

Projet 2 : Automatisation de la recherche e-commerce

Robot Framework & SeleniumLibrary pour la boutique Lumina

Établissement :
ENET'COM

Étudiants :
Ahmed AbdDayme AHMEDBOUHA

Moussa Mahmoud BA

Classe : 3 GT TST

Enseignante : Mme INES Jemal

Année universitaire : 2024/2025

— Dépôt : https://github.com/ahmedabddayme3752/projet_test/

Table des matières

1	Introduction	2
2	Vue d'ensemble du projet	2
2.1	Objectif	2
2.2	Périmètre	2
2.3	Lien du dépôt GitHub	2
3	Stack technique	2
4	Structure du dépôt	3
5	Mise en route	3
5.1	Prérequis	3
5.2	Lancer le site local	3
5.3	Exécuter les tests	3
6	Scénarios testés	3
7	Principaux mots-clés Robot (extraits)	4
8	Captures / Reporting	4
9	Captures d'écran	4
9.1	Captures des rapports Robot Framework	6
9.2	Trace de la commande Robot	7
10	Conclusion	8

1 Introduction

Ce document couvre le **Projet 2** donné par l'enseignante : automatisation de la fonction de recherche sur un site e-commerce (boutique Lumina). Les tests sont écrits avec Robot Framework et SeleniumLibrary pour valider l'expérience utilisateur côté front.

2 Vue d'ensemble du projet

2.1 Objectif

Automatiser la validation de la recherche produit (résultats existants, catégories, aucun résultat, correspondance partielle) pour garantir fiabilité et régression rapide.

2.2 Périmètre

- Site vitrine statique (HTML/CSS/JS) avec logique de recherche côté client.
- Tests end-to-end via navigateur piloté par SeleniumLibrary.
- Génération de rapports Robot Framework dans le dossier `results/`.

2.3 Lien du dépôt GitHub

Le code source complet du projet est disponible sur GitHub :

- Dépôt : <https://github.com/ahmedabddayem/projetTest>

3 Stack technique

- **Frontend** : HTML5, CSS3 (glassmorphism), JavaScript vanilla.
- **Automation** : Robot Framework, SeleniumLibrary, Python.
- **Outils** : Navigateur contrôlé par WebDriver.

4 Structure du dépôt

```

projetTest/
├── ecommerce_project/
│   ├── website/           # The Application Under Test
│   │   ├── index.html      # Main storefront
│   │   ├── style.css        # Styling
│   │   └── script.js        # Search logic & Data
│   └── tests/              # Test Automation Code
│       ├── search_tests.robot # Test Suite
│       └── resources.robot    # Reusable Keywords
└── results/                # Test Execution Reports (Log/Report)

```

FIGURE 1 – **Figure X** : Structure du projet projetTest/

5 Mise en route

5.1 Prérequis

```
1 pip install robotframework robotframework-seleniumlibrary
```

Listing 1 – Installation des dépendances

5.2 Lancer le site local

```
1 cd ecommerce_project/website
2 python3 -m http.server 8000
3 # Accès : http://localhost:8000
```

Listing 2 – Serveur local

5.3 Exécuter les tests

```
1 robot -d results ecommerce_project/tests/search_tests.robot
```

Listing 3 – Lancer la suite Robot

Les rapports HTML (log/report) sont générés dans `results/`.

6 Scénarios testés

- **Produit existant** : vérifie l'affichage correct des résultats.
- **Recherche par catégorie** : filtrage (ex. “Audio”).
- **Aucun résultat** : état vide et message utilisateur.
- **Correspondance partielle** : tolérance aux mots partiels.

7 Principaux mots-clés Robot (extraits)

- Ouverture du navigateur et navigation vers `http://localhost:8000`.
- Saisie d'un terme de recherche et déclenchement de la recherche.
- Vérification du nombre et du contenu des cartes produits affichées.
- Validation du message d'état quand aucun produit n'est trouvé.

8 Captures / Reporting

- Les exécutions produisent `report.html` et `log.html` dans `results/`.
- Les captures d'écran peuvent être ajoutées dans `results/` via SeleniumLibrary si nécessaire.

