

Présentation du Projet d'Automatisation En Test Logiciel.



Auteurs : Mohamed Reguigui & Firas Taktak

Université de Sfax — Année académique : 2025/2026

Plan du Présentation – Projet d'Automatisation De Test



Vue d'Ensemble du Projet

Portée et Techniques

- Tests automatisés d'une application web avec Selenium et Python.
- Plus de **20 cas de test** au total pour une couverture exhaustive.
- Les **4 suites de test** principales incluent :
 - Tests de performance
 - Tests multi-navigateurs
 - Tests de design réactif
 - Tests de stress et de charge

Méthodologies & Livrables

- Les techniques clés utilisées sont :
 - Analyse des valeurs limites (AVL)
 - Tests de transition d'état
 - Tests de configuration
 - Mesure de charge de base
- Les livrables du projet comprennent :
 - Les tests exécutés
 - Un rapport de bugs détaillé
 - Des résultats documentés

Site Web Sous Test (SWUT)

Nous avons sélectionné une application web publique pour nos tests, dont les caractéristiques garantissent la pertinence pour les techniques de boîte noire.

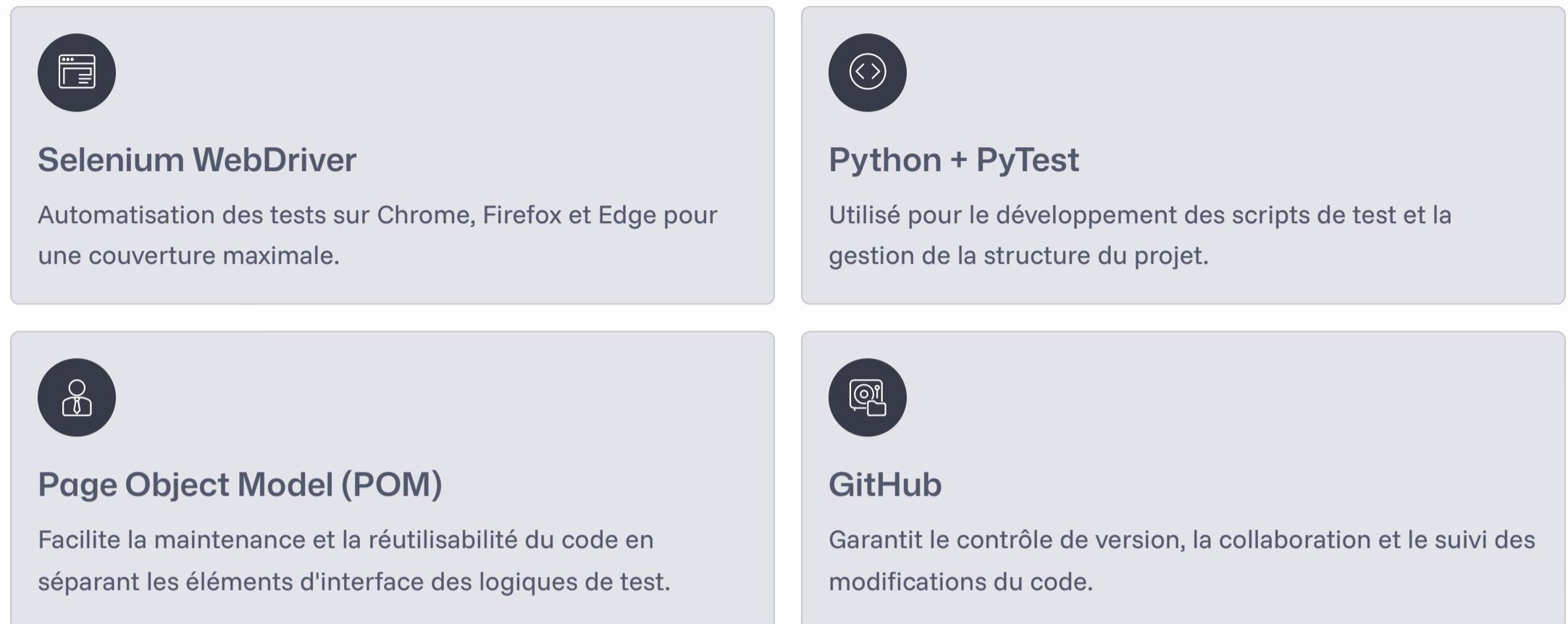
Détails de l'SWUT

- **Nom du site :** Data.Gov.tn
- **Nature :** site web publique accessible à tous.
- **Les fonctionnalités clés testées comprennent :**
 - Authentification des utilisateurs
 - Fonctionnalités de recherche
 - Navigation dans le site
 - Soumission de formulaires
 - Réactivité de l'interface utilisateur (UI)
- **Raison de la sélection :** Sa structure stable et sa représentation du monde réel en font un excellent candidat pour l'application des techniques de test boîte noire.

The screenshot shows the homepage of data.gov.tn. At the top, there's a navigation bar with links for "Demandes d'ouverture des données", "FAQ", "Contact", "Plan de site", and accessibility options (A-, A, A+). On the right side of the header is a search bar and a "Connexion" button. The main content area features a search bar with placeholder text "Entrer un mot clé, un titre ...", two dropdown menus for "Choisir un thème de donnée" and "Choisir un producteur de données", and three large circular statistics: "2784 Jeux de données", "197 Producteurs de données", and "10 Cas de réutilisation". To the right, there's a section titled "Actualités" with a thumbnail image showing people working with data visualizations like charts and graphs. Below it, a news item is displayed: "Réunion de travail sur les données publiques ouvertes au profit du Ministère de la Santé" dated "30/7/2025". The footer contains links for "Privacy - Terms" and the Tunisian flag.

