Compte-rendu hebdomadaire

Le principe de la Mairie de Saint-André

- ***1ère Semaine***

1er jour :

Cahier des charges

Service informatique de la mairie de de Saint-André

I-Présentation

Ce projet a pour objectif de créer un site web pour faire les arrêtés de chaque service qui vont être stockés en PDF et qu'on pourra consulter.

II-Fonctionnalités (MVP)

– -Créer les arrêtés.

– -Consulter les données sur les arrêtés.

– -Créer une authentification pour accéder au site.

III-Visuel

page d'authentification :

Dans ce style là.

page pour créer un arrêté

Dans ce style là, avec les informations nécessaires et correspondant aux arrêtés.

Adresse postale

Pour mettre à jour notre liste de diffusion, nous avons besoin que vous remplissiez ce formulaire

Titre

Please Select

Nom

Prénom

Nom de famille

Email

ex: myname@example.com

example@example.com

Numéro de téléphone

(000) 000-0000

Adresse

N° et rue

Ligne d'adresse additionnelle

Ville

État/Région

Code Postal

Soumettre

IV-Optimisation

– -Consulter l'historique des activités.

– -Créer des filtres et une barre de recherche pour faciliter la consultation.

– -Créer une mise en forme du site.

V-Délais

Pour terminer le MVP : 2 semaines / 2 semaines et demie.

Pour terminer le MVP et les optimisations : 3 semaines / 3 semaines et demie.

Logiciel à installer et autre pour la mission.

– Compte pour pouvoir effectuer la mission et notre stage.

– -InstalléeVisual Studio Code (VS Code).

– Installation de Wamp.

Faire la mise en œuvre de la mission.

– Création de la page du formulaire pour les arrêter.

– Création de la page du login.

– Code pour prendre et enregistrer les données du formulaire.

– Création de la page PHP consultation (pour avoir l'historique des données entrées).

– -Ajout de mise en forme (style CSS).

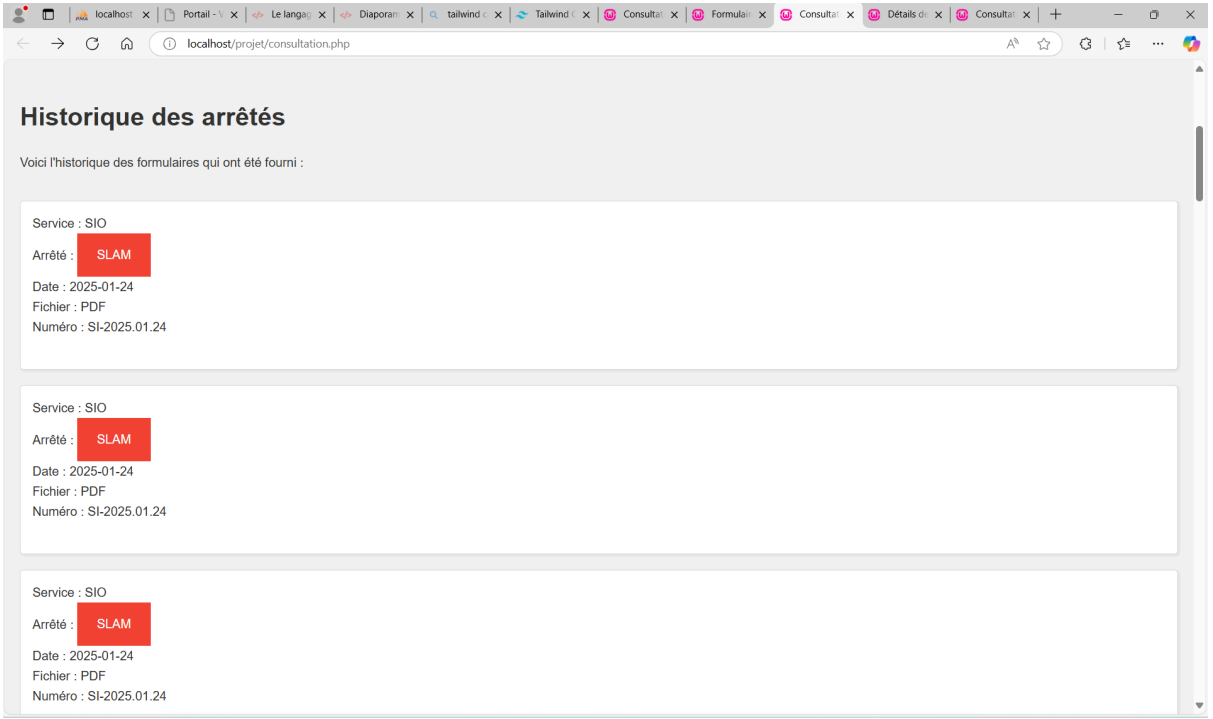
Cette semaine, j'ai pu créer le formulaire avec un début d'historique, en ajoutant du code CSS pour ajouter une mise en forme du site et la création de la base de donnée. (Je n'ai pas pris de screen du début de mon travail dans la première semaine.

❖ *2ème Semaine*

Cette semaine, j’ai pu faire la page d'historique et j'ai ajouté du code CSS pour ajouter une mise en forme de la page , ajouter une barre de navigation , ajouter un filtre dans l’historique.

Page d’historique

Cette semaine, j’ai pu faire la page d'historique et j'ai ajouté du code CSS pour ajouter une mise en forme de la page.



Au début, j’avais un problème. La page d’historique me montre l’historique de session, ce qui fait que quand je me suis connecté le lendemain, il n’y avait plus d’historique, alors que dans la base de données, l’historique était toujours présent .

Pour cela, j’ai dû faire un lien avec la base de données pour que, quand l’ utilisateur entre les informations demandées dans le formulaire, ça s’envoie dans la base de données qui le stockera, puis la page d’historique se connectera à la base de données pour avoir l'historique des anciens formulaires remplis.

Coté programmation (Page d’historique)

*****screen du code de la 4ème/3ème semaine et le reste des screen en dessous*****

consultation.php (page d’historique)

