsi leur exécution régulière et leur évolutivité.



3. Framework de Test - Pytest :

L'utilisation de Pytest, un framework de test en Python, a été un pivot majeur dans le processus d'automatisation. Pytest a introduit un cadre organisé pour l'écriture et l'exécution de tests, simplifiant l'écriture d'assertions, la gestion des fixtures et la production de rapports détaillés. Sa nature modulaire a permis une mise en œuvre efficace des tests unitaires et fonctionnels, renforçant ainsi la qualité des applications testées.



4. Interfaces Utilisateur (Interfaces Graphiques ou Autres) :

Pour évaluer les applications web de manière holistique, j'ai interagi avec les interfaces utilisateur via des navigateurs web tels que Google Chrome et Mozilla Firefox. Cette approche a permis de simuler les interactions d'utilisateurs réels, garantissant la validité des tests. Cette interaction a également permis de mettre en évidence les éventuels problèmes d'ergonomie et d'utilisabilité.

L'ensemble de ces outils et interfaces a été orchestré de manière synergique pour élaborer des tests automatisés complets et précis. L'utilisation judicieuse de ces ressources a permis d'assurer la qualité, la stabilité et la performance des applications web de l'entreprise. Ces outils ont grandement favorisé ma collaboration avec l'équipe, la création de tests sophistiqués et la production de résultats significatifs pour évaluer la qualité des applications.



11. Documentation

11.1. Document technique

Au cours de mon stage chez Epiconcept, j'ai eu l'opportunité de m'impliquer activement dans la réalisation de tests d'automatisation pour les applications web de l'entreprise. Pour mener à bien cette mission, j'ai pu compter sur une gamme de documents techniques qui ont joué un rôle essentiel dans la planification, l'exécution et la gestion de ces tests.

1. *Attribution des Tâches via la Plateforme JIRA :*

La plateforme JIRA s'est avérée être un outil de gestion de projet indispensable. Grâce à JIRA, nous avons pu attribuer des tâches spécifiques à chaque membre de l'équipe, suivre l'avancement de ces tâches, et maintenir une traçabilité complète de toutes les activités liées aux tests d'automatisation.

Cette plateforme a permis une collaboration efficace au sein de l'équipe et une gestion transparente des priorités.

2. *Descriptions des Objectifs des Tâches :*

Chaque tâche assignée était accompagnée d'une description détaillée de ses objectifs. Ces descriptions nous ont fourni un contexte clair sur ce qui était attendu de chaque test, y compris les critères de réussite et les éléments spécifiques à vérifier.

Cela a grandement facilité la préparation et l'exécution des tests en veillant à ce que nous restions concentrés sur les résultats escomptés.

11.2. Manuel d'Utilisation de l'Application d'Automatisation de Tests

1. Introduction

Ce manuel utilisateur vous guidera à travers le processus d'utilisation de l'application d'automatisation de tests pour les applications web Voozanoo et NeonatLabo. Vous apprendrez comment utiliser l'application de test automatisé pour effectuer des tests sur ces applications, vérifier les résultats et analyser les rapports générés.

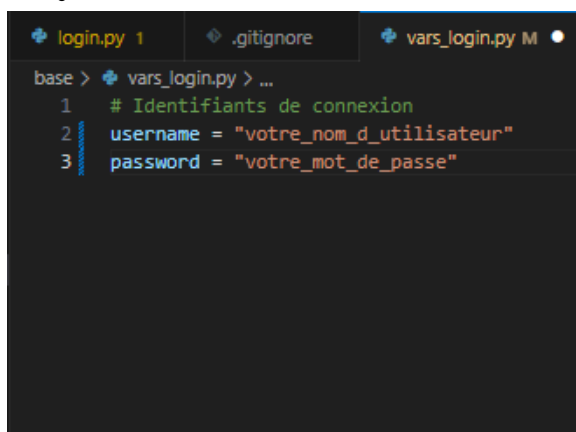
2. Préparation de l'Environnement

2.1 Ouverture des Codes Sources avec VSCode

- Pour commencer, lancez l'application VSCode sur votre système.
- Ensuite, ouvrez le menu "File" (Fichier) dans la barre de menu et sélectionnez "Open Folder" (Ouvrir un Dossier).
- Parcourez votre système de fichiers pour localiser le dossier contenant les codes sources des tests. Sélectionnez ce dossier pour l'ouvrir dans VSCode.

