Une image contenant texte, capture d’écran, conception, graphisme

Description générée automatiquement

**Encadrée par : NADA MAKKI**

**TEST QA**

**4 -ème Année ERP-BI**

**Hadil dridi**

**Sommaire**

1. **Contexte général**

* 1.1 Introduction
* 1.2 Organisme d’accueil
  + 1.2.1 Présentation de l’ESPRIT
* 1.3 Problématique
  + 1.3.1 Processus de Test Existant
  + 1.3.2 Critique de l’existant
* 1.4 Solution Proposée
* 1.5 Choix Méthodologique
* 1.6 Conclusion

2. **Fondamentaux des Tests**

* Étude de Cas avec Katalon et OpenCart
  + Tests d’ergonomie
  + Tests de compatibilité
  + Tests de performance
* Définition du Test
* Tester pendant le Cycle de Vie d’un Logiciel
* Adaptation des Tests à OpenCart
* Objectifs et Bénéfices des Tests
* Bugs courants et Résolutions

3. **Machine Learning**

* 9.1 Objectif du Test
* 9.2 Configuration de l’Environnement
* 9.3 Étapes du Test avec Machine Learning
  + a. Collecte des Données de Test
  + b. Entraînement du Modèle Machine Learning
  + c. Utilisation des Prédictions ML dans Katalon Studio
  + d. Adaptation Dynamique des Tests
* 9.4 Bénéfices de l’Approche Machine Learning
* 9.5 Conclusion

4. **Diagrammes de cas d’utilisation**

* Diagramme de cas d’utilisation globale
* Détail du cas d’utilisation « Gérer l’exécution des cas de tests »
* Détail du cas d’utilisation « Gérer les bugs »

5.**Conclusion générale**

# Introduction

Ce chapitre a pour objectif de placer notre projet dans son contexte global. Nous commencerons par une présentation générale de l’organisme d’accueil. Ensuite, nous identifierons et structurerons le problème, avant de présenter la solution proposée. Pour terminer, nous détaillerons la méthodologie de travail adoptée pour notre projet.

# Organisme d’accueil

Dans ce chapitre, nous allons présenter l’organisme d’accueil, l’École Supérieure Privée d’Ingénierie et de Technologies (ESPRIT), en mettant un accent particulier sur sa Direction des Systèmes d’Information (DSI). Cette présentation a pour but de fournir un aperçu complet de l’institution, de son organisation interne, de ses missions, et de l’importance de la DSI dans le fonctionnement global de l’école.

* + 1. **Présentation de l’ESPRIT**

❍ **Historique** : Fondée en [année de fondation], l’ESPRIT s’est rapidement imposé comme une institution de premier plan dans le domaine de l’ingénierie en Tunisie. Elle offre une formation de qualité en ingénierie et technologies, préparant ses étudiants à devenir des professionnels compétents et innovants.

❍ **Mission et Vision :** La mission de l’ESPRIT est de former des ingénieurs polyvalents, capables de répondre aux besoins du marché de l’emploi. Sa vision est de devenir un leader régional en matière de formation en ingénierie et technologies. La Figure 1.1 représente le logo d’esprit Une image contenant texte, Police, logo, capture d’écran

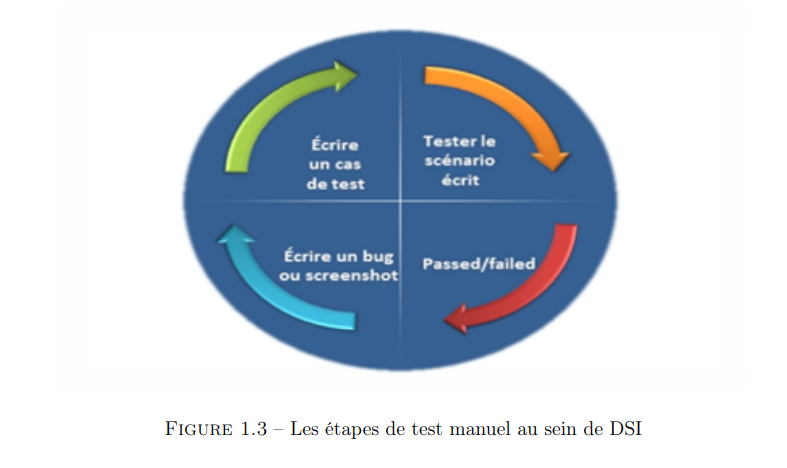
Description générée automatiquement

**1.3** **Problématique :**  
Dans un contexte où la qualité des applications web est devenue un facteur déterminant pour garantir la satisfaction des utilisateurs, il est indispensable d’adopter des processus de tests rigoureux pour identifier et corriger les anomalies avant la mise en production. Tester manuellement un site e-commerce comme OpenCart peut rapidement devenir inefficace et chronophage, surtout lorsque des mises à jour fréquentes ou des variations de configurations sont nécessaires.

