



BTS SIO – Option SLAM

Documentation d'épreuve

Rédacteur	Version	Date	Nb pages
DIETRICH Johann	1.1	12/02/2023	15

Mise en place d'une page administrateur
permettant d'éditer les méta description des
pages web

SOMMAIRE

1	CAHIER DES CHARGES	3
1.1	Introduction.....	3
1.2	Expression fonctionnelle du besoin	4
1.3	Contraintes.....	5
1.4	Gestion des droits d'accès.....	6
2	DESCRIPTION DES ENVIRONNEMENTS	7
3	METHODOLOGIE.....	8
3.1	Méthodologie et versioning.....	8
3.2	Gestion des tests de la solution	8
3.3	Gestion de projet	9
4	MISE EN OEUVRE	10
4.1	La création de la page administrateur.....	10
4.2	La/les pages voulues.....	12
4.3	Tester le référencement SEO de notre site.....	13
5	GESTION DE LA MAINTENANCE (CORRECTIVE / EVOLUTIVE).....	14
5.1	Mise à jour de la documentation du SI.....	14
5.2	Evaluation de la qualité de la solution.....	14
5.3	Gestion des tests de mise à jour.....	14
6	BILAN DU PROJET.....	15
6.1	Validation des exigences point par point.....	15
6.2	Axes d'amélioration	15
6.3	Compétences acquises	15

1 Cahier des charges

1.1 Introduction

Type de mission
Mission effectuée dans le temps libre
Contexte
Un site a besoin d'une manière plus simple pour mettre des balises méta data pour le référencement web.
Demande du client
Un moyen de mettre des balises sur les pages.
Budget disponible
Budget 0

1.2 Expression fonctionnelle du besoin

Liste des fonctionnalités attendues :

Front office

Page aléatoire 1 :

- Avoir les balises méta affichées

Page aléatoire 2 :

- Avoir les balises méta affichées

Page administrateur :

- Page permettant de choisir la page et définir le titre et la description de celle-ci

1.3 Contraintes

Générales
Quelques journées
Juridiques
Aucune informations sensibles stockés donc aucune contrainte RGPD
Techniques
Simplement utilisation de PHP
Ergonomique
Visible et compréhensible de manière basique

1.4 Gestion des droits d'accès

Administrateur
Accès à la page administratrice

2 Description des environnements

Environnement de développement
Purement en live server PHP

3 Méthodologie

3.1 Méthodologie et versioning

Utilisation de github

3.2 Gestion des tests de la solution

unitaires

Créer une page avec le header associé

Choisir la page liée dans la page administrative et lui donner les métadonnées voulues

Voir si la page les a reçues

3.3 Gestion de projet

3.3.1 Budget

Pas de réel budget car ce n'est qu'un projet simulé, a utiliser sur un plus grand projet.

4 Mise en oeuvre

Pour effectuer ce projet, il nous faut plusieurs choses mises en place. La page administrateur, les pages qui vont avoir les balises.

4.1 La création de la page administrateur

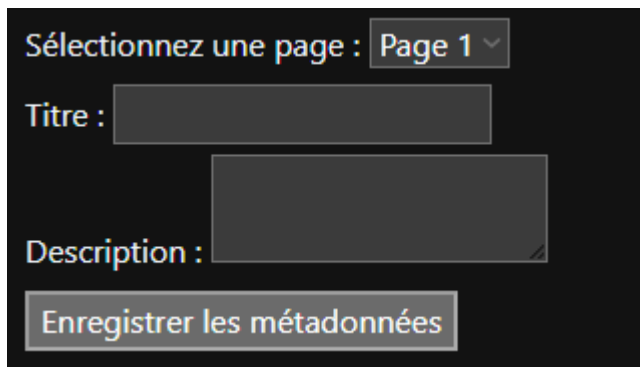
En première partie, on va créer le sélecteur de page en créant une array définissant les pages.

```
$pages = [  
    'page1.php' => 'Page 1',  
    'page2.php' => 'Page 2',  
    'page3.php' => 'Page 3'  
];
```