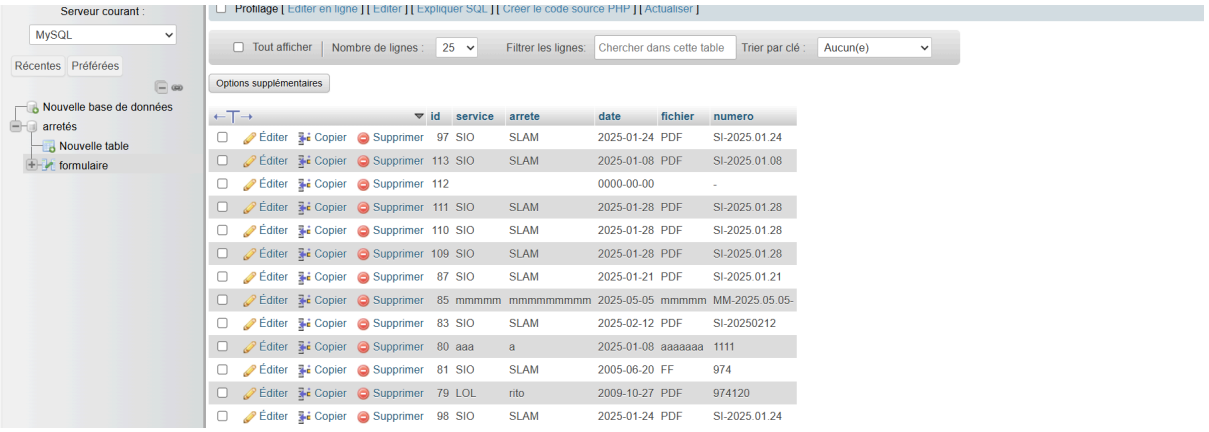
consultation.php > html > body

```html
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <title>Consultation de l'historique</title>
    <link rel="stylesheet" href="styles_2.css">
</head>

<body>

<div class="nav-links">
        <div1 class="left-links">
            <a href="http://localhost/projet/formulaire.php" target="_blank">Formulaire</a>
            <a href="http://localhost/projet/consultation.php" target="_blank">Consulter l'historique</a>
        </div1>
        <div1 class="right-links">
            <a href="logout.php" class="logout">Se déconnecter</a>
        </div1>
    </div>

    <br><br>

    <!-- Un formulaire qui permet de filtrer l'historique (service, libelle, numéro) -->
    <form method="GET" action="">
    <h1 id="mid">Filtrer l'historique</h1>
    <br>

    <label for="service">Filtrer par service:</label>
    <select name="service" id="service">
        <option value="">-- Sélectionner un service --</option>
        <option value="service1" <?php echo (isset($_GET['service']) && $_GET['service'] == 'service1') ? 'selected' : ''; ?>>Service 1</option>
        <option value="service2" <?php echo (isset($_GET['service']) && $_GET['service] == 'service2') ? 'selected' : ''; ?>>Service 2</option>
        <option value="service3" <?php echo (isset($_GET['service']) && $_GET['service] == 'service3') ? 'selected' : ''; ?>>Service 3</option>

    </select>

    <br><br>

    <input type="text" name="libelle" placeholder="Filtrer par libelle..." value="<?php echo isset($_GET['libelle']) ? htmlspecialchars($_GET['libelle']) : ''; ?>">
    <input type="text" name="numero" placeholder="Filtrer par numéro..." value="<?php echo isset($_GET['numero']) ? htmlspecialchars($_GET['numero']) : ''; ?>"><br>

    <input type="submit" value="Filtrer">
</form>


    <br><br>
    <h1>Historique des libelles</h1>

    <p>Voici l'historique des formulaires qui ont été fourni :</p>
    <br>
```

```php
<?php
session_start();

// Connexion à la base de données
$servername = "localhost";
$username = "noah";
$password = "1511";
$dbname = "arretés";

// Créer la connexion
$conn = new mysqli($servername, $username, $password, $dbname);

// Vérifie la connexion
if ($conn->connect_error) {
    die("Connexion échouée : " . $conn->connect_error);
}

// Récupères les filtres
$service_filter = isset($_GET['service']) ? $_GET['service'] : '';
$libelle_filter = isset($_GET['libelle']) ? $_GET['libelle'] : '';
$numero_filter = isset($_GET['numero']) ? $_GET['numero'] : '';

// Construire la requête SQL avec des filtres
$sql = "SELECT * FROM formulaire WHERE 1=1";

if (!empty($service_filter)) {
    $sql .= " AND service LIKE '%" . $conn->real_escape_string($service_filter) . "%'";
}

if (!empty($libelle_filter)) {
    $sql .= " AND libelle LIKE '%" . $conn->real_escape_string($libelle_filter) . "%'";
}

if (!empty($numero_filter)) {
    $sql .= " AND numero LIKE '%" . $conn->real_escape_string($numero_filter) . "%'";
}

$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while ($entree = $result->fetch_assoc()) {
        echo '<div>';
        echo 'Utilisateur : ' . htmlspecialchars($entree['utilisateur']) . '<br>';
        echo 'Service : ' . htmlspecialchars($entree['service']) . '<br>';
        echo 'Libelle : ' . htmlspecialchars($entree['libelle']) . '<br>';
        echo 'Date : ' . htmlspecialchars($entree['date']) . '<br>';
        echo 'Fichier : <a href="detail.php?id=' . $entree['id'] . '" class="small-button">' . htmlspecialchars($entree['fichier']) . '</a><br>';
        echo 'Numéro : ' . htmlspecialchars($entree['numero']) . '<br>';
        echo '<br>';
        echo '</div>';
    }
} else {
    echo 'Aucune donnée d\'historique disponible.<br>';
}
$conn->close();
?>

<a href="http://localhost/projet/signalement.html" target="_blank">Signaler un problème</a>
</body>
</html>
```

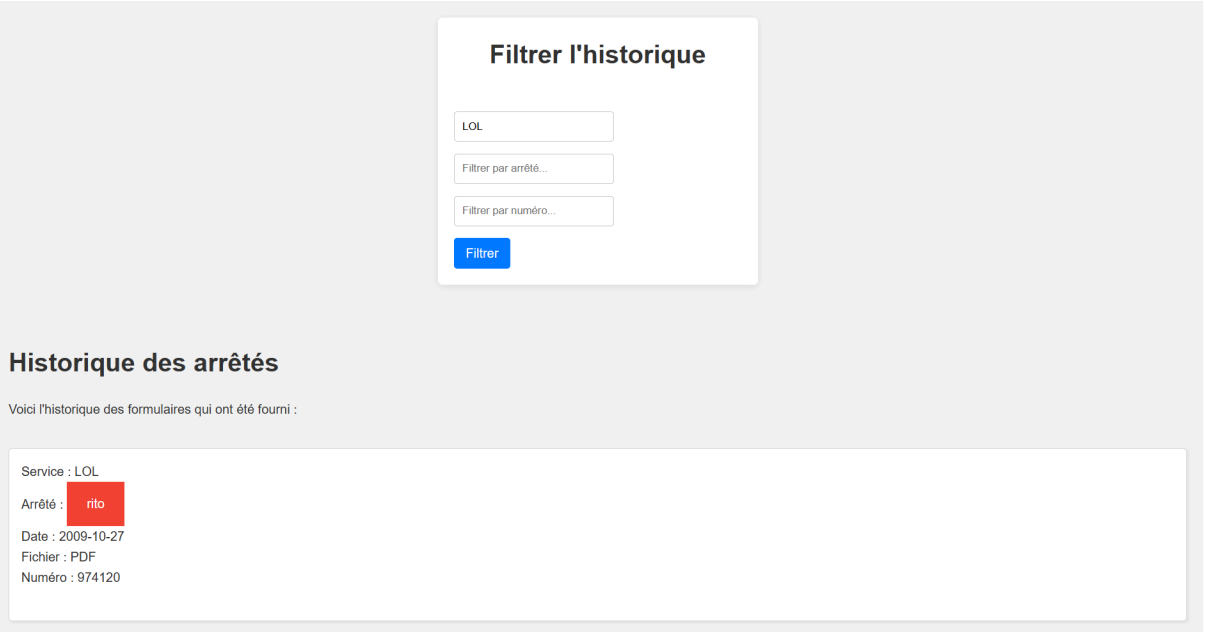
Base de donnée

Ici, on peut voir là où sont stockées les informations émises par le formulaire, et c’est ici que la page d’historique vient récupérer les informations pour les afficher.



Page d’historique (filtre)

J’ai aussi créé un filtre par service, libellé et numéro. Ce qui permet de rapidement trouver ce qui nous intéresse.



Coté programmation (Page d'historique "focus sur le filtre")

consultation.php au niveau de l'HTML (page d'historique)

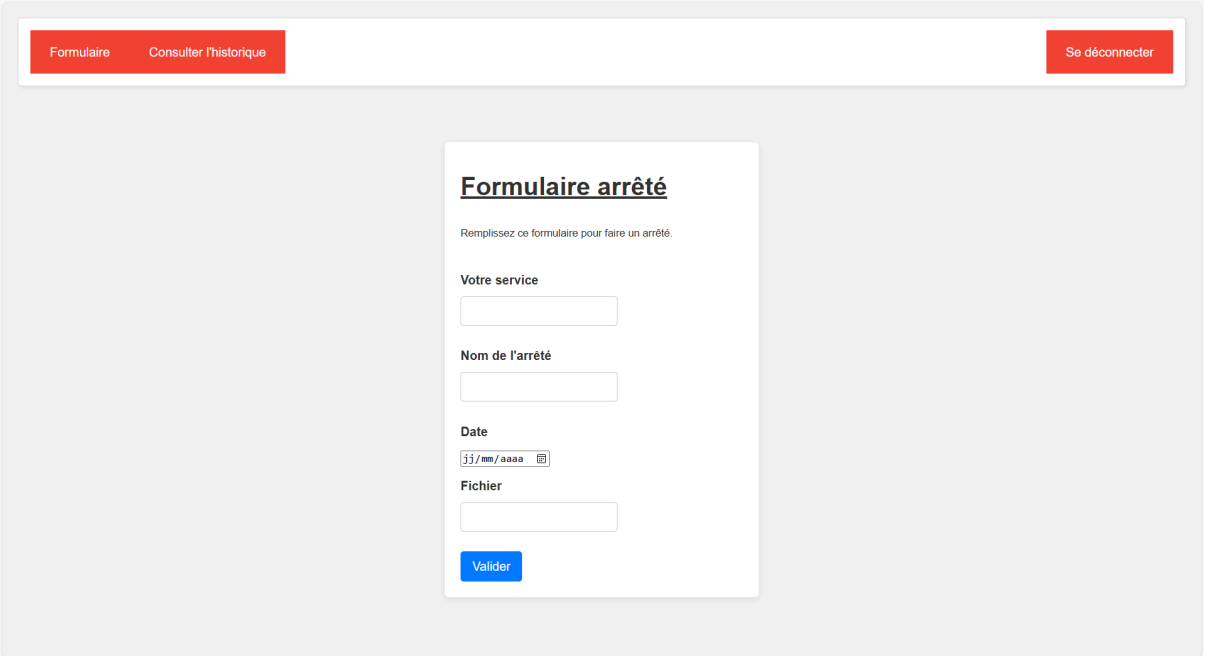
```html

    <!-- Un formulaire qui permet de filtrer l'historique (service, libelle, numéro) -->
    <form method="GET" action="">
    <h1 id="mid">Filtrer l'historique</h1>
    <br>

    <label for="service">Filtrer par service:</label>
    <select name="service" id="service">
        <option value="">-- Sélectionner un service --</option>
        <option value="service1" <?php echo (isset($_GET['service']) && $_GET['service'] == 'service1') ? 'selected' : ''; ?>>Service 1</option>
        <option value="service2" <?php echo (isset($_GET['service']) && $_GET['service'] == 'service2') ? 'selected' : ''; ?>>Service 2</option>
        <option value="service3" <?php echo (isset($_GET['service']) && $_GET['service'] == 'service3') ? 'selected' : ''; ?>>Service 3</option>

    </select>

    <br><br>

    <input type="text" name="libelle" placeholder="Filtrer par libelle..." value="<?php echo isset($_GET['libelle']) ? htmlspecialchars($_GET['libelle']) : ''; ?>">
    <input type="text" name="numero" placeholder="Filtrer par numéro..." value="<?php echo isset($_GET['numero']) ? htmlspecialchars($_GET['numero']) : ''; ?>"><br>

    <input type="submit" value="Filtrer">
</form>