C’est dans ce cadre que l’utilisation d’outils de test automatisé, tels que **Katalon Studio**, se révèle indispensable. Katalon Studio offre la possibilité de simplifier et d’accélérer le processus de test en automatisant des scénarios répétitifs tout en assurant une couverture exhaustive des fonctionnalités critiques du site. Le défi principal réside dans l'identification et l'implémentation des scénarios les plus pertinents pour détecter les anomalies liées à la compatibilité, aux performances, et à l’ergonomie, tout en intégrant ces tests dans un processus agile de développement et de livraison continue.

**1.3.1 Processus de Test Existant**

Au sein de la Direction des Systèmes d’Information (DSI), le rôle du testeur consiste à vérifier manuellement l’application, sans recourir à des scripts de test automatisés. Le testeur agit comme un utilisateur final pour examiner le logiciel et identifier les comportements inattendus ou les bogues. Il utilise des scénarios de test pour s’assurer que les résultats obtenus sont conformes aux spécifications demandées. La Figure 1.3 illustre les différentes étapes du test manuel réalisées par l’équipe qualité



**1.4 Solution Proposée**

La solution adoptée repose sur l'utilisation de Katalon Studio pour mettre en place une stratégie de test structurée, afin de coordonner efficacement les activités de test et garantir un niveau de qualité optimal pour le site OpenCart. Cette approche intègre des tests manuels et automatisés tout en s'appuyant sur les fonctionnalités avancées de Katalon Studio pour améliorer l'efficacité et la précision des tests.

Avec l’utilisation de Katalon Studio, la stratégie proposée permet de :

* Effectuer des tests manuels via Katalon pour valider les fonctionnalités critiques du site.
* Analyser les exigences du produit afin de traduire celles-ci en scénarios de test pertinents.
* Concevoir et organiser des cas de test dans Katalon TestOps, en priorisant les scénarios essentiels.
* Préparer et gérer les données de test dans Katalon pour garantir des exécutions fiables et réalistes.
* Automatiser les scénarios de test répétitifs ou complexes à l’aide de Katalon Recorder ou du Studio IDE.
* Générer des rapports détaillés à partir des résultats des tests automatisés pour une traçabilité optimale.

**1.5 Choix Méthodologique :**

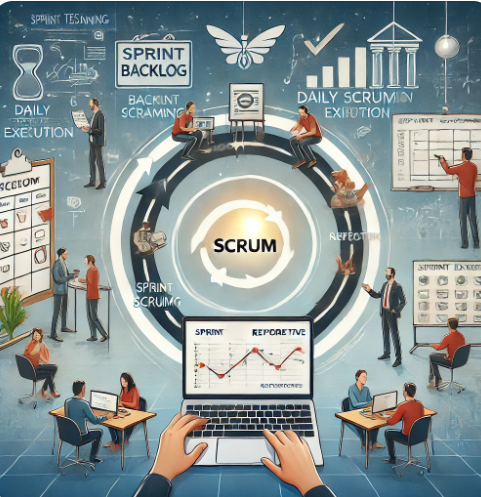
**Application de Scrum dans le Contexte des Tests**