9 Captures d'écran

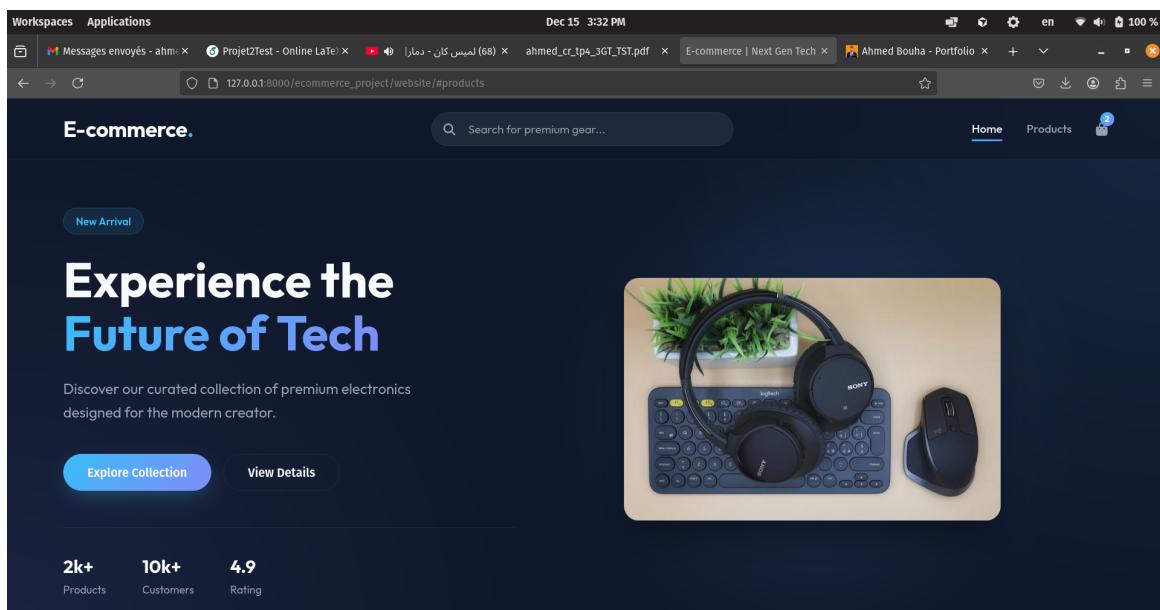


FIGURE 2 – **Figure 1 :** Accueil de la boutique (recherche par défaut)

