



Rapport de fin de projet

Test Site web Decathlon.tn



Réalisé par :

Ayoub Ben Hadj Youssef

&

Mohamed Aziz El Ouni

[Méthode de travail 3](#_Toc106984230)

[Test Fonctionnel 4](#_Toc106984231)

[Périmètre d'intervention de l'activité 4](#_Toc106984232)

[Recette associée au projet 4](#_Toc106984233)

[Les éléments du produit/ logiciel qui seront testés 4](#_Toc106984234)

[Les éléments qui sont exclus de la stratégie 4](#_Toc106984235)

[Implémentation des tests 4](#_Toc106984236)

[Exécution des tests 5](#_Toc106984237)

[Clôture des tests 5](#_Toc106984238)

[Difficultés & Contraintes trouvés 5](#_Toc106984239)

[Test Auto Cypress 6](#_Toc106984240)

[Environnement 6](#_Toc106984241)

[Scénario de test automatisé 6](#_Toc106984242)

[Contrainte 6](#_Toc106984243)

[Conception des cas de test 6](#_Toc106984244)

[Implémentation des cas de test 6](#_Toc106984245)

[Exécution des cas de test 7](#_Toc106984246)

[Conclusion 8](#_Toc106984247)

[Test auto sur Selenium 8](#_Toc106984248)

[Périmètre d'intervention de l'activité 8](#_Toc106984249)

[Environnement de test 8](#_Toc106984250)

[Difficultés & Contraintes trouvés 9](#_Toc106984251)

[Test Auto SELENIUM IDE 9](#_Toc106984252)

[Environnement 9](#_Toc106984253)

[Scénario de test automatisé 9](#_Toc106984254)

[Contrainte 9](#_Toc106984255)

[Conception des cas de test 9](#_Toc106984256)

[Exécution des cas de test 10](#_Toc106984257)

[Scénario Login avec informations valide 10](#_Toc106984258)

[Scénario Login avec mail invalide 11](#_Toc106984259)

[Scénario Login avec mot de passe invalide 12](#_Toc106984260)

[Conclusion 13](#_Toc106984261)

[Test de Performance 13](#_Toc106984262)

[Environnement 13](#_Toc106984263)

[Scénario de test et données de test 14](#_Toc106984264)

[Contrainte 14](#_Toc106984265)

[Implémentations du test 14](#_Toc106984266)

[Exécution des cas de test 15](#_Toc106984267)

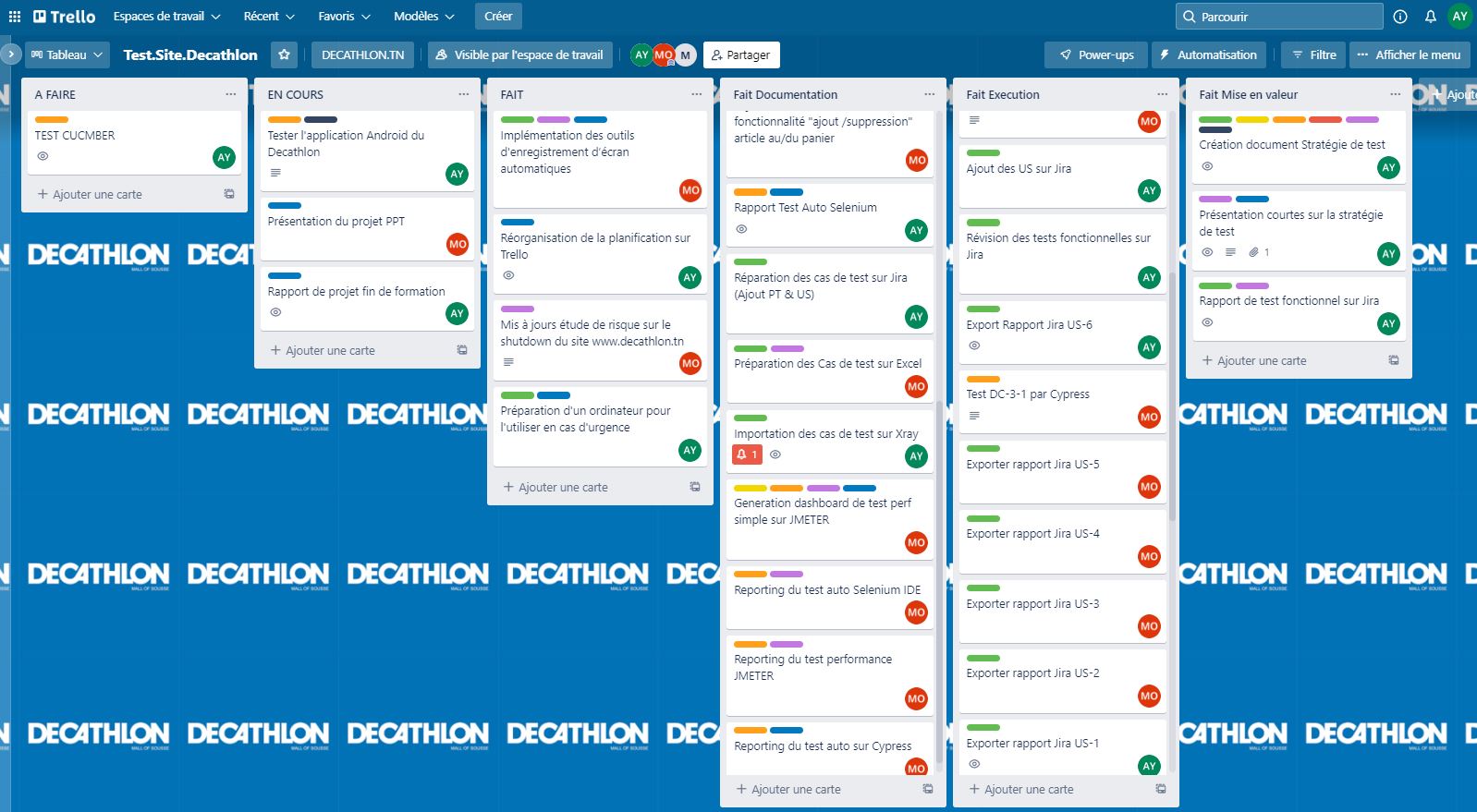
[Génération du rapport de test 15](#_Toc106984268)

[Conclusion 17](#_Toc106984269)

[Conclusion générale 17](#_Toc106984270)

# Méthode de travail

* Utiliser Trello pour la gestion des taches



* Taches planifiées et sa répartition :

|  |  |
| --- | --- |
| **Taches** | **Répartition** |
| Création de stratégie de test | Ayoub |
| Répartition des taches | Ayoub |
| Rédaction de recette de test | Mohamed Aziz |
| Révision et importation de la recette de test | Ayoub |
| Test US1 sur Jira | Mohamed Aziz |
| Test US2 sur Jira | Mohamed Aziz |
| Test US3 sur Jira | Ayoub |
| Test US4 sur Jira | Ayoub |
| Test US5 sur Jira | Ayoub |
| Test US6 sur Jira | Ayoub |
| Exporter le rapport de test fonctionnel | Ayoub |
| Création Dashboard des résultats sur Jira | Ayoub |
| Test auto de la fonctionnalité “Recherche par catégorie” sur Cypress | Mohamed Aziz |
| Test auto de la fonctionnalité “Accès au panier” sur Cypress | Mohamed Aziz |
| Test auto de la fonctionnalité “Recherche et sélection d’un article” sur Cypress | Mohamed Aziz |
| Test auto du fonctionnalité “Passage commande” sur Selenium | Ayoub |
| Test auto de la fonctionnalité “Remplir Message” sur Selenium | Ayoub |
| Test auto de la fonctionnalité “Modifier les infos personnelles” sur Selenium IDE | Mohamed Aziz |
| Test auto de performance sur Jmeter | Mohamed Aziz |

# Test Fonctionnel

## Périmètre d'intervention de l'activité

L’activité de test vise à tester les fonctionnalités estimées comme majeures du site Decathlon selon la disponibilité du temps du projet.

## Recette associée au projet

Les fonctionnalités testées sont :

* Connexion & déconnexion à un compte Decathlon
* Créer un compte sur le site Decathlon
* Rechercher et filtrer un article
* Passer une commande
* Modifier les informations personnelles
* Ajout & Suppression d'un article au/du panier sur le site Decathlon
* Remplir un message

## Les éléments du produit/ logiciel qui seront testés

Dans le site Decathlon, les scénarios à testés sont principalement “Connexion/Déconnexion” “Passage d’une commande” et “Remplir un message”.