```

consultation.php au niveau du PHP (page d'historique)

```php
    session_start();

    // Connexion à la base de données
    $servername = "localhost";
    $username = "noah";
    $password = "1511";
    $dbname = "arretés";

    // Créer la connexion
    $conn = new mysqli($servername, $username, $password, $dbname);

    // Vérifie la connexion
    if ($conn->connect_error) {
        die("Connexion échouée : " . $conn->connect_error);
    }

    // Récupères les filtres
    $service_filter = isset($_GET['service']) ? $_GET['service'] : '';
    $libelle_filter = isset($_GET['libelle']) ? $_GET['libelle'] : '';
    $numero_filter = isset($_GET['numero']) ? $_GET['numero'] : '';

    // Construire la requête SQL avec des filtres
    $sql = "SELECT * FROM formulaire WHERE 1=1";

    if (!empty($service_filter)) {
        $sql .= " AND service LIKE '%" . $conn->real_escape_string($service_filter) . "%'";
    }

    if (!empty($libelle_filter)) {
        $sql .= " AND libelle LIKE '%" . $conn->real_escape_string($libelle_filter) . "%'";
    }

    if (!empty($numero_filter)) {
        $sql .= " AND numero LIKE '%" . $conn->real_escape_string($numero_filter) . "%'";
    }

    $result = $conn->query($sql);


```

Page du formulaire (barre de navigation)

J'ai aussi créé une barre de navigation dans la page formulaire qui permet de naviguer de page en page et qui se retrouve aussi dans la page historique. (Voici à quoi ressemble mon formulaire.)



Coté programmation (Page formulaire "focus sur la barre de navigation")

formulaire.php (page du formulaire)

formulaire.php > html > body > form > p

```html
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8" />
    <title>Formulaires d'arrêté</title>
    <link rel="stylesheet" href="styles_2.css">
</head>

<body>

    <div class="nav-links">
        <div1 class="left-links">
            <a href="http://localhost/projet/formulaire.php" target="_blank">Formulaire</a>
            <a href="http://localhost/projet/consultation.php" target="_blank">Consulter l'historique</a>
        </div1>
        <div1 class="right-links">
            <a href="logout.php" class="logout">Se déconnecter</a>
        </div1>
    </div>

```

dashboard.php (bouton pour se déconnecter)

```php
<!DOCTYPE html>
<html lang="fr">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tableau de bord</title>
</head>

<body>

    <?php
    session_start();

    // Vérifiez si l'utilisateur est connecté
    if (!isset($_SESSION['user'])) {
        header("Location: login.php"); // Redirige vers la page de connexion si non connecté
        exit();
    }

    $userEmail = $_SESSION['user']['email'];
    ?>

    <h1>Bienvenue, <?php echo $userEmail; ?>!</h1>
    <p><a href="logout.php">Se déconnecter</a></p>
</body>

</html>
```

❖ 3ème Semaine

Ils m'ont demandé des choses en plus dans le cahier des charges.

Update :

– Créer un historique sur la page formulaire où on voit les 10 derniers formulaires remplis.

– Créer un login admin.

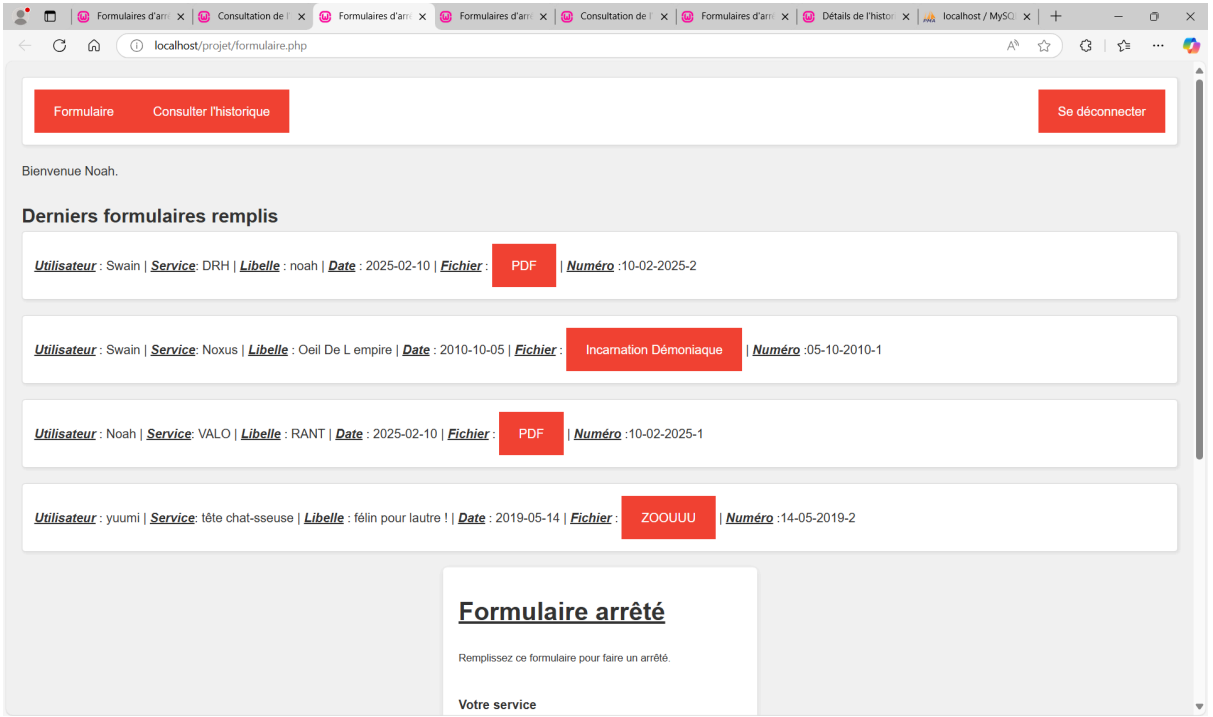
– Pour le numéro de l'arrêté (date – incrémentation d'un nombre).

– -créer un bouton de signalement ;

– voir quel utilisateur a fait le formulaire dans l'historique ;

Page du formulaire(10 derniers formulaire remplis)

On m'a demandé de créer un historique des 10 derniers formulaires qui doit être fait en haut de la page du formulaire pour que les utilisateurs puissent voir rapidement ce qui a déjà été fait. (j'ai tester pour voir si sa marchais)



