

**RAPPORT DE PROJET**

**PROGRAMMATION**

**WEB**

### **Encadré par :**

* Mr. SAADI MOSTAFA

### **Réalisé par :**

* HROUCH AYMANE (N : 27)
* GHOUNDAL YOUSSEF (N : 23)

**REMERCIEMENTS**

Ont tien tout d’abord à remercier notre encadrant Mr. SAADI Mostafa et Nous souhaitons à exprimer notre gratitude aussi à avoir donné l'occasion pour faire un projet qui nous a permet d’acquérir des nouvelles connaissances et de développer nos compétences, surtout dans ce module qui contient nouveaux connaissances

# SOMMAIRE:

* Introduction
* Outils Utilisés
* Structure
* Schéma utilisé
* Front-Office
* Back-Office
* Conclusion

# INTRODUCTION :

  Notre projet consiste à la conception et la mise en œuvre d'un site web personnel d’un groupe du musique qui s’appelle ‘’Foo Fighters’’.

### C’est un site constitué de deux espaces :

* Moderateur (/mod)
* Utilisateur

L’espace Utilisateur est de 5 pages:

* Acceuil : Page d’acceuil
* Gallery : Gallerie du groupe
* Albums : Albums du groupe.
* Fans Zone : Espace consacré aux commentaires des fans.
* News : Actualités du groupe.

# Outils Utilisés :

* HTML5/CSS3
* XML
* PHP7
* La base de données MySQL
* JQUERY PLUGINS (JInvertScroll , et JQuery Modal)
* Fontawesome.

# Structure :

.

├── SQL

│   └── foofighters.sql

├── assets

│   ├── common

│   │   ├── css

│   │   ├── img

│   │   ├── js

│   │   └── xml

│   └── libs

│   ├── bootstrap-4.3.1-dist

│   ├── font-awesome-4.7.0

│   └── jquery

├── mod

│   ├── config.php

│   ├── delete.php

│   ├── edit.php

│   ├── header.php

│   ├── index.php

│   ├── login.php

│   ├── logout.php

│   ├── new.php

│   ├── reset-password.php

│   └── signup.php

├── albums.php

├── article.php

├── fans.php

├── footer.php

├── gallery.php

├── index.php

├── lyrics.php

├── navbar.php

└── news.php

# Schéma Utilisé:

<Navbar>

<header>

<Main>

<section>

<section>

<div> <div>

<footer>

# **FRONT**-OFFICE :

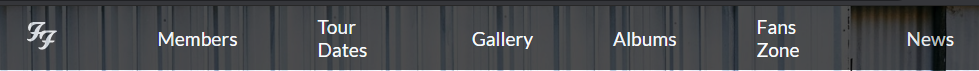
On va présenter les pages de notre sites avec explication, des lignes de notre code et des captures d’écran pour bien représenté

# Acceuil:



navbar:

<?php include("navbar.php"); ?>



Elle contient 7 **composants** :

-**Acceuil** : C’est le lien de la page d’acceuil avec le logo du groupe

<a class="logo" href="index.php">

        <img src="assets/common/img/logo.svg" alt="" />

-**Members**: Lien nous prend au la section de members( on reste dans la même page (Acceuil)

 <a href="index.php#members">Members</a>

-**Tour Dates** : Nous prend au la section de tour dates qui situe dans la même page (acceuil) aussi.

-**Gallery** : C’est la page de galerie

 <a href="gallery.php">Gallery</a>

-**Albums** : Qui contient les albums avec leurs dates.

<a href="albums.php">Albums</a>

-**Fans Zone** : zone des fans.

<a href="fans.php">Fans Zone</a>

-**News** : Actualités.