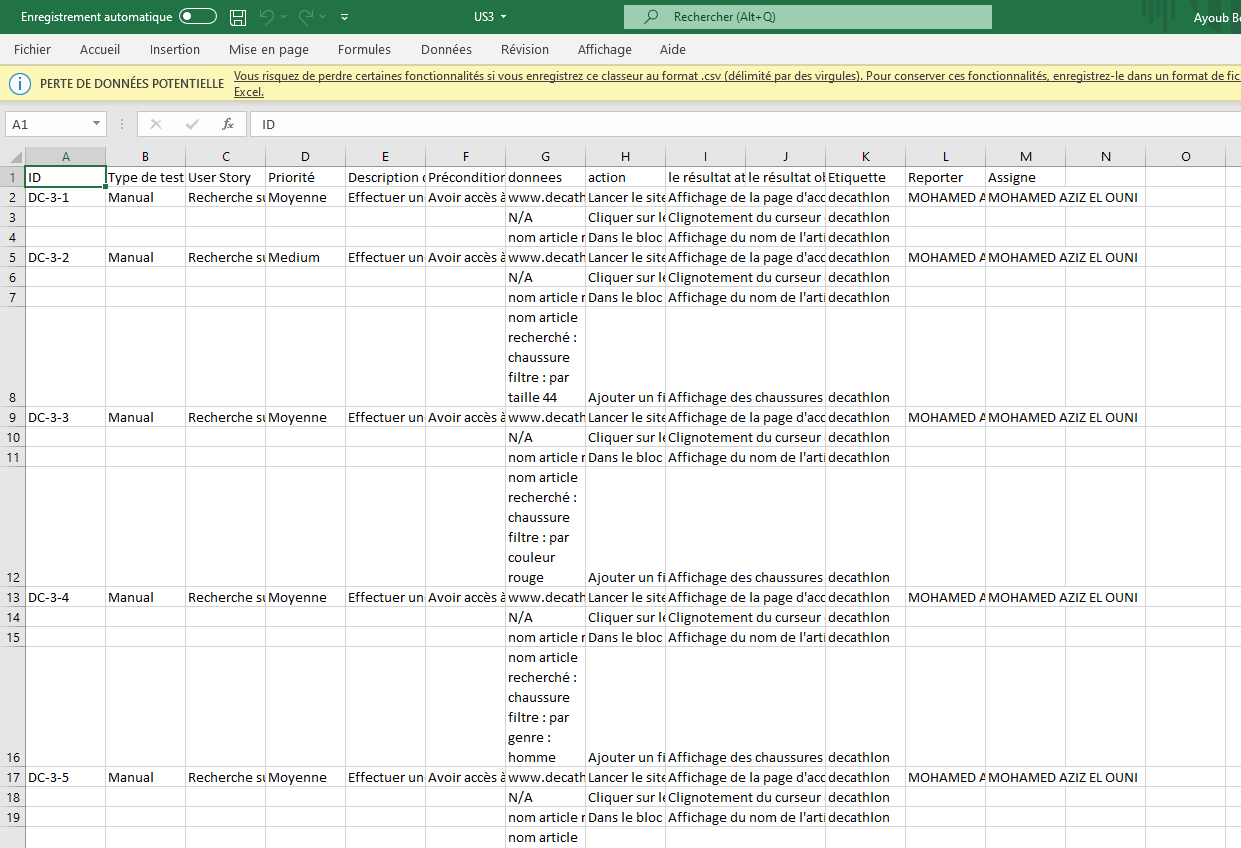
## Les éléments qui sont exclus de la stratégie

* Test de API (on n’a pas l’accès à la partie BackEnd).
* Tests de stress avec un nombre d’utilisateurs élevé (Risque de bloquer les site).
* Test auto de la fonctionnalité “Connexion au site” (La connexion se fait avec un mail de validation).

## Implémentation des tests

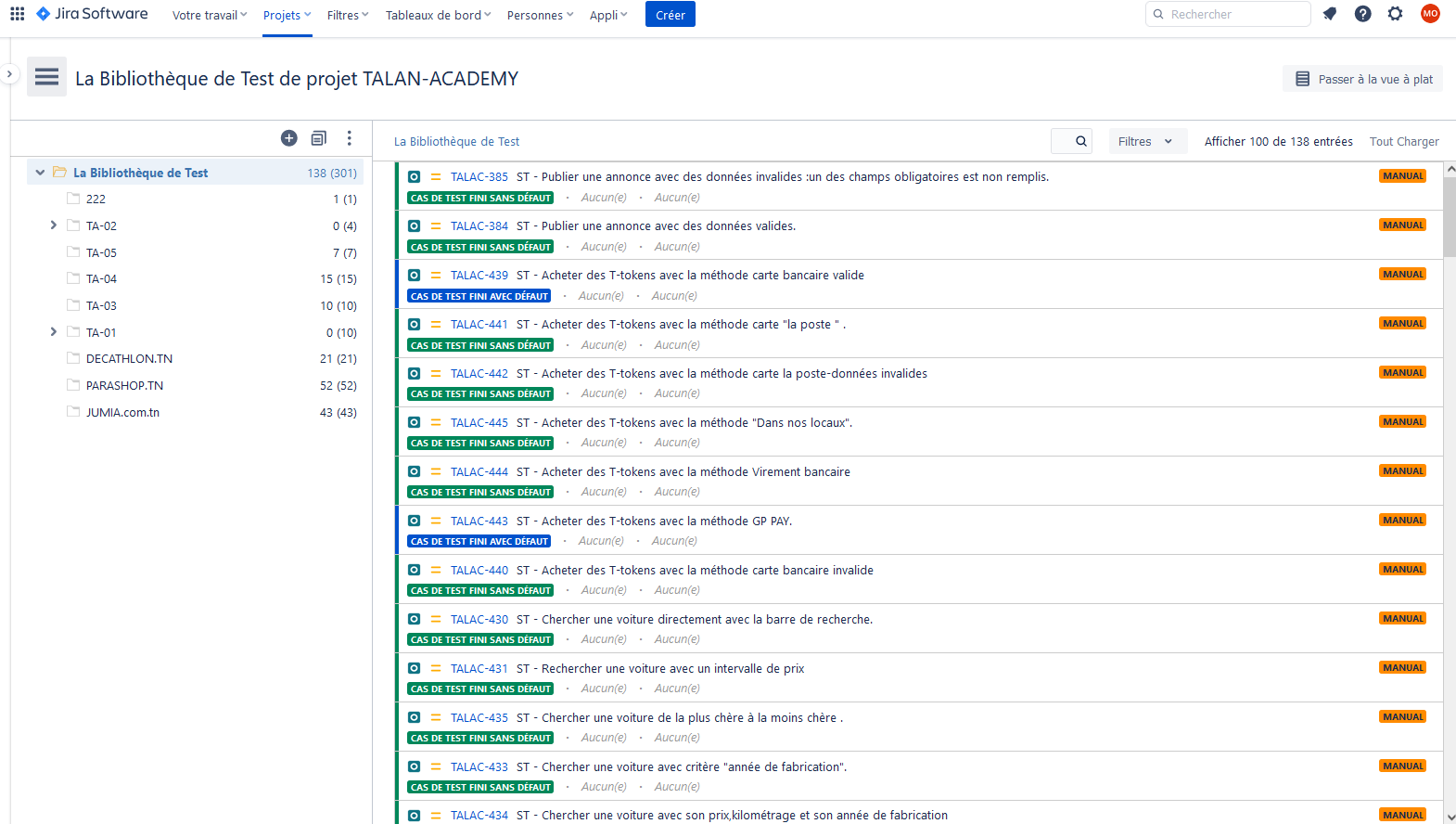
Les cas de tests sont rédigés sur GoogleSheet et reformuler selon le besoin du Jira :

Chaque cas de test a été rédigé dans un seul fichier Excel, restructuré et convertit en CSV.

