



GÉSTION DES DÉPENSES

Mini-projet XML-PHP

ENCADRÉ PAR
Mr. Elmakhefi Nouredine

Réalisé par:
El Khayati El Mehdi

Contents

| | |
|-----------------------------------|---|
| Introduction | 2 |
| Présentation du mini-projet | 2 |
| 1. À propos !..... | 2 |
| 2. Technologies utilisées. | 2 |
| a. HTML, CSS et Bootstrap | 3 |
| b. Font awesome..... | 3 |
| c. PHP..... | 3 |
| d. XML | 4 |
| e. DTD..... | 4 |
| 3. Fonctionnalités..... | 5 |
| a. Ajouter une dépense..... | 5 |
| b. Modifier une dépense..... | 5 |
| c. Supprimer une dépense..... | 6 |
| Conclusion..... | 6 |

Table des figures

| | |
|---|---|
| Figure 1 - interface de l'application | 3 |
| Figure 2 - Code addition d'une dépense | 4 |
| Figure 3 - fichier xml de l'application..... | 4 |
| Figure 4 - fichier dtd de l'application | 5 |
| Figure 5 - ajouter une dépense | 5 |
| Figure 6 - modifier une dépense | 6 |
| Figure 7 - supprimer dépense | 6 |

Introduction

Aux débuts d'Internet, les ordinateurs et les programmes échangeaient des données en utilisant des fichiers. Malheureusement, ces fichiers avaient bien souvent des règles de formatage qui leur étaient propres. Par exemple, les données étaient séparées par des points, des virgules, des espaces, des tirets, etc.

Le problème avec ce système est qu'il fallait sans cesse adapter les programmes au format du fichier ce qui représentait une charge de travail importante.

Il a donc fallu régler ce problème rapidement. Le langage **SGML** ou **Standard Generalized Markup Language** est alors né. C'est un langage puissant, extensible et standard qui permet de décrire à l'aide de **balises** un ensemble de données. Mais ce langage, très complexe, n'était pas forcément compatible pour effectuer des échanges sur le web.

Un groupe d'informaticiens ayant de l'expérience dans le SGML et le web a alors décidé de se pencher sur le sujet. Le langage **XML** est donc né. Le XML 1.0 est devenu une recommandation du W3C (le "World Wide Web Consortium") le 10 février 1998.

Depuis, les spécifications du langage ont évolué, et la version 1.1 a été publiée le 4 février 2004. C'est pourtant la version 1.0 du langage XML qui est encore la plus utilisée aujourd'hui et c'est cette version que nous allons étudier dans ce tutorial.

Alors, quel est l'objectif de XML ?

L'objectif du XML est de faciliter les échanges de données entre les machines. A cela s'ajoute un autre objectif important : décrire les données de manière aussi bien compréhensible par les hommes qui écrivent les documents XML que par les machines qui les exploitent.

Présentation du mini-projet

1. À propos !

Ce mini-projet a but de maîtriser l'utilisation de la technologie XML afin de l'exploiter dans les autres projets informatique. Il s'agit d'un système de gestion des dépenses, où on peut créer une nouvelle dépense avec des attributs Id, Montant, Personne et description, modifier une dépense, supprimer une dépense et lister tous les dépenses stocker dans le fichier XML.

2. Technologies utilisées.

a. HTML, CSS et Bootstrap

C'est technologies sont indispensable pour la création des pages WEB. HTML et CSS sont la base, alors que bootstrap à but de faciliter le styling des pages Web.

Gestion des dépenses d'un utilisateur

+ Nouvelle dépense

Une action de dépenses a été ajoutée

| ID dépense | Montant | Personne | Description | Action |
|------------|----------|----------|-----------------------|------------------------------|
| 2 | 123 | Hamza | Desc | <div>ModifierSupprimer</div> |
| 3 | 75.00 | Mohamed | S'est fait le jour XX | <div>ModifierSupprimer</div> |
| 4 | Ferdaous | 13.5 | Jour YY | <div>ModifierSupprimer</div> |
| 5 | 550 | Rayan | Jour ZZ | <div>ModifierSupprimer</div> |

This simple XML CRUD project was made with ❤️ by El Mehdi El Khayati 🌐 - ENSAH GI2

Figure 1 - interface de l'application

b. Font awesome

Il s'agit juste d'un API des icons. Tous les icons itégrées dans l'application s'est appelée grâce au stylesheet font-awesome intégré avec les styles de l'application.

c. PHP

C'est le moteur de cette application, c'est le langage responsable de la passation des données à travers les requêtes \$_POST et \$_GET.

```

1  <?php
2      session_start();
3      if(isset($_POST['add'])){
4          //open xml file
5          $depenses = simplexml_load_file('files/depenses.xml');
6          $depense = $depenses->addChild('depense');
7          $depense->addChild('id', $_POST['id']);
8          $depense->addChild('montant', $_POST['montant']);
9          $depense->addChild('personne', $_POST['personne']);
10         $depense->addChild('description', $_POST['description']);
11         file_put_contents('files/depenses.xml', $depenses->asXML());
12         $_SESSION['message'] = 'Une action de dépenses a été ajoutée';
13         header('location: index.php');
14     }
15     else{
16         $_SESSION['message'] = 'Veuillez remplir la forme SVP!';
17         header('location: index.php');
18     }
19
20  ?>