<a href="news.php">News</a>

#### CSS de navbar:

#main-navbar { #main-navbar a:link,

  height: 4rem; #main-navbar a:visited,

  width: 100%; #main-navbar a:active {

  position: fixed;   text-decoration: none;

  padding: 0 8rem;    transition: background-color 0.5s;

  font-size: rem;   margin-left: 3rem;

  background-color: #000;   color: #fff;

  opacity: 0.7;    font-size: 1.2rem;

  display: flex;   padding: 0.5rem 1rem;

  align-items: center;    border-radius: 3px;

  justify-content: center; #main-navbar a:hover {

  z-index: 1;   color: #fff;

}   border-bottom: #fff solid 2px;}

#main-navbar a img {

  width: 60%;

}

## Header :

<header id="main-presentation" class="presentation">

      <h1>Foo Fighters</h1>

    </header>

#### CSS:

#main-presentation { .presentation {

  background: url("img/mainbg.jpg")    height: 90vh;

no-repeat; width: 100%;

  background-size: cover;   justify-content: center;

  background-position: center;  align-items: center;}

}

.presentation h1 { .presentation h1:hover {

  font-size: 10rem;   cursor: none;

  color: red;  opacity: 0.1;}

  background-color: rgba(0, 0, 0, 0.1);

  width: 100%;

  text-align: center;

}

## Members:

<section id="members">

        <h1>Members</h1>

        <div>

          <div class="member">

            <img src="assets/common/img/members/1-min.jpg" alt="" />

            <h3>Dave Grohl</h3>

            <h4>Guitar</h4>

          </div>

          <div class="member">

            <img src="assets/common/img/members/2-min.jpg" alt="" />

            <h3>Rami Jaffee</h3>

            <h4>Organ</h4>

          </div>

.

.

<div class="member">

            <img src="assets/common/img/members/8-min.jpg" alt="" />

            <h3>Franz Stahl</h3>

            <h4>Lead Guitar</h4>

          </div>

        </div>

    </section>

#### CSS

#members {

  background: lineargradient(rgba(0, 0, 0, 0.541), rgba(0, 0, 0, 0.445)), url("img/membersbg.jpg");

  padding: 2.6rem 0;

}

#members div {      #members h1 {

  width: auto;   font-size: 3rem;

  display: flex;  text-align: center;

  flex-wrap: wrap;   padding: 1rem 0;

  justify-content: center;   color: #fff;

}

La partie de CSS est très long pouvez vous voir le fichier **style.css**

## Section 1:

 <section id="infos">

        <div class="text">

          <p>

            Foo Fighters is an American rock band formed in Seattle, Wa shington, in 1994. It was................

.

.

</p>

        </div>

        <img src="./assets/common/img/dave.jpg" alt="" />

      </section>

## Tour dates:



<section id="tour-dates">

        <div class="table">

          <h2>Tour Dates</h2>

          <table>

            <tr>

              <td>07/29/21</td>

              <td>

                LOLLAPALOOZA<br />

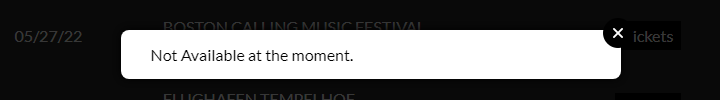
                CHICAGO, US

              </td>

              <td><a href="#modal" rel="modal:open">Tickets</a></td>

            </tr>

Lorsque on clique sur button ‘’tickets’’ il affiche un message de la disponibilité du ticket



<div id="modal" class="modal">

      <p>Not Available at the moment.</p>

    </div>

## Footer:

<?php include("footer.php") ?>



<footer>

      <div>

        <a href="https://www.facebook.com/foofighters/">

          <i class="fa fa-facebook-official" aria-hidden="true"></i>

        </a>

        <a href="https://twitter.com/foofighters"><i class="fa fa-twitter" aria-hidden="true"></i></a>

        <a href="https://www.instagram.com/foofighters/"><i class="fa fa-instagram" aria-hidden="true"></i></a>

      </div>

      <p>Copyright Reserved - Foo Fighters <?php echo date("Y");?></p>

</footer>

# Gallery:

Dans cette page on a utilisé **Javascript** pour apporter les images.