2.2 Configuration des Identifiants d'Utilisateur

- Dans les codes sources des tests, repérez le fichier nommé "login.py".
- À l'intérieur de ce fichier, vous trouverez une ligne marquée **username =**. Remplacez le texte **votre_nom_d_utilisateur** par votre propre nom d'utilisateur pour l'application.
- De même, localisez la ligne **password =** et remplacez **votre_mot_de_passe** par votre mot de passe de connexion.



```
login.py 1  .gitignore  vars_login.py M
base > vars_login.py > ...
1  # Identifiants de connexion
2  username = "votre_nom_d_utilisateur"
3  password = "votre_mot_de_passe"
```

3. Exécution des Tests Automatisés

3.1 Lancement des Scénarios de Test

- Ouvrez le terminal intégré de VSCode. Pour ce faire, allez dans le menu "View" (Affichage) et sélectionnez "Terminal".
- Entrez la commande de lancement du test correspondant à l'application que vous souhaitez tester. Voici les commandes pour les deux applications :
 - Pour Voozanoo : **pytest -s Voozanoo/TEST.py --html=report.html**
 - Pour NeonatLabo : **pytest -s NeonatLabo/TEST.py --html=report.html**

```
▼ TERMINAL

PS D:\MCCI SIO\Stages\Epiconcept\oi-automated-tests> pytest -s NeonatLabo/TEST.py --html=report.html
===== test session starts =====
platform win32 -- Python 3.10.11, pytest-7.3.2, pluggy-1.0.0
rootdir: D:\MCCI SIO\Stages\Epiconcept\oi-automated-tests
plugins: forked-1.6.0, html-3.2.0, metadata-3.0.0, ordering-0.6, rerunfailures-11.1.2, xdist-3.3.1, seleniumbase-4.15.3
collected 5 items

NeonatLabo\TEST.py .....

----- generated html file: file:///D:/MCCI%20SIO/Stages/Epiconcept/oi-automated-tests/report.html -----
===== 5 passed in 139.74s (0:02:19) =====
```

3.2 Vérification des Résultats

- Après avoir saisi la commande de lancement du test, le processus de test automatisé sera initié.
- Un navigateur web sera automatiquement ouvert par le webdriver et dirigé vers l'URL spécifiée pour l'application.
- Le test automatisé effectuera les étapes de connexion en utilisant les informations d'identification fournies.
- Si les informations d'identification sont correctes, le test poursuivra en effectuant les actions définies dans les scénarios de test.
- Les résultats de chaque test seront enregistrés, et un rapport détaillé sera généré dans un fichier HTML nommé "report.html".

report.html

Report generated on 31-Jul-2023 at 15:45:02 by [pytest-html](#) v3.2.0

Summary

5 tests ran in 160.61 seconds.

(Un)check the boxes to filter the results.

☒ 5 passed, ☒ 0 skipped, ☒ 0 failed, ☒ 0 errors, ☒ 0 expected failures, ☒ 0 unexpected passes, ☒ 0 rerun

Results

[Show all details](#) / [Hide all details](#)

▲ Result	▼ Test	▼ Duration	▼ Links
Passed (show details)	NeonatLabo/TEST.py::NeonatLaboTestClass::test_import_results_button	4.85	
Passed (show details)	NeonatLabo/TEST.py::NeonatLaboTestClass::test_login	53.70	
Passed (show details)	NeonatLabo/TEST.py::NeonatLaboTestClass::test_main	77.13	
Passed (show details)	NeonatLabo/TEST.py::NeonatLaboTestClass::test_title_accueil_HBS	12.29	
Passed (show details)	NeonatLabo/TEST.py::NeonatLaboTestClass::test_title_import_results	12.46	

4. Gestion des Rapports de Tests

4.1 Ouverture des Rapports Générés

- Une fois que les tests sont terminés, recherchez le fichier "report.html" dans le dossier spécifié.
- Double-cliquez sur ce fichier pour l'ouvrir automatiquement dans votre navigateur par défaut.

4.2 Analyse des Résultats

- Le rapport affiche une vue détaillée des tests effectués, indiquant si chaque test a réussi ("passed") ou échoué ("failed").
- Utilisez ce rapport pour évaluer les performances de l'application testée et identifier les domaines nécessitant une attention particulière.

5. Installation des Prérequis

5.1 Python et pip

- Assurez-vous d'avoir Python 3.x installé sur votre système.
- Pour vérifier, ouvrez un terminal et entrez la commande **python --version**.
- Si Python n'est pas installé, téléchargez-le depuis le site officiel de Python et suivez les instructions d'installation.

