IUT de Lens
16 Rue de l'Université
62307, Lens

Association des Naturalistes de la Gohelle
CCS Jules Grare, les Marichelles
62800, Liévin

RAPPORT DE STAGE

Projet Refonte d'un Site Web

Théo VANBANDON

Du 21 Mai au 15 Juillet 2024

Maître de stage : M. Romain Wallon, maître de conférences

Année universitaire : 2023-2024

Avant-propos

La recherche d'un stage a été une période particulièrement difficile pour moi. Malgré mes efforts, je n'avais toujours pas réussi à trouver une opportunité. C'est alors que l'équipe pédagogique du département informatique m'a proposé une alternative : un stage au sein de "Association des Naturalistes de la Gohelle". Je tiens à souligner l'importance de cette opportunité qui a été cruciale pour la réussite de mon parcours.

Remerciements

Je souhaite exprimer ma gratitude à plusieurs personnes sans qui cette expérience n'aurait pas été possible. Je remercie mon tuteur de stage, M. Romain Wallon, pour son encadrement et ses précieux conseils tout au long de mon stage.
Un grand merci également à l'équipe pédagogique de mon établissement pour leur soutien et pour m'avoir proposé cette alternative qui s'est avérée bénéfique.
Je remercie aussi Mme. Membré pour ses conseils de recherche de stage qui, même si je n’ai pas réussi à trouver un stage cette année, me seront utiles pour mes prochaines candidatures dont le stage de troisième année.
Enfin, je remercie mon collègue de stage Lucas Copin pour sa collaboration et son aide précieuse, rendant cette expérience encore plus enrichissante et agréable.

Table des matières

1 Note de synthèse....4
1.1 Présentation de l'association....4
1.1.1 Objectifs de l'association....4
1.1.2 Collaborations et partenariats....5
1.1.3 Contribution à la connaissance de la nature....5
1.2 Conditions de stage....5
1.3 Contenu du stage....6
1.3.1 Présentation de l'existant....6
1.3.2 Démarche adoptée....8
1.3.3 Difficultés rencontrées....9
1.3.4 Mise en production....9
1.3.5 Bilan général....10
2 Rapport technique....11
2.1 Introduction....11
2.2 Refonte de la base de données....12
2.2.1 État de la base de données lors de la récupération du projet....12
2.2.2 Solutions apportées....12
2.2.3 Résultats....14
2.3 CRUD pour la page "Observations"....15
2.3.1 Affichage des observations....17
2.3.2 Affichage des informations pour une observation donnée....18
2.3.3 Création d'une observation....19
2.3.4 Modification d'une observation....23
2.3.5 Suppression d'une observation....25
2.3.6 Routes....27
2.4 Gestion des rôles....28
2.5 Génération de fichiers sous format PDF....30
2.5.1 Explication et installation....30
2.5.2 Fonctionnement....30
2.6 Gestion des mails....31
2.6.1 Explication et problèmes rencontrées....31
2.6.2 Solution finale....31
2.7 Déploiement du site sur l'hébergeur....34
2.7.1 Importation de la base de données....34
2.7.2 Transfert des fichiers....35
3 Annexes....37
3.1 Portfolio....37
3.2 Lettre de mission....53
3.3 Exemple de tract pour une sortie....54

1 Note de synthèse

1.1 Présentation de l'association

L'Association des Naturalistes de la Gohelle (ANG) fut créée le 10 Janvier 1987 par une poignée de naturalistes motivés par l'étude et la protection des oiseaux et de la nature en général.

L'action de l'ANG se concentre principalement sur le pays de la Gohelle, un territoire s'étendant autour des villes de Lens et de Liévin, dans le département du Pas-de-Calais, au nord de la France.

Logo de l'association

1.1.1 Objectifs de l'association

L'ANG a pour vocation de rassembler des naturalistes de diverses spécialités dans le but de :

- Entreprendre ensemble des travaux d'étude et de connaissance de la faune et de la flore régionales : L'association réalise des études scientifiques approfondies sur les milieux naturels de la région, par exemple au terril de Pinchonvalles.
- S'ouvrir à tous pour une initiation et une formation aux sciences naturelles : Elle offre des programmes d'initiation et de formation ouverts à tous, contribuant ainsi à la sensibilisation et à l'éducation environnementale.
- Participer aux actions de protection et de gestion de l'environnement et du cadre de vie : L'ANG s'engage activement dans des projets de conservation et de gestion environnementale en collaboration avec diverses institutions et associations locales.