1. Rôles dans Scrum appliqués aux tests
   * Product Owner : Définit les priorités des tests en fonction des besoins métiers et des critères d’acceptation.
   * Scrum Master : Facilite l’exécution des tests en supprimant les obstacles, en veillant au respect des délais et en optimisant les processus.
   * Équipe de test : Conçoit, exécute, et analyse les tests (manuels et automatisés) en utilisant Katalon Studio.
2. Sprints et Backlog
   * Product Backlog : Comprend l’ensemble des exigences, des cas de test, et des scénarios à valider.
   * Sprint Backlog : Sélection des tests prioritaires pour le sprint en cours, en fonction des critères d'acceptation et des risques.
   * Durée des sprints : Généralement de 1 à 2 semaines, permettant une livraison fréquente de résultats testés et validés.
3. Planification des Tests dans Scrum
   * Réunion de planification du sprint :
     + Identification des fonctionnalités à tester.
     + Définition des scénarios de test et des tâches liées (tests manuels ou automatisés).
   * Définition of Done (DoD) : Les tests sont considérés comme terminés lorsque :
     + Tous les scénarios définis sont exécutés.
     + Les anomalies sont signalées et suivies.
     + Les rapports de test sont générés et partagés.
4. Exécution des Tests dans Scrum
   * Daily Scrum : État d’avancement des tests (progression, blocages, nouveaux cas identifiés).
   * Utilisation de Katalon Studio :
     + Automatisation des scénarios de régression ou répétitifs.
     + Gestion des cas de test et des suites de tests pour une exécution structurée.
     + Génération de rapports de tests automatisés pour les réunions quotidiennes.
5. Rétrospective et Amélioration Continue
   * Analyse des résultats des tests exécutés pendant le sprint.
   * Identification des points à améliorer (ex. : outils, délais, couverture des tests).
   * Planification des ajustements pour le prochain sprint.

Avantages de l’approche Scrum appliquée aux tests avec Katalon Studio

* Flexibilité : Adaptation rapide aux changements dans les exigences du site OpenCart.
* Efficacité : Tests continus grâce à l'intégration des tests automatisés dans chaque sprint.
* Collaboration : Meilleure coordination entre les développeurs, les testeurs, et les parties prenantes.
* Amélioration constante : Apprentissage d’un sprint à l’autre pour optimiser la qualité et les performances des tests.

L’approche Scrum, combinée à Katalon Studio, assure une validation rapide et itérative des fonctionnalités tout en maintenant une qualité optimale du produit.



**1.6 Conclusion :**

Ce chapitre introductif a permis de poser les bases de la problématique rencontrée dans le cadre des tests du site OpenCart, où la qualité et la fiabilité du produit sont essentielles pour garantir une expérience utilisateur optimale. L’absence de tests rigoureux peut entraîner des anomalies coûteuses et des insatisfactions, d’où la nécessité d’intégrer une stratégie de test structurée et adaptée.

La solution proposée repose sur l’utilisation de **Katalon Studio**, combinant des tests manuels et automatisés, et s’appuyant sur une méthodologie agile comme **Scrum**. Ce choix méthodologique assure une couverture complète des scénarios critiques tout en optimisant l’efficacité et la collaboration au sein de l’équipe.

**2.1 Fondamentaux des Tests :**

**Étude de Cas avec Katalon et OpenCart**

Dans le cadre de mon projet, j’ai utilisé **Katalon Studio** pour tester le site web **OpenCart**, une solution e-commerce open-source. Les tests réalisés incluent plusieurs aspects essentiels pour garantir la qualité et la performance du site.

**Tests d’ergonomie (usability)**

L'**ergonomie** ou la convivialité d'un site est cruciale pour garantir une expérience utilisateur fluide et agréable. Dans ce contexte, j'ai effectué des tests visant à évaluer l’intuitivité et la facilité de navigation sur le site OpenCart. Les utilisateurs doivent pouvoir naviguer sur le site sans confusion, trouver rapidement les produits, ajouter des articles au panier, et procéder à l'achat sans difficultés majeures. J’ai vérifié des éléments comme l’accessibilité des menus, la réactivité des boutons, la clarté des informations, et la fluidité de l’interaction globale.

**Tests de compatibilité (compatibility)**

La **compatibilité** est un autre aspect fondamental des tests, particulièrement dans le cadre des applications web. OpenCart étant une plateforme accessible depuis divers appareils et navigateurs, il est essentiel de vérifier son comportement sur différentes configurations. J’ai effectué des tests de compatibilité en simulant l’accès au site depuis divers navigateurs (Chrome, Firefox, Safari, etc.) et appareils (PC, tablettes, smartphones). L’objectif était de m’assurer que le site reste fonctionnel, que l'affichage soit correct et que les performances ne varient pas de manière significative d’un environnement à l’autre.