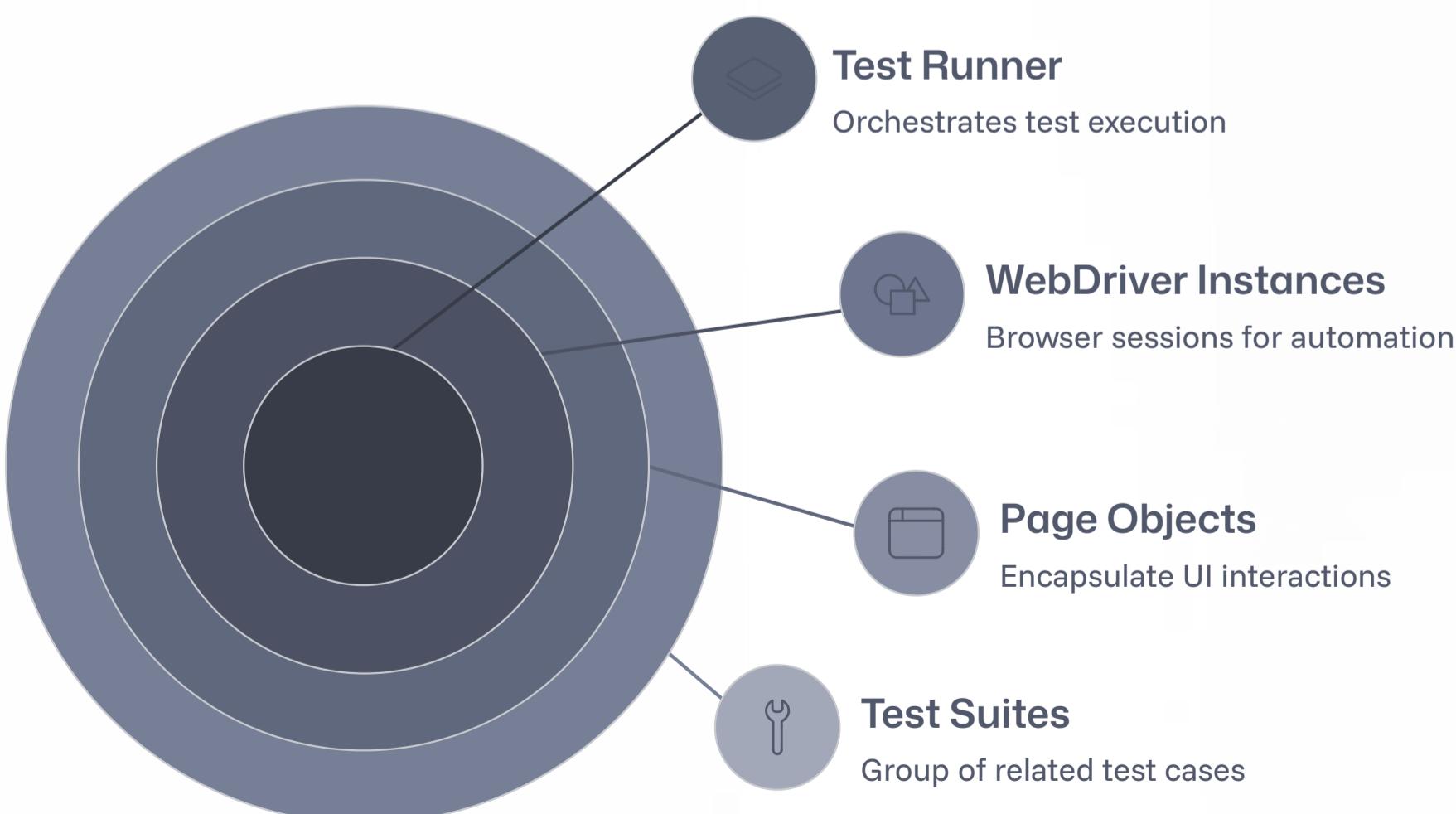
Architecture de Test

Notre architecture de test est conçue pour l'efficacité et la maintenabilité, en s'appuyant sur des outils robustes et des pratiques de développement éprouvées.



Structure du Projet

- `/tests` : Contient tous les scripts de test exécutables.
- `/pages` : Héberge les définitions des Page Objects.
- `/drivers` : Contient les exécutables des navigateurs (WebDriver).
- `/utils` : Regroupe les fonctions utilitaires et les aides.
- `/reports` : Stocke les rapports d'exécution des tests.



Techniques de Test Boîte Noire Utilisées

1

Partition en Classes d'Équivalence

Division des données d'entrée en partitions valides et invalides.

2

Analyse des Valeurs Limites

Test des valeurs aux limites de chaque partition pour détecter les erreurs courantes.

3

Test de Transition d'État

Vérification des transitions entre les états de l'application.

4

Test de Table de Décision

Couverture des combinaisons de conditions et d'actions complexes.

5

Devinette d'Erreurs (Error Guessing)

Utilisation de l'expérience pour anticiper les erreurs potentielles.

6

Test de Configuration

Évaluation des performances et fonctionnalités sur diverses configurations (navigateurs, résolutions).