Les noms et les légendes de ces images sont stockées dans un fichier **XML(*gallery.xml*)**, qui ressemble à ça:

<?xml version="1.0"?>

<gallery>

    <image>

        <caption>légende1</caption>

        <src>1.jpg</src>

    </image>

    <image>

        <caption>légende2</caption>

        <src>2.jpg</src>

    </image>

    <image>

        <caption>légende3</caption>

        <src>3.jpg</src>

    </image>

</gallery>

et en **javascript** nous avons utilisé [*XMLHttpRequest()*](#) comme suivant:

[// On obtient depuis le fichier XML src et caption](#)

[let xhr = new XMLHttpRequest();](#)

[xhr.open("GET", "./assets/common/xml/gallery.xml");](#)

[xhr.onreadystatechange = function () {](#)

et on procéde à apporter les images sur le fichier HTML avec ses légendes.

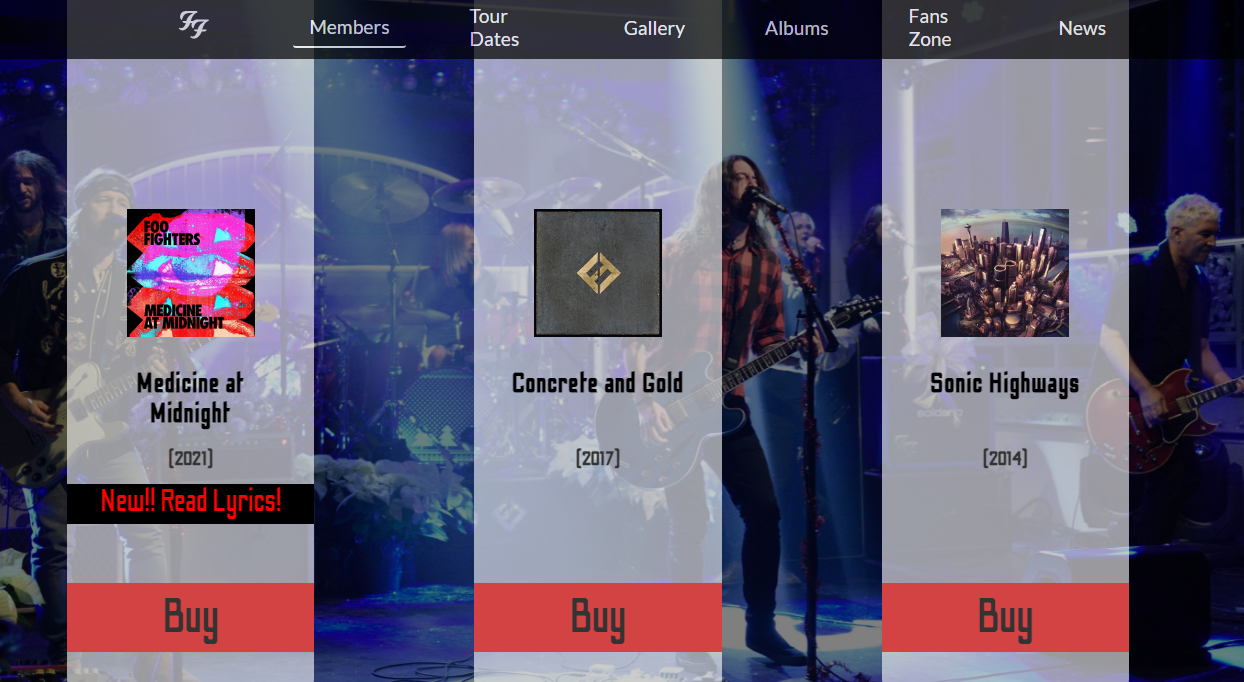
# Albums:

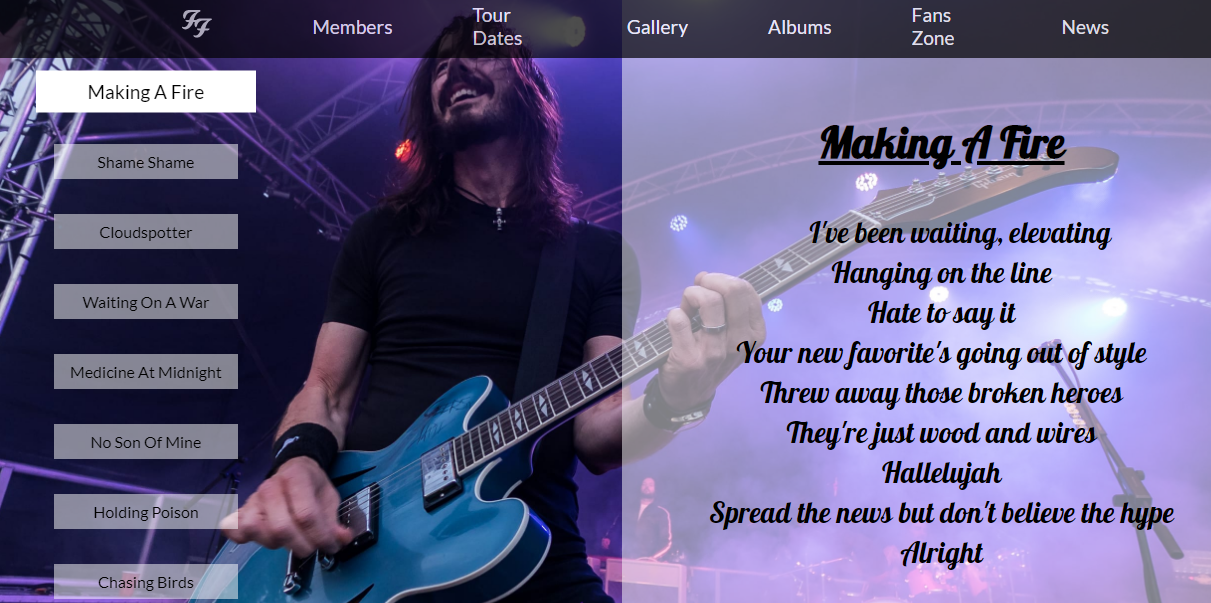
C’est une page qui expose les albums.

Cette page utilise deux JQuery plugins:

**JQueryModal** qui offre une fenêtre contextuelle JavaScript légère et polyvalente, personnalisable et réactive. Ils peuvent être utilisés pour afficher des fenêtres contextuelles d'alerte, des vidéos et des images sur un site Web.

**JInvertScroll :**C'est un plugin léger pour jQuery qui vous permet de vous déplacer horizontalement avec un effet de parallaxe tout en faisant défiler vers le bas.

Le utilisateurs peuvent accéder la pages des paroles du dernier album «Medicine at Midnight» en cliquant sur **‘’New! Read Lyrics!’’.**

**Lyric****s :**

Dans Cette page on a utilisé des fonctions de **javascript** pour utiliser le fichier **XML** qui contient les titres et lyrics des chansons.

# BACK-OFFICE :

* *config.php*

D’abord, on a le fichier **config.php** qui est responsable à se connecter au base de donnée en utilisant **PDO**.

Ce fichier va être inclus dans tout les fichiers qui ont besoin d’établir une connexion avec la base de données, on l’inclut en utilisant *require\_once.*

<?php

/\* Database credentials \*/

define('DB\_SERVER', 'localhost');

define('DB\_USERNAME', 'root');

define('DB\_PASSWORD', '');

define('DB\_NAME', 'foofighters');

Pour se connecter:

try{

    $pdo = new PDO("mysql:host=" . DB\_SERVER . ";dbname=" . DB\_NAME, DB\_USERNAME, DB\_PASSWORD);

    // Set the PDO error mode to exception

    $pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch(PDOException $e){

    die("ERROR: Could not connect. " . $e->getMessage());

