

## RAPPORT DE CONSTRUCTION DU PLAN DE TEST

### **LISTE DES PARTICIPANTS :**

- AWATI MAFOUO SUZY IVANA : 21T2306
- AMBASSA GUY MATTHIEU : 20U2654
- NGAH NOMO GUY ROGER : 20U2610
- KENMOE SIMO BORIS : 19M2066

## **LE PROCESSUS DE CONSTRUCTION DES PLANS DE TESTS POUR LES FICHIERS JAVASCRIPT SUIVANTS :**

User.test.js; basket.test.js; exception.test.js; exchange.test.js ; promotion.test.js ;  
event.test.js ; filter.test.js ; search.test.js

Que nous avons utilisés est le suivant :

### **1. ANALYSE DES EXIGENCES ET DU CODE :**

**Compréhension des fonctionnalités :** La première étape a consisté à comprendre les fonctionnalités et les objectifs de chaque fichier JavaScript.

**Identification des points critiques :** Nous avons identifié les fonctions critiques, les cas d'utilisation importants et les points potentiels de défaillance dans chaque fichier.

**Analyse du code :** Nous avons examiné le code source pour identifier les dépendances, les exceptions potentielles et les interactions avec d'autres modules.

### **2. DÉFINITION DES OBJECTIFS DE TEST :**

**Couverture des cas d'utilisation :** Les objectifs de test visaient à couvrir tous les cas d'utilisation importants et les scénarios possibles pour chaque fichier.

**Validation du comportement attendu :** Les tests ont été conçus pour valider le comportement attendu des fonctions et des modules.

**Détection des erreurs et des exceptions :** Les tests ont été conçus pour détecter les erreurs potentielles, les exceptions et les cas limites.

### **3. CHOIX DES OUTILS ET DES TECHNIQUES DE TEST :**

**Choix du framework de test :** Nous avons choisi le framework de test JavaScript vite test adapté aux besoins du projet

**Utilisation de mocks** : Pour isoler les dépendances et simplifier les tests, nous avons utilisé des mocks pour simuler les interactions avec les modules externes.

#### **4. ÉCRITURE DES TESTS UNITAIRES :**

**Création de tests pour chaque fonction** : Des tests unitaires ont été écrits pour chaque fonction et chaque module, en couvrant les différents cas d'utilisation et les scénarios possibles.

**Vérification des assertions** : Les assertions vérifient les résultats attendus des tests et détecter les erreurs.

**Gestion des exceptions** : Des tests ont été écrits pour gérer les exceptions potentielles et vérifier leur traitement correct.

#### **5. EXÉCUTION ET ANALYSE DES TESTS :**

**Exécution des tests** : Les tests ont été exécutés régulièrement pour détecter les erreurs et les régressions.

**Analyse des résultats** : Les résultats des tests ont été analysés pour identifier les erreurs, les points faibles et les zones à améliorer.

**Correction des bugs** : Les erreurs détectées par les tests ont été corrigées et les tests ont été mis à jour pour refléter les changements.

Le processus de construction des plans de tests que nous avons utilisé a été méthodique et a permis de garantir une couverture complète des fonctionnalités et des cas d'utilisation critiques.