



Rapport de SAE

Slave Narrative

SAE effectué du 02 Octobre 2023 au 13 Octobre 2023

Destinataires :

Pendaries Esther

Bruel Jean-Michel

Sommaire

User Story traitées / issues	3
Formulaire de contact	3
Option de recherche	3
Documentation utilisateur	4
Avant / Après	4
Formulaire de contact	4
Option de recherche	5
Exemple de code commenté	7
Formulaire de contact	8
Option de recherche	9
Justification	10
Prise de risque / les freins	11
Conseils pour la suite	12

User Story traitées / issues

Formulaire de contact

La première issue à laquelle j'ai été affecté concerne le formulaire de contact.

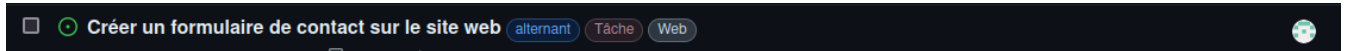


Figure 1: capture d'écran de l'issue de formulaire de contact sur GitHub

Elle implique donc la User Story ci-dessous.



Figure 2: user story concernant le formulaire de contact

En effet, il est essentiel pour les utilisateurs de pouvoir contacter le propriétaire du site web. C'est pourquoi le bouton pour accéder à cette page est situé dans le footer¹ du site.

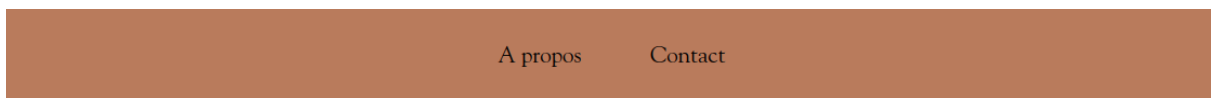


Figure 3: footer du site

L'achèvement réussi de cette User Story garantit que les utilisateurs peuvent facilement communiquer avec le propriétaire du site web, ce qui améliorera leur expérience globale sur la plateforme.

Option de recherche

Pour cette seconde partie, j'ai été affecté à une issue qui concerne la barre de recherche pour la liste de récits. On pourra donc rechercher un récit en fonction du nom de l'esclave, de la date de publication ou encore du titre du récit.

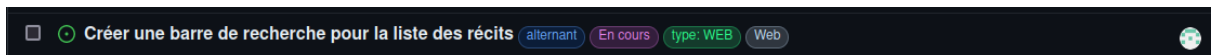


Figure 4: issue de la barre de recherche

Elle implique donc elle aussi une user story en rapport.



¹ footer: partie situé en tout en bas du site web

Figure 5: user story barre de recherche

Documentation utilisateur

Lors de ce projet, j'ai également été affecté à la documentation utilisateur concernant toute les fonctionnalités développées lors des 2 semaines de travail comme la page de statistique, la barre de recherche, la page de contact ou encore la refactorisation du code CSS en SASS.