}

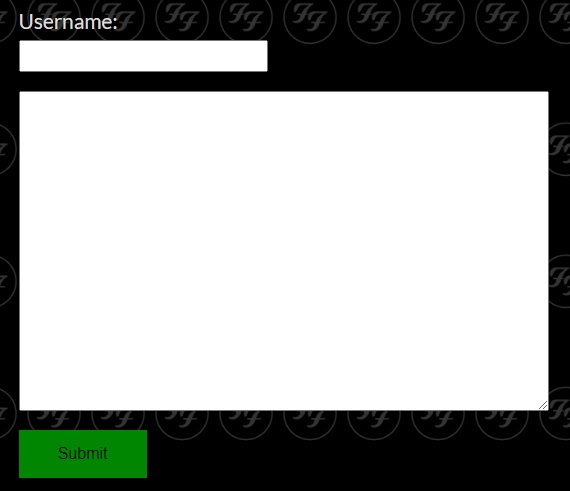
?>

*Avec affichage d’un message d’erreur si quelque chose ne marche pas par  die("ERROR: Could not connect. " . $e->getMessage());*

# Fanszone:

C’est une page dans laquelle les fans peuvent laisser des commentaires.

Pour ça l’utilisateur peut utiliser une formulaire, où il doit écrire son nom et son commentaire.

Voilà le code de cette formulaire.

On donne comme valeur à l’attribut “action” le chemin de ce fichier lui-même.

<form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="post">

        <label for="user">Username:</label>

        <input type="text" name="user" id="user" />

        <textarea name="msg" id="msg" cols="30" rows="10"></textarea>

        <input type="submit" />

      </form>

On a utilisé javascript,

On conserve le nom de l'utilisateur dans le localStorage, pour faciliter l'utilisation future.

Donc si l'utilisateur veut écrire un autre message, ça va être suffisant d'écrire seulement le message par ce que le champ de l'utilisateur va contenir le dernier nom utilisé.

Pour ça on doit écouter le champ du nom d'utilisateur.

<script>

      let userField = document.getElementById("user");

      if (localStorage.user) {

        userField.value = localStorage.user;

      }

      userField.addEventListener("input", () => {

        localStorage.user = userField.value;

      });

</script>

Au haut de ce fichier en vérifie donc la méthode de request,

if($\_SERVER["REQUEST\_METHOD"] === "POST")

si c’est « POST » on va ajouter traiter les données que l’utilisateur a envoyé depuis cette formulaire.

Puis il verifie si l’utilisateur a entré quelque chose si oui il effectue une insertion à la base de données:

// Vérification si l'utilisateur a entré quelque chose

  if (strlen($\_POST['user']) >= 2 and strlen($\_POST['msg']) >= 1 ) {

  // Insert values to DB

  $que = $pdo->prepare('INSERT INTO comments(user, msg) VALUES(:user, :msg)');

  $que->execute(array(

    'user' => $\_POST['user'],

    'msg' => $\_POST['msg'],

    ));

  }

Les 10 derniers commentaires sont affichés dans la même page :

Pour **l’affichage** des commentaires on fait une requete SQL, alors tout d’abord on va inclure le fichier *config.php* en utisant *require\_once()*.

require\_once("mod/config.php");

$answer = $pdo->query('SELECT id, user, msg, date\_format(date,\'[%d/%m/%Y/%H:%i:%s]\') AS date FROM comments ORDER BY id DESC LIMIT 0, 10');

Puis afficher en utilisant une boucle while:

while ($infos = $answer->fetch()) {

        ?>

      <div class="message">

        <span class="user"><?php echo htmlspecialchars($infos['user']) ? ></span>

        <p>

            <?php echo $infos['msg'] ?>

        </p>

        <span class="date"><?php echo $infos['date'] ?></span>

      </div>

# News:

C’est une page sur laquelle on trouve les actualités de **foofighters,** Seulement les **admins** peuvent manipuler cette page.