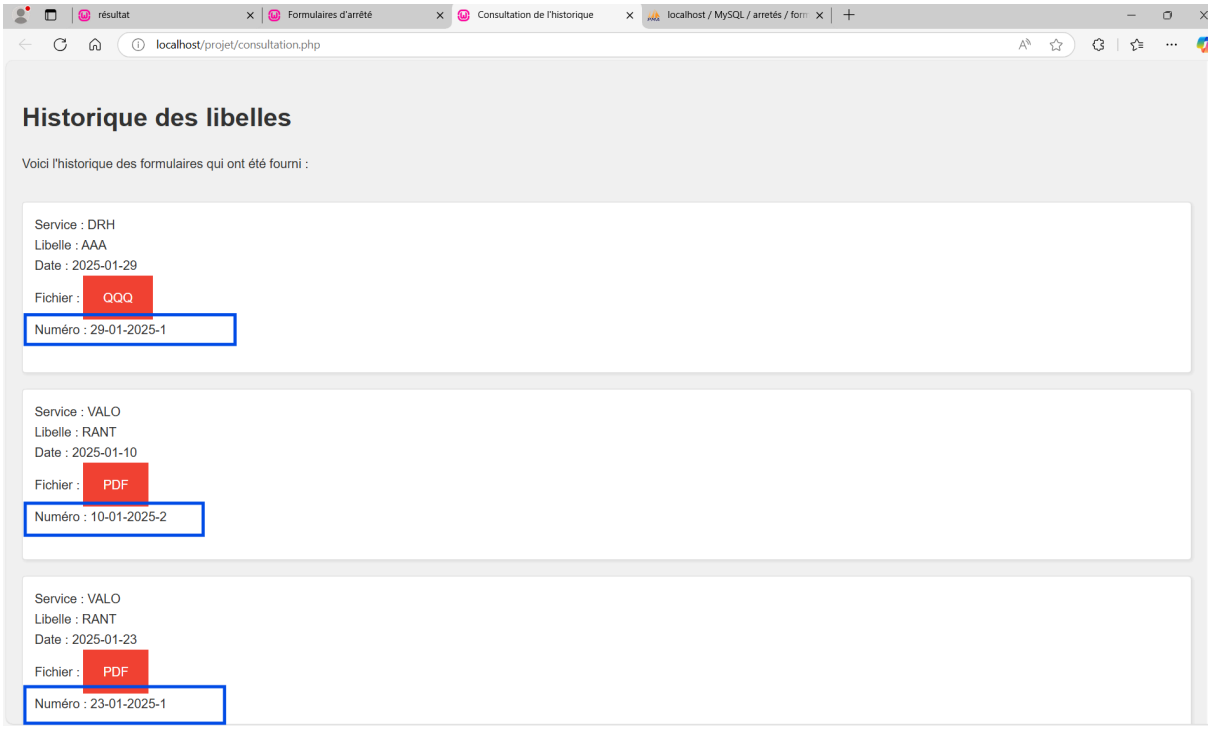
Coté programmation (Page formulaire “focus sur les 10 derniers formulaire remplis”)

formulaire.php (page du formulaire)

```php
<?php session_start(); // Démarrer la session
echo 'Bienvenue '.$_SESSION['user']['utilisateur'].'.'; ?>
<br><br>

<?php

// Connexion à la base de données
$servername = "localhost";
$username = "noah";
$password = "1511";
$dbname = "arretés";

// Créer la connexion
$conn = new mysqli($servername, $username, $password, $dbname);

// Vérifier la connexion
if ($conn->connect_error) {
    die("Connexion échouée : " . $conn->connect_error);
}

// Récupérer les 10 derniers formulaires
$sql = "SELECT * FROM formulaire ORDER BY id DESC LIMIT 10";
$result = $conn->query($sql);
?>

<h2>Derniers formulaires remplis</h2>
<?php


// Vérifier si des données existent
if ($result->num_rows > 0) {
    while ($entree = $result->fetch_assoc()) {
        echo '<div>';
        echo '<b><i><u>Utilisateur</u></i></b> : ' . htmlspecialchars($entree['utilisateur']) . "   |     ";
        echo ' <b><i><u>Service</u></i></b>:  ' . htmlspecialchars($entree['service']) . "   |     ";
        echo ' <b><i><u>Libelle</u></i></b> : ' . htmlspecialchars($entree['libelle']) . "   |     ";
        echo ' <b><i><u>Date</u></i></b> : ' . htmlspecialchars($entree['date']) . "     |   ";
        echo ' <b><i><u>Fichier</u></i></b> : <a href="detail.php?id=' . $entree['id'] . '">' . htmlspecialchars($entree['fichier']) . '</a>' . "    |    ";
        echo ' <b><i><u>Numéro</u></i></b> :' . htmlspecialchars($entree['numero']) . "       ";
        echo '</div>';
    }
} else {
    echo 'Aucun formulaire rempli récemment.<br>';
}

$conn->close();
?>
```

Page de consultation(numéro automatique)

Puis j'ai pu réaliser la création de numéro automatique lors d'un remplissage de formulaire qui comprend la date et une incrémentation. (ex. : 27-01-2025-7).



Coté programmation (Page résultat “focus sur les numéro automatique”)

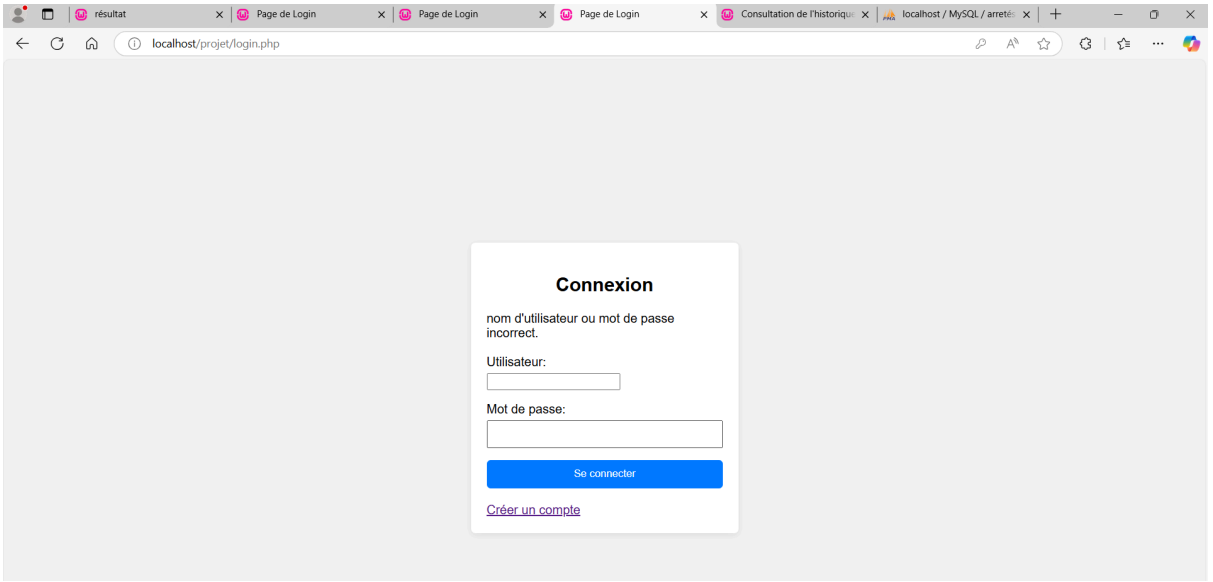
formulaire.php (page du formulaire)

```
formulaire.php   resultat.php   consultation.php   envoie_email.php   login.php   signalement.html   detail.php   admin_page.php   register.php   dashboard.ph
resultat.php > html > body
<html lang="fr">
<body>
        $dateForCount = date("Y-m-d", strtotime($date)); // Pour la requête
    } else {
        $dateFormatted = "";
    }

    // Prépare la requête SQL pour compter les formulaires déjà soumis pour cette date
    $countSql = "SELECT COUNT(*) as count FROM formulaire WHERE date = '$dateForCount'";
    $countResult = $conn->query($countSql);
    $countRow = $countResult->fetch_assoc();
    $incrementNumber = $countRow['count'] + 1; // Incrémenter le nombre

   // Ajoutez cette ligne pour récupérer l'utilisateur de la session
$utilisateur = $_SESSION['user']['utilisateur'];

// Prépare la requête SQL pour insérer les données dans la base
$sql = "INSERT INTO formulaire (service, libelle, date, fichier, utilisateur) VALUES ('$service', '$libelle', '$dateForCount', '$fichier', '$utilisateur')";



// Après avoir réussi à insérer les données dans la base de données
if ($conn->query($sql) === TRUE) {
    // Génère le numéro : date + numéro incrémenté
    $numero = $dateFormatted . '-' . $incrementNumber;

    // Insère le numéro dans la base de données
    $sql = "UPDATE formulaire SET numero = '$numero' WHERE id = " . $conn->insert_id;

    // Exécute la requête de mise à jour
    $conn->query($sql);

    // Affiche le message de succès
    echo '<div>';
    echo "<b><i><u>Les données ont été enregistrées avec succès.</u></i></b>"  . "<br><br>";

    // Ajoute les données dans la session
    $_SESSION['historique'][] = [
        'service' => $service,
        'libelle' => $libelle,
        'date' => $date,
        'fichier' => $fichier,
        'numero' => $numero
    ];
} else {
    echo "Erreur : " . $sql . "<br>" . $conn->error;
}
```

Puis j’ai fait fonctionner le login pour que les personnes puissent créer un compte puis se logger avec leur compte pour accéder au formulaire, et j’ai créé aussi un compte admin qui l’envoyera sur une page que seul l’admin peut atteindre. Puis on m'a dit de changer l’email par utilisateur pour se connecter.

