BTS SERVICES INFORMATIQUES AUX ORGANISATIONS **SESSION 2025**

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)

Épreuve E6 - Conception et développement d'applications (option SLAM)

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE	**N° réalisation : 1**
Nom, prénom : PARAIN Pierre	**N° candidat : 02150202107**
Épreuve ponctuelle ☒ **Contrôle en cours de formation** ☐	**Date :** 28 / 04 /2025

Organisation support de la réalisation professionnelle

SimplEduc est une ESN (Entreprise de Services du Numérique) créée en 2013 à Arras (62000). Spécialisée dans le développement de logiciels innovants pour le secteur de l'éducation, elle regroupe des développeurs experts en nouvelles technologies, notamment dans la conception de sites web et d'applications mobiles. Afin d'optimiser son organisation interne, certains de ses développeurs ont mis au point une solution permettant de gérer l'ensemble des activités de l'entreprise. Cet outil a pour objectif de faciliter la gestion des projets destinés aux clients ainsi que le suivi des collaborateurs impliqués dans leur réalisation.

Intitulé de la réalisation professionnelle : Simpleduc

Ce projet a été développé en **PHP** avec des requêtes **SQL** pour gérer la connexion, la déconnexion et l'inscription au site Share, nécessitant une confirmation par email. La gestion des utilisateurs repose sur un système d'héritage permettant de différencier les contacts et les développeurs, avec la mise en place des fonctionnalités d'ajout, de modification, de suppression et d'affichage en liste. La gestion des coordonnées inclut l'ajout et la modification des informations personnelles. JavaScript a été utilisé pour effectuer des contrôles sur les formulaires d'inscription et d'ajout de personnes, ainsi que pour ajouter une animation sur la page d'accueil lors du défilement. Plusieurs améliorations ont été apportées, notamment la refonte de la gestion des contacts, l'affichage de la liste des contacts, la gestion des fichiers partagés et l'affichage des fichiers par utilisateur.

Période de réalisation : Janvier à décembre 2024 **Lieu** : Epsi Arras, France

Modalité : ☐ **Seul(e)** ☒ **En équipe**

Compétences travaillées

☒ Concevoir une solution d'infrastructure réseau

☒ Installer, tester et déployer une solution d'infrastructure réseau Exploiter,

☒ dépanner et superviser une solution d'infrastructure réseau

Conditions de réalisation[1] (ressources fournies, résultats attendus)

Les ressources mises à disposition pour ce projet comprenaient des cours sur PHP, SQL et JavaScript, un cahier des charges détaillé, un serveur web ainsi qu'une présentation de l'entreprise.
L'objectif attendu était le développement d'une application web permettant de suivre en temps réel l'avancement des projets de l'entreprise ainsi que les équipes en charge de leur réalisation.

Description des ressources documentaires, matérielles et logicielles utilisées[2]

• Pour héberger le site : Utilisation du nuage pédagogique : https://nuage-pedagogique.fr/

• Afin de travailler en mode projet : Pour la gestion des tâches : https://trello.com/ et pour la collaboration et le versionnage : https://github.com/

Modalités d'accès aux productions[3] et à leur documentation[4*]

Lien GitHub : https://github.com/tom-billet/share
Lien du site du serveur de développement : https://s4-8048.nuage-peda.fr/sharePro/
Lien du site du serveur de production :
Lien de la documentation :
Administrateur pour se connecter : pierre.parain315@gmail.com / Alpipach3183**
Utilisateur pour se connecter : pierreparain.pro@gmail.com / Alpipach3183**

[1] En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Administration des systèmes et des réseaux » 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 nécessaires 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, par exemples schéma complet de réseau mis en place et configurations des services.

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS **SESSION 2025**

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (verso, éventuellement pages suivantes)

Épreuve E6 - Conception et développement d'applications (option SLAM)