7

Mesure de Charge/Temps de Réponse

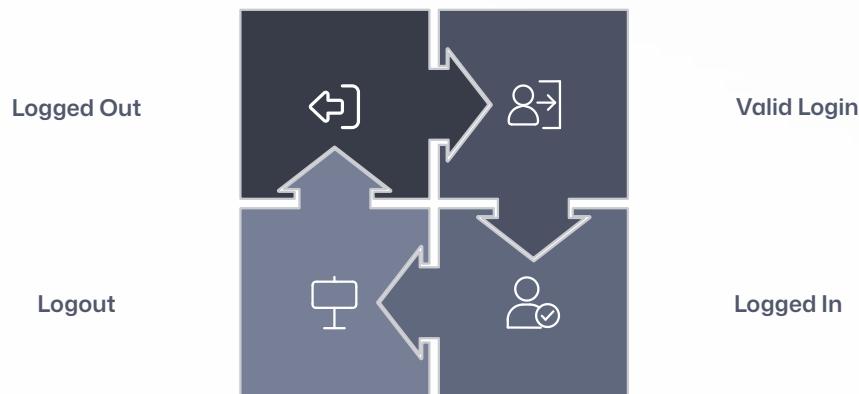
Évaluation de la réactivité de l'application sous différentes charges.

Test de Transition d'État

Scénario : Connexion → Tableau de bord → Déconnexion

Les tests de transition d'état évaluent la capacité du système à passer d'un état à l'autre en fonction d'événements spécifiques.

Diagramme de la Machine à États



Ce diagramme illustre le parcours simple de l'utilisateur à travers les états de connexion et de déconnexion.

Cas Testés

- **Connexion invalide** : L'utilisateur reste dans l'état **[Déconnecté]**.
- **Connexion puis fermeture de l'onglet** : La session est invalidée, l'utilisateur revient à l'état **[Déconnecté]** lors de la prochaine tentative.
- **Déconnexion explicite** : L'état est réinitialisé, l'utilisateur passe de **[Connecté]** à **[Déconnecté]**.

Chaque transition et état est minutieusement vérifié pour s'assurer d'un comportement prévisible et sécurisé.

Suite de Test 1 : Tests Fonctionnels

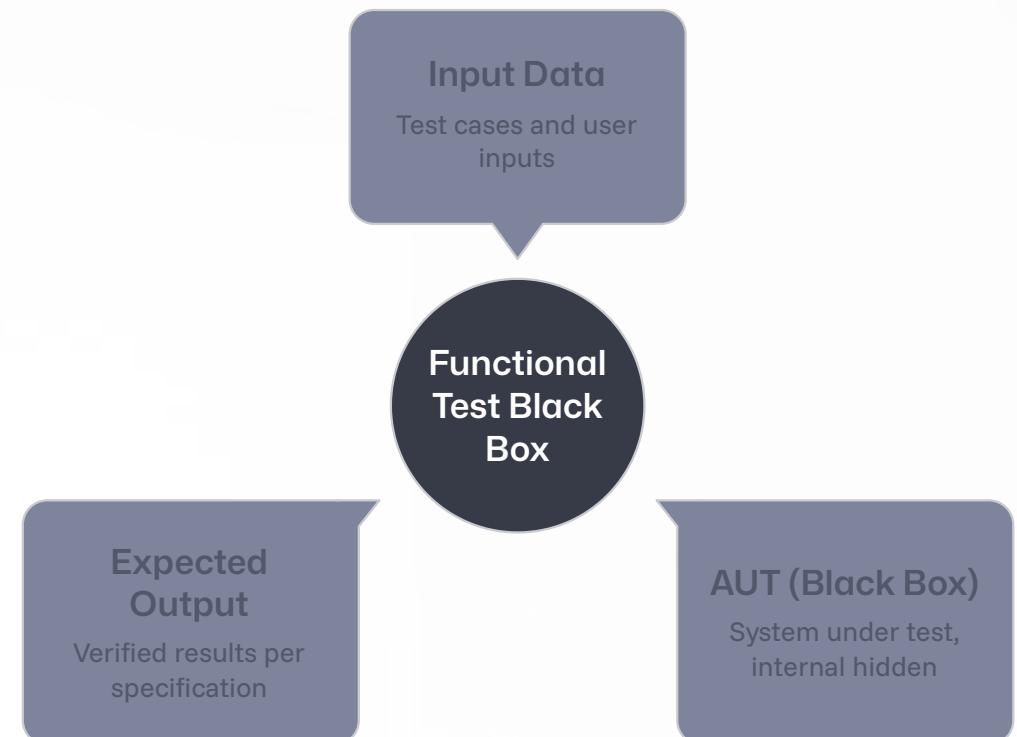
Valider les fonctionnalités principales de la Web site

Cette suite se concentre sur la vérification que chaque fonctionnalité de l'application se comporte comme prévu selon les spécifications.

