

# BTS SIO 2022

## Support et mise à disposition de services informatiques (E4)

PAGE DE PRÉSENTATION DU DOSSIER

N° de candidat<sup>1</sup> : | \_ | \_ | \_ | \_ | \_ | \_ | \_ | \_ | \_ | \_ | \_ | \_ |

NOM : Zatric.....

Prénom : Nihad.....

Date de passage <sup>1</sup> : ..... / ..... /2022	Heure de passage <sup>1</sup> : .....h.....
--	---

CATEGORIE CANDIDAT <sup>2</sup> (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

<sup>1</sup> Informations communiquées sur votre convocation envoyée en mars-avril 2022

<sup>2</sup> 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 4 : REQUETE ASYNCHRONE EN VANILLA JS..... 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 ..... 11

# 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"><li>- Le cahier des charges</li><li>- La solution proposée</li><li>- La gestion de projet</li><li>- La mise en œuvre</li><li>- Le bilan du projet</li></ul>

## 2 Mission 4 : Requête asynchrone en vanilla JS

### 2.1 Cahier des charges

<b>Type de mission</b>
Mission effectuée en alternance.
<b>Contexte</b>
Dans le cadre de la réalisation d'un WYSIWYG permettant aux commerciaux de l'entreprise de générer des fichiers PDF présentant nos offres promotionnelles, j'ai intégré un formulaire permettant de créer des caractéristiques (description technique d'un article).
<b>Demande du client</b>
Mettre en place un système d'ajout de caractéristique sans rechargement de page.
<b>Expression du besoin</b>
Un bouton pour créer une caractéristique.
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 caractéristique est créée : création dynamique de l'élément dans le dom pour palier le non rechargement de la page.
<b>Budget disponible</b>
Mon cout salarial sur un jour.
<b>Outils disponibles</b>
Un poste avec un serveur local installé et configuré (Wampserver).
<b>Contraintes</b>
Temps : 1 jour.
Responsive : non.
Ressources : un jour de travail, seul.
<b>Confidentialité</b>

## 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, j'ai mis en place la méthode AJAX en place afin d'augmenter l'expérience utilisateur en évitant le rechargement de la page.

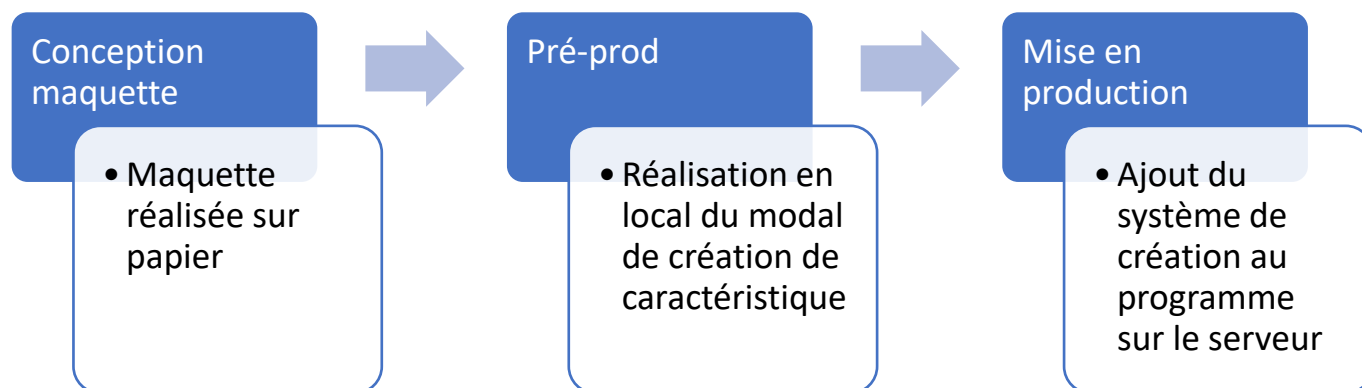
## 2.3 Gestion de projet

---

### 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 jour 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 id="crud_caracteristique" class="d-none div_modal" method="POST">
  <div id="div_crud_caracteristique" class="div_crud">
    <h2 id="h2_crud_caracteristique" class="h2_crud">Ajoutez le contenu de la nouvelle caractéristique :</h2>
    <input type="text" class="form-control" id="champ_caracteristique" name="caracteristique" placeholder="Caracteristique">
    <button type="button" id="valider_creation_caracteristique" onclick="fonction_valider_creation_caracteristique()">Créer</button>
  </div>
</form>
```

On obtient ce modal lorsqu'on clique sur le bouton de création :

Cliquez sur les caractéristiques à afficher :



Ajoutez le contenu de la nouvelle caractéristique :

Créer la caractéristique

Lorsqu'on crée une caractéristique, elle est ajoutée à la liste de caractéristiques de façon dynamique :

Meuble monté tourillonné collé, suspendu par boîtiers d'accrochage réglables anti-décrochement. Vide sanitaire de 52mm.	✓		
Corps de meuble et façades en panneau aggloméré hydrofuge mélaminé de 18mm d'épaisseur	✓		
Portes avec amortisseurs intégrés. Poignées métal chromées brillantes.			
Tiroirs double parois métallique, sortie totale amortie	✓		
Lot de 2 pieds aluminium	✓		
Plan vasque marbre reconstitué brillant, système de trop plein moulé intégré	✓		



Cette fonction Javascript permet d'ajouter via le fichier `ajax_creer_caracteristique.php` la caractéristique à la base de données de façon asynchrone.