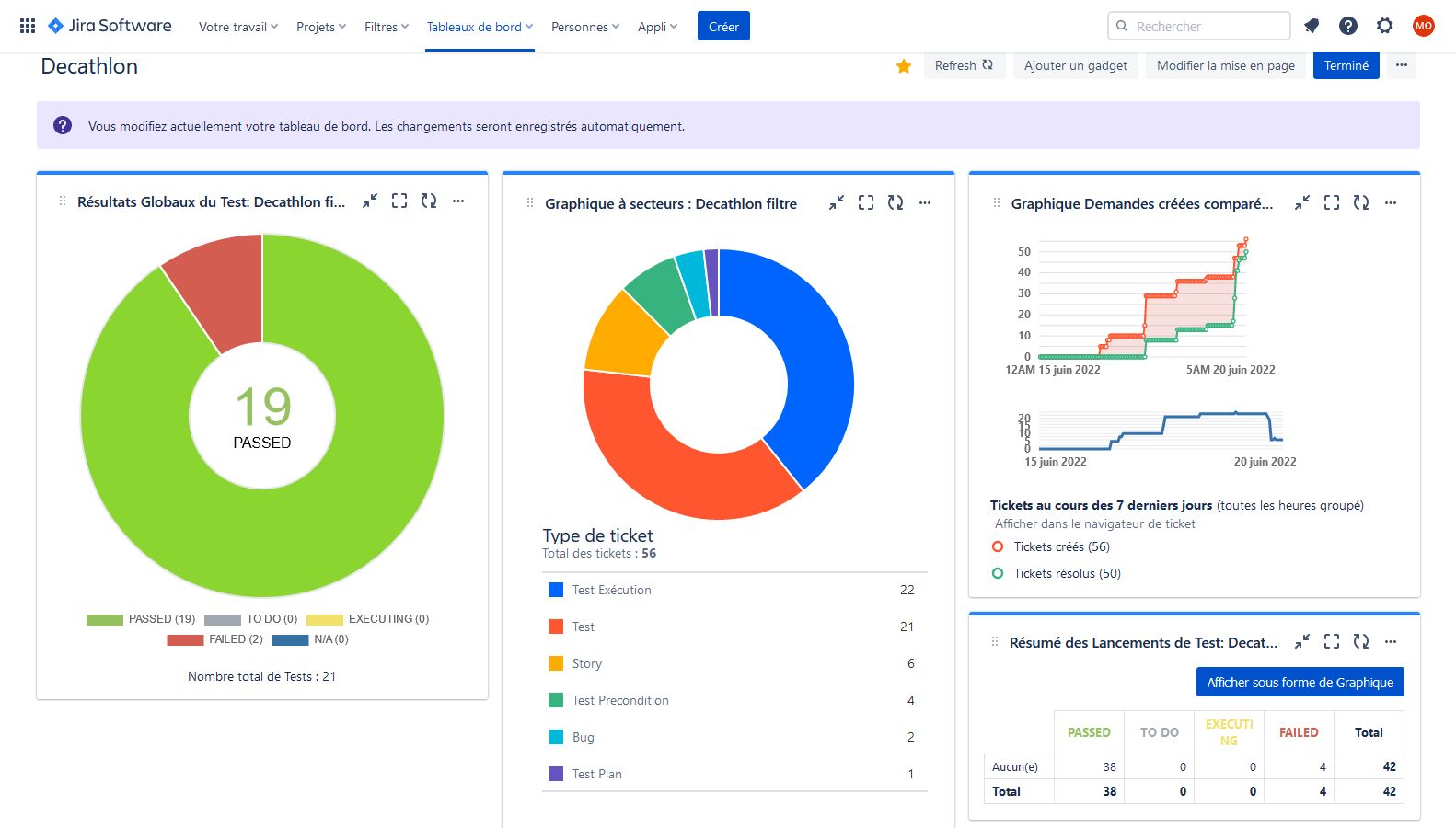


## Exécution des tests

Une fois on a importé les cas de tests sur Jira, on vérifie les détails de chaque cas de test et on assure la connexion entre le plan de test, les US et les cas de tests.



On lance l’exécution des cas de tests et on génère le Dashboard et le rapport d’exécution



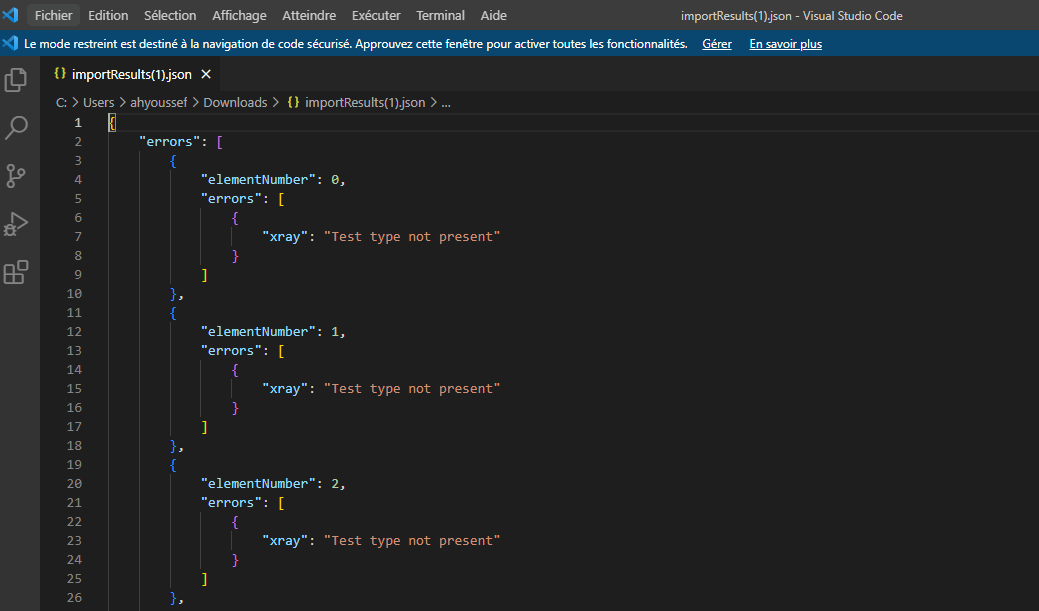
Les tests résultent deux anomalies d’un total de 21 cas de tests.

## Clôture des tests

En vérifiant les critères de sortie, le clôture de test fonctionnel est validé. Les résultats de tests vont être vérifier par les outils test automatique.

## Difficultés & Contraintes trouvés

|  |  |
| --- | --- |
| **Contrainte** | **Solution proposée** |
| Importation des cas de test sur Xray doit être dans les normes demandés de l’outil | Suivre le rapport d’erreur et corriger |
| Un seul projet pour tester de tous les sites | Créer un filtre par étiquette “Decathlon” afin de distinguer nos tickets |



# Test Auto Cypress

## Environnement

Windows 10 Professional

Google Chrome Version 103.0.5060.53 (Build official) (64 bits)

## Scénario de test automatisé

* Accès au panier
* Recherche par catégorie
* Recherche et sélection d’un article

## Contrainte

|  |  |
| --- | --- |
| **Contrainte** | **Solution proposée** |
| Chute du site [www.decathlon.tn](http://www.decathlon.tn/) | Lancer les scénarios sur [www.decathlon.fr](http://www.decathlon.fr/) |
| Impossible d’afficher les résultats de générer un rapport sur le test effectué :  Essai sur VSC effectué sans résultat favorable | Enregistrement par capture vidéo des scénarios de test et préparation d’un fichier avec les codes rédiger |

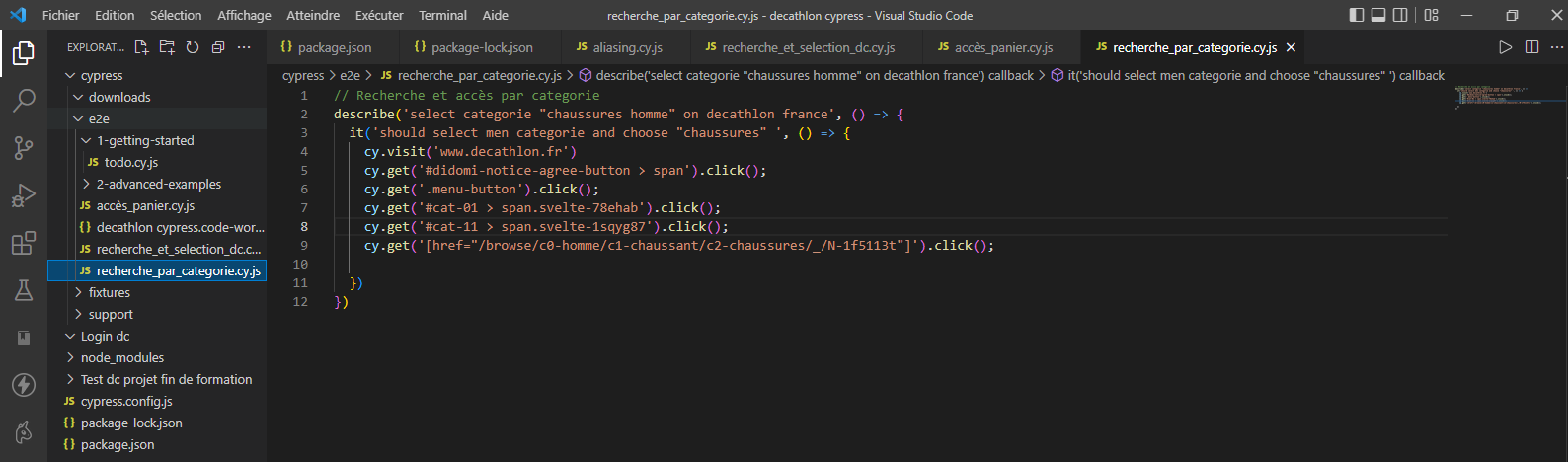
## Conception des cas de test

* Scénario accès au panier
* Scénario sélection par catégorie
* Scénario Recherche par catégorie