Présentation du Projet

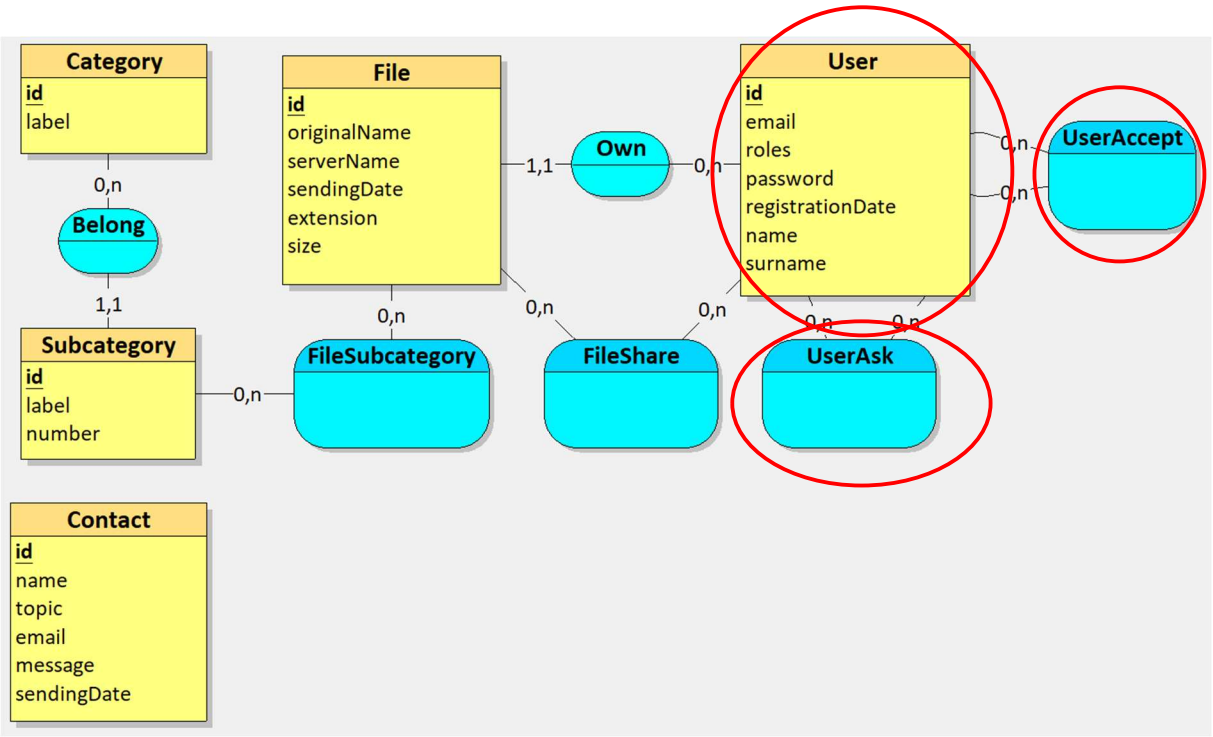
Le projet **Share** a pour objectif de faciliter la gestion et le partage de fichiers entre utilisateurs tout en assurant un contrôle précis des accès et des permissions. J'ai mis en place un système d'authentification incluant la connexion, la déconnexion et l'inscription, avec une confirmation par email pour sécuriser l'accès.

J'ai développé la gestion des utilisateurs avec une distinction entre contacts et développeurs, en intégrant un système d'héritage pour organiser ces rôles. Les fonctionnalités principales incluent l'ajout, la modification, la suppression et l'affichage en liste des contacts et des développeurs. J'ai également ajouté la gestion des coordonnées des utilisateurs, permettant leur ajout et leur modification.

Pour renforcer la sécurité et l'expérience utilisateur, j'ai intégré des contrôles JavaScript sur les formulaires d'inscription, d'ajout de personnes et de mise à jour des coordonnées. De plus, une animation a été mise en place sur la page d'accueil lors du défilement afin d'améliorer l'aspect visuel et la fluidité de navigation.

Enfin, j'ai optimisé la gestion des fichiers en permettant leur stockage, leur suppression et leur partage entre utilisateurs, avec une gestion affinée des permissions et un affichage organisé des fichiers par utilisateur.

Base de données



Mes Parties

La partie PHP et MySQL

Je vais donc commencer par expliquer comment une personne se crée un compte et se connecte

