

BTS SIO 2022

Support et mise à disposition de services informatiques (E4)

PAGE DE PRÉSENTATION DU DOSSIER

N° de candidat¹ : | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ |

NOM : Zatric.....

Prénom : Nihad.....

Date de passage ¹ : / /2022	Heure de passage ¹ :h.....
--	---

CATEGORIE CANDIDAT ² (UNE CASE A COCHER)	
<input type="checkbox"/> Scolaire <input checked="" type="checkbox"/> Apprenti <input type="checkbox"/> Formation professionnelle continue <input type="checkbox"/> Expérience professionnelle 3 ans	<input type="checkbox"/> Ex-scolaire <input type="checkbox"/> Ex-apprenti <input type="checkbox"/> Ex-formation professionnelle continue

¹ Informations communiquées sur votre convocation envoyée en mars-avril 2022

² Informations communiquées sur votre confirmation d'inscription

Tampon de
l'établissement



BTS SIO – Dossier Etudiant
Justificatif d'acquisition des compétences

Epreuve E4
Support et mise à disposition de
services informatiques

SOMMAIRE

- 1 INTRODUCTION 4
- 2 MISSION 3 : CONNEXION UTILISATEUR 5
 - 2.1 Cahier des charges..... 5
 - 2.2 Etude et conception de la solution 6
 - 2.3 Gestion de projet 7
 - 2.4 Mise en œuvre..... 8
 - 2.5 Bilan 12

1 Introduction

L'objectif de ce document est de vous présenter les missions professionnelles que j'ai effectué dans le cadre de ma formation BTS SIO à l'école IRIS de Strasbourg.

Ces missions peuvent être de trois types :

- Effectuées en entreprise durant une alternance
- Effectuées en stage en entreprise
- Effectuées à l'école (compte-rendu de TP, projet collaboratif)

Le type de la mission sera précisé dans chaque cahier des charges.

Ce document se compose des parties suivantes :

Chapitres	Contenu
Chapitres 2 à ...	<p>Présentation des missions, avec pour chacune :</p> <ul style="list-style-type: none">- Le cahier des charges- La solution proposée- La gestion de projet- La mise en œuvre- Le bilan du projet

2 Mission 3 : Connexion utilisateur

2.1 Cahier des charges

Type de mission
Mission effectuée en alternance.
Contexte
Dans le cadre de la réalisation d'un programme de service après-vente, j'ai mis en place un système de connexion administrateur.
Demande du client
Mettre en place un système de connexion administrateur de sorte à accéder à des fonctionnalités avancées du programme.
Expression du besoin
Un bouton de connexion dans le header (en haut à gauche de la page).
Un modal (formulaire caché) s'affichant lorsqu'on clique sur le bouton permettant d'entrer les informations.
Une vérification des informations entrées par l'utilisateur.
Si la connexion est réussie : on remplit des cookies avec les informations de l'utilisateur, lui permettant d'accéder à des fonctionnalités avancées du programme.
Budget disponible
Mon cout salarial sur une semaine.
Outils disponibles
Un poste avec un serveur local installé et configuré (Wampserver).
Contraintes
Temps : 1 semaine.
Responsive : non.
Ressources : une semaine de travail, seul.
Confidentialité

2.2 Etude et conception de la solution

2.2.1 Les solutions possibles

Langages de programmation :

- Php
- Javascript
- HTML

Frameworks :