Exemples de Cas de Test

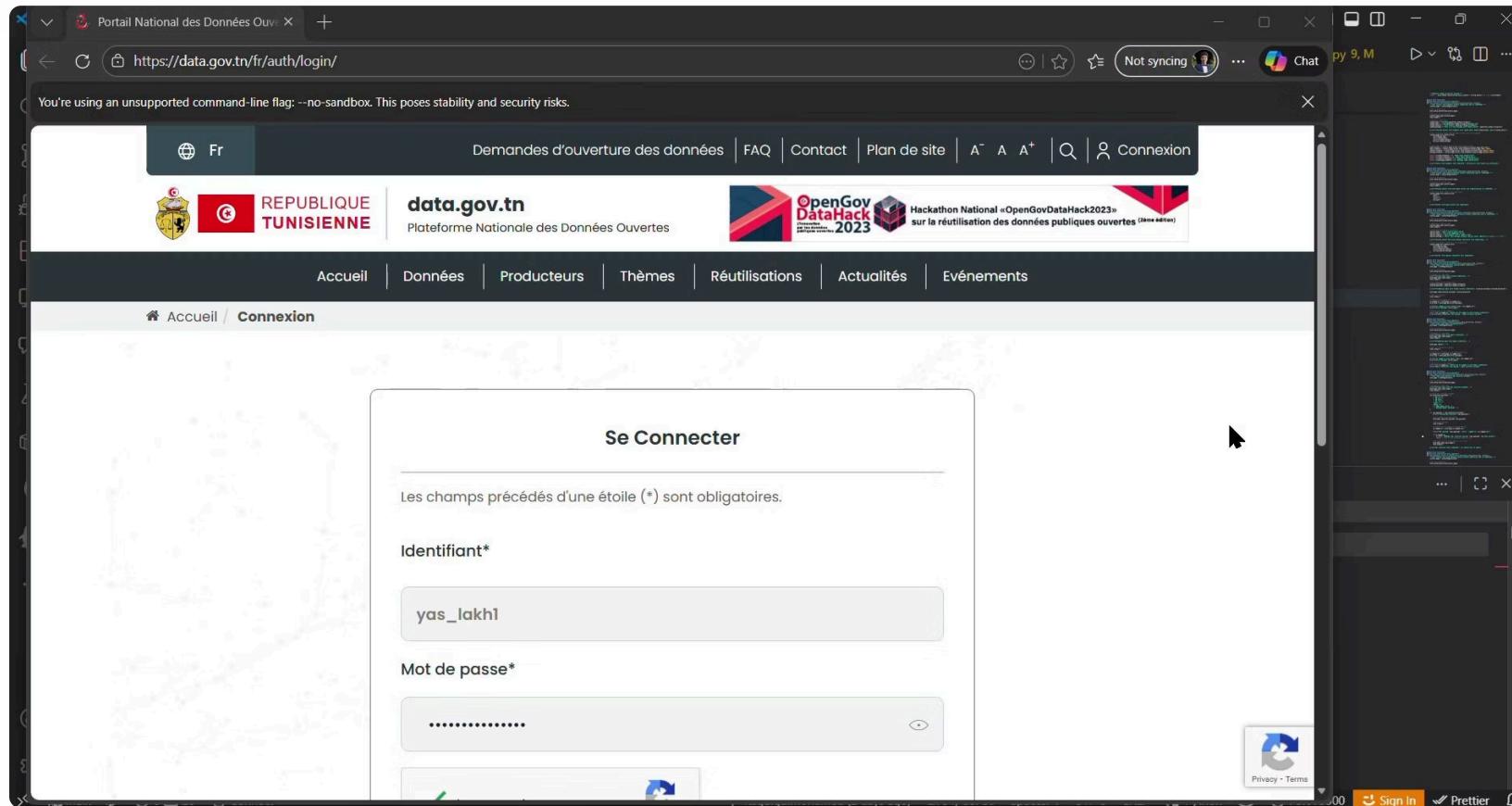
- **Connexion** : Test des scénarios valides et invalides.
- **Soumission de Formulaire** : Vérification de la bonne gestion des données.
- **Validation des Champs** : Assurer que les entrées respectent les formats attendus.
- **Navigation** : S'assurer que tous les liens et chemins fonctionnent correctement.
- **Recherche** : Validation de la précision et de l'efficacité des résultats.
- **Messages d'Erreur** : Vérification de l'affichage des messages appropriés.
- **Permissions/Contrôle d'Accès** : S'assurer que les utilisateurs ont les bons niveaux d'accès.

Modèle de Boîte Noire pour Tests Fonctionnels



Chaque test fonctionnel traite l'application comme une "boîte noire", se concentrant uniquement sur les entrées et les sorties sans connaître sa structure interne.

Login Test (demo)



Exemple de Cas de Test : Test de Valeur Limite

Scénario : Champ de formulaire "nom d'utilisateur" (1 à 20 caractères)

Le test de valeur limite est crucial pour identifier les défauts aux frontières des domaines de données valides et invalides.

1	Entrée Invalidé Chaîne vide "" → Rejeté
2	Entrée Valide (Min) "A" (1 caractère) → Accepté
3	Entrée Valide (Max) "AAAAAAAA...20" (20 caractères) → Accepté
4	Entrée Invalidé Chaîne de 21 caractères → Rejeté

Cette méthode aide à s'assurer que les systèmes gèrent correctement les limites des entrées autorisées, une source courante de bugs.

Search Test demo

The screenshot shows a web browser window displaying the Tunisian Open Data Catalog (catalog.data.gov.tn/fr/dataset/?q=santé+et+éducation&sort=score+desc%2C+metadata_modified+desc). The page is in French (Fr). The header includes links for 'Demandes d'ouverture des données', 'FAQ', 'Contact', 'Plan de site', and 'Connexion'. It also features the Tunisian flag and the text 'Plateforme Nationale des Données Ouvertes'. A banner for the 'OpenGov DataHack 2023' hackathon is visible.

The main navigation bar has links for 'Accueil', 'Données', 'Producteurs', 'Thèmes', 'Réutilisations', 'Actualités', and 'Événements'. The current page is 'Jeux de données'.

A sidebar titled 'Producteurs' lists datasets from various Tunisian government entities:

- Ministère de l'Agri... (890)
- Ministère des affair... (307)
- Ministère de l'Indu... (222)
- Commune de Kélibia (95)
- Commune de Jemmel (90)
- Commune d'Enfidha (87)

The search bar contains the query 'santé et éducation'. The results section displays 2784 datasets found, sorted by 'Pertinence'. One dataset is highlighted:

Station Climatique [Bou Salem Balta] DGAICTA (Ministère de l'Agriculture, ...)