```

Figure 2 - Code addition d'une dépense

d. XML

C'est l'outil de stockage dans cette application. Même s'il n'est pas puissant comme XML, mais il a montré qu'il peut jouer le rôle d'un moyen de stockage.

```

1  <?xml version="1.0"?>
2  <depenses>
3      <depense>
4          <id>2</id>
5          <montant>123</montant>
6          <personne>Hamza</personne>
7          <description>Desc</description>
8      </depense>
9      <depense>
10         <id>3</id>
11         <montant>75.00</montant>
12         <personne>Mohamed</personne>
13         <description>S'est fait le jour XX</description>
14     </depense>
15     <depense>
16         <id>4</id>
17         <montant>Ferdaous</montant>
18         <personne>13.5</personne>
19         <description>Jour YY</description>
20     </depense>
21     <depense>
22         <id>5</id>
23         <montant>550</montant>
24         <personne>Rayan</personne>
25         <description>Jour ZZ</description>
26     </depense>
27 </depenses>

```

Figure 3 - fichier xml de l'application

e. DTD

Le fichier dtd à but de vérifier si le fichier XML est bien écrit et qu'il respecte la forme choisie pour l'écriture du fichier xml et ses attributs.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!ELEMENT dépenses (depense*)>
3 <!ELEMENT depense (id, montant, personne, description)>
4 <!ATTLIST depense id ID #REQUIRED>
5 <!ELEMENT montant (#PCDATA)>
6 <!ELEMENT personne (#PCDATA)>
7 <!ELEMENT description (#PCDATA)>
8
```

Figure 4 - fichier dtd de l'application

3. Fonctionnalités

a. Ajouter une dépense

The screenshot displays a web application interface for managing expenses. A modal window titled "Add New" is open, allowing users to add a new expense. The modal contains four input fields: "ID:", "Montant:", "Personne:", and "Description:". At the bottom of the modal are "Cancel" and "Save" buttons. In the background, a table lists existing expenses with columns for "ID dépenses", "Montant", "Personne", and "Description". The table has five rows, with the first row being a header and the subsequent four rows containing data. The first data row shows an ID of 2, a Montant of 550, a Personne named Rayan, and a Description of Jour ZZ. Below the table, there are buttons for "Modifier" (Modify) and "Supprimer" (Delete) for each row. At the bottom of the page, a footer message states: "This simple XML CRUD project was made with ❤️ by El Mehdi El Khayati - ENSAH GI2".

Figure 5 - ajouter une dépense

b. Modifier une dépense

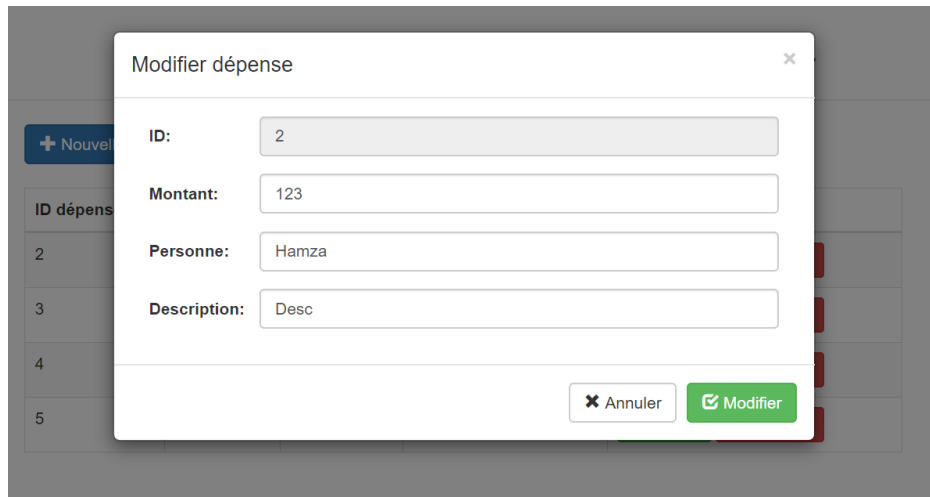


Figure 6 - modifier une dépense

c. Supprimer une dépense

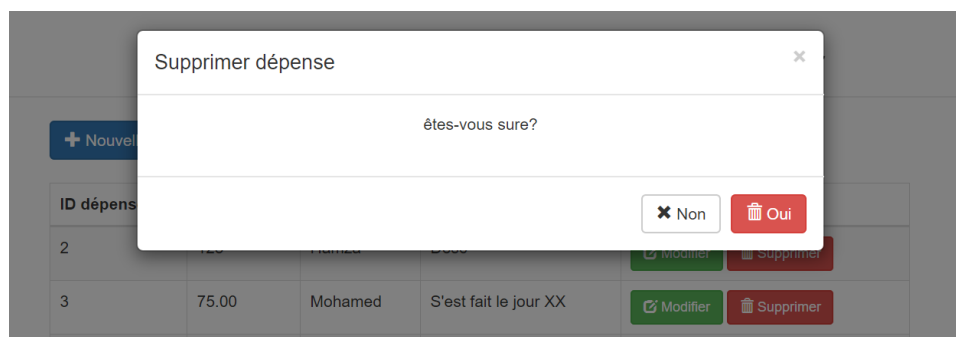


Figure 7 - supprimer dépense

Conclusion

J'espère que ce rapport a été bien clair. Ce petit projet à été fait sous l'encadrement de Mr Nouredine.

Le code source s'est uploadé dans mon GitHub @kaygi22.