Figure 6: issue de la documentation utilisateur

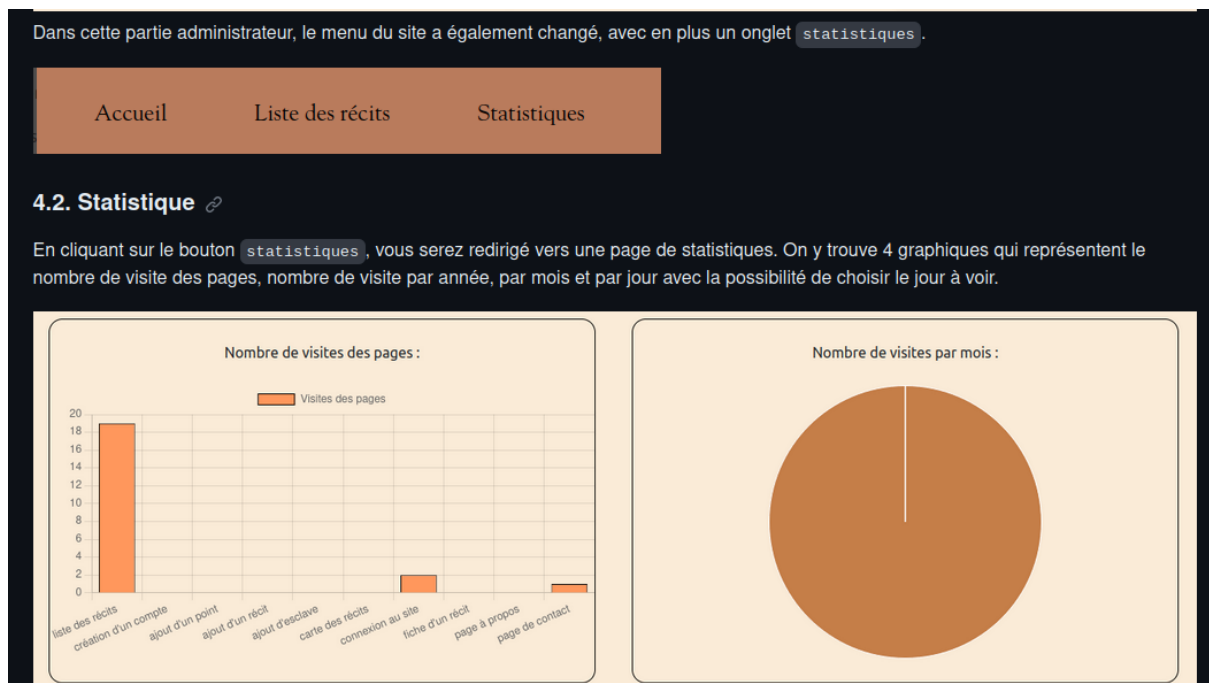


Figure 7: exemple de contenu de la documentation pour la partie statistique.

Avant / Après

Formulaire de contact

Voici la page de liste des récits avant.

Recensement des récits		
Nom de l'esclave	Date de publication	Titre
Cugoano, Ottobah	1787	Thoughts and Sentiments on the Evil of Slavery and Other Writings.
Equiano, Olaudah	1789	The Interesting Narrative of the Life of Olaudah Equiano, or Gustavus Vassa, the African. Written by Himself.
King, Boston	1798	Memoirs of the Life of Boston King, a Black Preacher, Written by Himself, during His Residence at Kingswood-School.
Smith, Venture	1798	A Narrative of the Life and Adventures of Venture, a Native of Africa: : But Resident above Sixty Years in the United States of America. Related by Himself.
White, George	1810	A Brief Account of the Life, Experience, Travels, and Gospel Labours of George White, an African; Written by Himself, and Revised by a Friend.

Figure 8: Page de liste des récits

Et la voici maintenant.

Recensement des récits		
Rechercher...	Rechercher	
Nom de l'esclave	Date de publication	Titre
Cugoano, Ottobah	1787	Thoughts and Sentiments on the Evil of Slavery and Other Writings.
Equiano, Olaudah	1789	The Interesting Narrative of the Life of Olaudah Equiano, or Gustavus Vassa, the African. Written by Himself.
King, Boston	1798	Memoirs of the Life of Boston King, a Black Preacher, Written by Himself, during His Residence at Kingswood-School.
Smith, Venture	1798	A Narrative of the Life and Adventures of Venture, a Native of Africa: : But Resident above Sixty Years in the United States of America. Related by Himself.
White, George	1810	A Brief Account of the Life, Experience, Travels, and Gospel Labours of George White, an African; Written by Himself,

Figure 9: Page de liste des récits avec la barre de recherche

Voici un exemple d'utilisation de la barre de recherche. Comme on peut le voir, on retrouve bien les résultats correspondant à la recherche.

cugo	Rechercher	
Nom de l'esclave	Date de publication	Titre
Cugoano, Ottobah	1787	Thoughts and Sentiments on the Evil of Slavery and Other Writings.
Cugoano, Ottobah	1825	Narrative of the Enslavement of Ottobah Cugoano, a Native of Africa; Published by Himself in the Year 1787.

