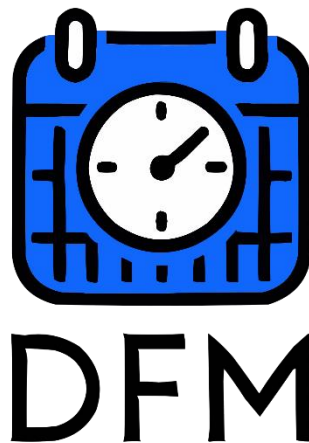


AUDOUARD Raphaël TD A – Formation Initiale
GUIHENEUF Mattin TD A – Formation Initiale
MOURGUE Clément TD A – Formation Initiale
ROUYER Johan TD A – Formation Initiale
VICTORAS Dylan TD A – Formation Initiale



S6.01

Valorisation d'une application



Sujet Don't Forget Me
Commanditaire : M. Antoine MOULIN
Cahier de recette

B.U.T Informatique – Semestre 6 (2024 – 2025)

I.U.T de Bayonne et du Pays Basque

Département Informatique

Cahier de recette

Nom du projet : Don't Forget Me

Version : 1.0

Dernière mise à jour : 20/02/2025

Ce cahier contient l'ensemble des tests automatisés que nous avons pu mettre en place ainsi que leurs descriptions.

1. Introduction

Le cahier de recette vise à vérifier que chaque fonctionnalité du site de prise de rendez-vous respecte les spécifications techniques et fonctionne correctement.

2. Objectifs

- Définir et exécuter des scénarios de test spécifiques au site.
- Identifier les écarts par rapport aux attentes.
- Suivre les anomalies et leurs résolutions.

3. Organisation des tests

Utilisation de PHPUnit

L'exécution des tests pour le site de prise de rendez-vous repose sur l'utilisation de **PHPUnit**, permettant de garantir la fiabilité des fonctionnalités développées.

Le processus suit plusieurs étapes méthodiques :

1. Identification des cas de test

Chaque fonctionnalité est analysée afin d'identifier les scénarios à tester. Les cas de test incluent les flux principaux et les cas limites (ex. valeurs invalides, erreurs de validation).

2. Création d'une branche dédiée

Une nouvelle branche Git est créée pour chaque série de tests afin de maintenir un historique clair et faciliter l'intégration après validation.

3. Création du fichier de test

Les tests sont organisés dans des fichiers spécifiques à chaque fonctionnalité, respectant la convention `Feature/` ou `Unit/` selon le type de test. Si nécessaire, un dossier `Api/` est utilisé pour les tests liés aux endpoints d'API.

4. Écriture des tests avec PHPUnit

Si une fonction Helper est requise, elle est définie pour éviter la duplication de code et standardiser les données d'entrée.

Les tests sont rédigés en suivant le format **GIVEN / WHEN / THEN**, avec des commentaires clairs :

GIVEN : Contexte du test (ex. utilisateur connecté, données valides ou invalides).

WHEN : Action effectuée (ex. soumission d'un formulaire, appel d'une API).

THEN : Résultat attendu (ex. enregistrement réussi, message d'erreur renvoyé).

5. Validation des résultats

Vérifier que les messages d'erreur sont affichés correctement et traduits en fonction des langues disponibles.

Assurer que les retours respectent les exigences fonctionnelles définies.

6. Exécution et validation des tests

Lancer les tests en local via `phpunit` ou `php artisan test`.

Vérifier que toutes les assertions passent et que les erreurs éventuelles sont pertinentes.

Corriger les éventuelles anomalies et réexécuter les tests avant validation finale.

Ce processus garantit la robustesse du système et facilite la gestion des évolutions en maintenant une couverture de test efficace.

Si nécessaire, plus de détails sont présents dans le fichier "DFM-Notice de tests".

4. Tests

4.1. Test de connexion

12 cas ont été testés voir détails ci-joint dans le fichier *Test des fonctionnalités.xlsx* du dossier Tests/ dans la feuille *Connexion au compte*.

4.2. Test de création de compte client

14 cas ont été testés voir détails ci-joint dans le fichier *Test des fonctionnalités.xlsx* du dossier Tests/ dans la feuille *Création compte client*.

4.3. Test de modification d'informations de l'utilisateur

10 cas ont été testés voir détails ci-joint dans le fichier *Test des fonctionnalités.xlsx* du dossier Tests/ dans la feuille *Modifier informations user*.

4.4. Test de réservation

10 cas ont été testés voir détails ci-joint dans le fichier *Test des fonctionnalités.xlsx* du dossier Tests/ dans la feuille *Réserver*.

4.5. Test d'ajout d'activité

6 cas ont été testés voir détails ci-joint dans le fichier *Test des fonctionnalités.xlsx* du dossier Tests/ dans la feuille *Ajouter une activité*.

4.6. Test de l'appel de l'API

6 cas ont été testés voir détails ci-joint dans le fichier *Test des fonctionnalités.xlsx* du dossier Tests/ dans la feuille *Appel API*.