Et le sélecteur grâce à cet array :

```
<form method="post" action="accueil.php">  
    <label for="page">Sélectionnez une page :</label>  
    <select name="page" id="page">  
        <?php foreach ($pages as $uri => $name) { ?>  
            <option value="<?php echo $uri; ?>"><?php echo $name; ?></option>  
        <?php } ?>  
    </select>  
    <br>  
    <label for="title">Titre :</label>  
    <input type="text" name="title" id="title">  
    <br>  
    <label for="description">Description :</label>  
    <textarea name="description" id="description"></textarea>  
    <br>  
    <button type="submit">Enregistrer les métadonnées</button>  
</form>
```

Ce qui nous donne :



Maintenant en seconde partie, on doit indiquer ce qu'il va se passer quand l'on ajoute les données qu'on veut. Celles-ci sont transformées en fichiers avec les informations voulues dans les balises méta :

```
// Si le formulaire est soumis, enregistrer les métadonnées dans un fichier
if (isset($_POST['page']) && isset($_POST['title']) && isset($_POST['description'])) {
    $page = $_POST['page'];
    $title = $_POST['title'];
    $description = $_POST['description'];

    // Enregistrer les métadonnées dans un fichier
    $metadata_file = "$page.metadata";
    file_put_contents($metadata_file, "$title\n$description");
}
```

4.2 La/les pages voulues

Pour la page concernée, qui doit avoir une balise méta, il suffit donc, d'appeler le fichier contenant les informations voulues, de les prendre, et de les afficher dans les balises montrées dans la Head de la page.

Ceci est fait assez facilement grâce à une fonction permettant de récupérer les données d'un fichier.

```
<?php
// Lire les métadonnées depuis le fichier "page1.php.metadata"
$metadata_file = "page1.php.metadata";
if (file_exists($metadata_file)) {
    $metadata = file($metadata_file, FILE_IGNORE_NEW_LINES);
    $title = $metadata[0];
    $description = $metadata[1];
} else {
    $title = "Titre par défaut";
    $description = "Description par défaut";
}
```

Ceci permettant de récupérer les données ou de mettre des données par défaut au cas où.

Puis il reste plus qu'à afficher les balises méta à l'aide des variables décidées dans le Head comme ci-dessous.

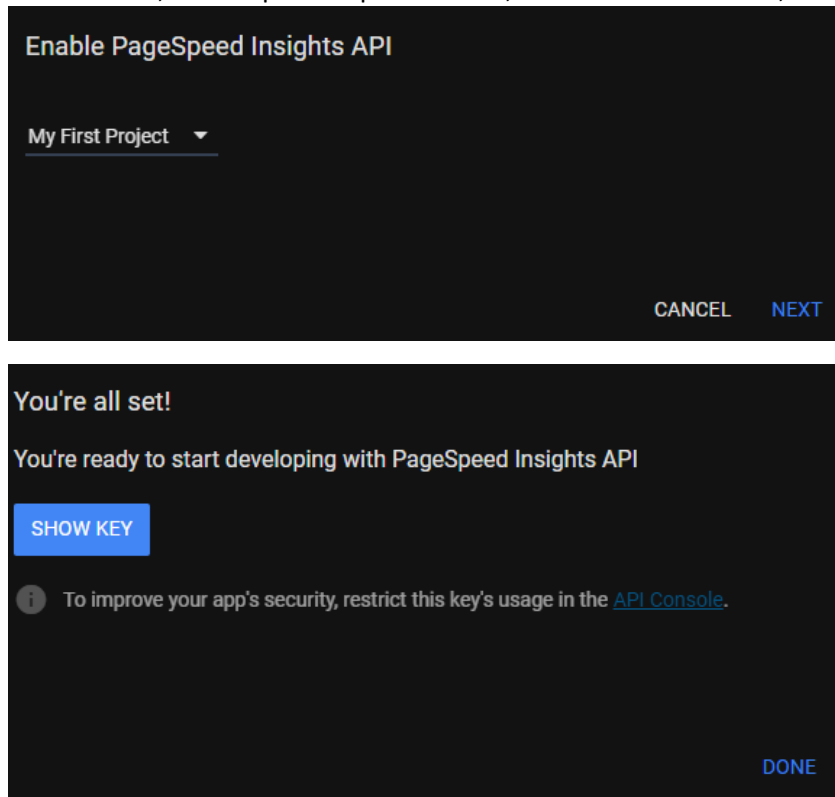
```
<head>
    <title><?php echo $title;?></title>
<meta name="description" content="<?php echo $description; ?>">
<link rel="stylesheet" href="..\css\css.css"/>
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <script src="..\JS\js.js"></script>
</head>
```