1.1.2 Collaborations et partenariats

Tournée vers l'extérieur et l'échange, l'ANG est membre de plusieurs organisations, dont le Groupe Ornithologique et Naturaliste, le CPIE Chaîne des Terrils, et le Conservatoire d'Espaces Naturels Hauts-de-France. Elle collabore également avec des associations locales telles que Nature et Vie, Association Découverte Nature, et Lestrem Nature, partageant informations et sorties guidées.

1.1.3 Contribution à la connaissance de la nature

L'ANG participe régulièrement à des enquêtes naturalistes régionales et nationales, indispensables à la connaissance du patrimoine naturel régional. Parmi ces projets, on peut citer l'atlas des oiseaux nicheurs, le recensement international des oiseaux d'eau en hiver, et les inventaires des amphibiens, chauves-souris, libellules, et busards Saint-Martin en hiver. L'association joue également un rôle important dans l'observatoire de l'environnement de Lens-Liévin, fournissant des données essentielles pour aider les décideurs locaux.

1.2 Conditions de stage

Étant donné que l'Association des Naturalistes de la Gohelle ne dispose pas de bureau, j'ai effectué mon stage au sein de l'IUT de Lens. Ce stage a été réalisé en équipe de deux puisque je l'ai fait avec Lucas Copin, également étudiant de BUT Informatique 2ème année. Une salle nous a été attribuée, où les techniciens ont installé des ordinateurs pour que nous puissions travailler. D'autres étudiants étaient également présents dans la salle, effectuant eux aussi un stage. Au total, nous étions quatre à travailler dans la même salle.

Concernant les horaires de travail, nous n'avions pas d'horaires stricts, mis à part un début de journée à 8h30. La plupart du temps, nous commencions à cette heure, avec une pause de 30 minutes pour le déjeuner. De cette manière, nous terminions notre journée à 16h00, mais il nous arrivait de travailler chez nous lors de problèmes non réglés durant la journée.

Notre encadrement a été assuré par M. Romain Wallon qui est de maître de conférence et fait parti de l'association. Nous travaillions en autonomie et en cas de questions, nous avions la possibilité de contacter notre tuteur via l'application de messagerie instantanée "Mattermost". De plus, dès que M. Wallon était à l'IUT, il venait nous voir pour nous conseiller, observer les avancées et nous faire part des souhaits de l'association concernant les fonctionnalités du site. Nous disposions également de diverses ressources, tel que le logiciel PhpStorm pour mener à bien notre projet de refonte du site web. On peut également citer GitLab en tant qu'outil de gestion de projet.