Page de connexion

On peut voir ici que si on rentre un utilisateur ou un mot de passe incorrect, il y aura un message d'erreur.



Coté programmation (Page connexion“focus sur la connexion d’un utilisateur ”)

login.php (page de connexion)

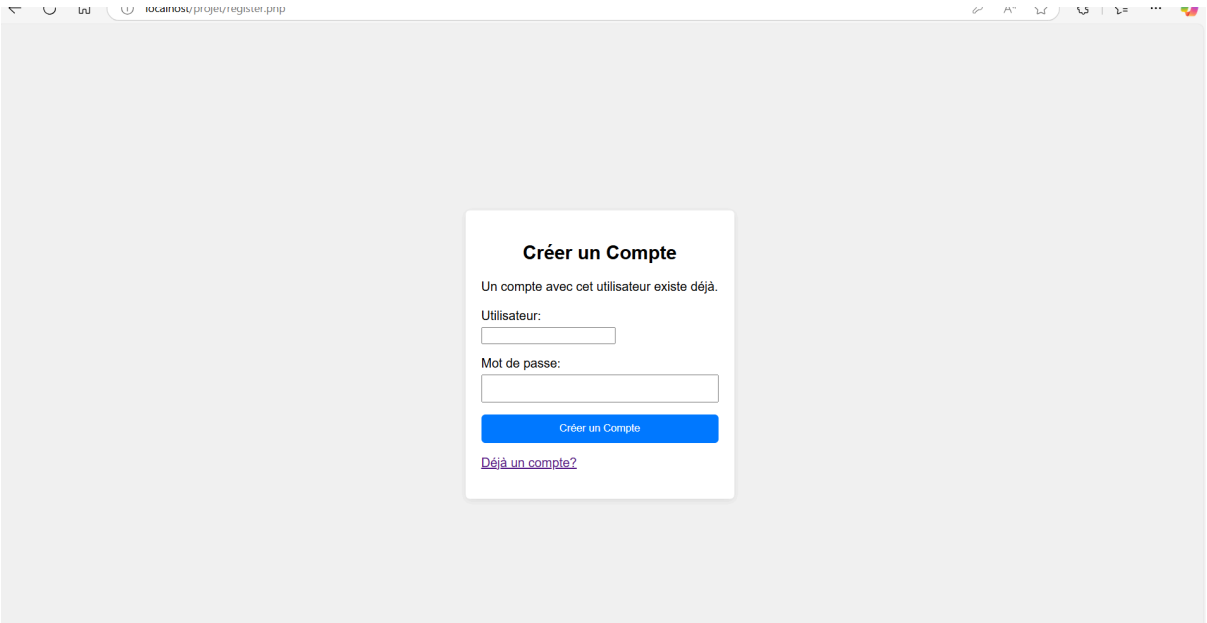
login.php > html > body

```php
<!DOCTYPE html>
<html lang="fr">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Page de Login</title>
    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <?php
    session_start();
    $postData = $_POST;
    $loggedUser = null;
    $errorMessage = null;

    $servername = "localhost";
    $username = "noah";
    $password = "1511";
    $dbname = "compte";

    // Créez la connexion
    $conn = new mysqli($servername, $username, $password, $dbname);

    // Vérifiez la connexion
    if ($conn->connect_error) {
        die("Connexion échouée : " . $conn->connect_error);
    }

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        if (isset($postData['utilisateur']) && isset($postData['password'])) {
            if (!filter_var($postData['utilisateur'])) {
                $errorMessage = 'Il faut un utilisateur valide pour soumettre le formulaire.';
            } else {
                $stmt = $conn->prepare("SELECT * FROM utilisateurs WHERE utilisateur = ?");
                $stmt->bind_param("s", $postData['utilisateur']);
                $stmt->execute();
                $result = $stmt->get_result();

                if ($result->num_rows === 1) {
                    $user = $result->fetch_assoc();
                    // Vérifie le mot de passe
                    if (password_verify($postData['password'], $user['password'])) {
                        $loggedUser = [
                            'utilisateur' => $user['utilisateur'],
                        ];
                        $_SESSION['user'] = $loggedUser;

```

```php
                    // Vérifie si l'utilisateur est admin
                    if ($user['utilisateur'] === 'admin') {
                        header("Location: admin_page.php");
                    } else {
                        header("Location: formulaire.php");
                    }
                    exit();
                } else {
                    $errorMessage = 'nom d\'utilisateur ou mot de passe incorrect.';
                }
            } else {
                $errorMessage = 'nom d\'utilisateur ou mot de passe incorrect.';
            }
        }
    }
    ?>

    <div class="container">
        <h2>Connexion</h2>
        <?php if ($errorMessage): ?>
            <div class="error"><?php echo $errorMessage; ?></div>
        <?php endif; ?>
        <form action="login.php" method="POST">
            <div class="form-group">
                <br>
                <label for="utilisateur">Utilisateur:</label>
                <input type="utilisateur" id="utilisateur" name="utilisateur" required>
            </div>
            <div class="form-group">
                <label for="password">Mot de passe:</label>
                <input type="password" id="password" name="password" required>
            </div>
            <button type="submit">Se connecter</button>
            <div1 class="right-links">
                <br><br>
                <a href="http://localhost/projet/register.php" target="_blank">Créer un compte</a>
            </div1>

        </form>
    </div>
</body>

</html>
```

Page de création de compte

On peut voir ici que si on rentre un compte qui existe déjà, il y aura un message d'erreur.



Coté programmation (Page création de compte “focus sur la création d’un utilisateur ”)

register.php (page de connexion)

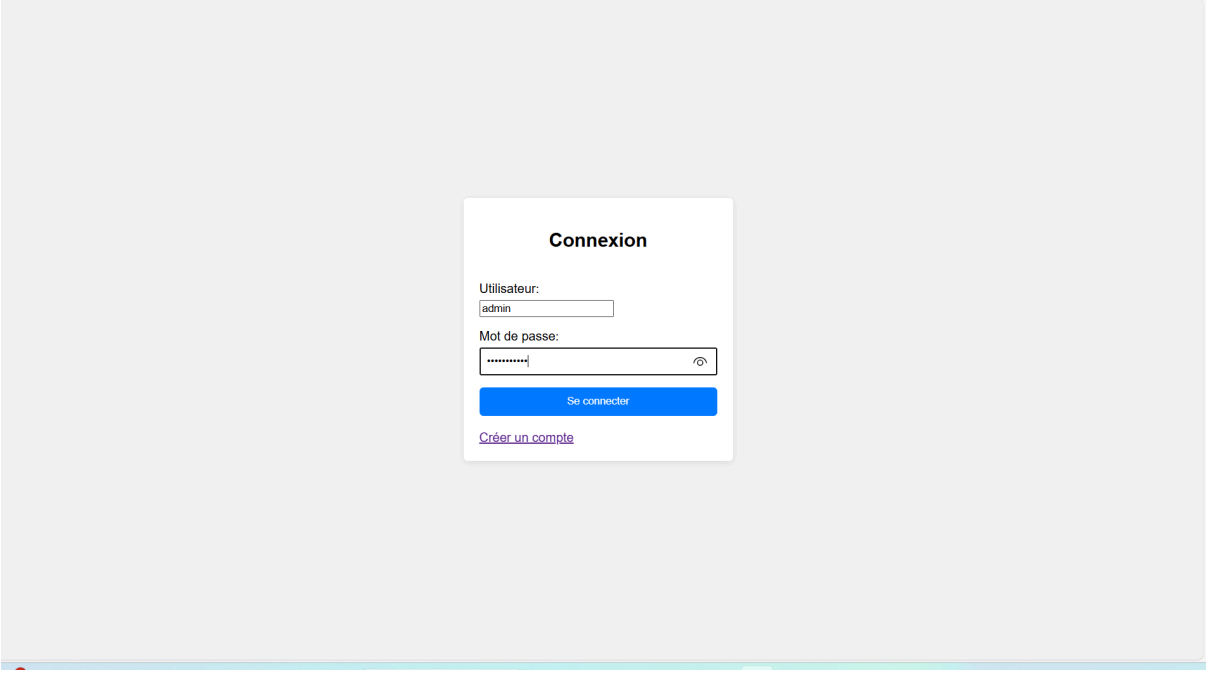
register.php > html > body

```php
<!DOCTYPE html>
<html lang="fr">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Créer un Compte</title>
    <link rel="stylesheet" href="styles.css">
</head>

<body>

    <?php
    session_start();
    $postData = $_POST;
    $errorMessage = null;
    $successMessage = null;

    $servername = "localhost";
    $username = "noah";
    $password = "1511";
    $dbname = "compte";

    // Créez la connexion
    $conn = new mysqli($servername, $username, $password, $dbname);

    // Vérifiez la connexion
    if ($conn->connect_error) {
        die("Connexion échouée : " . $conn->connect_error);
    }

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        if (isset($postData['utilisateur']) && isset($postData['password'])) {
            if (!filter_var($postData['utilisateur'])) {
                $errorMessage = 'Il faut un utilisateur valide pour soumettre le formulaire.';
            } elseif (strlen($postData['password']) < 6) {
                $errorMessage = 'Le mot de passe doit contenir au moins 6 caractères.';
            } else {
                // Vérifiez si l'utilisateur existe déjà
                $stmt = $conn->prepare("SELECT * FROM utilisateurs WHERE utilisateur = ?");
                $stmt->bind_param("s", $postData['utilisateur']);
                $stmt->execute();
                $result = $stmt->get_result();

                if ($result->num_rows > 0) {
                    $errorMessage = 'Un compte avec cet utilisateur existe déjà.';
                } else {
                    // Hachez le mot de passe pour la sécurité
                    $hashedPassword = password_hash($postData['password'], PASSWORD_DEFAULT);