Published on 11 Décembre 2025, 20:31 | 3.5 ★

This dataset contains meteorological observations recorded every 15 minutes at the Bou Salem - Balta station, including temperature, ...

Viewed 393 times, downloaded 265 times, favorited 7 times, and shared 0 times.

Suite de Test 2 : Tests Multi-Navigateurs

Cette suite garantit que l'application offre une expérience utilisateur cohérente et fonctionnelle sur différents environnements de navigation web, une étape cruciale pour atteindre une large audience.

Navigateurs Testés

- Google Chrome
- Mozilla Firefox
- Microsoft Edge

Aspects Validés

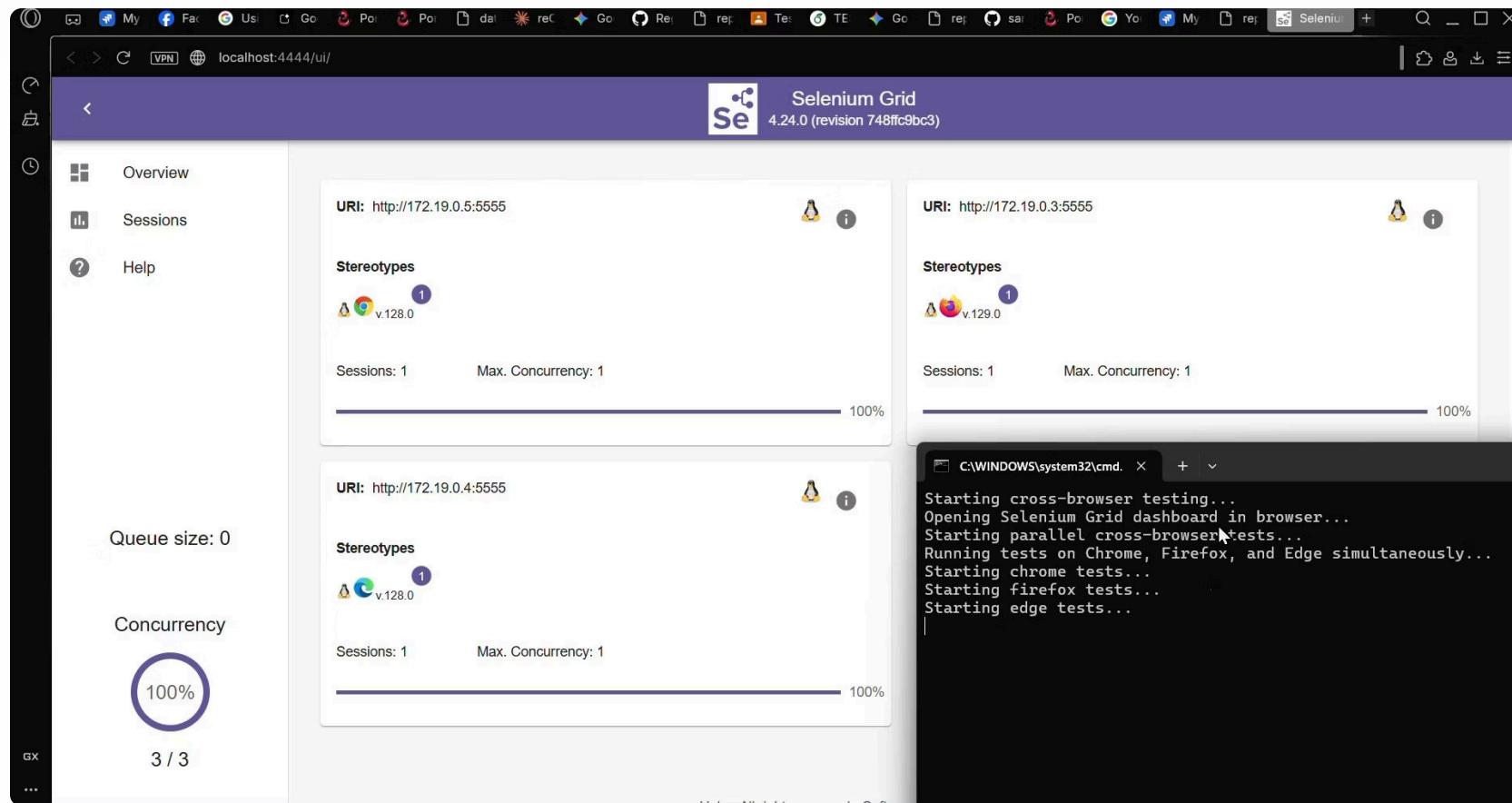
- Cohérence de la mise en page
- Visibilité et fonctionnalité des boutons
- Rendu correct du CSS
- Compatibilité JavaScript

Matrice des Résultats des Tests Multi-Navigateurs

	Chrome	Firefox	Edge
Login	✓	✓	✓
Search	✓	✓	✓
Form Submit	✓	✓	✓
Responsive	✓	!	✓

Le symbole ! indique que des différences visuelles ou de comportement ont été détectées et nécessitent une investigation plus approfondie.

Cross browser Testing (demo)



Suite de Test 3 : Tests de Conception Responsive

Cette suite de tests vérifie la capacité de l'application à s'adapter et à fonctionner correctement sur une variété de tailles d'écran et d'appareils, garantissant une expérience utilisateur optimale quel que soit le point d'accès.