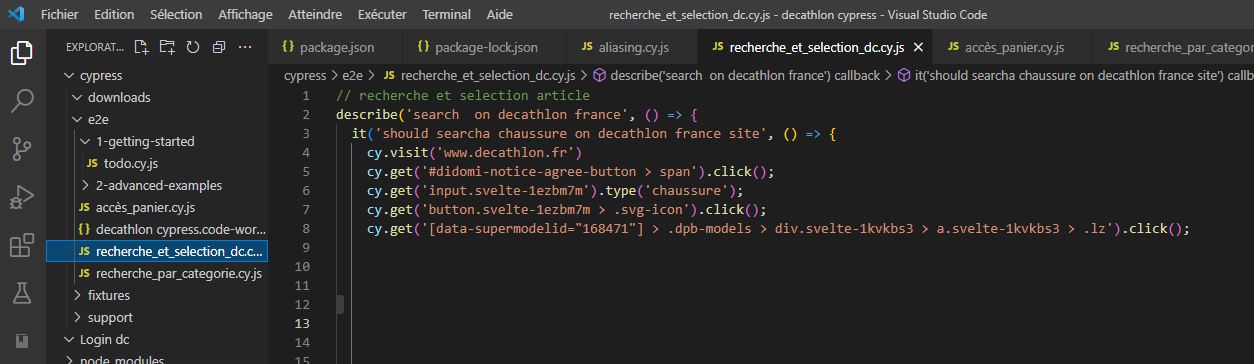
## Implémentation des cas de test

A l’aide de l’outil Visual Studio Code, les cas de test ont été rédigé et implémenter

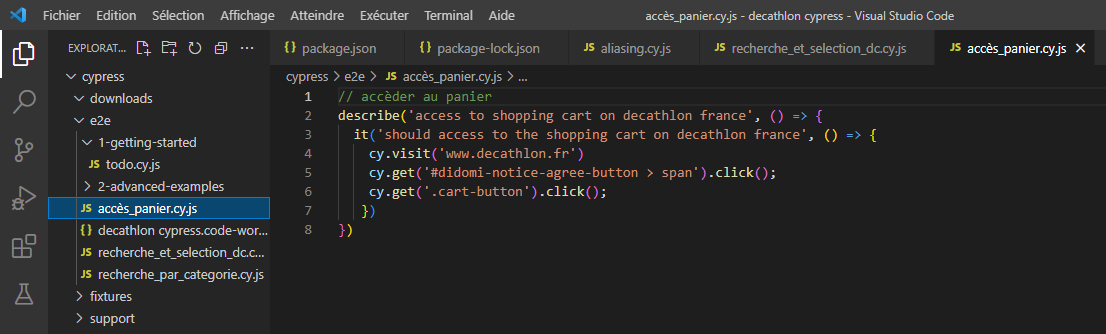
* Scénario Recherche par catégorie :



* Scénario Recherche et sélection :



* Scénario Accès au panier :



## Exécution des cas de test

L'exécution a été effectué en lançant Cypress à partir du terminal de Visual Studio Code

(À regarder la vidéo dans le dossier partagé)

## Conclusion

Malgré les contraintes rencontrées durant l'exécution de ces cas de test nous avons pu anticiper ces derniers grâce à une étude des risques et nous avons pu effectuer une couverture complète des cas de test implémenter.

# Test auto sur Selenium

## Périmètre d'intervention de l'activité

Ce test auto vise à valider les résultats de test fonctionnel des deux fonctionnalités suivantes :

1. Passage d’une commande : Recherche d’un article, Commander, Passer à une autre recherche d’article, choisir la quantité à acheter, et passer la commande.
2. Remplir un message : Passage à la page cible, Remplir les cases, Envoyer le message.

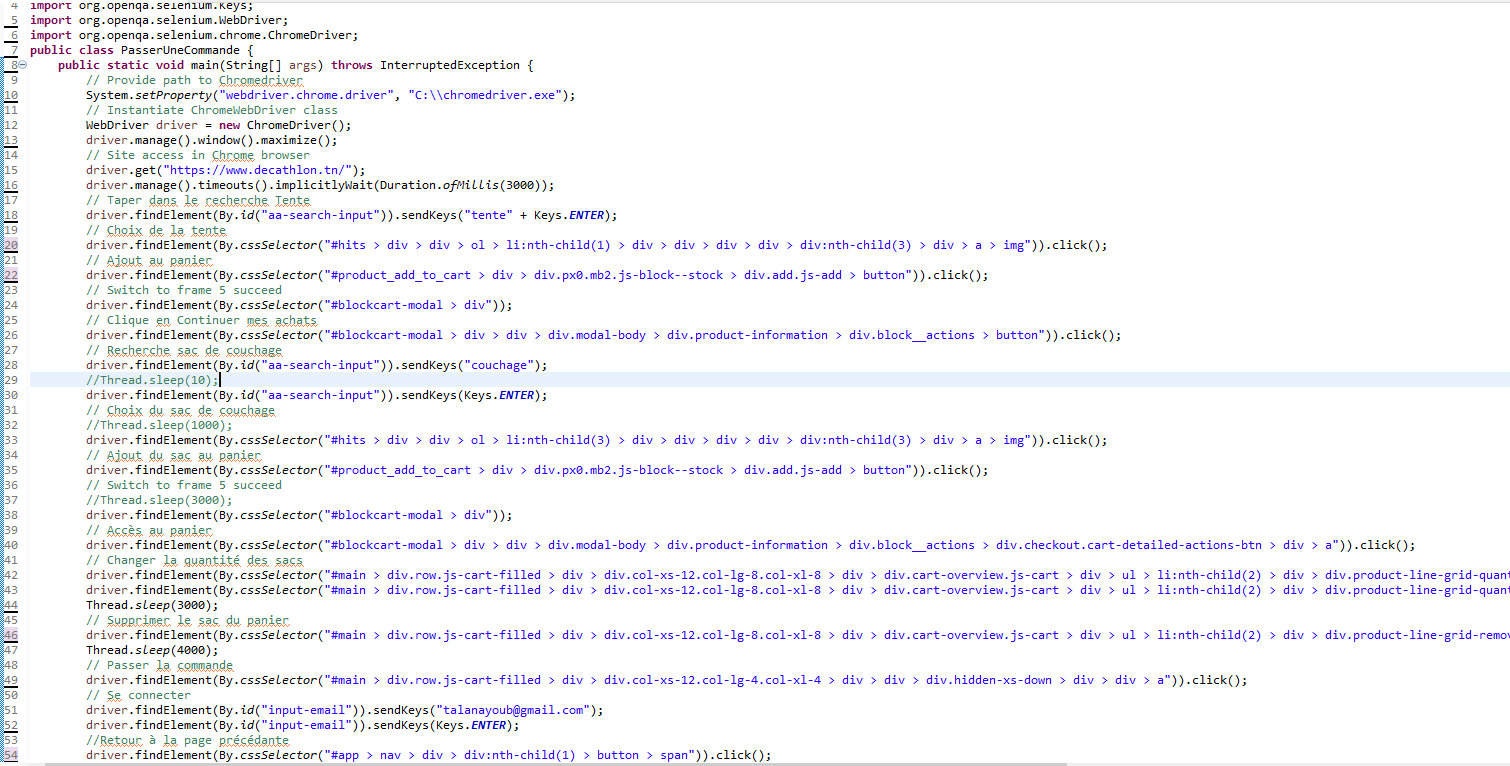
## Environnement de test

Le code est rédigé sur IDE Eclipse en langage Java en ajoutant les librairies Selenium et les Web drivers des navigateurs Chrome et Microsoft Edge.

* Code du Scénario “Remplir un message”



* Code du Scénario “Passage d’une commande”



## Difficultés & Contraintes trouvés

|  |  |
| --- | --- |
| **Contrainte** | **Solution proposée** |
| Changement de la version du navigateur | Vérifier la version du navigateur avant d’exécuter |
| Vitesse d’exécution du test est importante  Non-détection du l’élément suivant | Utiliser les outils “Wait” dans le code pour ralentir l'exécution. |
| Changement de Frame | Trouver la commande pour se basculer du Frame principal à un autre |

# Test Auto SELENIUM IDE

## Environnement

Windows 10 Professional

Google Chrome Version 103.0.5060.53 (Build official) (64 bits)

## Scénario de test automatisé

* Login avec informations Valide
* Login avec mail invalide
* Login avec mot de passe Invalide

