**Une image contenant texte

Description générée automatiquement**

Test de performances Mascotte

4IW3

Le 28/06/2022

## Projet à tester :

## « https://www.mascotte-assurances.fr/ »

**Groupe de travail :**

* Taguemount Kamelia
* Oussadi Sonia

# Préambule :

## Description globale du projet :

Mascotte assurance est un site web qui décrit les différents parcours de souscription en ligne, aux différents contrats d’assurance qu’offre l’entreprise.

Les différents contrats que propose l’entreprise sont :

* Assurance automobile de collection ou ancienne
* Assurance moto de collection
* Assurance cyclomoteur de collection
* Assurance scooter de collection
* Assurance flotte auto de collection
* Assurance flotte moto de collection
* Assurance flotte de véhicules de collection

## Objectif de l’application :

L’application a pour objectif de créer des devis en ligne et de permettre à l’utilisateur de souscrire à un contrat d’assurance

Type d'utilisateurs :

* Les utilisateurs de l’application
* Les administrateurs
* Les partenaires

# 2.Architecture de l'application :

## Technologies/Langages utilisés :

Le site mascotte-assurance est développé avec le CMS Wordpress (une base de données MySQL et le langage PHP)

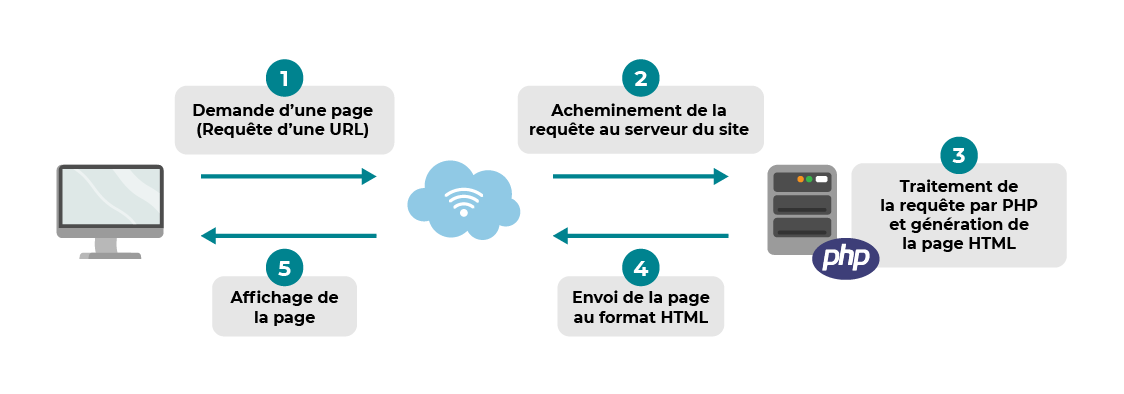


## 

## Les principales fonctionnalités du site sont :

* Un système d’authentification pour les utilisateurs partenaires
* Plusieurs parcours de création de devis et de souscription en ligne

Schéma décrivant les interactions entre les différentes parties de l’application :



# 3. Exigences du test :

## Objectif :

Test de performance pour vérifier la cohérence du comportement basique de l’application, le temps de réponse des requêtes sur la réalisation des différents parcours de souscription.

## Exigences :

|  |  |  |  |
| --- | --- | --- | --- |
| **Business Transactions** | User Load | Response time | Transaction /heure |
| Accéder à la page création de devis auto | 500 | 1s | 1000 |
| Accéder aux pages des différentes étapes du devis | 500 | 1s | 1000 |

# 4. Environnement de test :

Environnement de test Wordpress en local :

* Création d’une copie du site en production vers le site local
* Utilisation des images docker (Nginx, WordPress, MySQL, phpMyAdmin)
* Utiliser la configuration serveur en production pour configurer les différents services de docker.

Utilisation de l’outil JMeter pour l’exécution des différents tests.

# 5. Planification des tests :

L’utilisation de l’application reste moyenne, car c’est une application conçue uniquement pour une seule catégorie de collectionneurs (véhicule à moteur).

Le type de test nécessaires pour l’application est le **Load Testing**.

Les métriques que l’on souhaite surveiller sont :

* Response Time
* Thread counts ??
* Throughput ??

Les métriques qui définissent le succès ou l’échec du test sont :

* Response Time
* ??

# 6. Etapes des tests

|  |  |
| --- | --- |
| **Etape #** | **Accéder à la page création de devis auto** |
| **1** | Page d’accueil |
| **2** | Click sur le lien « CREATION DE DEVIS EN 3 MIN » |
| **3** | Click sur le lien « assurance auto-collection » |

|  |  |
| --- | --- |
| **Etape #** | **Accéder aux pages des différentes étapes du devis** |
| **1** | Page d’accueil |
| **2** | Click sur le lien « CREATION DE DEVIS EN 3 MIN » |
| **3** | Click sur le lien « assurance auto-collection » |
| **4** | Répondre aux questions et passer les étapes |
| **5** | Souscription |