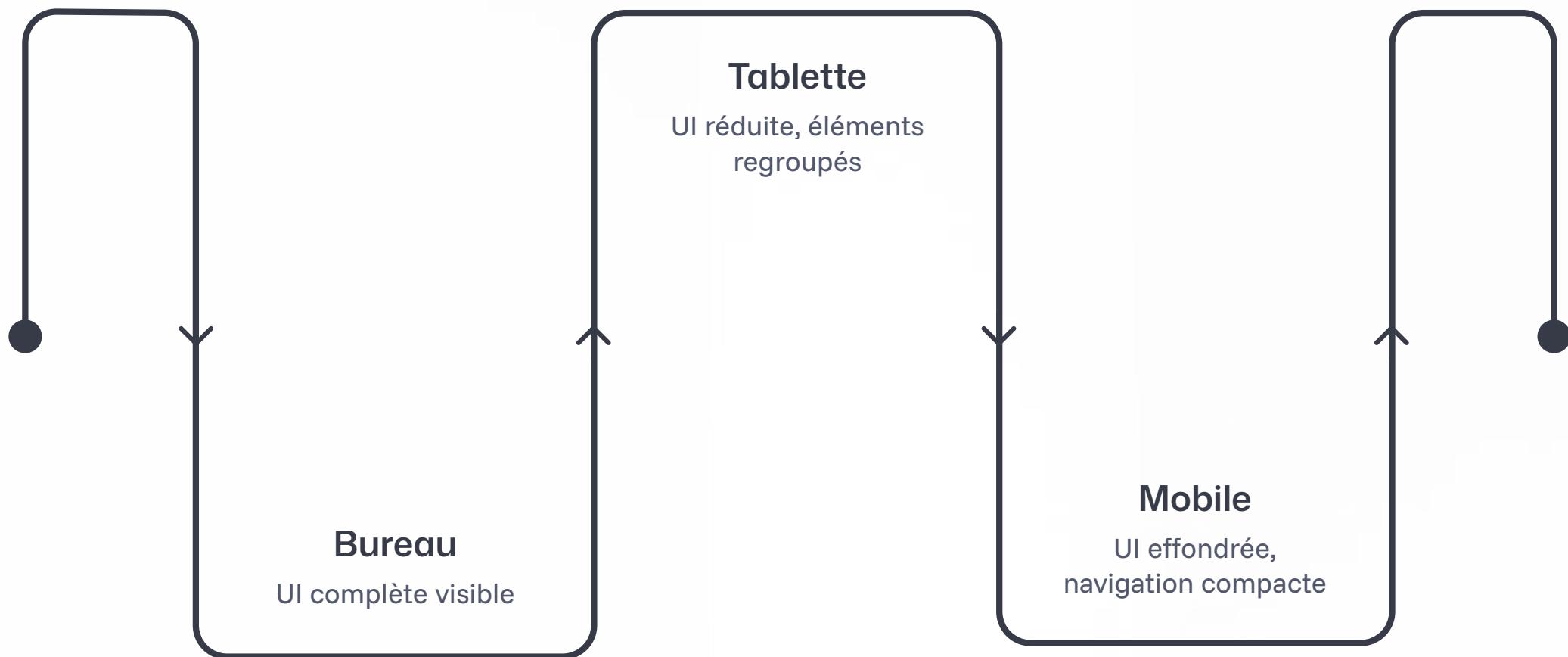
Résolutions Utilisées

- 1920 × 1080 (Ordinateur de bureau)
- 1366 × 768 (Ordinateur portable)
- 768 × 1024 (Tablette)
- 375 × 812 (Mobile)

Points d'Attention

- Effondrement de la mise en page
- Éléments cachés ou tronqués
- Comportement du menu "hamburger"
- Interactions tactiles

Aperçu du Test de Responsive Design



Ce diagramme illustre comment l'interface utilisateur de l'application est testée et validée à travers différents viewports, de l'expérience complète sur ordinateur de bureau aux interfaces optimisées pour tablette et mobile.

Test Responsive (demo)

The screenshot displays a developer's workspace with two main windows. On the left is a browser window titled 'Portail National des Données' showing the URL data.gov.tn/fr/. The page content includes the Tunisian Republic logo, the 'data.gov.tn' logo, and a banner for the 'OpenGov DataHack 2023'. On the right is a code editor window titled 'PROJET-Test-Logiciel' containing Python test code for a contact form. The code uses Selenium to interact with elements like NAME_INPUT, EMAIL_INPUT, SUBJECT_INPUT, and MESSAGE_INPUT. The code editor interface shows multiple tabs and a sidebar labeled 'OUTPUT' with several SSL handshake errors. The bottom status bar of the code editor indicates the author is 'ReguiguiMohamed' (2 days ago), the file is 'test_contact_form_elements_only.py', and the code is in Python.

```
responsive_layout.py report.html test_security_boundary_tests.py 1, M test_search_advanced.py 8, M confest.py 9, M
test_security_boundary_tests.py > test_contact_form_elements_only(auto_setup_monitoring, browser):
    random_name,
    random_email,
    ct=random_subject,
    ge=random_message

    # Check that the form elements exist and can be filled
    assert len(contact_page.driver.find_elements(*contact_page.NAME_INPUT)) > 0, "Name input should exist"
    assert len(contact_page.driver.find_elements(*contact_page.EMAIL_INPUT)) > 0, "Email input should exist"
    assert len(contact_page.driver.find_elements(*contact_page.SUBJECT_INPUT)) > 0, "Subject input should exist"
    assert len(contact_page.driver.find_elements(*contact_page.MESSAGE_INPUT)) > 0, "Message input should exist"

    # Contact form elements test completed - verified all form fields are accessible

```

Suite de Test 4 : Performance et Stress

Cette suite évalue la réactivité, la stabilité et l'évolutivité de l'application sous différentes charges de travail, en identifiant les goulets d'étranglement et les points de défaillance potentiels.