```

```
52                      $stmt = $conn->prepare("INSERT INTO utilisateurs (utilisateur, password) VALUES (?, ?)");
53                      $stmt->bind_param("ss", $postData['utilisateur'], $hashedPassword);
54                      if ($stmt->execute()) {
55                          $successMessage = 'Votre compte a été créé avec succès!';
56                          header("Location: login.php");
57                          exit();
58                      } else {
59                          $errorMessage = 'Erreur lors de la création de votre compte. Veuillez réessayer.';
60                      }
61                  }
62              }
63          }
64      }
65      ?>
66
67      <div class="container">
68          <h2>Créer un Compte</h2>
69          <?php if ($errorMessage): ?>
70              <div class="error"><?php echo $errorMessage; ?></div>
71          <?php endif; ?>
72          <?php if ($successMessage): ?>
73              <div class="success"><?php echo $successMessage; ?></div>
74          <?php endif; ?>
75          <form action="register.php" method="POST">
76              <br>
77              <div class="form-group">
78                  <label for="utilisateur">Utilisateur:</label>
79                  <input type="utilisateur" id="utilisateur" name="utilisateur" required>
80              </div>
81              <div class="form-group">
82                  <label for="password">Mot de passe:</label>
83                  <input type="password" id="password" name="password" required>
84              </div>
85              <button type="submit">Créer un Compte</button>
86          </form>
87          <p> <a href="login.php">Déjà un compte?</a></p>
88      </div>
89  </body>
90
91  </html>
```

Page de connexion de l'administrateur

en écrivant le bon nom d'utilisateur et le mot de passe , cela enverra l'admin dans une page réservé à lui



Coté programmation (Page de connexion "focus sur la différenciation entre l'administrateur et les utilisateurs")

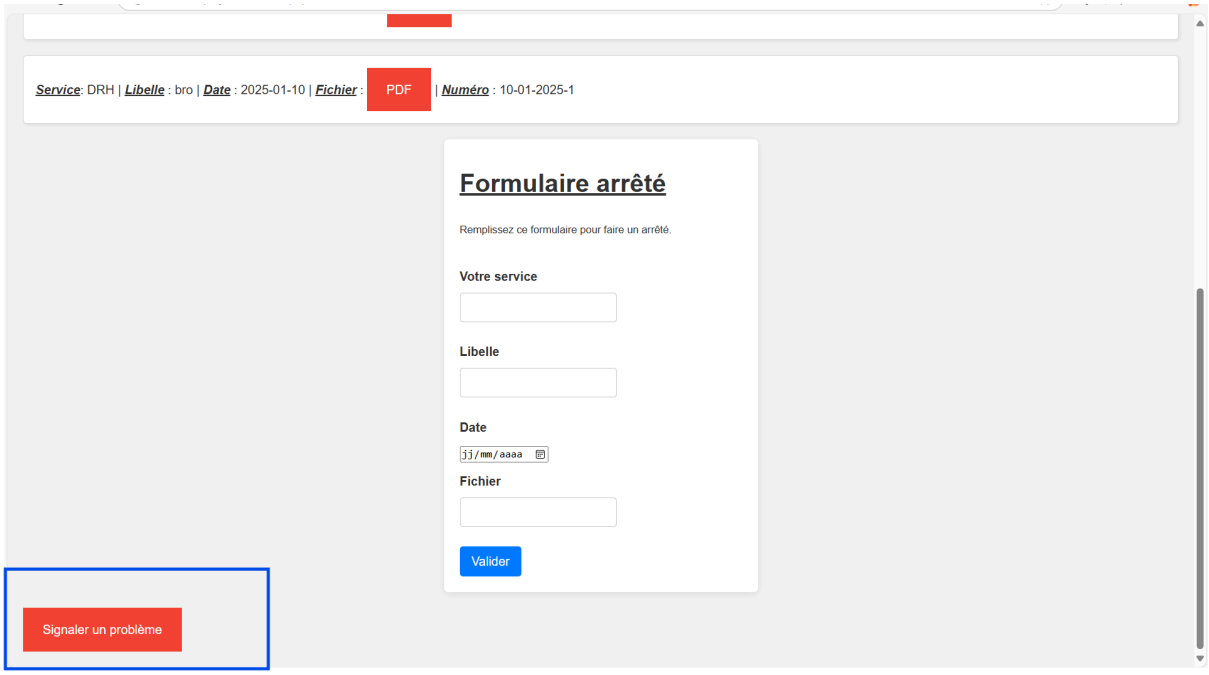
Après avoir vérifié que le mot de passe est correct et que l'utilisateur est correct, il vérifie ensuite si l'utilisateur est un admin grâce à une égalité.

login.php (page de connexion)

```php
if ($result->num_rows === 1) {
    $user = $result->fetch_assoc();
    // Vérifie le mot de passe
    if (password_verify($postData['password'], $user['password'])) {
        $loggedUser = [
            'utilisateur' => $user['utilisateur'],
        ];
        $_SESSION['user'] = $loggedUser;

        // Vérifie si l'utilisateur est admin
        if ($user['utilisateur'] === 'admin') {
            header("Location: admin_page.php");
        } else {
            header("Location: formulaire.php");
        }
        exit();
    } else {
        $errorMessage = 'nom d\'utilisateur ou mot de passe incorrect.';
    }
```

Page de formulaire(bouton de signalement)

On m'a aussi demandé de créer un bouton de signalement en cas de problème qui enverra un mail au responsable pour qu'il répare le problème. J'ai mis ces boutons de signalement dans le formulaire et l'historique.



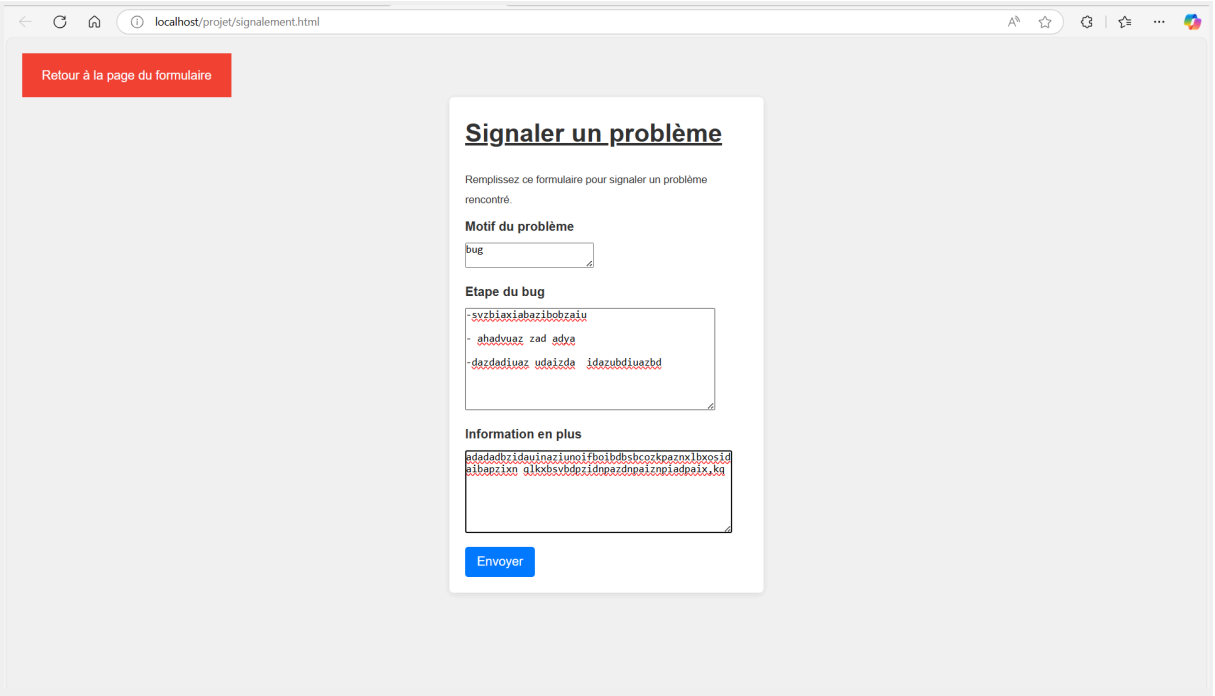
Coté programmation (Page du formulaire“focus sur le bouton signalement”)

formulaire.php (page de formulaire)

```
104 } else {
105     echo 'Aucune donnée d\'historique disponible.<br>';
106 }
107
108
109
110     $conn->close();
111     ?>
112
113 <a href="http://localhost/projet/signalement.html" target="_blank">Signaler un problème</a>
114 </body>
115
116 </html>
117
```