**Tests de performance**

Les **tests de performance** permettent de mesurer les performances du site dans des conditions réelles ou simulées. En testant OpenCart, j’ai évalué les **temps de réponse** du site sous différentes charges (nombre d’utilisateurs simultanés, volume de données traitées, etc.). Ces tests ont pour but de détecter d’éventuels goulets d'étranglement qui pourraient ralentir le site, nuire à l’expérience utilisateur ou empêcher l'accès pendant les périodes de forte affluence. Par exemple, des tests ont été réalisés pour voir comment OpenCart réagit lorsqu'un grand nombre d'utilisateurs consulte des pages simultanément, ou encore comment le temps de chargement des pages varie en fonction de la taille du catalogue de produits.

**2.2.1 Définition du Test**

Le **test logiciel** est un processus crucial dans le développement de tout système informatique. Il permet d’évaluer un logiciel ou ses composants afin de vérifier qu’il répond aux exigences définies. Les tests peuvent être réalisés manuellement, mais l’automatisation des tests est de plus en plus utilisée, notamment pour les tests répétitifs et à grande échelle.

Il existe deux types de tests principaux :

* **Validation** : Ce type de test cherche à vérifier si le logiciel répond aux besoins fonctionnels des utilisateurs. En d’autres termes, il permet de répondre à la question : "Est-ce que le logiciel fait ce que l'utilisateur attend ?"
* **Vérification** : Ce test a pour objectif de vérifier si le logiciel a été conçu correctement, selon les spécifications initiales. Il répond à la question : "Est-ce que le logiciel fonctionne comme prévu ?"

Les tests sont essentiels pour garantir la **qualité** d’un logiciel, prévenir les anomalies et assurer qu’il réponde aux attentes des utilisateurs tout en respectant les exigences techniques ou contractuelles.

**2.2.2 Tester pendant le Cycle de Vie d’un Logiciel**

Les tests doivent être effectués tout au long du cycle de vie du logiciel pour garantir que le produit final soit fiable et conforme aux attentes. Voici les principales étapes où les tests sont intégrés :

* **Expression des besoins** : Identification et clarification des attentes des utilisateurs et des parties prenantes.
* **Spécifications** : Traduction des besoins en exigences détaillées et compréhensibles pour les équipes de développement et de tests.
* **Développement** : Création du logiciel en suivant les spécifications définies.
* **Validation** : Les tests sont effectués pour garantir que le logiciel correspond aux attentes des utilisateurs. Ces tests peuvent être :
  + **Tests fonctionnels** : Pour vérifier que le logiciel réalise correctement les tâches qu’il est censé accomplir.
  + **Tests non fonctionnels** : Pour analyser des aspects comme la performance, la sécurité, ou encore l’ergonomie.
* **Déploiement** : Une fois les tests validés, le logiciel est mis à disposition des utilisateurs finaux.

**2.2.3 Adaptation des Tests à OpenCart**

Lors des tests réalisés sur le site OpenCart, j’ai combiné des **tests manuels** et des **tests automatisés** pour garantir la qualité du site.

* **Tests manuels** : Utilisés pour les évaluations d'ergonomie et pour détecter d’éventuels problèmes visuels ou fonctionnels imprévus. Ces tests permettent de simuler l’expérience d’un utilisateur réel et de repérer les problèmes qui ne peuvent pas être facilement détectés par des outils automatisés.
* **Tests automatisés avec Katalon** : Ces tests ont été utilisés pour des vérifications répétitives, telles que la validation des flux utilisateurs sur le site (connexion, ajout au panier, validation de la commande) ainsi que des tests de performance, pour simuler différentes charges et analyser la réactivité du site. L’automatisation est particulièrement efficace pour tester rapidement de grandes quantités de données ou pour des scénarios qui doivent être validés à chaque nouvelle version du site.

**2.2.4 Objectifs et Bénéfices des Tests**

Les tests sur le site OpenCart avaient plusieurs objectifs principaux :

* Garantir une **expérience utilisateur fluide**, en s’assurant que la navigation et l’interaction avec le site sont intuitives.
* Vérifier la **compatibilité multi-navigateurs et appareils**, pour garantir que les utilisateurs puissent accéder au site de manière optimale depuis n’importe quel environnement.
* Assurer une **performance optimale sous diverses charges**, notamment pour prévenir les risques de lenteur ou d’indisponibilité du site lors de pics de trafic.