5.2 Selenium et Webdriver

- Pour exécuter les tests automatisés, vous aurez besoin de Selenium WebDriver.
- Installez Selenium en ouvrant un terminal et en exécutant la commande **pip install selenium**.

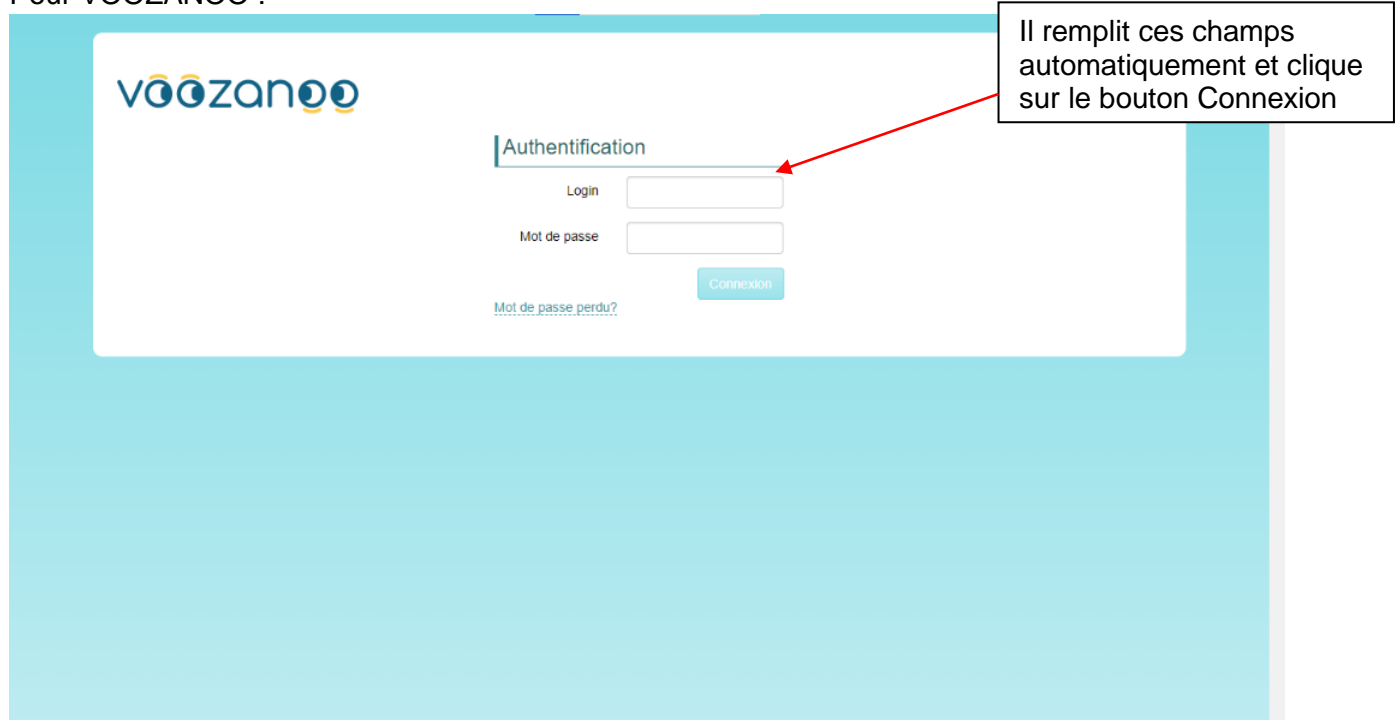
5.3 Pytest

- Pour exécuter les tests avec Pytest, installez-le en ouvrant un terminal et en exécutant la commande **pip install pytest**.

6. Scénarios de Test

- Le test automatisé ouvre la page de connexion de l'application.

Pour VOOZANOO :



Pour NEONATLABO :

Il remplit ces champs automatiquement et clique sur le bouton Connexion

Authentification

Login

Mot de passe

Connexion

Mot de passe perdu?