```
function fonction_valider_creation_caracteristique() {
    let description = champ_caracteristique.value;
    if (description !== '') {
        let ajax = new XMLHttpRequest();

        ajax.open("POST", "connexions/ajax_creer_caracteristique.php", true);
        ajax.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
        ajax.send("description=" + description);

        ajax.onreadystatechange = function () {
            if (this.readyState == 4 && this.status == 200) {
                /*console.log(this.responseText);*/
                let id = this.responseText;
                fonction_creer_ligne_caracteristique(id);
            }
        }
        crud_caracteristique.classList.add('d-none');
    }
    else {
        alert('Le champ ne doit pas être vide.');
```

De plus, si l'ajout est effectué avec succès, la fonction `fonction_creer_ligne_caracteristique(id)` est exécutée :

Elle permet via la méthode javascript `document.createElement` de créer une ligne de caractéristique supplémentaire dans le DOM. De ce fait, on compense le fait qu'il n'y a pas d'actualisation de la page.

Fichier `ajax_creer_caracteristique.php` :

```
include ('connexion_mysql.php'); // connexion à la base de données

$query = "INSERT INTO offre_caracteristiques (description) VALUES (:description) "; // requête sql

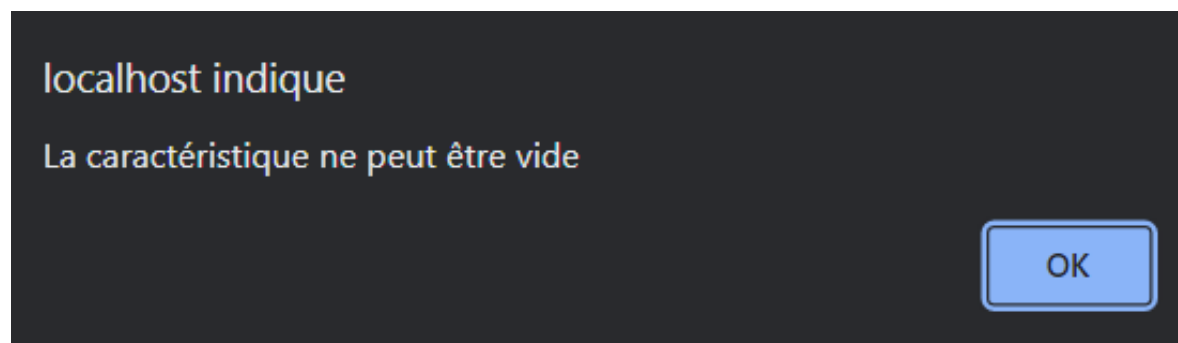
$result = $bdd->prepare($query);
$result->bindParam(':description', $_POST['description'], PDO::PARAM_STR, 255); // protection de l'intégrité des données
$result->execute(); // execution de la requête sql
$result->closeCursor(); // fermeture de la connexion à la base de données

echo $bdd -> lastInsertId();
```

## 2.4.2 Recettage

### Test fonctionnel

Contexte 1 : Je me mets à la place de l'utilisateur, je valide le formulaire sans entrer de caractéristique > le programme renvoie un message d'erreur :

















Contexte 2 : Je me mets à la place de l'utilisateur, je valide le formulaire en entrant une caractéristique :

Ajoutez le contenu de la nouvelle caractéristique :

Créer la caractéristique

La caractéristique est ajoutée à la liste :

Portes avec amortisseurs intégrés. Poignées métal chromées brillantes.			
Tiroirs double parois métallique, sortie totale amortie			
Lot de 2 pieds aluminium			
Plan vasque marbre reconstitué brillant, système de trop plein moulé intégré			
test			

## 2.5 Bilan

---

### 2.5.1 Validation des exigences point par point

---

Création du bouton de pour afficher le modal ☒

Création du modal ☒

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

Ajout de la caractéristique de façon asynchrone dans la BDD ☒

Ajout de la caractéristique dans le DOM ☒

### 2.5.2 Axes d'amélioration

---

Duplication + modification d'une ligne déjà existante plutôt que de recréer un élément de 0.

### 2.5.3 Compétences acquises

---

Maîtrise des requêtes de type POST.

Gestion de formulaires en PHP/Javascript.

Requête asynchrone en vanilla JS.