Les articles sont afficher par une requête:

$result = $pdo>query('SELECT n.id, u.name as author, n.title, n.article,date\_format(n.created\_at,\'%d/%m/%Y %H:%i\') AS date FROM news n, user u WHERE n.author = u.id ORDER BY n.created\_at DESC LIMIT 0, 10');

Puis afficher en utilisant une boucle while:

while ($data = $result->fetch()) {

        ?>

      <div class="message">

        <a href="<?php echo "article.php?id=". $data["id"] ?>" class="article-title"><?php echo htmlspecialchars($data['title']) ?></a>

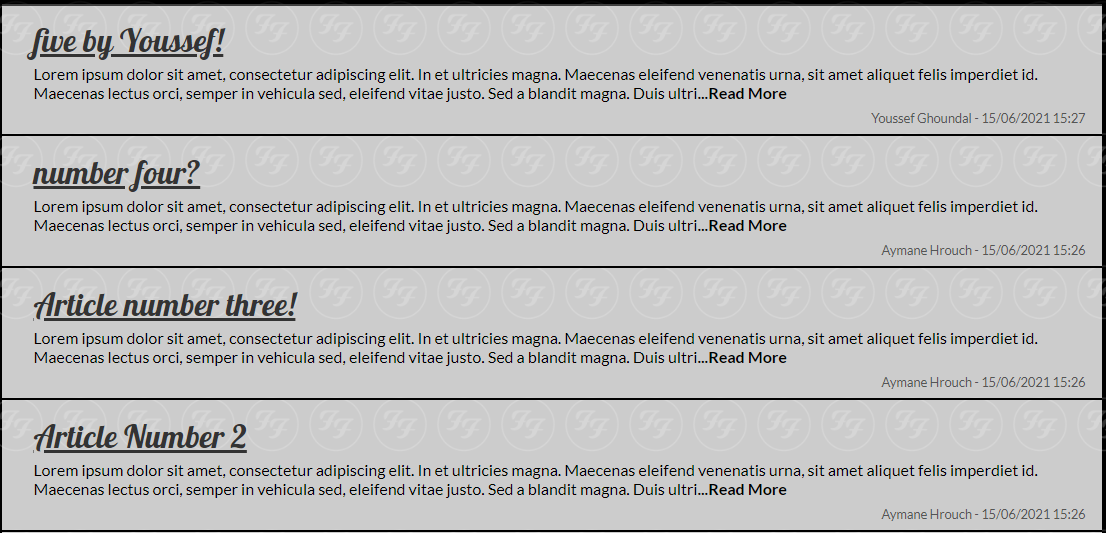
        <p>

            <?php echo substr(htmlspecialchars($data['article']), 0, 250) . "<a class='read-more' href=\"article.php?id=" . $data["id"] . "\">...Read More</a>" ?>

        </p>

        <span class="date"><?php echo $data["author"] . " - " . $data['date'] ?></span>

      </div>



Comment peuvent les admins manipuler ces article ?

Nous avons crée un espace admin.

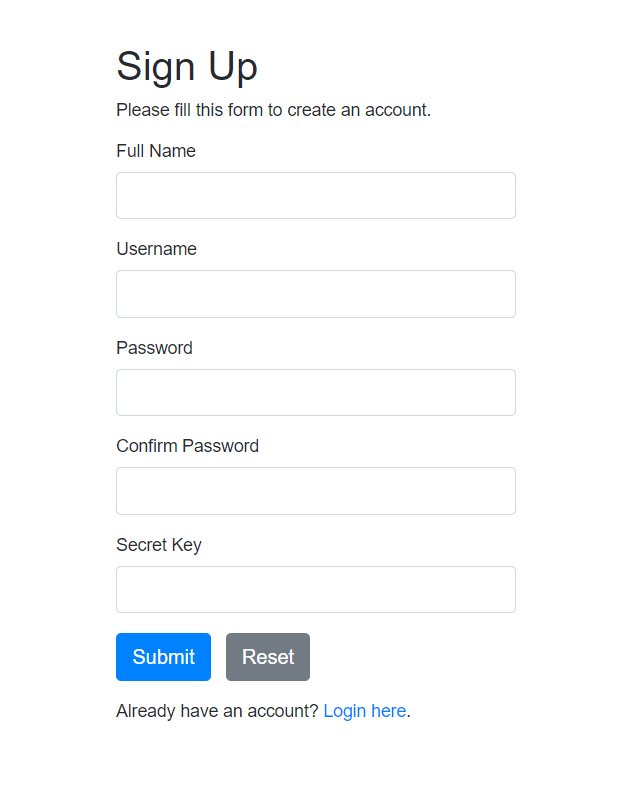
Cet espace est dans un dossier mod/

il contient les fichier suivant :