- JQuery
- VueJS

Technologies : AJAX

2.2.2 La solution retenue

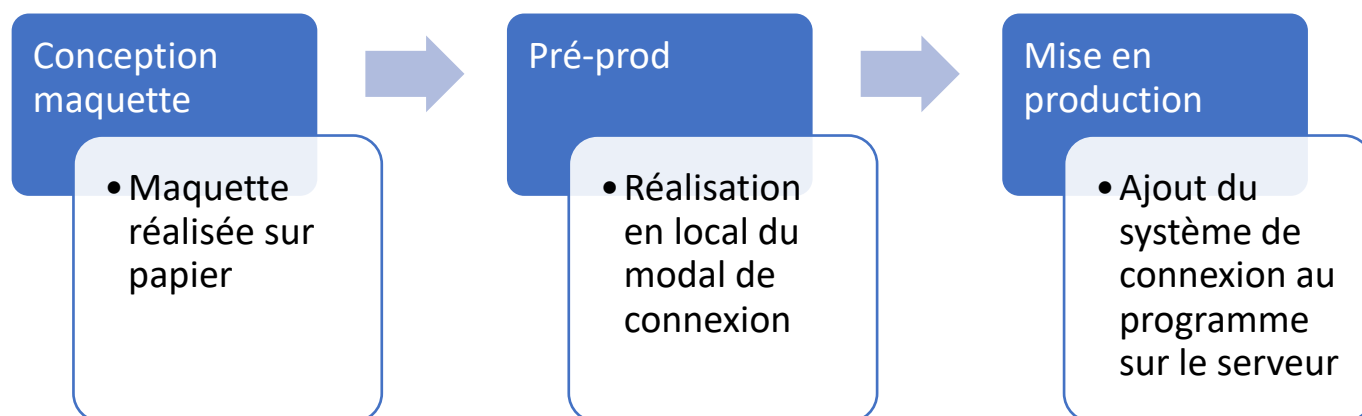
J'ai décidé de réaliser cette mission en PHP procédural, sans framework car la réalisation de ce système de pagination ne requiert pas énormément de lignes de codes donc il semble superflu d'intégrer un framework pour si peu.

De plus, je n'ai pas eu besoin de mettre en place de requête asynchrone (AJAX), puisque le contenu de la page est assez léger pour qu'une actualisation n'impacte pas négativement la fluidité du programme.

2.3 Gestion de projet

2.3.1 Planing de déploiement de la solution

Schéma de la réalisation de la mission :



2.3.2 Budget

Matériel : 1 poste avec un serveur installé et configuré (Wampserver).

Licences : PHPStorm.

Coût humain : 1 semaine de travail pour 1 personne.