Ces tests ont permis de détecter des problèmes potentiels avant leur déploiement, d'optimiser le site pour une meilleure expérience utilisateur et d’offrir une base solide pour son évolution future.

**2.3 Les bugs :**

Lors de l'utilisation de **OpenCart**, plusieurs types de bugs peuvent être rencontrés, comme dans toute autre application web. Ces bugs peuvent être liés à différents aspects du site, allant de l'interface utilisateur à la gestion des produits ou des paiements. Voici quelques bugs courants que les utilisateurs peuvent rencontrer sur OpenCart, ainsi que des solutions possibles.

**1. Problèmes d’affichage et d’ergonomie (UI/UX)**

* **Erreur de rendu des pages** : Parfois, le thème OpenCart ne s'affiche pas correctement en raison de conflits entre les fichiers CSS ou JavaScript. Cela peut entraîner des éléments mal alignés ou des pages qui ne se chargent pas correctement.
  + **Solution** : Vérifier les fichiers de thème, nettoyer le cache du navigateur, et s'assurer que les fichiers CSS/JS sont correctement minifiés et non corrompus.
* **Images de produits ne s'affichent pas** : Les images de produits peuvent ne pas apparaître sur certaines pages, ce qui donne une mauvaise expérience utilisateur.
  + **Solution** : Vérifier les permissions des dossiers où les images sont stockées, s’assurer que les images sont bien téléchargées et vérifier les liens dans le backend.

**2. Problèmes de panier et de commande**

* **Articles ajoutés au panier mais qui disparaissent** : Un bug fréquent dans OpenCart est que les produits ajoutés au panier ne sont pas conservés lors du passage d'une page à l'autre.
  + **Solution** : Ce problème peut être lié à la configuration des sessions sur le serveur ou à des cookies. Assurez-vous que la gestion des sessions est correctement configurée dans les paramètres d'OpenCart et que le cache du navigateur est vidé.
* **Problème avec la validation de la commande** : Certains utilisateurs signalent que la validation de commande ne fonctionne pas correctement, souvent à cause de modules de paiement non compatibles ou mal configurés.
  + **Solution** : Vérifier les paramètres des extensions et s'assurer que le module de paiement est bien activé et compatible avec la version actuelle d'OpenCart.

**3. Problèmes de performance**

* **Temps de chargement lents** : Lors de l’accès à certaines pages du site, le temps de chargement peut être excessif. Cela peut être dû à un trop grand nombre d'extensions, une mauvaise configuration du serveur, ou des images non optimisées.
  + **Solution** : Optimiser les images (compression et réduction de la taille), utiliser la mise en cache des pages, et revoir la configuration du serveur web pour s’assurer qu’il est capable de supporter la charge.
* **Problème de cache** : Parfois, OpenCart peut rencontrer des problèmes liés à la mise en cache, entraînant des versions obsolètes du site qui s'affichent aux utilisateurs.
  + **Solution** : Vider régulièrement le cache du système via l'admin d’OpenCart et s’assurer que les configurations de cache dans le backend sont correctes.

**4. Problèmes de compatibilité avec les extensions**

* **Conflits entre extensions** : Lorsqu'il y a trop d'extensions ou de modules ajoutés, certains peuvent entrer en conflit avec d’autres, entraînant des erreurs sur le site. Cela peut inclure des problèmes avec le panier, le paiement, ou la gestion des produits.
  + **Solution** : Désactiver les extensions une par une pour identifier celle qui provoque le problème. Tester les extensions sur un environnement de test avant de les déployer en production.
* **Modules de paiement qui échouent** : Un autre problème courant réside dans l'intégration de certains modules de paiement comme PayPal ou Stripe, qui peuvent échouer pendant le processus de commande.
  + **Solution** : Vérifier que les clés API et les configurations sont correctement entrées et que le module de paiement est à jour.