Figure 10: utilisation de la barre de recherche avec le mot "cugo"

Option de recherche

Voici le contenu de la page de contact avant

Récits d'esclaves

Chaque voix doit être entendue

Ceci est la page de contact

Et la voici après l'ajout du formulaire

Contactez-nous

Nom

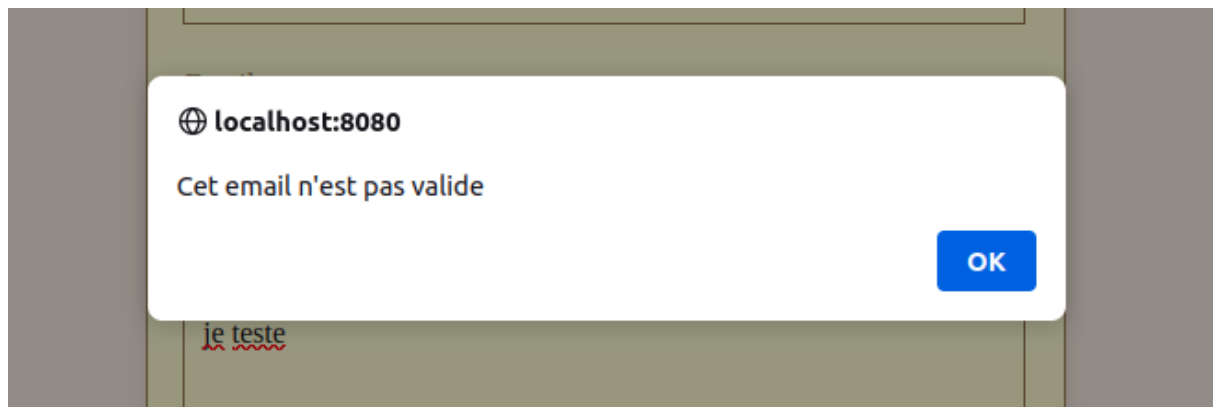
Email

Message

je teste

Envoyer

Lorsque l'email n'est pas valide, on prévient l'utilisateur avec un message et le mail ne sera pas envoyé.



Lorsque le formulaire est correctement rempli, on prévient également l'utilisateur que le message a bien été envoyé.



Voici le contenu du mail une fois reçu.

À moi ▼

Bonjour,

Voici un nouveau message de Quentin

Mail: test@gmail.com

je teste

Exemple de code commenté

Concernant le code, CodeIgniter, un framework a été utilisé. Il se base sur un modèle MVC².

² MVC : Modèle Vue Controller, c'est un modèle dans la conception de log

Formulaire de contact

Pour la partie du formulaire de contact, seulement le fichier de vue a été modifié.

Tout d'abord, le code pour inclure la librairie d'envoi de mail.

```
<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/@emailjs/browser@3/dist/email.min.js"></script>
<script type="text/javascript">
  (function() {
    emailjs.init("CFprpDhypfb61HdhQ"); //on donne la clé publique de notre compte EmailJS
  })();
</script>
```

Ensuite, on crée le formulaire. Le “lang” est utilisé pour la fonctionnalité de changement de langue dans l'application. On demande aussi à ce que chaque élément du formulaire soit obligatoire pour l'envoyer. Une fonction “sendMail” est utilisé lorsqu'on envoi le formulaire.

```
<body>
  <form id="myForm" method="post">
    <div class="login-container">
      <h2><?= lang('contact_resc.title') ?></h2>
      <div class="input-group">
        <label for="name"><?= lang('contact_resc.name') ?></label>
        <input type="text" id="name" name="name" required>
      </div>
      <div class="input-group">
        <label for="email"><?= lang('contact_resc.email') ?></label>
        <input type="email" id="email" name="email" required>
      </div>
      <div class="input-group">
        <label for="message"><?= lang('contact_resc.message') ?></label>
        <textarea id="message" name="message" rows="5" style="resize: none;" required></textarea>
      </div>
      <button type="button" onclick="sendMail()"><?= lang('contact_resc.send_button') ?></button>
    </div>
  </form>
```

Cette fonction est utilisée pour envoyer l'email grâce à Email JS.