2.4 Mise en œuvre

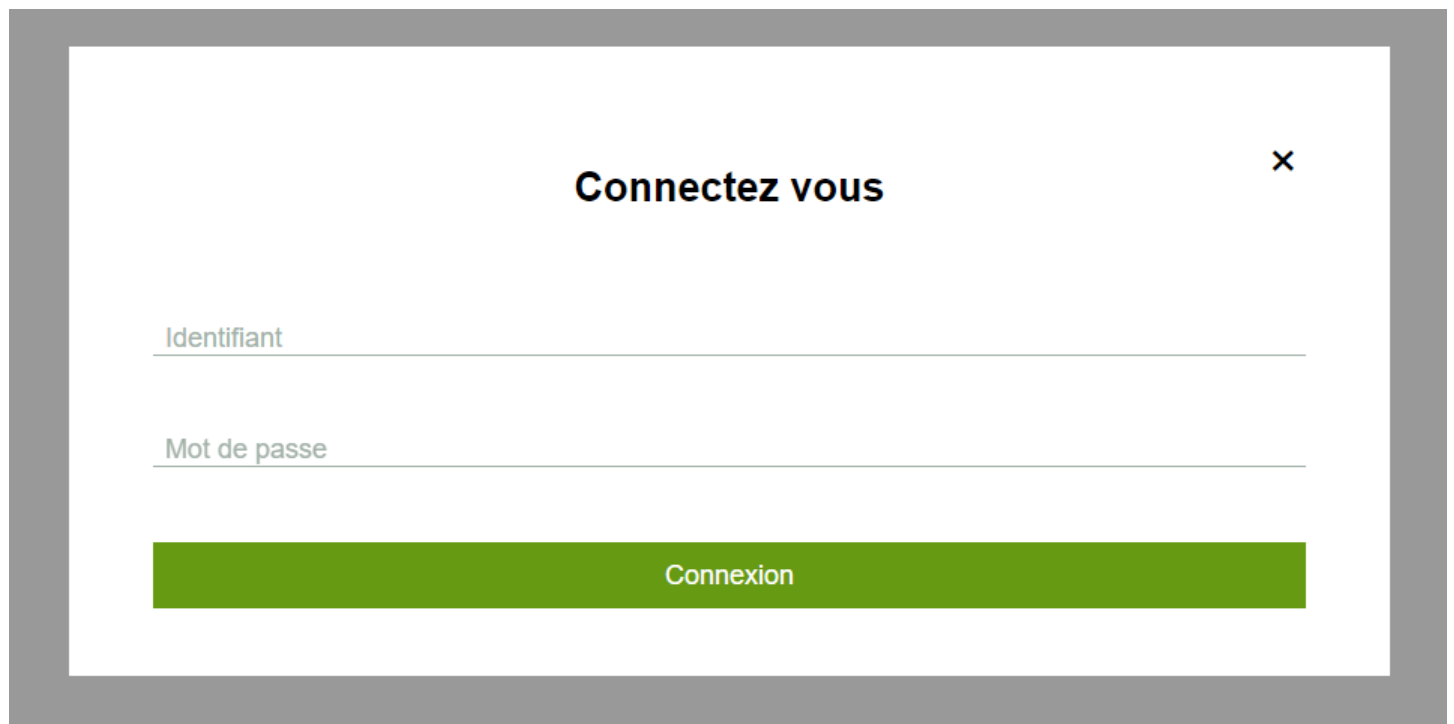
2.4.1 Implémentation de la solution

Tout d'abord, on crée le formulaire en HTML :

```
<form class="form d-none" method="post" id="modal_connexion">
  <div class="legende_form">
    <span id="close_modal_connexion" onclick="cacher_modal_connexion()"></span>
    <h2>Connectez vous</h2>
  </div>
  <div class="contenu_form">
    <div class="form-group">
      <input type="text" class="connexion_input" name="identifiant" id="identifiant" autocomplete="off" required />
      <label for="identifiant" class="label-name">
        <span class="content-name">Identifiant</span>
      </label>
    </div>
    <div class="form-group">
      <input type="password" class="connexion_input" name="mot_de_passe" id="mot_de_passe" autocomplete="off" required />
      <label for="mot_de_passe" class="label-name">
        <span class="content-name">Mot de passe</span>
      </label>
    </div>

    <button id="bouton_connexion" type="button" class="bouton" onclick="authentification()">Connexion</button>
  </div>
</form>
```

On obtient ce rendu, visible lorsqu'on clique sur le bouton de connexion :



The screenshot shows a modal window titled "Connectez vous" with a close button (X) in the top right corner. The form contains two input fields: "Identifiant" and "Mot de passe", both with placeholder text. Below the inputs is a green button labeled "Connexion". The modal is set against a dark gray background.

Une fois que l'on reçoit les informations renseignées par l'utilisateur, on sélectionne tous les utilisateurs stockés dans la base de données :

```
$query = "SELECT * FROM pointage_salaries";
```

Grâce à cette requête on remplit un tableau que l'on va balayer. Si on trouve une correspondance entre les noms de compte et mot de passe entrés par l'utilisateur et un de ceux compris dans la base de données, on autorise l'accès (\$acces = true) :

```
// on remplit les variables à partir du formulaire de connexion, avec chaque première lettre en majuscule
// par exemple Jean pierre devient Jean Pierre
$identifiant = strtolower($_POST['identifiant']);
$mot_de_passe = $_POST['mot_de_passe'];

$query = "SELECT * FROM pointage_salaries"; // tableau contenant tous les comptes utilisateurs
$result = $bdd->query($query);

$acces = false;
while($tab_ligne = $result->fetchALL( mode: PDO::FETCH_NUM )){ // on parcourt le tableau des comptes utilisateurs
    // pour chaque ligne du tableau, on remplit les variables nom et prénom (avec chaque première lettre en majuscule)
    for($i=0; $i<count($tab_ligne); $i++){
        $identifiant_bdd = strtolower($tab_ligne[$i][9]);
        // si l'identifiant et le mot de passe (haché dans la bdd, que l'on vérifie grâce à password_verify) correspondent
        if($identifiant == $identifiant_bdd && password_verify($mot_de_passe, $tab_ligne[$i][10])) {
            // alors on accorde l'accès
            $acces = true;

            $nom = ucwords(strtolower($tab_ligne[$i][1]));
            $prenom = ucwords(strtolower($tab_ligne[$i][2]));
            $role = $tab_ligne[$i][11];
            $id = $tab_ligne[$i][0];
            break;
        }
    }
}
```