**5. Problèmes de mise à jour et d'installation**

* **Échec de mise à jour** : Lors de la mise à jour vers une version plus récente d’OpenCart, certains utilisateurs rencontrent des erreurs ou des incompatibilités, souvent liées à des modules tiers.
  + **Solution** : Avant de mettre à jour, sauvegarder toutes les données et extensions. Assurez-vous également que toutes les extensions installées sont compatibles avec la version d'OpenCart vers laquelle vous mettez à jour.
* **Problèmes lors de l’installation de modules** : Certains modules peuvent ne pas s'installer correctement, ce qui entraîne des erreurs dans l'interface d'administration ou sur la boutique en ligne.
  + **Solution** : Vérifier les fichiers d’installation du module et s’assurer que les permissions des fichiers et des dossiers sont correctement définies pour permettre l’installation.

**6. Problèmes liés à la gestion des utilisateurs et des permissions**

* **Utilisateurs ne pouvant pas accéder à certaines parties de l’administration** : Parfois, des utilisateurs avec des permissions spécifiques n'ont pas accès à certaines fonctionnalités de l'admin, même si cela devrait être permis.
  + **Solution** : Vérifier les groupes d’utilisateurs et les permissions dans le panneau d’administration d’OpenCart pour s’assurer qu’elles sont correctement configurées.
* **Problème de connexion et de mot de passe** : Des utilisateurs peuvent rencontrer des problèmes de connexion à leur compte, souvent dus à des erreurs dans la base de données ou des configurations incorrectes.
  + **Solution** : Vérifier les logs d’erreurs d'OpenCart et vérifier les paramètres de la base de données (par exemple, les préfixes des tables ou les paramètres de session).

**2.3.1 résoudre les bugs**

* **Vérification des logs** : OpenCart génère des logs d'erreurs que tu peux consulter dans l'administration ou dans le dossier système pour trouver des détails spécifiques sur les erreurs rencontrées.
* **Test en environnement de développement** : Avant de déployer des mises à jour ou des modules sur ton site en production, teste-les dans un environnement local ou sur un serveur de développement pour éviter les interruptions de service.
* **Mettre à jour régulièrement** : Assure-toi que tu utilises toujours la version la plus récente d’OpenCart et que les extensions que tu utilises sont compatibles avec la version actuelle du logiciel.
* **Optimisation des ressources serveur** : Si ton site est lent ou rencontre des problèmes de performance, pense à améliorer les ressources serveur (mémoire, CPU, disque SSD) et optimise les configurations de PHP et du serveur web (Apache ou Nginx).

En suivant ces bonnes pratiques, tu pourras résoudre la plupart des bugs courants et améliorer la stabilité et les performances de ton site OpenCart.

**3.1 Diagramme de cas d’utilisation globale**

Le diagramme de cas d’utilisation représente les fonctionnalités globales d’un système et délimite son champ d’action. Il met également en évidence les interactions entre le système et les utilisateurs qui interagissent avec lui.

La Figure 3.7 illustre le diagramme de cas d’utilisation global.

Une image contenant texte, diagramme, ligne, capture d’écran

Description générée automatiquement

**3.2 Détail du cas d’utilisation « Gérer l’exécution des cas de tests » :**

**Une image contenant texte, diagramme, capture d’écran, ligne

Description générée automatiquement**

**3.3 Détail du cas d’utilisation « Gérer les bugs »**

**Une image contenant texte, diagramme, ligne, capture d’écran

Description générée automatiquement**

1. **Performance Testing :**

**Une image contenant texte, capture d’écran, ligne, nombre

Description générée automatiquement**

**5.usability testing :**

**Une image contenant texte, ligne, Tracé, nombre

Description générée automatiquement**

**6.comptability testing :**

**Une image contenant texte, ligne, nombre, Police

Description générée automatiquement**

**7.Tester sur katalon Studio :**

**\*** **Automatisation d'un test sur un site OpenCart :**

**Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement**

**\*Tester navigate to URL :**

**Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement**

**\* Maximize window :**

**Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement**

**\*Click :**

**Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement**

**8.1 Machine Learning** :

Le Machine Learning est une branche de l’intelligence artificielle qui permet aux machines d’apprendre à partir des données et de s’améliorer avec l’expérience, sans avoir besoin d’être explicitement programmées pour chaque tâche. En apprentissage automatique, les algorithmes sont conçus pour repérer des modèles et des corrélations dans de grands ensembles de données, afin de prendre des décisions ou de faire des prédictions plus précises. Plus les algorithmes sont entraînés, meilleure est la performance de l’application de machine learning. Par ailleurs, la taille et la qualité des données accessibles aux algorithmes ont un impact direct sur leur précision.