## Contrainte

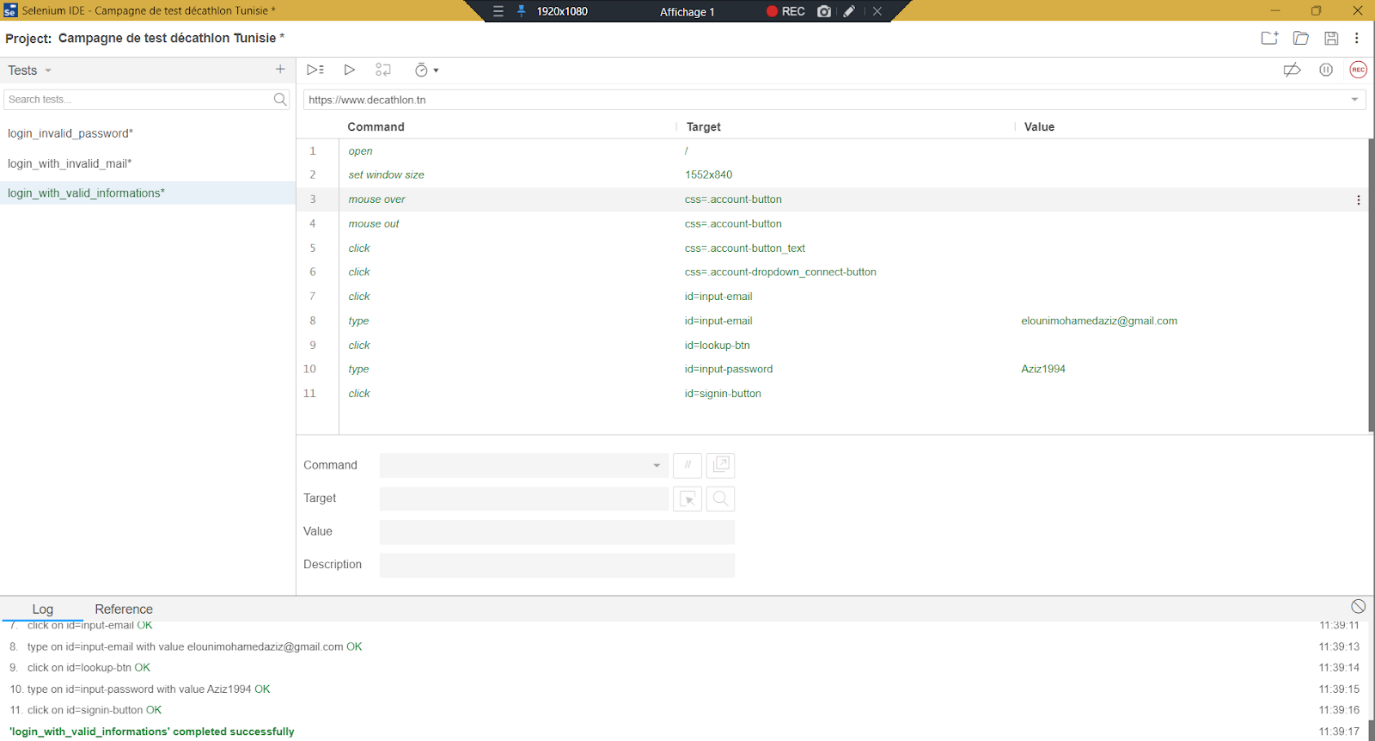
|  |  |
| --- | --- |
| **Contrainte** | **Solution proposée** |
| Impossibilité de générer des rapports de test après exécution  →indisponibilité de l'extension test report (selenium IDE) | Enregistrement par capture vidéo de l’exécutions des scénarios de test et des Screenshot et préparation d’un fichier avec les codes rédiger |

## Conception des cas de test

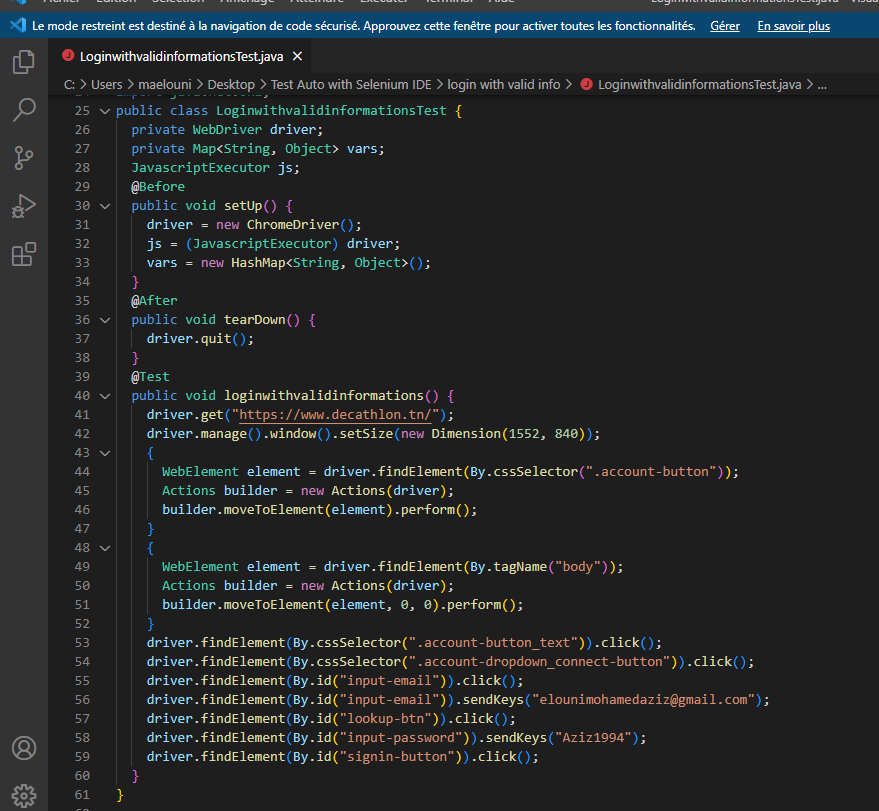
Avec l’outil selenium IDE, nous avons enregistré les scénarios de test et réalisé un test de bout en bout.

## Exécution des cas de test

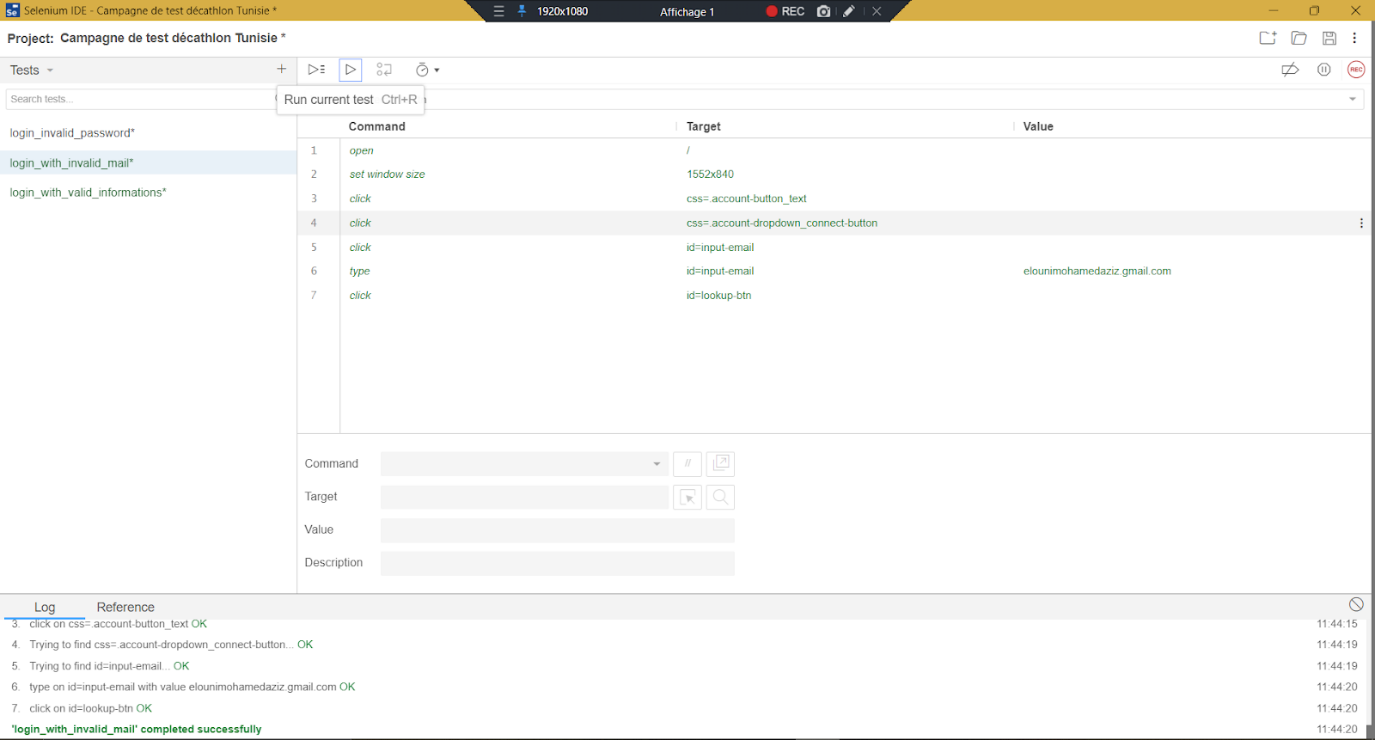
### Scénario Login avec informations valide