Bien sûr, tout en mettant toutes les informations d'autres fichiers nécessaires tel que le css ou le js.

4.3 Tester le référencement SEO de notre site

Maintenant que l'on a mis en place la partie gérant les balises SEO, il serait pratique de pouvoir voir l'avancement et l'utilité de celles-ci par rapport au référencement SEO de notre site. Pour cela, rien de plus simple et pratique qu'utiliser une autre API de google, pagespeed API.

Pour se faire, rien de plus simple. D'abord, comme toutes les API, il nous faut obtenir une clé.



Une fois que l'on a la clé en possession, il nous suffit de mettre notre code ci-dessous :

```
$url = 'https://www.dietrichportfolio.site'; // L'URL du site web à évaluer
$api_key = 'xxxx xxxx xxxx xxxx'; // Clé API pour l'API Google PageSpeed Insights
$api_url =
"https://www.googleapis.com/pagespeedonline/v5/runPagespeed?url={$url}&key={$api_key}";

// Appel de l'API et récupération des données
$data = file_get_contents($api_url);
$results = json_decode($data, true);

// Récupération du score de vitesse
$speed_score = $results['lighthouseResult']['categories']['performance']['score'] * 100;

// Affichage du score de vitesse
echo "Score de vitesse pour {$url} : {$speed_score}/100";
```

Ce petit code nous permettant, après avoir tout configuré (l'url à analyser, la clé, et l'url de l'api) de récupérer les informations sous format json qu'on décode et qu'on met sur un total de 100.

Ensuite, celui-ci est simplement affiché tel un pourcentage (/100).

Grâce à cela, on peut donc observer la qualité de référencement de notre site et savoir si l'on a besoin d'optimiser plus ou pas.

5 Gestion de la maintenance (corrective / évolutive)

5.1 Mise à jour de la documentation du SI

Mettre à jour des documentations technique et d'utilisation d'une solution

Logiquement aucune mise à jour sera nécessaire mais au cas où, la documentation de l'API reste toujours disponible sur le site officiel et donc pourra être utilisée en cas de changement.

5.2 Evaluation de la qualité de la solution

Évaluer la qualité d'une solution

La solution est simple mais fonctionnelle, on est capable de cette manière de modifier les balises importantes de chaque pages.

5.3 Gestion des tests de mise à jour

Élaborer et réaliser les tests des éléments mis à jour

Logiquement, aucun réel test sera nécessaire en cas de mise à jour à part tester purement et simplement la fonctionnalité de l'API et de la mise en place des balises.

6 Bilan du projet

6.1 Validation des exigences point par point

Capacité d'éditer des méta descriptions d'une page

Capacité de le faire facilement

Capacité d'analyser notre site afin de voir les changements/potentiels changement à faire

6.2 Axes d'amélioration

Trouver un moyen de voir directement dans notre site de quel manière améliorer le référencement SEO.

6.3 Compétences acquises

Se servir d'une API

Se servir d'un fichier afin de stocker des informations importantes

Savoir réutiliser se fichier et un array afin de définir (dans notre cas) les balises méta