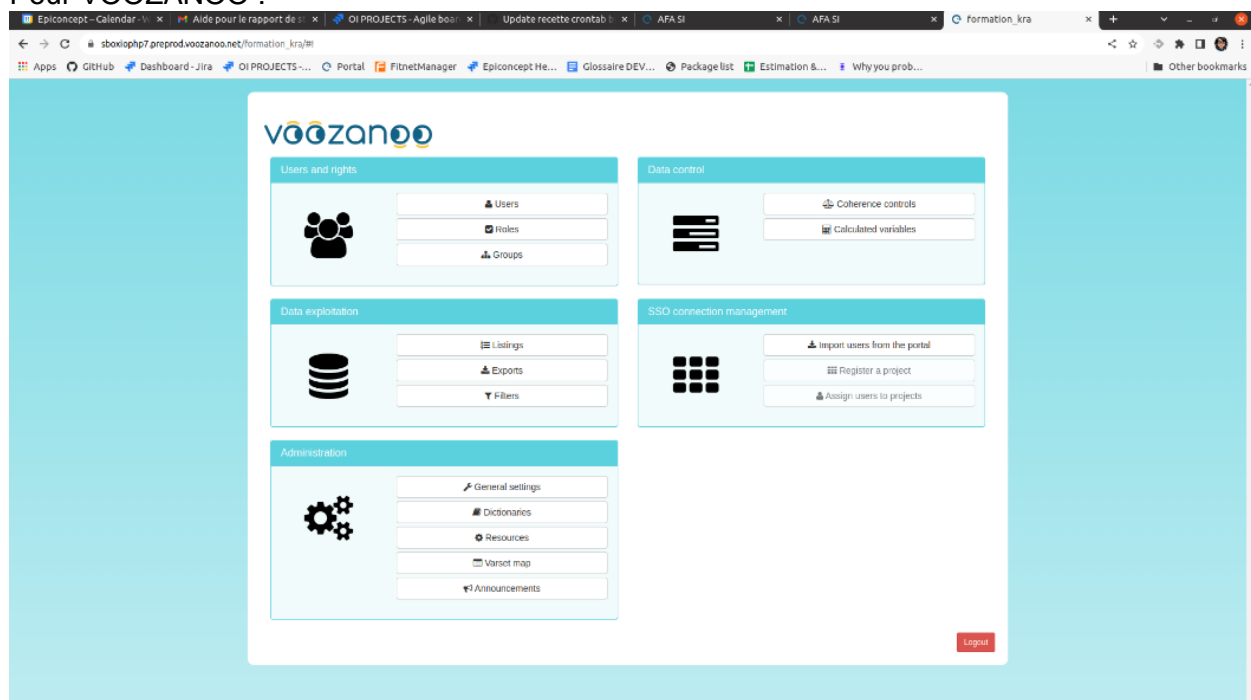
Authentification SSO

Connexion

Version : 1.3.0 | © voozanoo / epiconcept 2013

- Il saisit automatiquement le nom d'utilisateur et le mot de passe.
- Après avoir saisi les informations correctes, il clique sur le bouton "Entrer".
- Une fois connecté, l'application est dirigée vers la page d'accueil.

Pour VOOZANOO :





- Selon le scénario de test spécifique, il peut être ordonné de vérifier la présence d'un bouton.
- Si le bouton est présent, le test peut cliquer dessus pour vérifier son fonctionnement.



- Il clique sur le bouton « Import de résultats » s'il existe, et affiche la page Import de résultats



- Le test peut également vérifier la présence d'un titre ou d'une phrase spécifique.
- À la fin de chaque test, un rapport détaillé est généré sous forme d'une page web "report.html".
- Ce rapport indique si chaque test a réussi ("passed") ou échoué ("failed").

12. Formation des utilisateurs

Pendant mon stage chez Epiconcept Océan Indien, j'ai organisé une réunion de formation pour présenter en détail une application d'automatisation de tests à mon équipe. Cette réunion, qui s'est déroulée dans nos locaux, a rassemblé les développeurs, les chefs de projet et notre superviseure, Vanina. J'ai expliqué comment l'application améliorait la qualité de nos applications web grâce à l'automatisation des tests, soulignant ses avantages tels que la réduction des erreurs humaines et la rapidité des tests.

J'ai ensuite plongé dans les détails de son fonctionnement, montrant comment elle interagissait avec nos applications web. J'ai décrit le processus de création et d'exécution de tests automatisés, mettant en avant la simplicité de l'interface utilisateur. J'ai encouragé toute l'équipe, quel que soit son niveau d'expérience en tests automatisés, à utiliser cette application comme un outil essentiel.