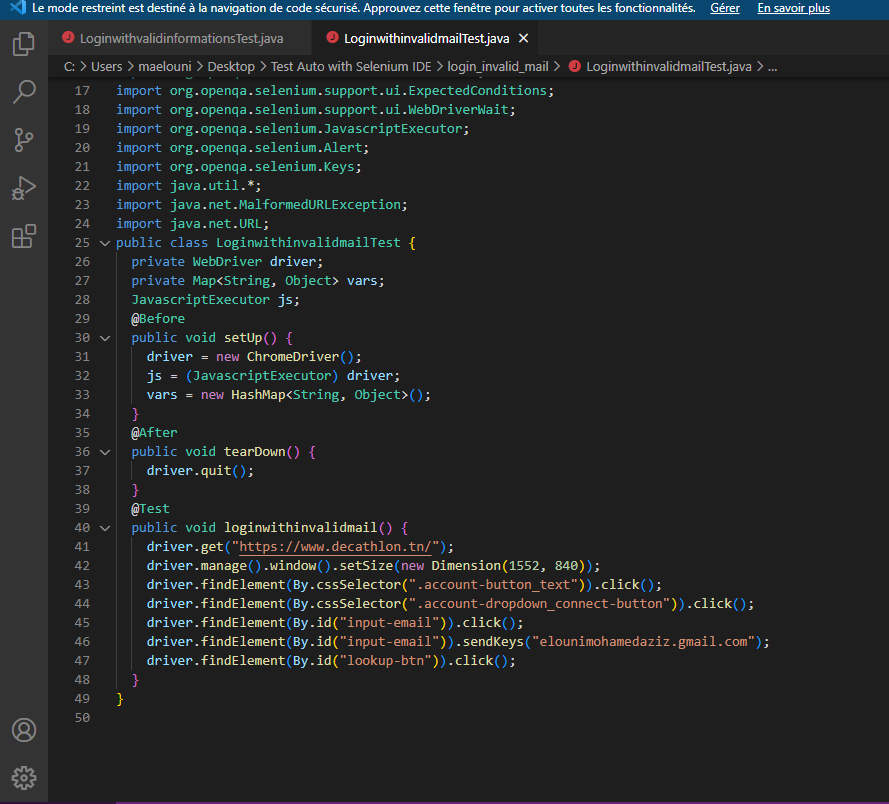
Pour donner suite à l’exécution, un code est automatiquement généré :



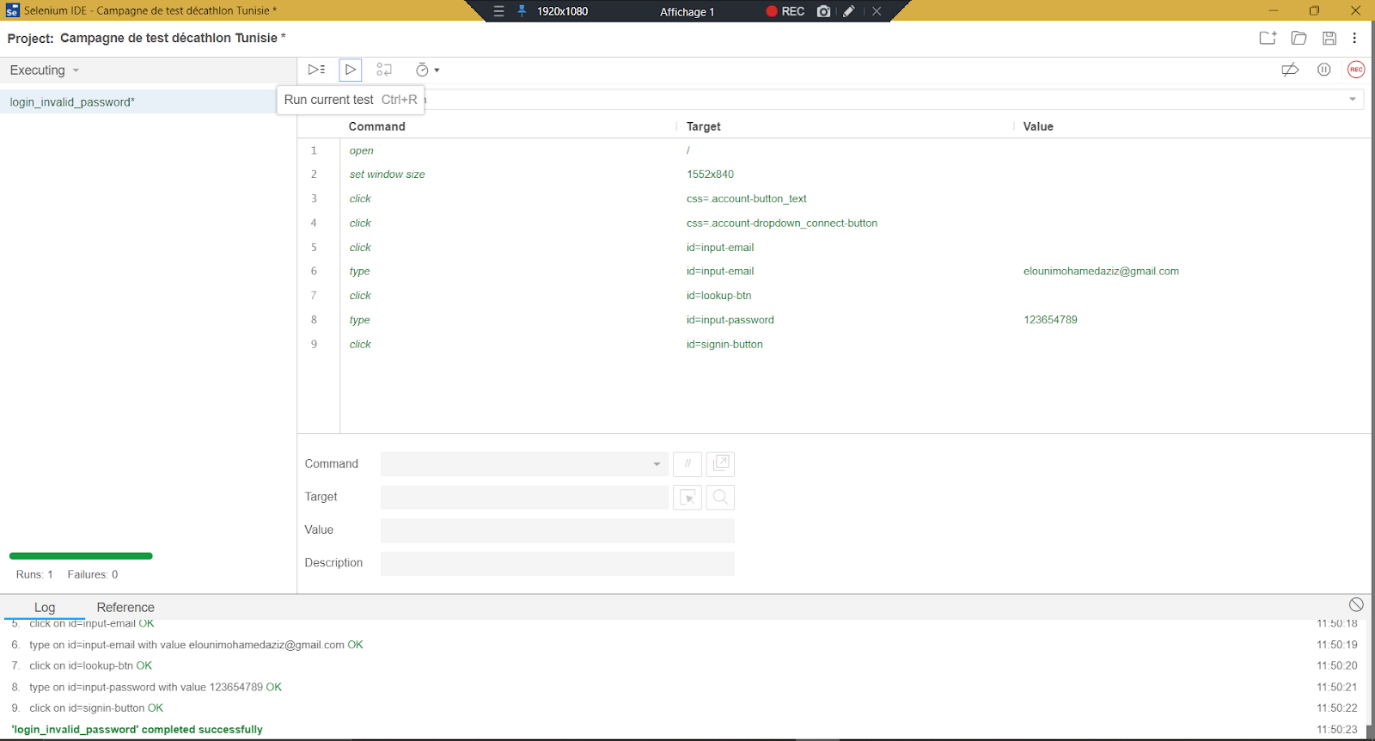
### Scénario Login avec mail invalide



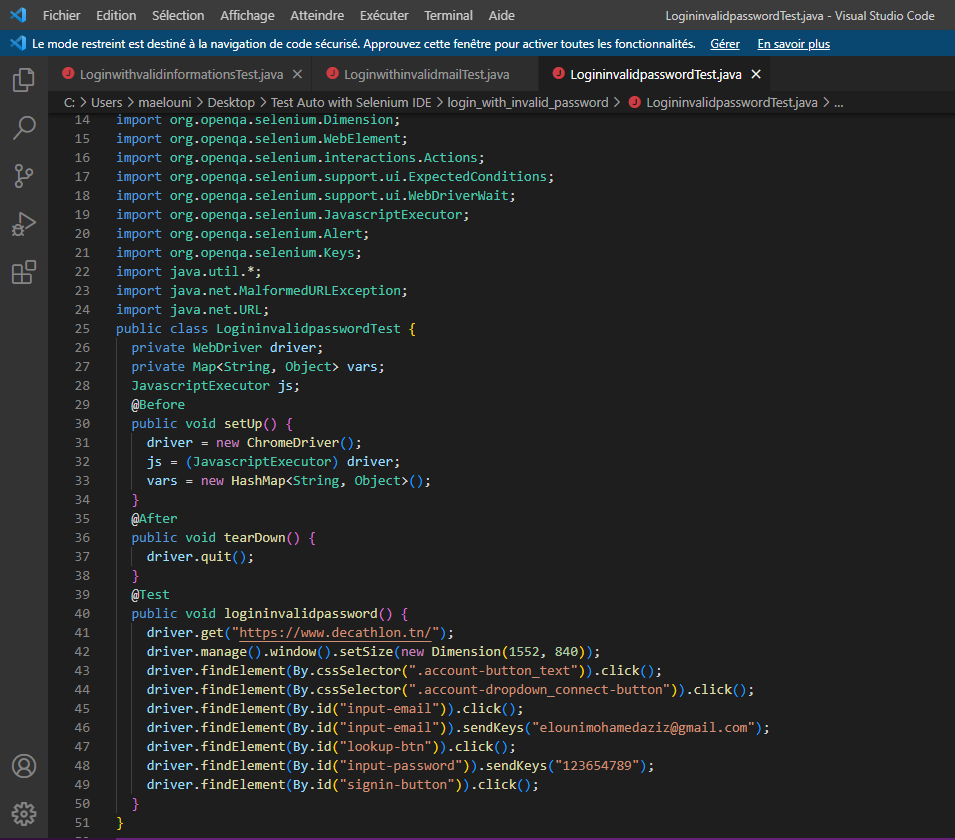
À la suite de l’exécution, un code est automatiquement est généré :



### Scénario Login avec mot de passe invalide



À la suite de l’exécution, un code est automatiquement généré :



Des vidéos de l'exécution ont été aussi enregistrés et seront remis avec les livrables

A la fin du test, Cypress génère un tableau récapitulatif comme suit



## Conclusion

Après enregistrement des scénarios, l'exécution a été effectuée avec succès malgré la contrainte rencontrés, la solution apportée a permis d’avoir une idée sur l'exécution de ces scénarios.

Pour les prochaines campagnes en utilisant Selenium IDE, un outil de reporting sera mis en place afin de faciliter la gestion des KPI.