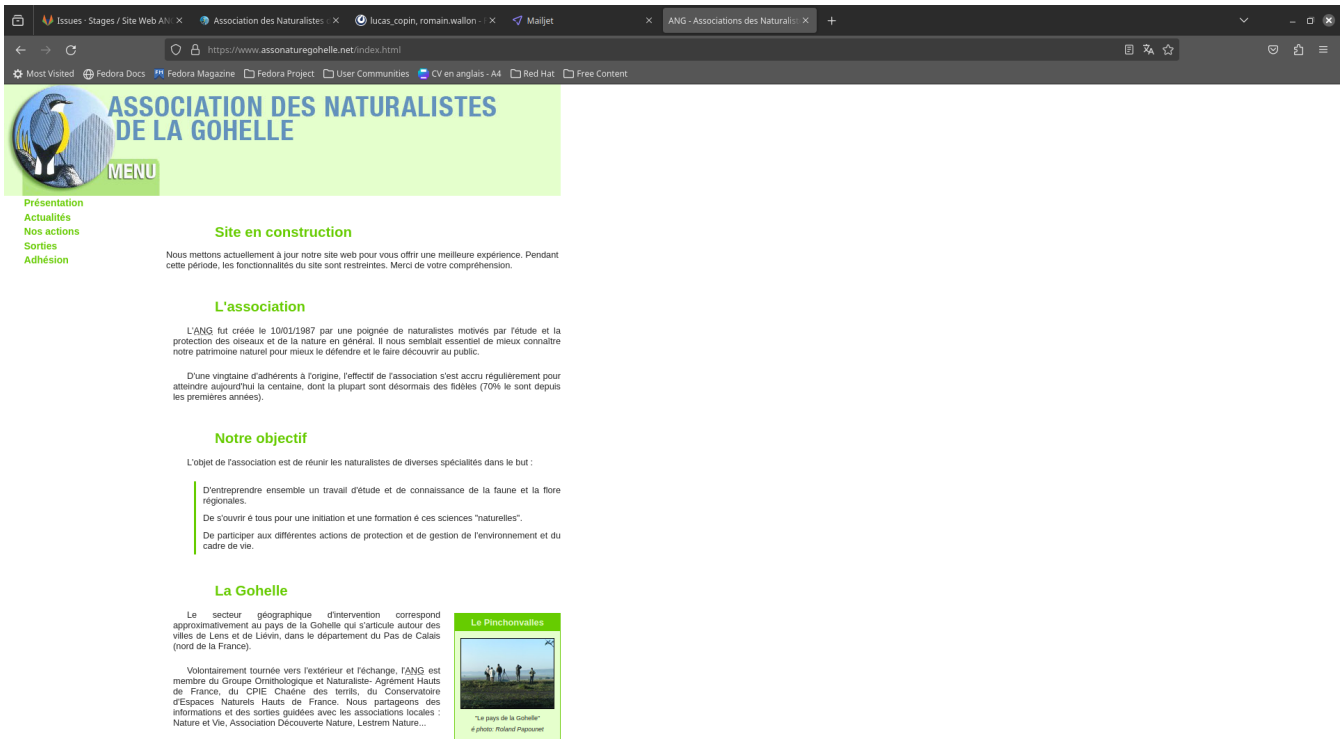
1.3 Contenu du stage

1.3.1 Présentation de l’existant

Le stage consistait à réaliser une refonte totale du site internet de l'association en PHP en utilisant le framework Laravel. Le site hébergé ne répondait plus aux besoins de l’association et était transformé en site vitrine. C’est-à-dire que l’administrateur du site ne pouvait plus ajouter de nouvelles informations car la partie serveur était désactivée, entraînant des erreurs lors de l'accès aux pages.

Capture d’écran montrant l'erreur sur le site hébergé avant le stage

De plus, le site avait un design vieillissant et n’était pas "responsive", ce qui signifie que l'affichage n'était pas adapté à tous les types d’écrans (smartphones, tablettes, ordinateurs).



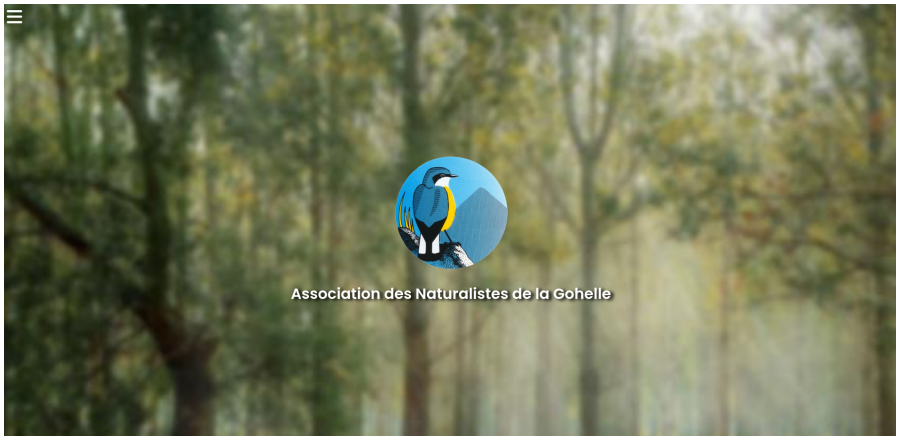
Capture d’écran du site hébergé avant le stage

Ce projet de refonte du site avait été commencé par d'autres étudiants il y a deux ans. Certaines fonctionnalités attendues par l'association étaient donc déjà réalisées, notamment l'authentification, qui est un point essentiel pour tout site internet.

Capture d'écran des formulaires d'authentification récupérés

Lorsque nous avons récupéré le projet, l'authentification fonctionnait bien et les formulaires étaient présents sur le site, comme vous pouvez le voir ci-dessus. Nous les avons conservés en les modifiant légèrement.