Page de signalement d’un problème

Sur cette page, l’utilisateur devra écrire le problème rencontré et, s’ils le veulent, ils peuvent rajouter des détails sur le problème rencontré. Cela enverra ensuite sur un mail d’un responsable



Coté programmation (Page de signalement“focus sur le bouton signalement”)

signalement.html (page de signalement)

signalement.html > html > body > form

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8" />
    <title>Page de signalement</title>
    <link rel="stylesheet" href="styles_2.css">
</head>
<body>
    <a href="http://localhost/projet/formulaire.php" target="_blank">Retour à la page du formulaire </a>

    <form action="envoie_email.php" method="post">
        <h1><u>Signaler un problème</u></h1>
        <p><small>Remplissez ce formulaire pour signaler un problème rencontré.</small></p>

        <p>
            <label>Motif du problème</label>
            <textarea name="bug_probleme" required></textarea>
        </p>

        <p>
            <label>Etape du bug</label>
            <textarea name="bug_stage" placeholder="facultative"></textarea>
        </p>

        <p>
            <label>Information en plus</label>
            <textarea name="bug_information" placeholder="facultative"></textarea>
        </p>

        <input type="submit" value="Envoyer">
    </form>

</body>
</html>
```

envoie_mail.php (code d'envoie vers le mail)

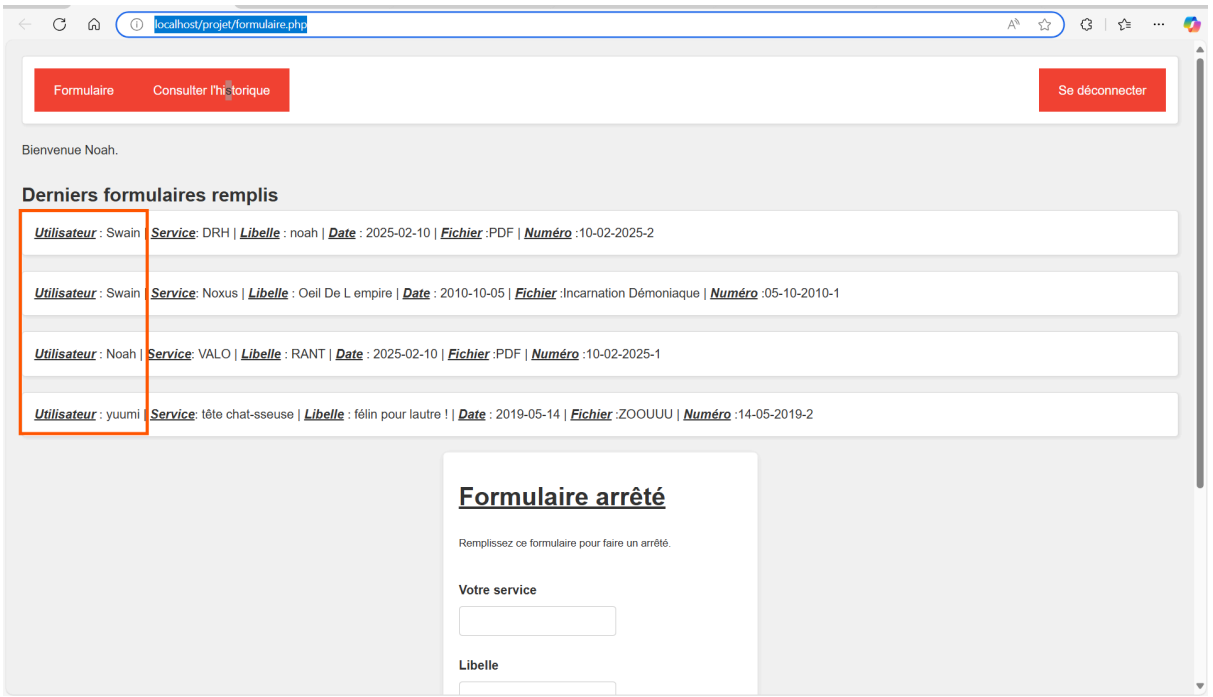
envoie_email.php > ...

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>mail</title>
</head>
<body>

    <?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $motif = htmlspecialchars($_POST['bug_probleme']);
    $etape = htmlspecialchars($_POST['bug_stage']);
    $information = htmlspecialchars($_POST['bug_information']);

    // Destinataire de l'email
    $to = "testl@example.com";
    $subject = "Signalement de problème";

    // Corps de l'email
    $message = "Motif du problème: $motif
";
    $message .= "Etape du bug: $etape
";
    $message .= "Information en plus: $information
";

    // En-têtes de l'email
    $headers = "From: test@example.com";

    // Envoi de l'email
    if (mail($to, $subject, $message, $headers)) {
        echo "Votre signalement a été envoyé avec succès.";
    } else {
        echo "Une erreur s'est produite lors de l'envoi de votre signalement.";
    }
} else {
    echo "Méthode de requête non valide.";
}
?>


</body>
</html>
```

❖ 4ème Semaine

Page de consultation/formulaire

Cette semaine, on m'a aidé à afficher l'utilisateur qui a rempli le formulaire, mais il y avait toujours un problème lors de l'affichage de l'utilisateur qui a rempli le formulaire : sa montre toujours le nom de l'utilisateur connecté, même si ce n'est pas celui qui a réellement soumis le formulaire. Du coup, il fallait créer une colonne en plus pour 'utilisateur' et le rajouter dans la requête SQL pour l'envoyer dans la base de données.



Coté programmation (Page de formulaire“focus sur l'affichage de l'utilisateur ”)

resultat.php (page de résultat)

resultat.php > html > body

```php
<html lang="fr">
<body>

// Vérifie si le formulaire a été soumis
if ($_SERVER["REQUEST_METHOD"] == "GET") {
    // Récupère les données du formulaire
    $service = isset($_GET['service']) ? $_GET['service'] : '';
    $libelle = isset($_GET['libelle']) ? $_GET['libelle'] : '';
    $date = isset($_GET['date']) ? $_GET['date'] : '';
    $fichier = isset($_GET['fichier']) ? $_GET['fichier'] : '';

    // Vérifie que la date est fournie
    if ($date) {
        $dateFormatted = date("d-m-Y", strtotime($date)); // Format de la date pour s'assurer qu'il est correct
        $dateForCount = date("Y-m-d", strtotime($date)); // Pour la requête
    } else {
        $dateFormatted = "";
    }

    // Prépare la requête SQL pour compter les formulaires déjà soumis pour cette date
    $countSql = "SELECT COUNT(*) as count FROM formulaire WHERE date = '$dateForCount'";
    $countResult = $conn->query($countSql);
    $countRow = $countResult->fetch_assoc();
    $incrementNumber = $countRow['count'] + 1; // Incrémenter le nombre

   // Ajoutez cette ligne pour récupérer l'utilisateur de la session
$utilisateur = $_SESSION['user']['utilisateur'];

// Prépare la requête SQL pour insérer les données dans la base
$sql = "INSERT INTO formulaire (service, libelle, date, fichier, utilisateur) VALUES ('$service', '$libelle', '$dateForCount', '$fichier', '$utilisateur')";


// Après avoir réussi à insérer les données dans la base de données
if ($conn->query($sql) === TRUE) {
    // Génère le numéro : date + numéro incrémenté
    $numero = $dateFormatted . '-' . $incrementNumber;

    // Insère le numéro dans la base de données
    $sql = "UPDATE formulaire SET numero = '$numero' WHERE id = " . $conn->insert_id;

    // Exécute la requête de mise à jour
    $conn->query($sql);

    // Affiche le message de succès
    echo '<div>';
    echo "<b><i><u>Les données ont été enregistrées avec succès.</u></i></b>"  . "<br><br>";

    // Ajoute les données dans la session
    $_SESSION['historique'][] = [
        'service' => $service,
        'libelle' => $libelle,
```