# Test de Performance

## Environnement

Windows 10 Professional

Google Chrome Version 103.0.5060.53 (Build official) (64 bits)

## Scénario de test et données de test



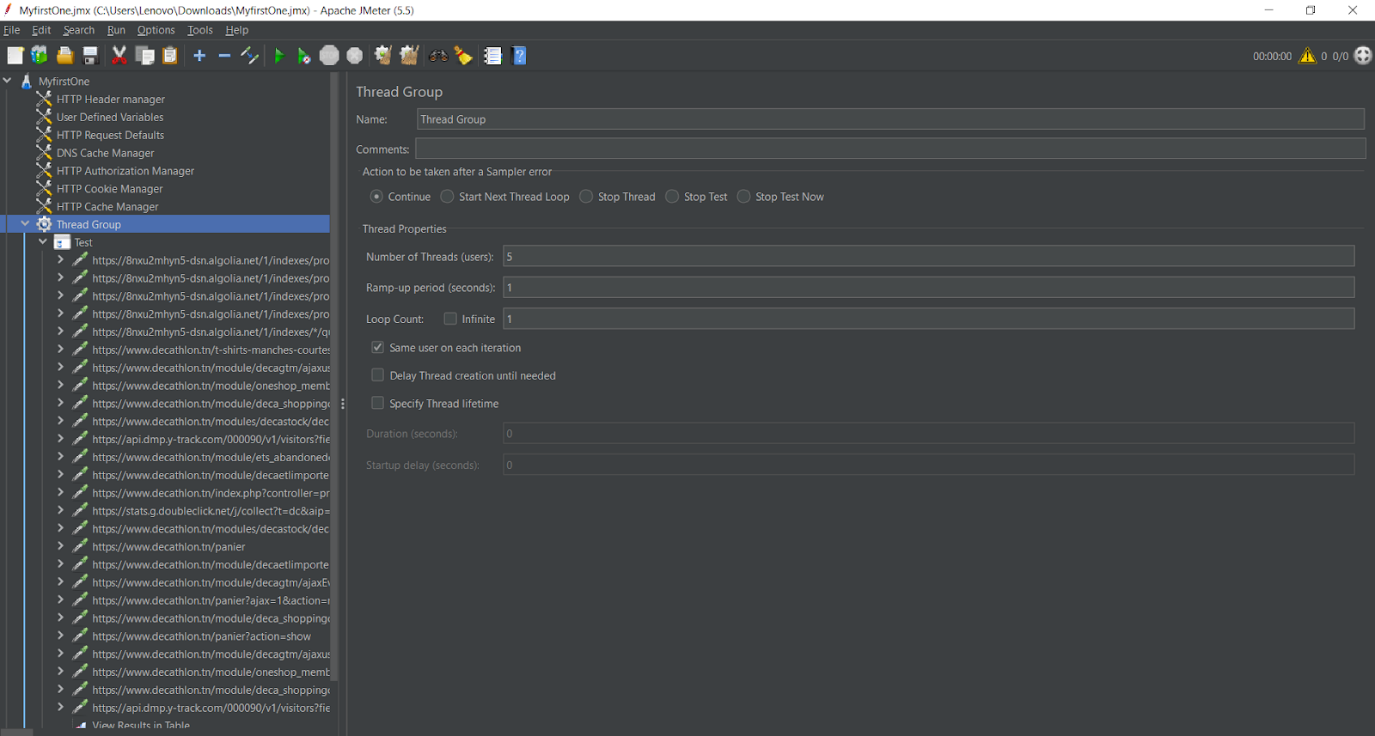
## Contrainte

|  |  |
| --- | --- |
| **Contrainte** | **Solution proposée** |
| Impossibilité de réaliser un test de performance en injectant un grand nombre d’utilisateur durant une période importante  →Risque de blocage de la part du site  → durée de génération de rapport trop longue | Réaliser un test de performance simple en diminuant le nombre d’utilisateur et le temps. |

## Implémentations du test

Rédaction du scénario à l’aide de l’extension chrome Blazemeter.

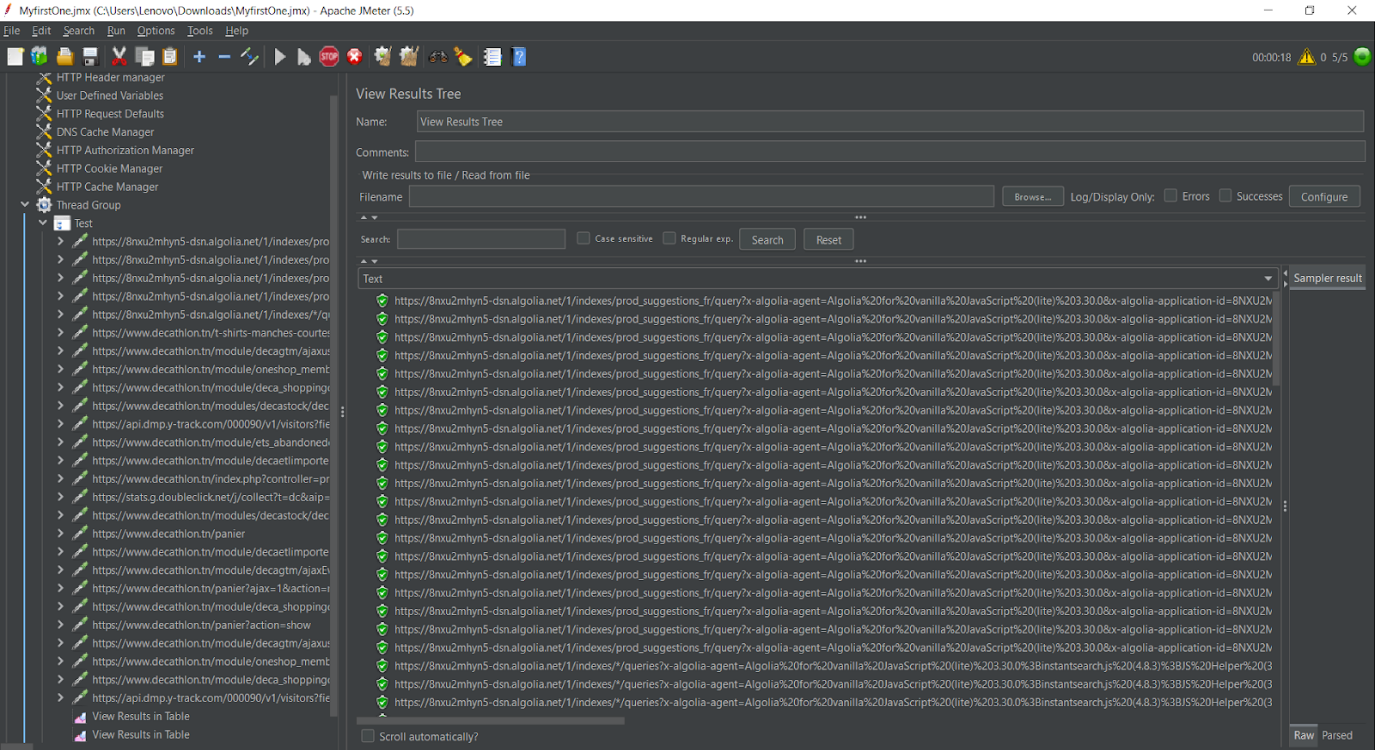
Implémenter le scénario enregistrer sur l’outil JMETER



Injection des paramètres de test pour les performances du site5 utilisateur chaque secondes pendant une minute.

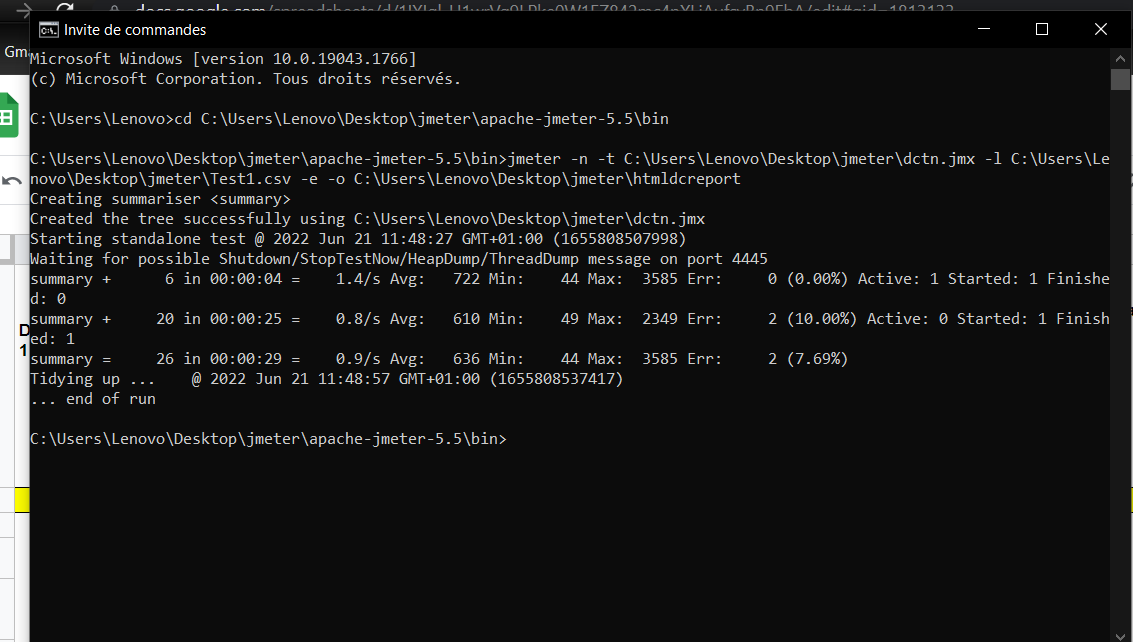
## Exécution des cas de test

L'exécution a été réalisée à l'aide de l’outil JMeter après insertion des paramètres et choix de l’affichage grâce à l’ajout de la fonctionnalité “*view results in tree* “ et “*view results in table”*