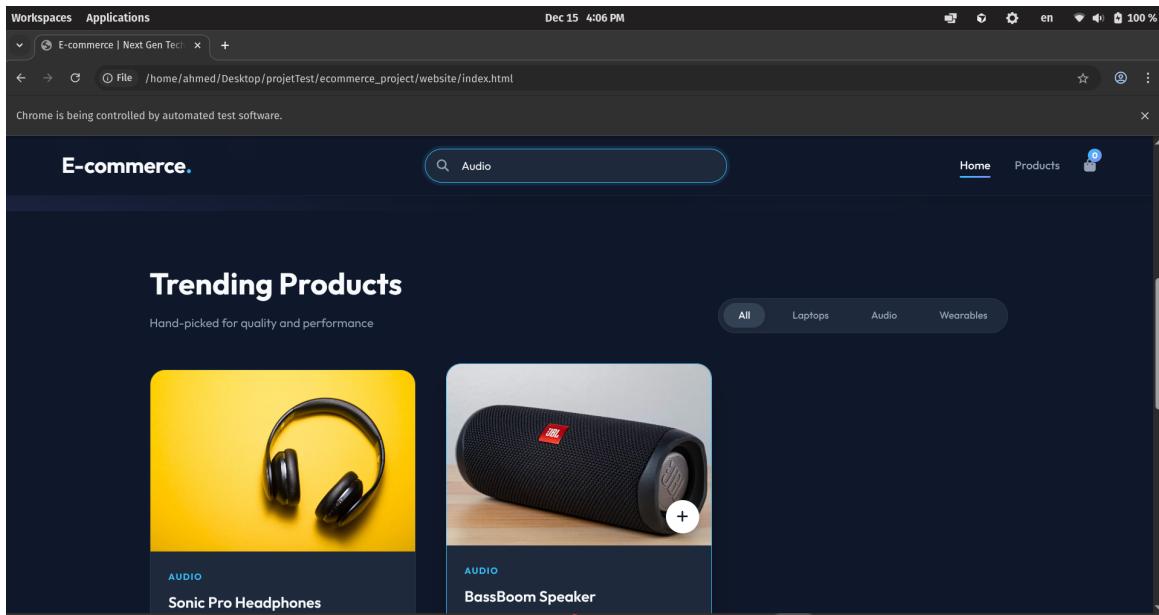


FIGURE 3 – Figure 2 : Résultats pour un produit existant

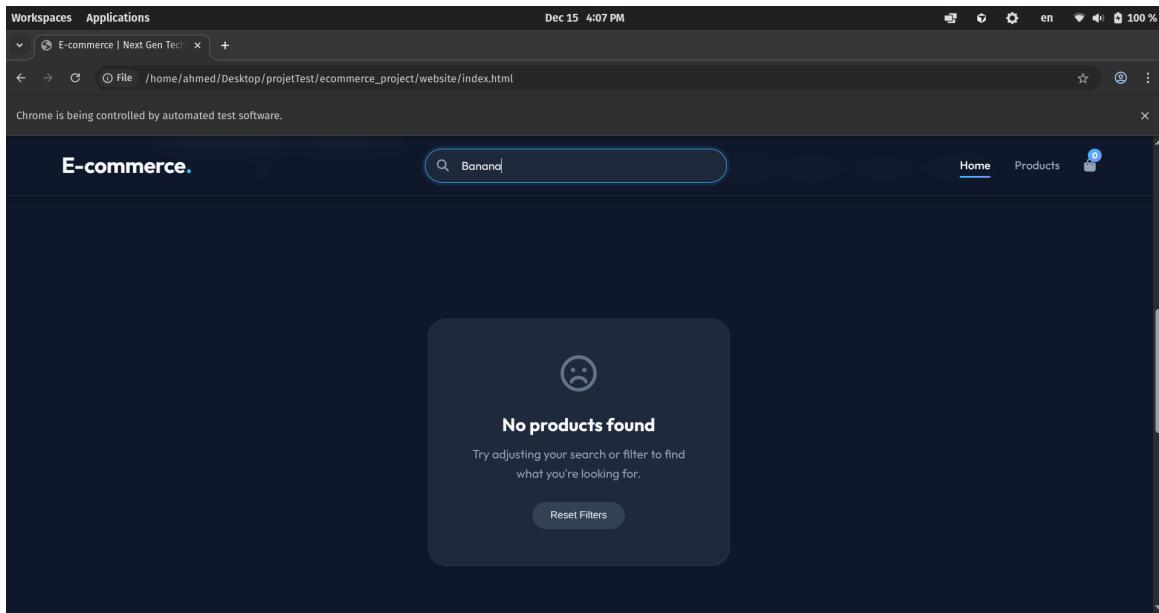


FIGURE 4 – Figure 3 : État vide quand aucun produit ne correspond

9.1 Captures des rapports Robot Framework

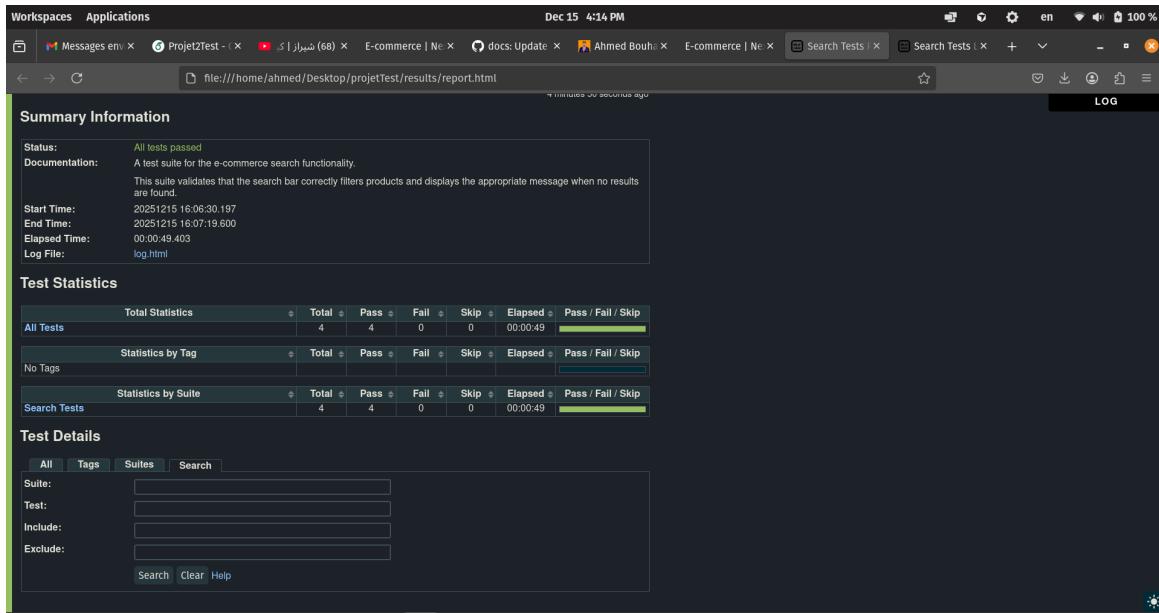


FIGURE 5 – Figure 4 : Rapport Robot Framework `report.html` avec 4 tests passés

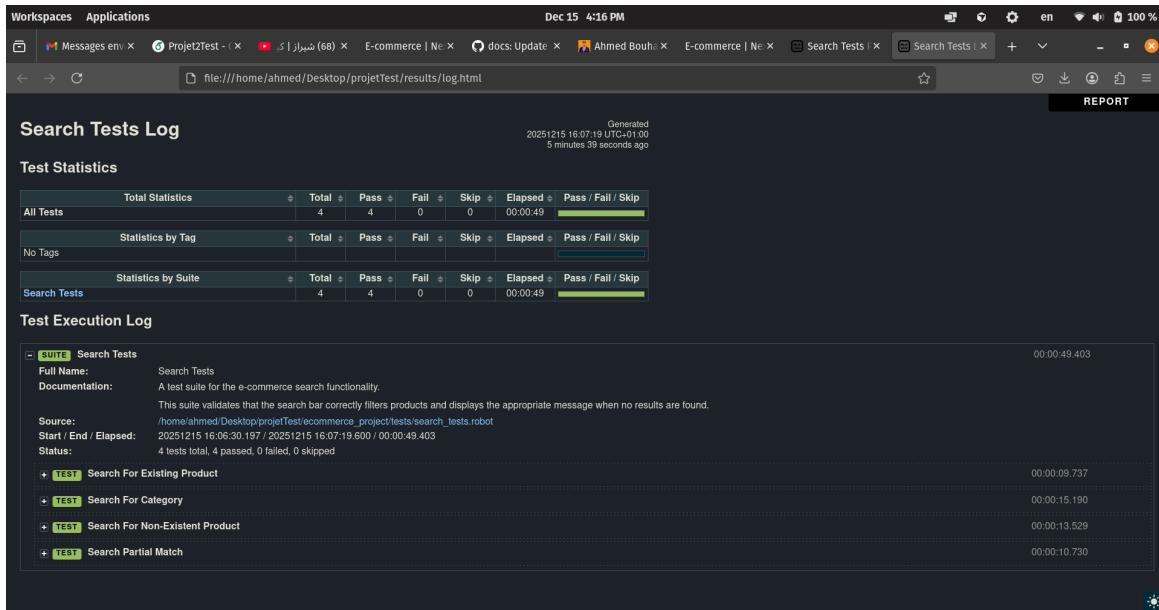


FIGURE 6 – Figure 5 : Détail d'exécution dans `log.html` (mots-clés et captures)

```
● ahmed@pop-os:~/Desktop/projetTest$ robot -d results ecommerce_project/tests/search_tests.robot
=====
Search Tests :: A test suite for the e-commerce search functionality.
=====
Search For Existing Product :: Test searching for a product that e... | PASS |
-----
Search For Category :: Test searching for a category (e.g., 'Audio'). | PASS |
-----
Search For Non-Existent Product :: Test searching for a term that ... | PASS |
-----
Search Partial Match :: Test searching with a partial string. | PASS |
-----
Search Tests :: A test suite for the e-commerce search functionality. | PASS |
4 tests, 4 passed, 0 failed
=====
Output: /home/ahmed/Desktop/projetTest/results/output.xml
Log:   /home/ahmed/Desktop/projetTest/results/log.html