Li 42, Col 11 (4 sélectionné) Espace

formulaire.php (page de résultat)

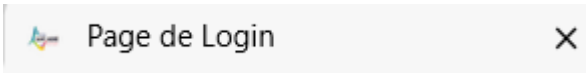
```
formulaire.php > html > body
2   <html>
10  <body>
37      // Vérifier la connexion
38      if ($conn->connect_error) {
39          die("Connexion échouée : " . $conn->connect_error);
40      }
41
42      // Récupérer les 10 derniers formulaires
43      $sql = "SELECT * FROM formulaire ORDER BY id DESC LIMIT 10";
44      $result = $conn->query($sql);
45      ?>
46
47      <h2>Derniers formulaires remplis</h2>
48      <?php
49
50
51      // Vérifier si des données existent
52      if ($result->num_rows > 0) {
53          while ($entree = $result->fetch_assoc()) {
54              echo '<div>';
55              echo '<b><i><u>Utilisateur</u></i></b> : ' . htmlspecialchars($entree['utilisateur']) . "   |     ";
56              echo ' <b><i><u>Service</u></i></b>:  ' . htmlspecialchars($entree['service']) . "   |     ";
57              echo ' <b><i><u>Libelle</u></i></b> : ' . htmlspecialchars($entree['libelle']) . "   |     ";
58              echo ' <b><i><u>Date</u></i></b> : ' . htmlspecialchars($entree['date']) . "     |   ";
59              echo ' <b><i><u>Fichier</u></i></b> :' . htmlspecialchars($entree['fichier']) . '</a>' . "    |    ";
60              echo ' <b><i><u>Numéro</u></i></b> :' . htmlspecialchars($entree['numero']) . "       ";
61              echo '</div>';
62          }
63      } else {
64          echo 'Aucun formulaire rempli récemment.<br>';
65      }
66
67      $conn->close();
68      ?>
69
70
71      <form method="GET" action="resultat.php">
72          <p>
73              <b>
74                  <h1><u>Formulaire arrêté</u>
75              </b></h1>
76          <p><small>Remplissez ce formulaire pour faire un arrêté.</small></p>
77
78          <p><br>
79              <label>Votre service</label> <input type="text" name="service" required>
80          </p>
81
82          <input type="hidden" name="utilisateur" value="<?php echo $_SESSION['user']['utilisateur']; ?>">
83
84          <p>
85              <label>Libelle </label> <input type="text" name="libelle" required>
```

Puis on m'a demandé d'améliorer l'esthétique de mon site, avec du style CSS, et de modifier certaines choses pour que ce soit plus lisible, et de mettre des logos qui représentent ou autre, comme par exemple quand il faut se connecter avec l'utilisateur (icône de petit bonhomme) et le mot de passe (cadenas).

Page de connexion

dans la page de connexion j'ai mis le logo d'utilisateur et mot de passe dans leurs champs et j'ai mis une photo de la mairie en fond .

Et j'ai mis un logo sur toutes les pages.



Coté programmation (Page connexion"focus l'esthétique de la page ")

Ce code permet d'afficher le logo.

login.php (page de connexion)

```
<link rel="stylesheet" href="styles.css">
<link rel="icon" href="Saint-Andre_logotype_rvb.png" type="image/png">
```

Ces codes permettent de mettre les icônes utilisateur et mot de passe.

style.css

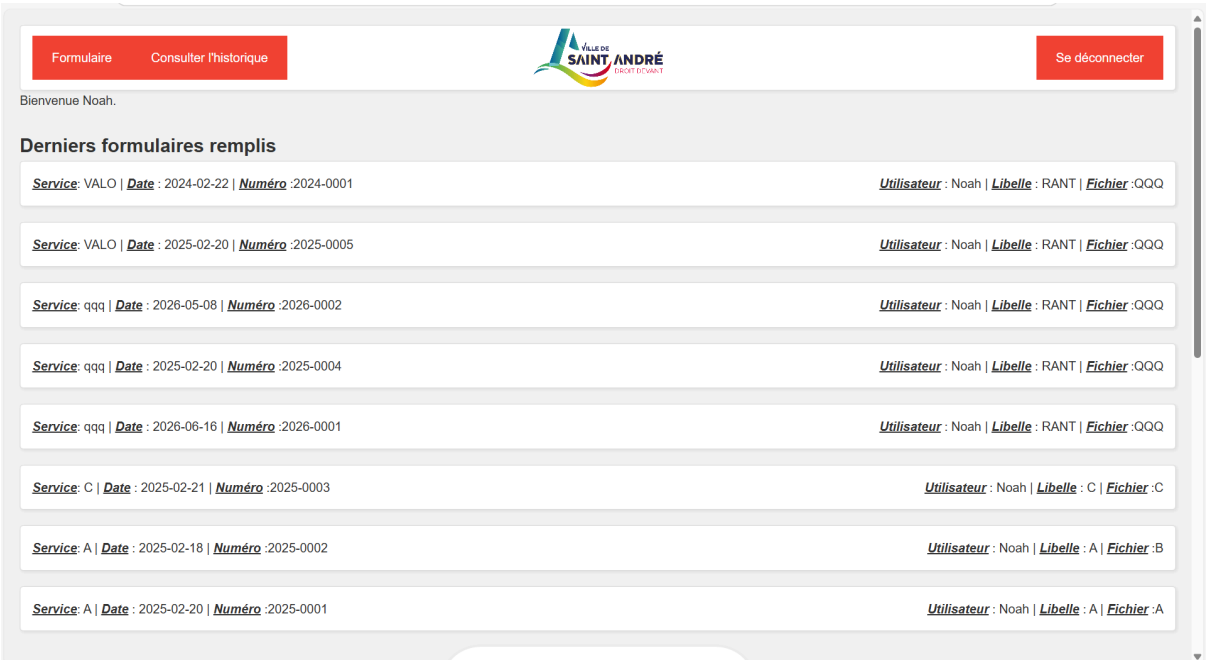
```
.input-utilisateur {
    background-image: url('utilisateur.PNG');
    background-repeat: no-repeat;
    background-position: 10px center;
    background-size: 20px 20px;
    padding-left: 40px;
    height: 40px;
    border: 1px solid #ccc;
    border-radius: 50px;
    box-sizing: border-box;

}

.input-password {
    background-image: url('bloquer.png');
    background-repeat: no-repeat;
    background-position: 10px center;
    background-size: 20px 20px;
    padding-left: 40px;
    height: 40px;
    border: 1px solid #ccc;
    border-radius: 50px;
    box-sizing: border-box;
}
```

Et c'est dans le body que j'ai mis la photo de la mairie.

```
body {
    font-family: Arial, sans-serif;
    background-image: url('mairie.jpg');
    background-size: cover;
    background-position: center;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
}
```

Page du formulaire

Ici j'ai mis le logo de saint-andré et j'ai séparé les informations dans l'historique.



Coté programmation (Page formulaire"focus l'esthétique de la page ")

formulaire.php (page de formulaire)

```
<style>
    .right-align {
        float: right;
    }
    .clearfix::after {
        content: "";
        clear: both;
        display: table;
    }
</style>


    <div class = 'logo'  class="nav-links">
```

```php
// Vérifier si des données existent
if ($result->num_rows > 0) {
    while ($entree = $result->fetch_assoc()) {
        echo '<div class="clearfix">'; // Utilisez clearfix pour gérer le flottement
        echo ' <b><i><u>Service</u></i></b>:  ' . htmlspecialchars($entree['service']) . "   |     ";
        echo ' <b><i><u>Date</u></i></b> : ' . htmlspecialchars($entree['date']) . "     |   ";
        echo ' <b><i><u>Numéro</u></i></b> :' . htmlspecialchars($entree['numero']) . "       ";
        echo '<span class="right-align">'; // Utilisez la classe pour aligner à droite
        echo '<b><i><u>Utilisateur</u></i></b> : ' . htmlspecialchars($entree['utilisateur']) . "   |     ";
        echo ' <b><i><u>Libelle</u></i></b> : ' . htmlspecialchars($entree['libelle']) . "   |     ";
        echo ' <b><i><u>Fichier</u></i></b> :' . htmlspecialchars($entree['fichier']) . "        ";
        echo '</span>'; // Fermez le span
        echo '</div>';
    }
} else {
    echo 'Aucun formulaire rempli récemment.<br>';
}

$conn->close();
?>
```