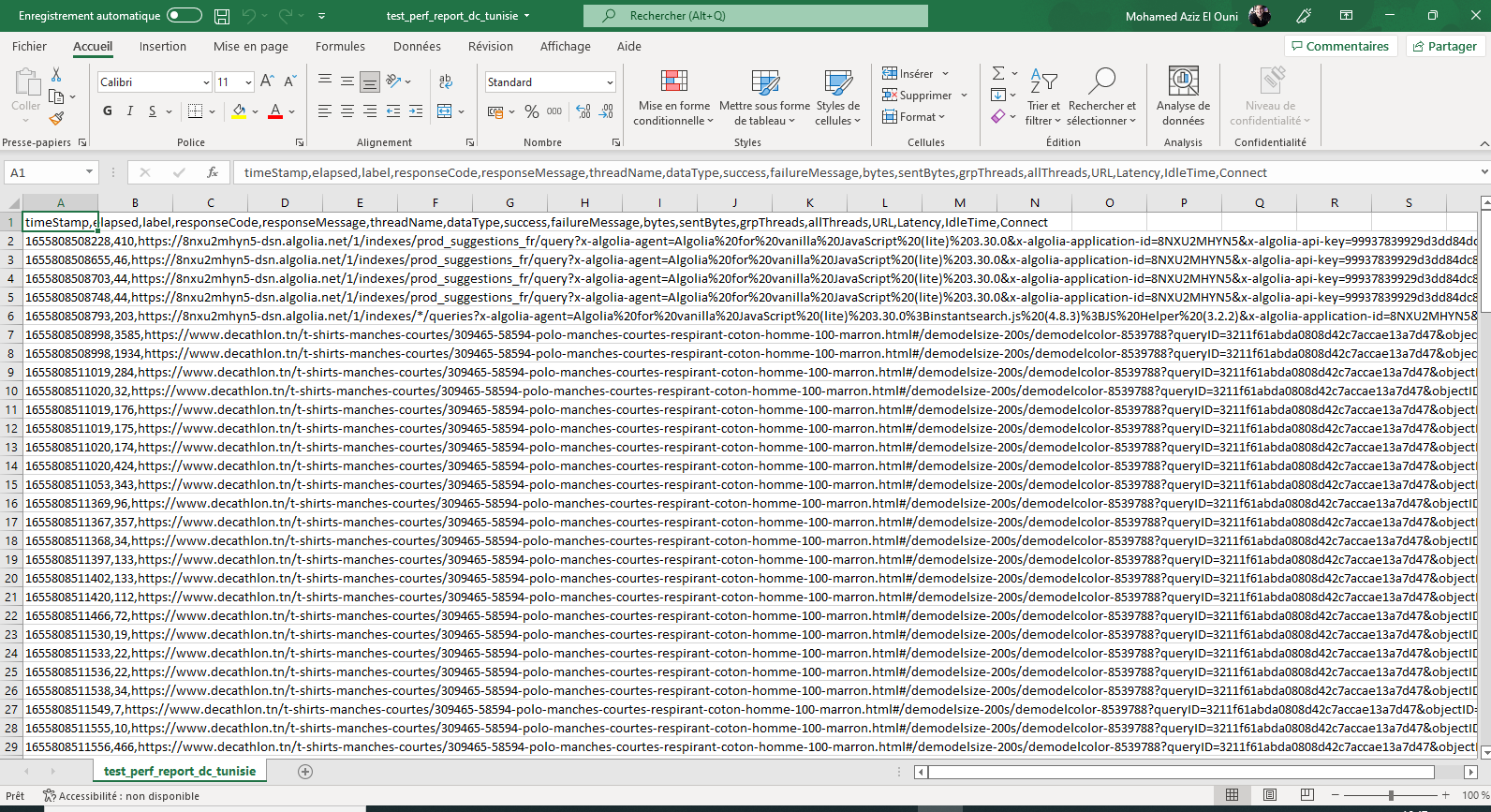
## Génération du rapport de test

Le rapport a été généré grâce à l’outil invite de commande

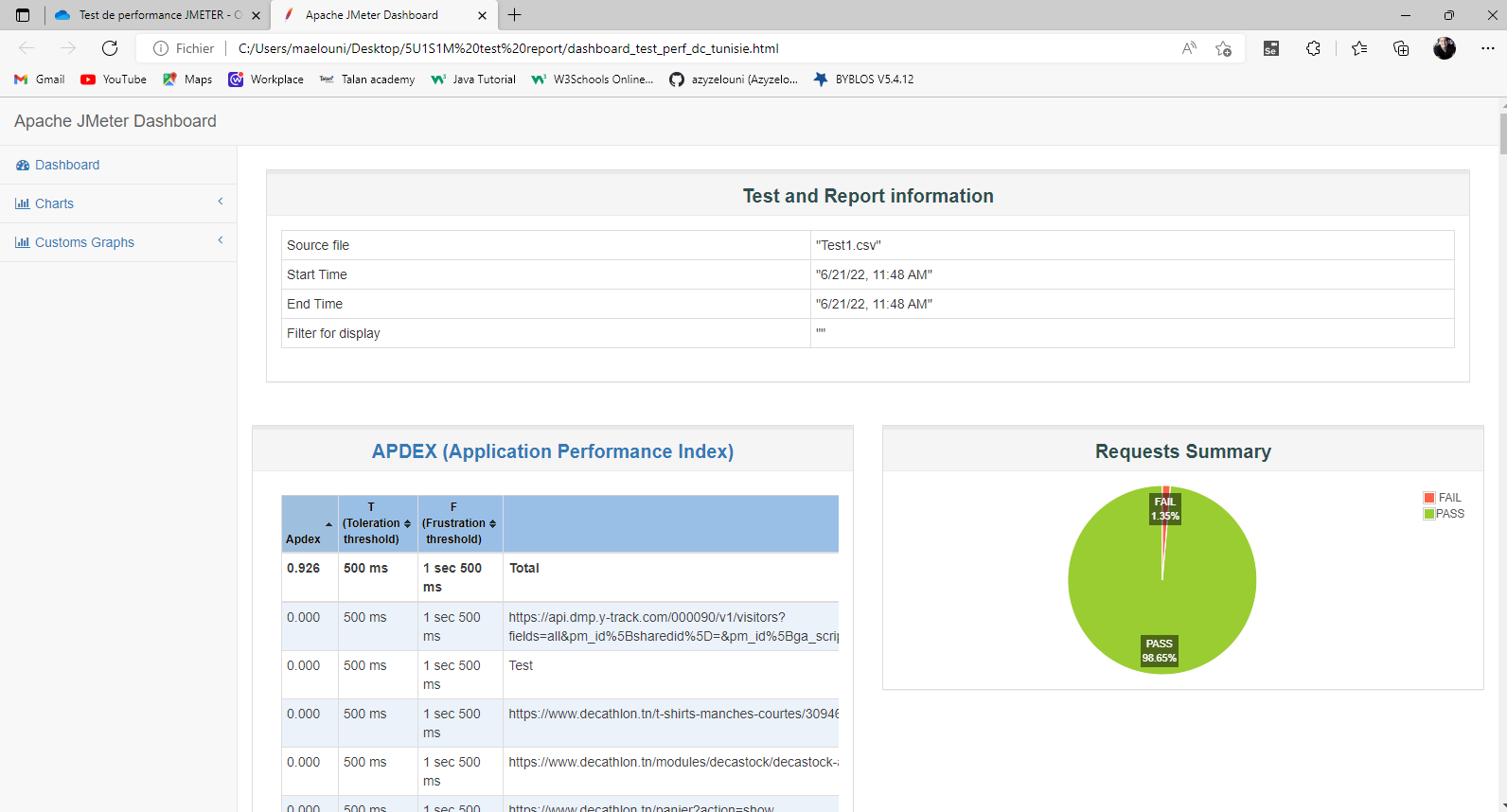


Les rapports sont fournis sous deux formes différentes :

* Un fichier .CSV



* Un Dashboard sous format HTML



## Conclusion

Le test de performance sur le site [Decathlon.tn](http://www.decathlon.tn/) a été réalisé de façon simple par suite de la période de génération des rapports qui posait contrainte pour l’avancement de la campagne et le temps estimé.

Un test plus robuste serait programmé lors d’un prochain sprint afin de bien tester la robustesse du site.

# Conclusion générale

Ce projet a mis en test les fonctionnalités majeurs du site Decathlon.tn afin de vérifier son bon fonctionnement et améliorer l’expérience du l’utilisateur final du site.

Les tests sont effectués de la façon manuelle à travers l’outil de planification des tests manuels Xray Jira, et d’une façon automatique par les outils de tests automatique (Cypress, Selenium, et Selenium IDE). Un test de performance par l’outil Jmeter met l'accent sur la résistance du site à l'accès simultané par un grand nombre d'utilisateurs.

En outre, le test du délai de réponse et le test de l'application Android du site sont importants pour valider la fiabilité du service.