* config.php
* delete.php
* edit.php
* header.php
* index.php
* login.php
* logout.php
* new.php
* reset-password.php
* signup.php

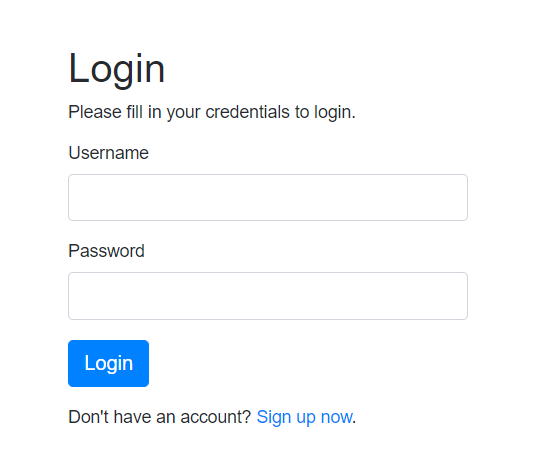
On éxplique les pages :

* signup.php

Une page qui contient une formulaire pour créer un compte comme admin du site web, pour empêcher quelconque de créer un compte l’utilisateur doit savoir le « secret key » qui est « ENSAKHOURIBGA»

* login.php

Une page qui contient une formulaire pour l’authentification



* index.php

La page principle, contient la liste des articles.

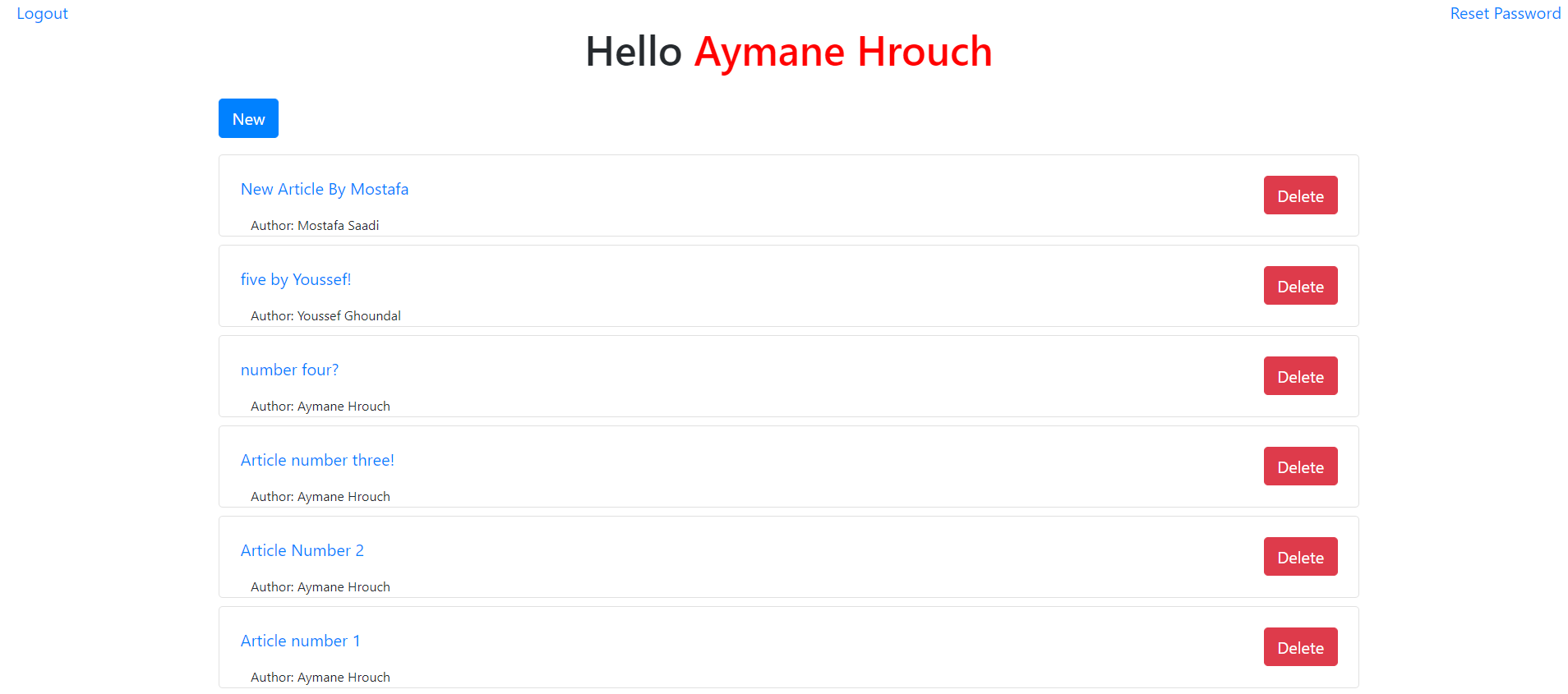
Si on clique la bouton « New » on ouvre la page new.php

