

Compte Rendu d'Activité

1. Introduction

Ce document témoigne de mon activité au cours des **deux dernières semaines** au sein de l'équipe de développement avancé de **SAE S5.A.01**. Notre projet principal, "Slaves Narratives", vise à améliorer un site web existant qui recense de manière exhaustive de nombreux récits d'esclaves.

Sommaire

- Compte Rendu d'Activité
 - 1. Introduction
 - 2. Issues / Tâches à faire
 - Liste résumée des tâches
 - Réalisations
 - Réorganisation du Repository GitHub
 - Standardisation de l'Envoi de Courriers électroniques
 - 3. Défis Rencontrés
 - 4. Recommandations
 - 5. Conclusion

2. Issues / Tâches à faire

Liste résumée des tâches

Tâche	Description
Standardisation de l'Envoi de Courriers électroniques	Remplacement du système natif d'envoi de courriers électroniques en utilisant la bibliothèque PHPMailer pour assurer un envoi de courriers électroniques plus fiable et robuste.
Système de Réinitialisation de Mot de Passe	Mise en place d'un système de réinitialisation de mot de passe avec envoi d'un courrier électronique contenant un lien de réinitialisation.
Formulaire de contact	Intégration du formulaire de contact avec l'envoi de courriers électroniques à l'administrateur.
Notification de Modification de l'Adresse Courrier électronique d'un Utilisateur	Envoi de courriers électroniques aux utilisateurs lorsqu'un administrateur modifie leur adresse courrier électronique.
Rénovation du README.md	Réorganisation, nettoyage et amélioration du contenu du fichier README du projet.
Restructuration du Repository	Réorganisation du contenu du repository pour le rendre plus lisible et cohérent, y compris l'ajout de fichiers tels que .gitignore et des exemples (.env.example) pour faciliter la mise en route du projet.

Les premiers jours de mon implication dans le SAE ont été consacrés à la prise en main du projet et à la compréhension de son fonctionnement.

Réalisations

Réorganisation du Repository GitHub

Le repository GitHub présentait une structure confuse, ce qui rendait la navigation et la compréhension difficiles. J'ai entrepris une réorganisation en adoptant une structure plus conventionnelle, que vous pouvez consulter [ici](#). J'ai également inclus un fichier `.gitignore` et des fichiers d'exemple (`.env.example`, etc.) pour faciliter la mise en route du projet. J'ai également déplacé les scripts de création de la base de données dans un dossier `sql/` afin de séparer les fichiers de code des fichiers de gestion de la base de données.

Le projet nécessite l'utilisation de [XAMPP](#) pour fonctionner, nécessitant lui-même le déplacement du dossier `slaves_narratives/` dans un autre répertoire pour y travailler, puis la manipulation pour le publier sur GitHub. Pour gagner du temps, j'ai créé deux scripts [PowerShell](#) permettant d'automatiser ces opérations.

```
copyToGit.ps1
```

```
# Check if memory.json exists
$jsonFilePath = Join-Path -Path $PSScriptRoot -ChildPath "memory.json"
if (-not (Test-Path -Path $jsonFilePath -PathType Leaf)) {
    Write-Host "The 'memory.json' file does not exist.
    Please create 'memory.json' based on 'memory.example.json' and fill in your values."
    exit 1
}

# Read the existing memory.json file
$jsonContent = Get-Content -Path $jsonFilePath | ConvertFrom-Json

# Define folder paths
$gitFolder = $jsonContent.GIT_FOLDER
$xamppFolder = $jsonContent.XAMPP_FOLDER
$folderToCopy = $jsonContent.FOLDER_TO_COPY

# Create an array of paths to check
$pathsToCheck = @($gitFolder, $xamppFolder)

# Check if all specified folders exist
if ($pathsToCheck | ForEach-Object { Test-Path -Path $_ -PathType Container }) {
    # Remove the old slave_narratives folder in the GIT_FOLDER if it exists
    $oldSlaveNarrativesPath = Join-Path -Path $gitFolder -ChildPath $folderToCopy
    if (Test-Path -Path $oldSlaveNarrativesPath -PathType Container) {
        Remove-Item -Path $oldSlaveNarrativesPath -Recurse -Force
    }

    # Copy the slave_narratives folder from XAMPP_FOLDER to GIT_FOLDER
    $newSlaveNarrativesPath = Join-Path -Path $gitFolder -ChildPath $folderToCopy
    Copy-Item -Path $xamppFolder\$folderToCopy -Destination $newSlaveNarrativesPath -
    Recurse

    Write-Host "The slave_narratives folder has been replaced in the GIT_FOLDER."
} else {
    Write-Host "One or both of the specified folders do not exist."
}
```

```
memory.json
```

```
{
  "GIT_FOLDER": "C:\\Users\\user\\Desktop\\SAE-S5-Slave-Narratives",
  "XAMPP_FOLDER": "C:\\xampp\\htdocs",
  "FOLDER_TO_COPY": "slave_narratives"
}
```