On remplit également des variables qui nous permettront de personnaliser le programme.

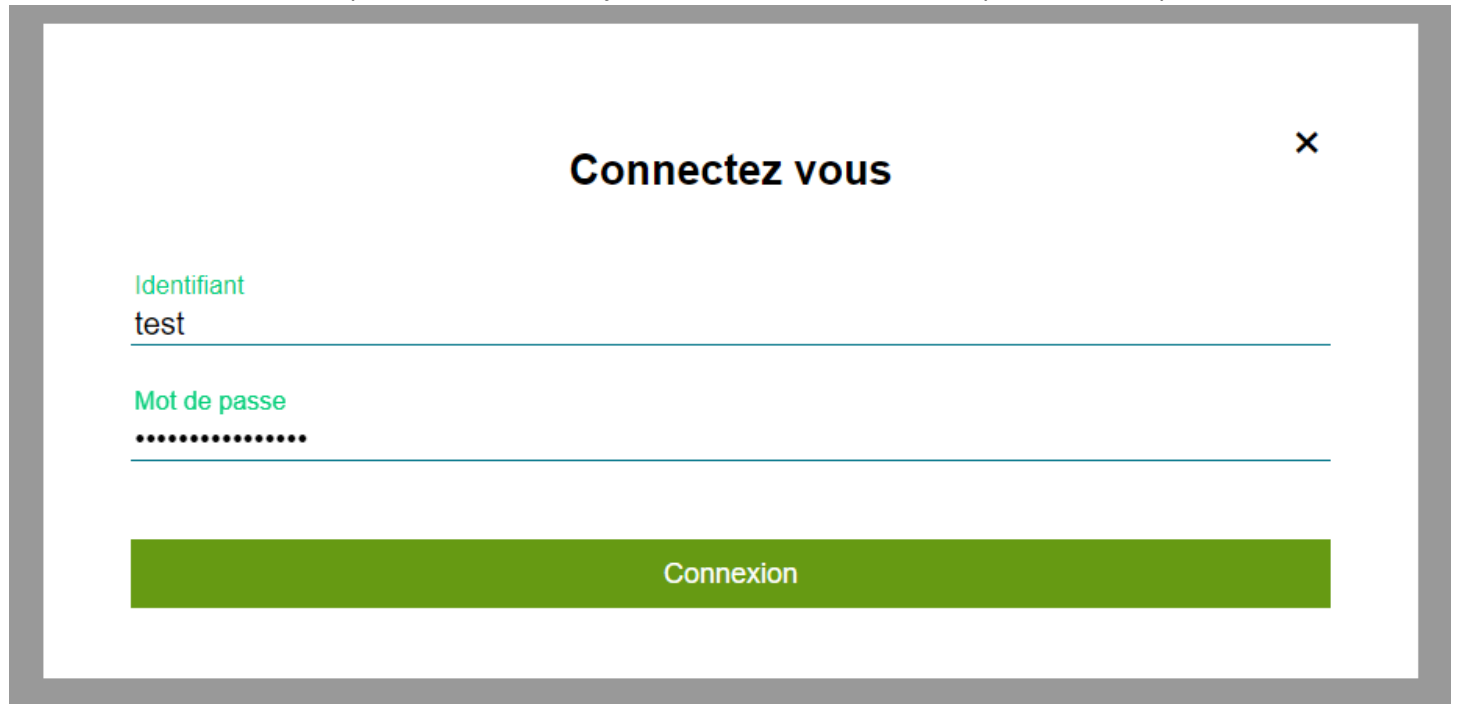
Enfin, si l'accès est autorisé, alors on remplit les cookies permettant à l'utilisateur d'accéder à des fonctionnalités avancées :

```
// si on a autorisé l'accès alors on définit 4 cookies
if($acces == true) {
    setcookie('nom', $nom, (time() + 604800));
    setcookie('prenom', $prenom, (time() + 604800));
    setcookie('identifiant', $identifiant, (time() + 604800));
    setcookie('id', $id, (time() + 604800));
    setcookie('affichage', 'user', (time() + 604800));
}
else {
    // sinon on renvoie une alerte popup
    echo<script>alert("Utilisateur ou mot de passe inconnu")</script>;
}
```