Si on clique un article on ouvre la page edit.php?id=x.

Si on clique « Delete » on ouvre la page delete.php?id=x

Si on clique « logout » au haut à gauche, on va se déconnecter.

avec ‘x’ la valeur de ID de l’article.



Les deux pages new.php et edit.php?id=x on la même structure, la seule différence c’est que les champs dans edit.php?id=x sont déjà remplis par les données de l’article dont l’id est ‘x’

edit.php?id=39

Dans les paragraphes suivantes, nous expliquons les pages du dossier mod/ en bref.

pour login.php

on vérifie si l’utilisateur est légitime, si oui on va initialiser une session et des variables de sessions, puis on réorienter l’utilisateur à la page index.php :

                            // Password is correct, so start a new session

                            session\_start();

                            // Store data in session variables

                            $\_SESSION["loggedin"] = true;

                            $\_SESSION["id"] = $id;

                            $\_SESSION["username"] = $username;

                            $\_SESSION["name"] = $row["name"];

                            // Redirect user to welcome page

                            header("location: index.php");

On commence dans toutes les autres pages par initialiser une session

// Initialize the session

session\_start();

pour les pages qui a besoin de l’authentification(index.php, new.php, edit.php, delete.php, logout.php), on vérifie si l’utilisateur s’a déjà connecté, si non on le réoriente vers la page login.php

    if(!isset($\_SESSION["loggedin"])){

        header("location: login.php");

        exit;

    }

Dans la page index.php on affiche la liste juste comme on a déjà fait dans new.php.

Dans les pages new.php et edit.php on vérifie si les champs sont accepté, les titres doit être au minimum d’une longueur 5 au minimum, les article d’un longueur 120. Si oui on va utiliser une commandes SQL pour insérer ( dans le cas de new.php ) ou modifier (dans le cas de edit.php )

La page delete.php?id=x peut supprimer l’article dont le id est ‘x’, en utilisant la commande DELETE de SQL :

require\_once 'config.php';

if(isset($\_GET["id"])) {

    $sql = "DELETE FROM news WHERE id=" . $\_GET["id"];

    $pdo->query($sql);

}

header("location: index.php")

La page logout.php a un rôle de se déconnecter, elle fait ça par détruire la session.

<?php

// Initialize the session

session\_start();

// Unset all of the session variables

$\_SESSION = array();

// Destroy the session.

session\_destroy();

// Redirect to login page

header("location: login.php");

exit;

?>

# Conclusion:

Tout au long de la préparation de notre projet, nous avons essayé de mettre en pratique les connaissances acquises durant la deuxième semestre dans le module développement web, et cela dans le but de réaliser un site réaliste et tendant le plus possible vers un rendu de qualité professionnelle.

Nous avons solidifié nos connaissances et notre approche à la programmation web.