copyToXampp.ps1 est identique à copyToGit.ps1 à l'exception de la ligne 33

Standardisation de l'Envoi de Courriers électroniques

Le projet implique divers scénarios d'envoi de courriers électroniques. Cependant, le système natif d'envoi de courriers électroniques s'est avéré peu fiable, voire inopérant. Mon travail a consisté à rechercher une alternative fiable pour remplacer la fonction native `mail()` de PHP. Après des recherches approfondies, j'ai opté pour l'utilisation de [PHPMailer](#) en raison de sa légèreté et de sa **robustesse**.

Une fois la bibliothèque de gestion des courriers électroniques déterminée, j'ai connecté les différentes parties de l'application au nouveau système. Cela a débuté par l'intégration du formulaire de contact, qui a été à l'origine de plusieurs autres tâches, notamment l'issue [\[US 1\] Page de contact #2](#).

```
trait_contact.php
```

```
<...>
```

```
$mail = new PHPMailer(); $mail->IsSMTP(); $mail->Mailer = "smtp";

try {
    // Server settings
    $mail->SMTPDebug = SMTP::DEBUG_SERVER; // Enable verbose debugging
    $mail->isSMTP();
    $mail->Host = $_ENV["MAIL_HOST"];
    $mail->SMTPAuth = true;
    $mail->Username = $_ENV["MAIL_USERNAME"];
    $mail->Password = $_ENV["MAIL_PASSWORD"];
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
    $mail->Port = $_ENV["MAIL_PORT"];

    // Recipients
    $mail->setFrom($_ENV["MAIL_USERNAME"], 'Site web - Slave Narratives');
    $mail->addAddress($_ENV["MAIL_TO"], 'Eric PHILIPPE');

    // Content
    $mail->isHTML(true);
    $mail->Subject = lang('TraitContact.title') . " - " . mb_convert_encoding($objet, 'ISO-8859-1', 'UTF-8');
    $mail->Body = '<p>'.lang('TraitContact.content_email').' <strong>'.$nom.' '.$prenom.' :</strong></p>'
        . '<p><strong>'.lang('TraitContact.email').' : </strong>'.$email.'</p>'
        . '<p><strong>'.lang('TraitContact.message').' : </strong></p>'
        . '<p>'. nl2br($message) .'</p>';

    $mail->send();
    $_SESSION['msg-contact'] = lang('TraitContact.email_send');
} catch (Exception $e) {
```

```
$_SESSION['erreur-contact'] = "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}
```

Une adresse Gmail a été spécialement créée pour le projet. Gmail offre une sécurité supplémentaire en cas de tentative de connexion suspecte, bien que sa configuration soit complexe.

Ce code a été appliqué de manière similaire aux autres parties de l'application nécessitant l'envoi de courriers électroniques, notamment :

- ☒ Formulaire de contact
- ☒ Mot de passe oublié
- ☒ Modification du mail utilisateur
- ☒ Modification du mot de passe utilisateur

3. Défis Rencontrés

Difficulté	Description
Langage du projet	L'intégralité du projet est codée en PHP, un langage que je maîtrise peu.
Framework CodeIgniter	Le projet repose sur le framework CodeIgniter, que je découvrais pour la première fois.
Problème de port SMTP	L'envoi de courriers électroniques s'est avéré être capricieux, fonctionnant sur certaines machines et pas sur d'autres en raison de problèmes liés au port SMTP.

4. Recommandations

Étant donné que l'équipe fonctionnait déjà de manière efficace avant mon arrivée et que le projet était bien avancé, je n'ai pas de recommandations particulières à faire.

5. Conclusion

Ce projet m'a permis de renforcer mes compétences en PHP et de découvrir de nouvelles bibliothèques. J'ai également pu apprendre le fonctionnement d'un projet déjà bien avancé, ainsi que les défis auxquels une équipe peut être confrontée.