Métriques de Performance Clés

- Temps de chargement de la page d'accueil :** Mesure la vitesse d'affichage initiale.
- Temps de connexion :** Évalue la réactivité du processus d'authentification.
- Temps d'affichage des résultats de recherche :** Vérifie l'efficacité de la fonction de recherche.

Tests de Résistance (Stress Test)

Un scénario de test de résistance simule une activité utilisateur intense en répétant des actions clés (connexion, navigation, recherche) en boucle, poussant l'application au-delà de ses limites opérationnelles normales pour observer son comportement.

Données Recueillies

Temps de Réponse Moyen

Indicateur global de la performance perçue.

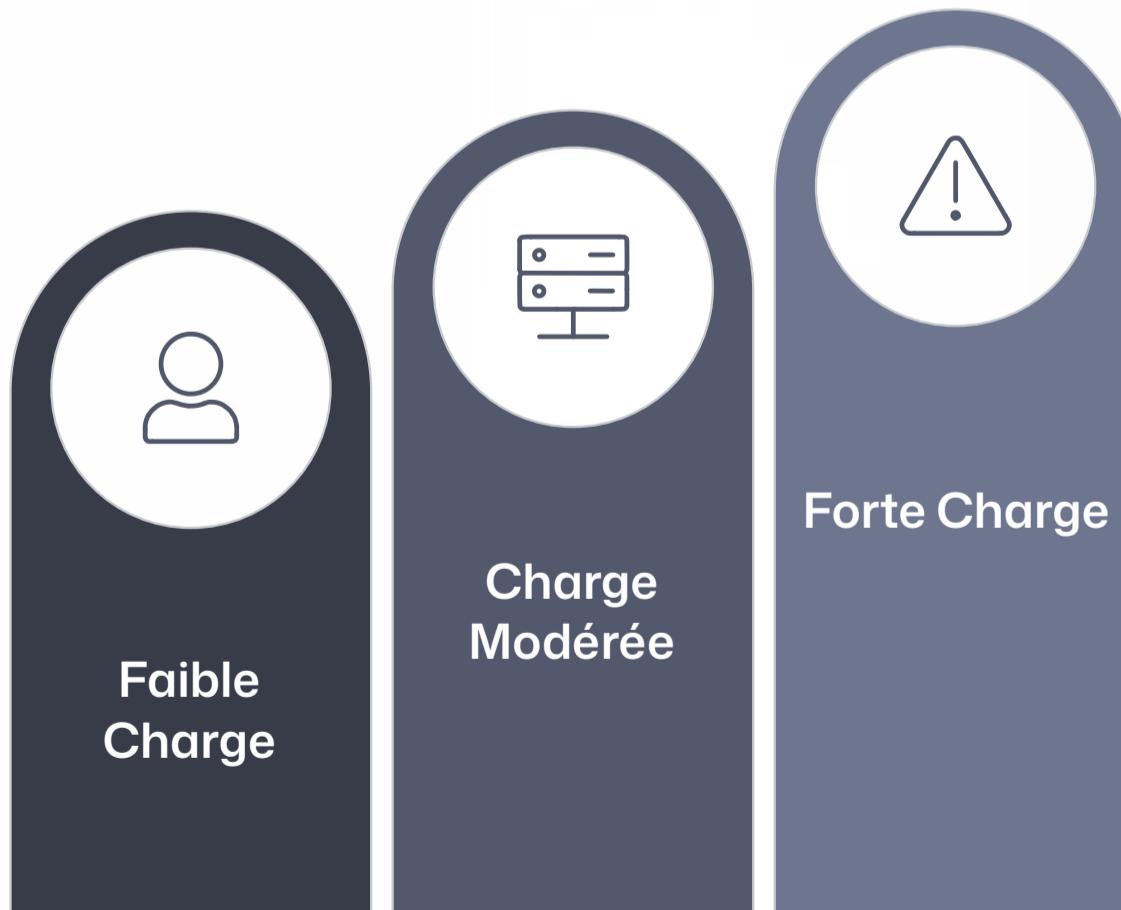
Temps de Réponse Maximal

Identifie les ralentissements extrêmes sous charge.

Nombre de Défaillances

Quantifie l'occurrence d'erreurs ou de pannes.

Modèle de Charge et Impact sur le Temps de Réponse



Ce diagramme illustre comment le temps de réponse de l'application est surveillé et analysé à mesure que la charge utilisateur augmente, révélant les seuils de performance.

Récapitulatif de l'Exécution des Cas de Test

Un aperçu des résultats globaux de l'exécution de nos suites de tests, mettant en évidence la progression et les domaines nécessitant une attention particulière.

25+

20

5

3

Cas de Test

Total exécuté

Réussis

Tests Passés

Échoués

Tests en Échec

Incidents

Problèmes détectés

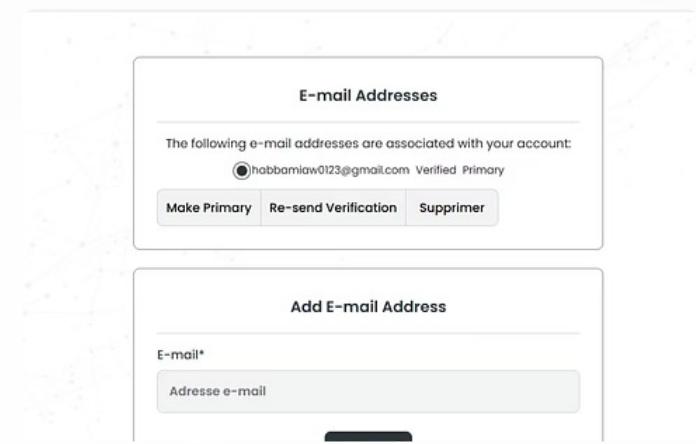
Exemple de Tableau d'Exécution

Cet exemple illustre la structure des rapports détaillés que nous générerons pour chaque exécution de test, facilitant le suivi et la correction des anomalies.