L'objet “params” permet de relier les éléments de notre formulaire avec id, au donnée dans le template sur Email JS.

Dans le “email.send”, on donne les paramètre de notre compte Email JS comme “contact_service”, pour relier notre email au formulaire et le “contact_form” qui est le nom de notre template, permettant de personnaliser le contenu du mail reçu.


```

<script>
function sendMail() {
    // Récupérer la valeur de l'e-mail
    var email = document.getElementById("email").value;

    // Vérifier si l'e-mail est valide en utilisant une regex
    var emailRegex = /^[a-zA-Z0-9!#$%&'*+\./=?^_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z0-9]{2,4}$/;
    if (!emailRegex.test(email)) {
        alert("Cet email n'est pas valide");
        return; // Arrêter l'envoi du formulaire si l'e-mail n'est pas valide
    }

    // Si l'e-mail est valide, continuer avec l'envoi du formulaire
    var params = {
        from_name: document.getElementById("name").value,
        email_id: email,
        message: document.getElementById("message").value
    };

    emailjs.send("contact_service", "contact_form", params).then(function (res) {
        alert("Message envoyé !");
    });
}
</script>

```

Option de recherche

Pour cette fonctionnalité, les 3 fichiers ont été modifiés.

Dans le fichier “Model”, une fonction qui prend en compte la barre de recherche a été ajoutée.

Si la barre de recherche est utilisée, on donne le nom de l’esclave, la date de publication et le titre à prendre en compte. Sinon, on affiche toute la liste d’esclaves.

```

public function getRecits($search = null)
{
    if ($search) {
        return $this->asArray() //on choisi les paramètre que la barre de recherche va prendre en compte
        ->like('nom_esc', $search)
        ->orLike('titre', $search)
        ->orLike('date_publi', $search)
        ->orderBy('date_publi')
        ->findAll();
    } else {
        return $this->asArray()
        ->orderBy('date_publi')
        ->findAll();
    }
}

```

Ensuite, on utilise cette fonction dans le fichier “Controller”.

On récupère la valeur de la recherche depuis l’URL en la mettant dans une variable puis on utilise la fonction “getRecits” avec en paramètre cette variable.

```

public function index()
{
    $model = model(ModelRecit::class);

    $search = $this->request->getGet('search'); // Récupère la valeur de recherche depuis l'URL

    $data = [
        'recits' => $model->getRecits($search),
    ];
}

```

On utilise le fichier de “Vue” pour créer la barre de recherche.

On crée la barre de recherche avec le bouton.

```

<form action="{base_url('recits')}" method="get">
    <input class="input-search"
        type="text"
        name="search"
        value="{php echo isset($_GET['search']) ? $_GET['search'] : ''}" placeholder="Rechercher...">
    <input class="button-search" type="submit" value="Rechercher">
</form><br>

```

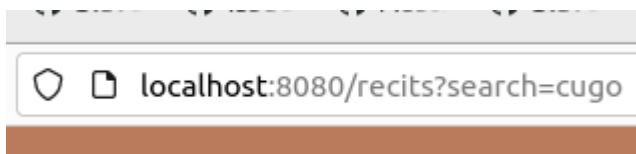


Si la recherche est utilisée, on récupère le résultat de la recherche pour filtrer en fonction.

```

<?php
if(isset($_GET['search'])){
    $filtervalues = $_GET['search'];
    $query = "SELECT * FROM tab_recits_v3 WHERE CONCAT(nom_esc,date_publi,titre) LIKE '%$filtervalues%' ";
}
?>

```



Justification

Pour le formulaire de contact, il est essentiel de pouvoir communiquer avec le propriétaire du site pour par exemple lui poser des questions par rapport au site, à la personne ou encore relayer un bug trouvé sur le site. Les 3 champs à remplir ont donc été choisis dans ce sens, pour que le propriétaire ait directement le nom de l'utilisateur, son mail pour lui répondre si besoin et bien sûr son message. Pour l'envoi du mail, j'ai choisi d'utiliser la librairie Email JS. Elle permettra au propriétaire du site d'afficher le contenu du mail comme

elle le souhaite grâce à un, template personnalisé et de ce fait, avoir de la personnalisation sans avoir besoin de coder.