Ce contrôleur Symfony gère l'inscription des utilisateurs en affichant un formulaire et en traitant les données soumises. Il vérifie si le formulaire est valide, hash le mot de passe avec `UserPasswordHasherInterface`, et enregistre l'utilisateur en base avec `EntityManagerInterface`. Enfin, il connecte l'utilisateur automatiquement après l'inscription.

```
$user = new User();
$form = $this->createForm(RegistrationFormType::class, $user);
$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {
    /** @var string $plainPassword */
    $plainPassword = $form->get('plainPassword')->getData();

    // encode the plain password
    $user->setPassword($userPasswordHasher->hashPassword($user, $plainPassword));

    $user->setRegistrationDate(new \Datetime());

    $entityManager->persist($user);
    $entityManager->flush();

    // do anything else you need here, like send an email

    return $security->login($user, AppCustomAuthenticator::class, 'main');
}
```

```php
class SecurityController extends AbstractController
{
    #[Route(path: '/login', name: 'app_login')]
    public function login(AuthenticationUtils $authenticationUtils): Response
    {
        // if ($this->getUser()) {
        //     return $this->redirectToRoute('target_path');
        // }

        // get the login error if there is one
        $error = $authenticationUtils->getLastAuthenticationError();
        // last username entered by the user
        $lastUsername = $authenticationUtils->getLastUsername();

        return $this->render('security/login.html.twig', ['last_username' => $lastUsername, 'error' => $error]);
    }

    #[Route(path: '/logout', name: 'app_logout')]
    public function logout(): void
    {
        throw new \LogicException('This method can be blank - it will be intercepted by the logout key on your firewall.');
    }
}
```

Ce code est un contrôleur Symfony pour la gestion de l'authentification.

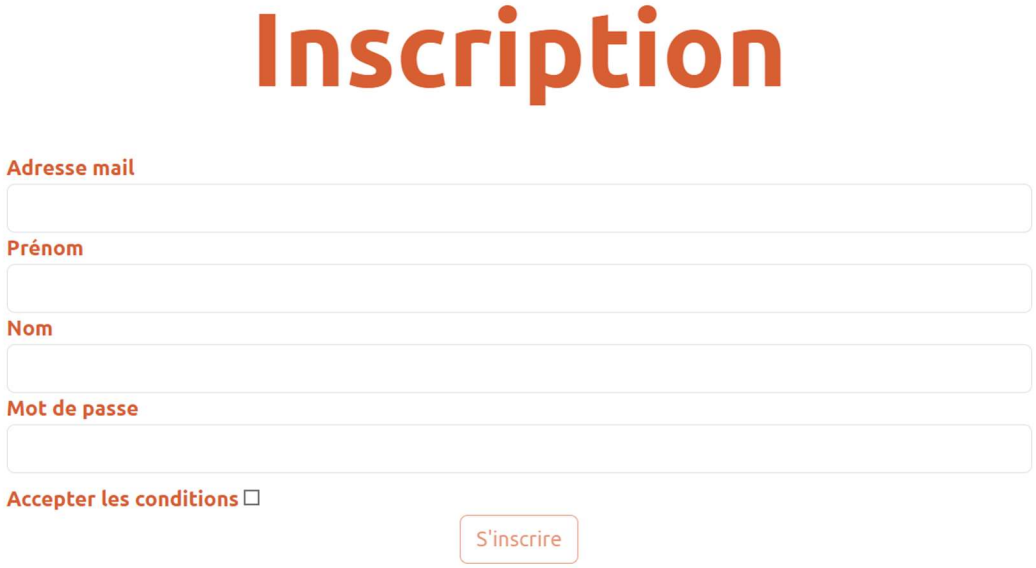
1. La méthode `login()` affiche la page de connexion et récupère les erreurs d'authentification ainsi que le dernier nom d'utilisateur saisi.
2. La méthode `logout()` est une route pour la déconnexion, mais elle est gérée automatiquement par le pare-feu Symfony.
3. Les routes `/login` et `/logout` sont définies avec des attributs PHP (`#[Route]`) pour être reconnues par Symfony.

```twig
<div class="col-12 col-md-6 bg-white p-4 m-0 text-primary">

    {{ form_errors(registrationForm) }}

    {{ form_start(registrationForm) }}
        {{ form_row(registrationForm.email) }}
        {{ form_row(registrationForm.name) }}
        {{ form_row(registrationForm.surname) }}
        {{ form_row(registrationForm.plainPassword, {
            label: 'Mot de passe'
        }) }}
        <div id="password-requirements" class="text-danger mt-2"></div>
        {{ form_row(registrationForm.agreeTerms) }}

        <div class="text-center">
            <button type="submit" class="btn" id="submit-button" disabled>S'inscrire</button>
        </div>
    {{ form_end(registrationForm) }}
</div>
</div>
```

Ce fichier Twig est un template pour une page d'inscription dans Symfony.

1. Il affiche un formulaire (`registrationForm`) contenant les champs email, nom, prénom, mot de passe et acceptation des conditions.
2. Un bouton d'inscription est désactivé par défaut et sera activé via un script JavaScript (`registerMDP.js`).
3. Il étend un template de base (`base.html.twig`) et inclut un script pour gérer les contraintes du mot de passe.