2.4.2 Recettage

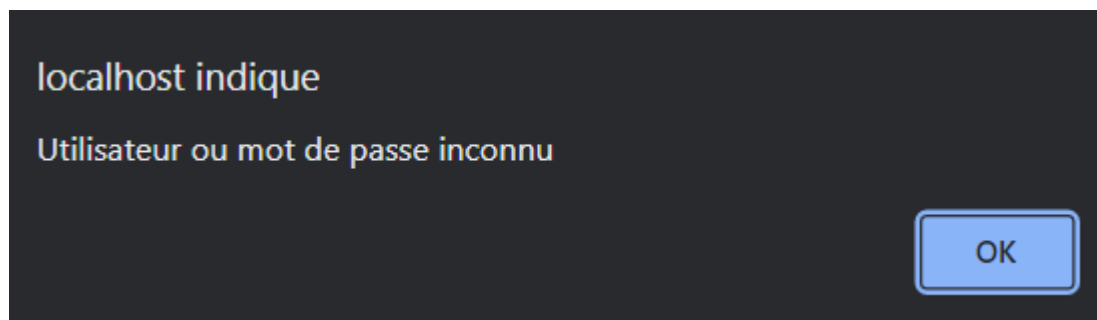
Test fonctionnel

Contexte 1 : Je me mets à la place de l'utilisateur, j'entre un mauvais nom de compte ou mot de passe :



The screenshot shows a login window titled "Connectez vous" with a close button (X) in the top right corner. It contains two input fields: "Identifiant" with the text "test" and "Mot de passe" with masked characters (dots). Below the fields is a green button labeled "Connexion".

Le programme renvoie un message d'erreur :



Contexte 2 : j’entre le bon nom de compte et mot de passe

×

Connectez vous

Identifiant

nihad

Mot de passe

.....

Connexion

La connexion est établie :



Zatric Nihad

Numéro de commande :



Tout afficher ☐

Liste des demandes de SAV

N° cmd	N° cmd client	Date BL	Date de saisie	Catégorie	Sous catégorie
186137	PR 0 CR4 27881	09/09/2021	07/10/2021	interne	fournisseur
186137	PR 0 CR4 27881	09/09/2021	01/10/2021	externe	transport
186137	PR 0 CR4 27881	09/09/2021	01/10/2021	interne	fournisseur
186137	PR 0 CR4 27881	09/09/2021	01/10/2021	interne	fournisseur
186137	PR 0 CR4 27881	09/09/2021	30/09/2021	interne	fournisseur

2.5 Bilan

2.5.1 Validation des exigences point par point

Création du bouton de connexion ☒

Création du formulaire de connexion ☒

Vérification de l'intégrité et de la correspondance des données de connexion ☒

Remplissage des cookies ☒

Validation de la connexion ☒

2.5.2 Axes d'amélioration

Connexion en AJAX pour améliorer l'expérience utilisateur.

2.5.3 Compétences acquises

Maîtrise des requêtes de type POST.

Gestion de formulaires en PHP/Javascript.

Hachage de mot de passe (bcrypt).