La réunion s'est transformée en une discussion interactive où les membres de l'équipe ont posé des questions pertinentes sur l'intégration de l'application dans nos projets et son impact sur notre workflow. Vanina a montré son enthousiasme quant aux avantages de l'application, ce qui a encouragé son adoption.

Nous avons également discuté des prochaines étapes, notamment la formation individuelle de chaque membre de l'équipe à l'utilisation de l'application et son intégration dans nos processus de développement.

En conclusion, la réunion de formation sur l'application d'automatisation de tests a été un succès. Elle a permis de sensibiliser l'équipe aux avantages de cette application, renforçant ainsi notre engagement envers la qualité et la performance de nos applications web tout en améliorant la collaboration au sein de l'équipe.

13. Conclusion : (Un Stage Riche en Apprentissage et en Réalisations)

Mon stage chez Epiconcept a été une expérience incroyablement enrichissante, m'offrant l'opportunité d'explorer en profondeur le domaine passionnant de l'automatisation des tests pour les applications web. Au cours de ces mois passés au sein de l'entreprise, j'ai pu acquérir des compétences techniques précieuses, travailler en étroite collaboration avec une équipe talentueuse et contribuer directement au processus d'amélioration de la qualité des produits.

L'une des réalisations dont je suis le plus fier est le développement de l'application d'automatisation de tests. Cette application, conçue pour évaluer les applications web de l'entreprise, représente un effort considérable de planification, de conception et d'exécution. Elle a non seulement permis d'optimiser les tests, mais également de garantir une meilleure stabilité, performance et fiabilité des applications. Cette réalisation est le fruit d'une collaboration fructueuse avec mes collègues développeurs et d'un engagement continu envers l'excellence technique.

Un autre aspect clé de mon stage a été l'interaction avec l'équipe Epiconcept. J'ai eu la chance de travailler avec des professionnels chevronnés, des chefs de projet engagés et une équipe de développement dévouée. Le partage des connaissances et des idées a été une constante tout au long de cette expérience, me permettant d'apprendre rapidement et de m'adapter aux meilleures pratiques de l'industrie.

En outre, la gestion attentive et le soutien de mon tuteur, Vanina, ont joué un rôle essentiel dans mon développement professionnel. Ses conseils, ses orientations et ses retours d'expérience m'ont aidé à surmonter les défis et à atteindre mes objectifs de stage.

Ce stage a également mis en lumière l'importance de la documentation, de la planification méticuleuse et de la communication transparente dans le domaine de l'automatisation des tests. Les documents techniques, tels que la liste des charges, les descriptions des objectifs des tâches et le site de page d'aide, ont été des ressources cruciales qui ont guidé notre travail vers le succès.

14. Références (Sources)

Langage de Programmation :

- [Python](#) - Site officiel de Python.
- [Python Wiki](#) - La documentation Python Wiki.

Frameworks de Test Automatisé :

- [Selenium](#) - Site officiel de Selenium.
- [Documentation Selenium](#) - Documentation officielle de Selenium.
- [Pytest](#) - Documentation officielle de Pytest.
- [Pytest - Getting Started](#) - Guide de démarrage de Pytest.

Base de Données et Langage SQL :

- [PostgreSQL](#) - Site officiel de PostgreSQL.
- [Documentation PostgreSQL](#) - Documentation officielle de PostgreSQL.

Environnement de Développement :

- [Visual Studio Code](#) - Site officiel de Visual Studio Code.
- [Visual Studio Code - Documentation](#) - Documentation de Visual Studio Code.

Gestion de Projet :

- [Atlassian JIRA](#) - Site officiel d'Atlassian JIRA.
- [Atlassian JIRA - Documentation](#) - Documentation d'Atlassian JIRA.
- [GitHub](#) - Site officiel de GitHub.
- [GitHub Guides](#) - Guides pour utiliser GitHub.

Documents Techniques :

- [Élaboration de Tests d'Automatisation](#) - Documentation interne de l'entreprise sur les tests d'automatisation.
- [Liste des Charges](#) - Document interne répertoriant les charges de travail pour le projet d'automatisation.
- [Description des Objectifs des Tâches](#) - Descriptions détaillées des objectifs des tâches attribuées.
- [Page d'Aide des Testeurs Automation](#) - Ressource interne développée par l'équipe des testeurs en automatisation de l'entreprise.

Autres Ressources :

- [Documentation Python Selenium](#) - Documentation pour l'utilisation de Selenium avec Python.
- [Guide Pytest](#) - Tutoriels complets sur l'utilisation de Pytest.