Figure 6: titre du mail reçu

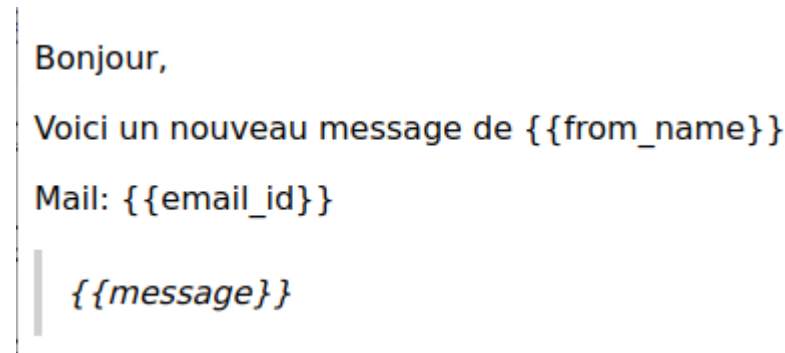


Figure : contenu du mail reçu (template personnalisé)

Pour la barre de recherche, je pense qu'il était essentiel, au vu des fonctionnalités développées par les élèves initiaux, de pouvoir rechercher un récit en particulier. En effet, les initiaux ont développé une partie administration qui permet notamment d'ajouter un récit.

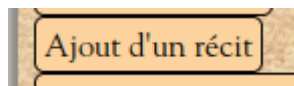


Figure : bouton pour ajouter un récit

De ce fait, le nombre de récits ne fera que croître au fil du temps avec donc de plus en plus de récit disponible sur le site. Une barre de recherche est donc essentielle pour que les utilisateurs prennent le moins de temps possible à trouver le récit qu'ils souhaitent. Ils pourront par exemple rechercher le nom d'un esclave en particulier, le titre d'un récit ou encore une date pour voir tous les récits écrits à la même date.

Prise de risque / les freins

Les prises de risque concernent les propositions de fonctionnalités qui n'étaient pas demandées par le client au début du projet. Après la prise d'initiative, un appel avec le client était effectué pour avoir son approbation la concernant. Pour ma part, j'ai cherché les fonctionnalités les plus pertinentes qui n'étaient pas encore développées et j'ai donc pris l'initiative de développer la barre de recherche pour la liste de récits, pour les raisons que j'ai citées précédemment.

Concernant les freins, la mise en place du projet a été un très gros frein pour plusieurs personnes du projet. Le fait que nous n'avions pas de serveur dédié pour le site a fait que nous avons dû trouver une autre solution pour l'héberger.

Le framework php "CodeIgniter" qui avait été utilisé par les précédents développeurs a aussi été un frein au vu de sa complexité se basant sur une architecture logicielle MVC. Il y a assez peu de documentation sur ce framework ce qui rendait son utilisation contraignante. D'autant plus que son utilisation n'était pas justifiée à mon sens, et a plus créer du trouble quant à la compréhension du code, alors que son but était de le structurer.

Conseils pour la suite

Pour la suite, je pense qu'il faudrait davantage prendre en compte l'environnement de travail, et mieux se préparer aux éventuels problèmes que l'on pourrait rencontrer. Il faudrait aussi préparer une documentation pour les personnes qui arrivent en cours de projet pour que dans notre cas les alternants, nous puissions démarrer le projet plus rapidement et rattraper le "retard" par rapport aux élèves initiaux concernant la SAE.