

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS		SESSION 2022	
Épreuve E5 - Conception et développement d'applications (option SLAM)			
Fiche descriptive de réalisation professionnelle (recto)			
DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE			N° réalisation : 1
Nom, prénom : DESWAERTE Florent			N° candidat :
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>		Date : 01 / 04 / 2022
Organisation support de la réalisation professionnelle « HubMarket » est une SAS créée en 2020. Son activité principale est la vente en ligne de différents produits en tout genre.			
Intitulé de la réalisation professionnelle La situation se concentre sur la création d'un panier, d'une page historique des commandes ainsi que la création d'un système de paiement (sous PHP, Framework Symfony), nous permettant ainsi d'effectuer des achats sur le site à l'aide du panier (en Javascript), du système de paiement et d'afficher les commandes réalisées.			
Période de réalisation : Janvier 2022 → Mars 2022		Lieu : ARRAS (France)	
Modalité : <input type="checkbox"/> Seul(e) <input checked="" type="checkbox"/> En équipe			
Compétences travaillées <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données			
Conditions de réalisation¹ (ressources fournies, résultats attendus) Travail réalisé en cours de PPE. Ressources fournies : serveur WEB, base de données, cahier des charges. Résultats attendus : Obtenir un système de paiement et de panier fonctionnel.			
Description des ressources documentaires, matérielles et logicielles utilisées² Documentation Symfony: https://symfony.com/doc/current/index.html Documentation Stripe: https://stripe.com/docs Serveur WEB: https://s4-8010.nuage-peda.fr/HubMarket-Site/public/ Logiciels: Visual Studio Code			
Modalités d'accès aux productions³ et à leur documentation⁴ Lien GitHub : https://github.com/Florent-Deswaerte/HubMarket-Site/ Lien du site : https://s4-8010.nuage-peda.fr/HubMarket-Site/public/ Documentation GitHub : https://github.com/Florent-Deswaerte/HubMarket-Site/wiki			

¹ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

² Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

³ Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve. ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁴ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

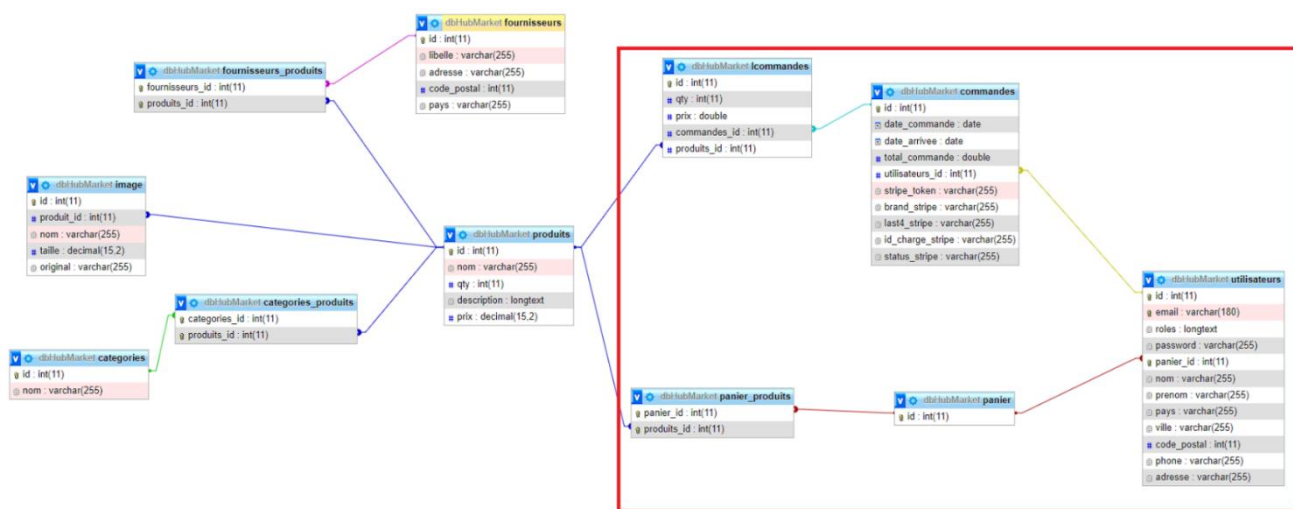
1-Présentation :

Ce projet a pour but de permettre aux utilisateurs de réaliser des achats sur le site de l'entreprise HubMarket.

Cette situation se concentre sur la gestion du panier, l'historique des commandes ainsi que le paiement des commandes (sous PHP, Framework Symfony), nous permettant ainsi d'effectuer des achats sur le site à l'aide du panier (en Javascript), du système de paiement et d'afficher les commandes réalisées.