**9. Tester le Site OpenCart avec Katalon Studio et Machine Learning :**

Dans ce projet, l'intégration de **Katalon Studio** avec des modèles de **Machine Learning** (ML) a permis d'automatiser les tests tout en prédisant certaines anomalies et comportements du site OpenCart sous différentes charges et scénarios utilisateurs. L'utilisation de ML a facilité l'optimisation de la stratégie de tests et la détection précoce de problèmes de performance.

**9.1. Objectif du Test**

Le but est d’automatiser les tests fonctionnels et de performance sur le site OpenCart tout en utilisant un modèle ML pour prédire et ajuster les scénarios de test en fonction des données collectées durant les tests précédents.

**9.2. Configuration de l'Environnement**

* **Katalon Studio** : utilisé pour l'automatisation des tests fonctionnels et de performance sur le site OpenCart.
* **Python + Scikit-learn** : utilisé pour entraîner un modèle de Machine Learning qui prédira la performance du site en fonction des variables collectées durant les tests (par exemple, nombre d'utilisateurs, temps de réponse des pages, etc.).
* **Katalon API** : pour l'intégration de Python avec Katalon et l'exécution de modèles ML à partir de Katalon Studio.

**9**.**3. Étapes du Test avec Machine Learning**

**a. Collecte des Données de Test**

Tout d'abord, nous exécutons un test de performance à l'aide de Katalon Studio pour collecter des données sur les performances du site OpenCart sous différentes charges (nombre d’utilisateurs simultanés, temps de réponse des pages, etc.). Ces données serviront d'entrées pour entraîner un modèle ML.

**b. Entraînement du Modèle Machine Learning**

Les données collectées pendant les tests sont ensuite utilisées pour entraîner un modèle ML en Python. Par exemple, un modèle de régression ou un arbre de décision peut être utilisé pour prédire le temps de réponse ou identifier les points faibles de performance sous diverses charges.

**c. Utilisation des Prédictions ML dans Katalon Studio**

Une fois le modèle ML entraîné, nous pouvons utiliser les prédictions pour ajuster la stratégie de tests dans Katalon. Par exemple, si le modèle prédit que le site sera lent sous une charge de 1000 utilisateurs simultanés, Katalon pourrait exécuter des tests supplémentaires dans cette configuration pour valider la performance.

L'intégration avec Katalon se fait via l'API ou en appelant directement le modèle Python depuis un script Groovy dans Katalon.

**d. Adaptation Dynamique des Tests**

En fonction des résultats du modèle, Katalon peut adapter les tests en temps réel. Par exemple, si le modèle prévoit une faible performance sous une certaine charge, Katalon peut :

* Augmenter la fréquence des tests de charge dans ces zones.
* Tester des fonctionnalités spécifiques plus intensivement, comme le processus de paiement ou les pages avec de nombreux produits

**9**.**4. Bénéfices de l'Approche Machine Learning**

* **Optimisation des tests** : En intégrant des modèles de ML, nous pouvons prédire les zones du site les plus susceptibles de poser problème, ce qui permet de concentrer les efforts de test sur ces zones critiques.
* **Tests intelligents** : Le modèle peut prédire les pires scénarios de performance, ce qui aide à ajuster les tests et à prévenir les défaillances avant la mise en production.
* **Amélioration continue** : À chaque nouveau test, les résultats peuvent être utilisés pour affiner le modèle et améliorer les prédictions et les tests futurs.

**9**.**5. Conclusion**

L'intégration de **Machine Learning** avec **Katalon Studio** a permis de renforcer l'automatisation des tests sur le site OpenCart en fournissant des prédictions intelligentes sur la performance du site sous diverses conditions. Cette approche a non seulement amélioré l'efficacité des tests, mais a également permis d'identifier des points faibles avant qu'ils n'atteignent les utilisateurs finaux. L’utilisation de la data science et du ML dans les tests logiciels représente un moyen puissant pour rendre les processus de validation plus dynamiques et adaptatifs.

**10.CONCLUSION :**

L'utilisation de Katalon Studio et Selenium pour automatiser les tests a garanti une couverture complète et efficace des différents scénarios, permettant d'identifier rapidement les points d'amélioration. Ces tests ont été essentiels pour assurer la qualité du site et optimiser son expérience utilisateur avant son déploiement**.**