Report: /home/ahmed/Desktop/projetTest/results/report.html
```

FIGURE 7 – Figure 6 : Capture du terminal lors de l'exécution de `robot -d results ...`

9.2 Trace de la commande Robot

```
1 cd /home/ahmed/Desktop/projetTest
2 robot -d results ecommerce_project/tests/search_tests.robot
3 =====
4 Search Tests :: A test suite for the e-commerce search functionality.
5 =====
6 Search For Existing Product :: Test searching for a product that e...
| PASS |
7 -----
8 Search For Category :: Test searching for a category (e.g., 'Audio').
| PASS |
9 -----
10 Search For Non-Existent Product :: Test searching for a term that ...
| PASS |
11 -----
12 Search Partial Match :: Test searching with a partial string.
| PASS |
13 -----
14 Search Tests :: A test suite for the e-commerce search functionality.
| PASS |
15 4 tests, 4 passed, 0 failed
16 =====
17 Output: /home/ahmed/Desktop/projetTest/results/output.xml
18 Log:   /home/ahmed/Desktop/projetTest/results/log.html
19 Report: /home/ahmed/Desktop/projetTest/results/report.html
```

Listing 4 – Exécution des tests depuis le terminal

10 Conclusion

L'automatisation garantit la fiabilité de la recherche e-commerce et accélère les cycles de régression. Les prochains pas peuvent inclure des scénarios multi-navigateurs et l'intégration continue.