ID du CT	Description	Statut
TC01	Connexion valide	Réussi
TC02	Connexion invalide	Réussi
TC03	Nom d'utilisateur long	Échec
TC04	Mise en page mobile	Échec
TC05	Rendu du formulaire Firefox	Réussi

Les statuts **Réussi** et **Échec** fournissent une visibilité immédiate sur la conformité de l'application aux exigences.

Exemples des Bugs



Implementation LLM

The screenshot shows a software application window titled "TEST CASE GENERATOR". Below the title is a subtitle, "An Instrument for the Methodical Examination of Software". A quote is displayed in the center: "To test, or not to test—that is never the question. The question is how thoroughly one must examine the workings of one's creation." At the bottom left, there is a section titled "INSTRUCTIONS FOR USE" containing a numbered list of five steps:

1. Select your preferred model from the available options
2. Provide your Gemini API authentication key
3. Compose your user story with clarity and precision
4. Enumerate the acceptance criteria with care
5. Generate your test cases and review the results

Rapport de Bugs Automatisé avec Jira

Notre approche de l'automatisation inclut une intégration transparente avec Jira pour un signalement des défauts rapide et efficace, améliorant la traçabilité et réduisant l'effort manuel.

Fonctionnement de l'Automatisation

Toutes les exécutions de tests sont automatiquement liées à Jira via son API REST. Une fonction Python personnalisée détecte les échecs, collecte les artefacts et crée des tickets Jira.



Exemple de Ticket Généré

Project: DATA-GOVTN-QA

Type de tâche: Bug

Résumé: TC03 – Le champ Nom d'utilisateur accepte 21 caractères

Description:

- Étapes automatiquement injectées
- Résultats attendus et réels
- Environnement de test (navigateur, OS, résolution)

Avantages Clés

Zéro Effort Manuel

Plus besoin de saisir manuellement les informations de bug.

Traçabilité Complète

Chaque échec est journalisé avec une structure cohérente.

Triage Accéléré

Les développeurs reçoivent des informations claires et détaillées.

Fiabilité Accrue

Aucune capture d'écran manquante ou étapes de reproduction inconsistentes.

Automated Jira (demo)

The screenshot shows the Jira software interface with a dark theme. On the left, there's a sidebar with various navigation options like 'For you', 'Recent', 'Starred', 'Apps', 'Plans', 'Spaces', and 'Teams'. The 'My Scrum Space' section is currently selected. The main area displays a 'Backlog' tab with four work items:

- APP-7 Test Failure: tests/functional/test_user_profile_bugs...
- APP-8 Test Failure: tests/functional/test_static_pages.py::t...
- APP-9
- APP-10 Test Failure: tests/functional/test_dataset_details...

On the right side, there's a detailed view of the fourth item (APP-10):

- Jira work item**: APP-10
- Test Failure:** tests/functional/test_dataset_details.py::test_dataset_download_link_is_accessible
- Severity:** Majeure
- Priority:** Moyenne
- Description:** Automated Test Failure Report
- Test Details:** tests/functional/test_dataset_details.py::test_dataset_download_link_is_accessible
- Time of Failure:** 2025-12-11T22:35:29.146496

Défis Rencontrés

Au cours de l'automatisation, nous avons identifié plusieurs défis techniques qui nécessitent une attention particulière pour assurer la fiabilité et la robustesse de nos suites de tests.

Incohérences des Pilotes de Navigateur

Les versions des pilotes peuvent introduire des comportements inattendus, compliquant la stabilité des tests cross-navigateurs.

Sélecteurs Instables

Les changements fréquents dans le code de l'application entraînent des modifications des sélecteurs, nécessitant des mises à jour constantes des scripts de test.

Délais de Chargement de Page

Les variations dans les temps de chargement peuvent provoquer des échecs intermittents si les attentes ne sont pas gérées dynamiquement.

Décalages des Points de Rupture

La détection précise des points de rupture responsive peut être complexe, exigeant des validations visuelles et fonctionnelles rigoureuses.

Différences CSS entre Navigateurs

Les rendus CSS subtilement différents entre navigateurs exigent une validation minutieuse de la cohérence visuelle.

Conclusion et Réalisations Clés

Ce projet d'automatisation a démontré l'application réussie des principes pour livrer un ensemble de tests robustes et efficaces, couvrant les aspects critiques de notre application.



Automatisation Complète

Plus de 20 cas de test boîte noire entièrement automatisés, garantissant une couverture de base solide.



Vérification Multi-Plateforme

Tests validés sur une variété de navigateurs et d'appareils, assurant une expérience utilisateur cohérente.



Robustesse et Évolutivité

Évaluations approfondies de performance et de stress incluses pour identifier les limites et garantir la stabilité.



Gestion des Anomalies

Un processus clair et efficace de rapport de bugs mis en place, facilitant le suivi et la résolution rapide des incidents.



Application des Principes De Test

Les techniques et méthodologies ont été appliquées et démontrées avec succès dans ce cadre d'automatisation réel.