La base de données :

Pour ce projet je me suis occupé des entités ci-dessous :



2-La partie PHP :

Je vais prendre par exemple la table « utilisateurs » :

• Les contrôleurs (permettant la récupération des données et le bon fonctionnement des pages) :

Contrôleur de la page paiement

```
//Page du paiement
#[Route('/paiement/{id}', name: 'paiement')]
public function panierPaiement(int $id, ProduitsManager $produitsManager): Response
{
    //Cherche la commande avec cette id
    $commande = $this->commandesRepository->findOneById($id);

    //Récupération information utilisateur
    $utilisateur = $this->getUser()->getData();

    //Récupération de la commande
    $intent_secret = $produitsManager->intentSecret($commande);

    return $this->render('panier/paiement.html.twig', [
        'utilisateur' => $utilisateur,
        'commande' => $commande,
        'intentSecret' => $intent_secret
    ]);
}
```

Contrôleur du formulaire utilisé pour le paiement

```
//Action formulaire
#[Route('/subscription/{id}', name: 'subscription_paiement')]
public function panierSubscription(int $id, Request $request, ProduitsManager $produitsManager): Response
{
    $commande = $this->commandesRepository->findOneById($id);
    //Récupérer le premier indice du tableau $commande pour avoir juste la commande que l'on veut
    $order = $commande[0];

    //Récupération information utilisateur
    /** @var Utilisateur $utilisateur
     * Pour pouvoir cliquer sur setPanier() (pour dire quel objet c'est)
     */
    $utilisateur = $this->getUser();
    if($request->getMethod() === "POST"){
        //Retourner la ressource et faire le traitement
        $ressource = $produitsManager->stripe($request->request->all(), $order);
        //Si ressource différent de null
        if($ressource){
            $produitsManager->create_subscription($ressource, $order, $utilisateur);
            //Retourne une réponse
            $panier = $utilisateur->getPanier();
            $utilisateur->setPanier($panier);
            $this->entityManager->flush();
            $this->helper->removeEntityObject($panier);

            $this->addFlash('success', "Paiement réussi !");
            return $this->redirectToRoute('panier_historique_commande', [
                'commande' => $commande,
            ]);
        }
    }

    return $this->redirectToRoute('panier_paiement', ['id' => $id]);
}
```

- La classe utilisateur (permettant les différentes interactions avec la base de données) :

Fonction `getData()` qui permet de récupérer certaines informations de l'utilisateur

```
public function getData(): array
{
    if(!is_null($this->getCommande())){
        $commandes = array();
        foreach($this->getCommande() as $commande){
            array_push($commandes, $commande->getData());
        }
    }
    else{
        $commandes = null;
    }
    if(is_null($this->panier)){
        $panier = null;
    }
    else{
        $panier = $this->panier->getData();
    }
    $data = array(
        'id'=>$this->id,
        'email'=>$this->email,
        'nom'=>$this->nom,
        'prenom'=>$this->prenom,
        'phone'=>$this->phone,
        'adresse'=>$this->adresse,
        'pays'=>$this->pays,
        'ville'=>$this->ville,
        'roles'=>$this->roles,
        'panier'=>$panier,
        'commandes'=>$commandes
    );
    return $data;
}
```

Extrait du formulaire de la page HTML du paiement

```
<form action="{{path('panier_subscription_paiement', {'id': commande.id}) }}" method="post" id="payment-form">
    <label>
        <div id="card-element" class="field is-empty my-6"></div>
        <span><span>Carte de crédit</span></span>
    </label>

    <button type="submit" class="site-btn btn-paiement">Acheter {{ commande.TotalCommande|number_format(2, ',', '.') }} €</button>

    <div class="outcome">
        <div class="error" role="alert"></div>

        <div class="success">
            Success! Your Stripe token is <span class="token"></span></div>
        </div>
    </form>
```

- Exemple de vues sur le site :

Paieement

```
//Page du paiement
#[Route('/paiement/{id}', name: 'paiement')]
public function panierPaiement(int $id, ProduitsManager $produitsManager): Response
{
    //Cherche la commande avec cette id
    $commande = $this->commandesRepository->findOneById($id);

    //Récupération information utilisateur
    $utilisateur = $this->getUser()->getData();

    //Récupération de la commande
    $intent_secret = $produitsManager->intentSecret($commande);

    return $this->render('panier/paiement.html.twig', [
        'utilisateur' => $utilisateur,
        'commande' => $commande,
        'intentSecret' => $intent_secret
    ]);
}
```

PAIEMENT

Carte de crédit
Numéro de carte MM / AA CVC

ACHETER 689,23 €

Affichage sur le module Stripe de l'achat

PAIEMENT

689,23 €EUR Réussi ✓

Date 15 mars à 17:44 Client Aucun Moyen de paiement Visa **** 4242 Évaluation des risques 25 Risque normal

Chronologie

- Paieement réussi 15 mars 2022 à 17:45
- Paieement démarré 15 mars 2022 à 17:44

Détails du paieement

Libellé de relevé bancaire HUBMARKET
Montant 689,23 €
Frais 20,24 €
Net 668,99 €
État Réussi
Description Aucune description Modifier

Moyen de paieement

ID pm_1KdduDFvEhgy6umFPK8k5js E-mail du propriétaire pro.flo.contact@gmail.com
Numéro **** 4242 Code postal 65365
Empreinte 41kYUxD5Ugw3hWjt Origine États-Unis
Date d'expiration 04 / 2024 Vérification CVC Réussi

Historique des commandes

Historique des commandes - Total des commandes : 5.157,49 €

ID	Date	Prix	Marque	Status
16	15/03/2022	2025,73 €	STRA	réussi
17	08/03/2022	1 741,30 €	STRA	réussi
11	02/03/2022	30,03 €	STRA	réussi
5	01/03/2022	130,03 €	STRA	réussi
6	06/02/2022	1.555,09 €	STRA	réussi
4	04/02/2022	1.500,33 €	STRA	réussi