Voilà le résultat



Un peu de JAVASCRIPT :

```javascript
document.addEventListener('DOMContentLoaded', () => {
    const passwordField = document.getElementById('registration_form_plainPassword'); // Vérifie bien que l'ID est correct
    const submitButton = document.getElementById('submit-button');
    const passwordRequirements = document.getElementById('password-requirements');

    passwordField.addEventListener('input', function () {
        const password = passwordField.value;
        console.log(password);

        // Validation des règles de sécurité pour le mot de passe
        const minLength = 12;
        const hasUpperCase = /[A-Z]/.test(password);
        const hasLowerCase = /[a-z]/.test(password);
        const hasNumber = /[0-9]/.test(password);
        const hasSpecialChar = /[!@#$%^&*(),.?":{}|<>]/.test(password);

        // Construction du message d'erreur
        let message = 'Le mot de passe doit contenir :<br>';
        let isValid = true;

        if (password.length < minLength) {
            message += `- Au moins ${minLength} caractères<br>`;
            isValid = false;
        }
        if (!hasUpperCase) {
            message += '- Une lettre majuscule<br>';
            isValid = false;
        }
        if (!hasLowerCase) {
            message += '- Une lettre minuscule<br>';
            isValid = false;
        }
        if (!hasNumber) {
            message += '- Un chiffre<br>';
            isValid = false;
        }
        if (!hasSpecialChar) {
            message += '- Un caractère spécial (e.g. !@#$%^&*)<br>';
            isValid = false;
        }
```

Ce script JavaScript valide le mot de passe en temps réel et active le bouton d'inscription uniquement si toutes les conditions sont remplies.

1. **Récupération des éléments HTML**
   - Il sélectionne le champ du mot de passe, le bouton d'inscription et la zone d'affichage des erreurs.
2. **Écoute des entrées utilisateur**
   - À chaque saisie, il récupère la valeur du mot de passe et teste plusieurs critères :
     - **Au moins 12 caractères**
     - **Au moins une majuscule et une minuscule**
     - **Au moins un chiffre**
     - **Au moins un caractère spécial**
3. **Affichage des erreurs et activation du bouton**
   - Si une condition n'est pas respectée, un message d'erreur détaillé est affiché sous le champ.
   - Tant que le mot de passe est invalide, le bouton reste désactivé.
   - Dès qu'il est conforme, le bouton d'inscription devient actif.

Retour au code php :

```twig
{% if is_granted('ROLE_ADMIN') or is_granted('ROLE_MOD') %}
    <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false">
    <div class="dropdown-menu">
        <a class="dropdown-item" href="{{path('app_contacts')}}">Liste contacts</a>
        <a class="dropdown-item" href="{{path('app_categories')}}">Liste catégories</a>
        <a class="dropdown-item" href="{{path('app_add_category')}}">Ajout catégorie</a>
        <a class="dropdown-item" href="{{path('app_add_subcategory')}}">Ajout sous-catégorie</a>
    </div>
    </li>
{% endif %}

{% if is_granted('ROLE_ADMIN')%}
    <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-expanded="false">
    <div class="dropdown-menu">
        <a class="dropdown-item" href="{{path('app_users')}}">Liste utilisateurs</a>
        <a class="dropdown-item" href="{{path('app_add_file')}}">Ajout fichier</a>
        <a class="dropdown-item" href="{{path('app_files')}}">Liste fichiers</a>
    </div>
    </li>
{% endif %}
```

Le code Twig utilise la fonction `is_granted()` pour vérifier si l'utilisateur connecté possède un rôle spécifique avant d'afficher certains éléments du menu. Le premier bloc vérifie si l'utilisateur a le rôle `ROLE_ADMIN` ou `ROLE_MOD` et affiche un menu déroulant contenant des liens vers la liste des contacts, la liste des catégories, ainsi que les pages d'ajout de catégories et sous-catégories. Le second bloc, réservé uniquement aux administrateurs (`ROLE_ADMIN`), affiche un autre menu déroulant avec des liens vers la liste des utilisateurs, l'ajout de fichiers et la liste des fichiers. Ces menus sont construits en utilisant Bootstrap pour la mise en forme et l'interactivité des dropdowns. Si un utilisateur n'a pas les rôles requis, les menus correspondants ne s'afficheront pas. Pour te donner le rôle admin, il faut vérifier dans ta base de données que ton utilisateur possède `ROLE_ADMIN` dans son champ des rôles, qui est stocké en format JSON (`DC2Type:json`).

Voici l’un comparé à l’autre