De même pour le thème, les étudiants précédents avaient déjà ajouté un fond avec des arbres.



Capture d'écran du fond arboré du site

Pour les textes, ils avaient ajouté un fond blanc opaque pour améliorer la lisibilité. Nous avons retravaillé ces parties tout en conservant l'idée d'origine.

Capture d'écran du fond des textes récupérés

De la même manière, nous avons également modifié le menu, tout en conservant globalement celui des anciens étudiants.

Capture d'écran du menu récupéré

1.3.2 Démarche adoptée

Pour la gestion du projet, nous avons utilisé GitLab, qui inclut un système de gestion de tickets. Chaque tâche à réaliser était répertoriée sous forme de tickets sur GitLab. Pour une meilleure visualisation, j'ai créé des labels : TO DO, DOING, DONE et TEST, permettant de distinguer facilement l'état des différentes tâches.

En ce qui concerne la répartition des tâches, nous avions la liberté de choisir la tâche que nous souhaitions réaliser, l'une après l'autre, selon nos préférences.

Nous avons commencé par analyser de A à Z le projet récupéré dans le but de :

- Déterminer quelles fonctionnalités étaient présentes ou non.
- Prendre connaissance de comment le code fonctionnait.
- Trouver des solutions pour implémenter les fonctionnalités futures.

1.3.3 Difficultés rencontrées

Durant ce stage, nous avons été confrontés à plusieurs difficultés, principalement liées à la compréhension du code récupéré.

Premièrement, il a fallu comprendre un code qui n'était pas le nôtre. Le code réalisé précédemment par les autres étudiants ne comportait pas de documentation. C'est à ce moment-là que l'on se rend compte de l'importance des commentaires et de la documentation.

De plus, les noms de fichiers n'étaient pas très clairs, ce qui compliquait davantage notre tâche. Par exemple, pour désigner les observations, il y avait un fichier nommé "post", et pour les activités, c'était "post_texts". Nous avons fini par renommer ces tables.

1.3.4 Mise en production

Au cours de ce projet, nous avons eu la chance de pouvoir mettre en production notre site. Habituellement lors des cours, nous apprenons à réaliser des sites, mais nous ne voyons pas comment les publier.

Avec M. Wallon, nous avons donc réservé une journée pour déployer le site sur l'hébergeur OVH et tout s'est bien passé. Les détails techniques du déploiement sont présentés plus bas dans le rapport.

1.3.5 Bilan général

- Bilan personnel

Grâce à ce stage, j'ai découvert de nombreuses choses sur le framework Laravel. Bien que nous l'ayons étudié durant les cours cette année, j'ai mieux compris comment établir des liaisons entre les tables.

J'ai également appris à générer un fichier PDF en y ajoutant les informations souhaitées. Cette fonctionnalité, très utile, est en réalité très facile d'utilisation.

De plus, j'ai réussi à envoyer des e-mails à partir d'une application Laravel, chose que je n'avais pas réussi à accomplir lors des TPs en classe. Cette compétence est particulièrement utile pour créer un formulaire de contact.

Pour finir, j'ai appris comment déployer un site qui n'est pas un site vitrine. J'avais déjà publié un site gratuitement sur GitHub mais c'était un site qui ne comportait pas de back-end, il se contentait juste d'afficher des informations. Donc c'est vraiment un point intéressant que j'ai découvert grâce au projet.

- Bilan pour l'association

Ce stage a eu un impact positif pour l'Association des Naturalistes de la Gohelle (ANG). En effet, la refonte du site web a permis de moderniser et de dynamiser la communication de l'association, rendant le site plus attrayant et fonctionnel. Grâce à notre travail, l'ANG dispose désormais d'une plateforme interactive et responsive, accessible sur tout type de dispositifs, ce qui améliore l'expérience utilisateur. L'association peut ainsi continuer à sensibiliser et à éduquer le public sur les enjeux environnementaux, à partager ses recherches et observations, et à renforcer ses collaborations avec d'autres organismes. En somme, ce projet de refonte a répondu aux besoins actuels de l'ANG, mais a permis de faciliter la mission de l'association dans la protection de la biodiversité régionale.

- Après l'IUT

Ce stage n'a pas modifié mon choix de continuer en BUT 3ème année car j'avais déjà la volonté de continuer en formation initale. En revanche, il m'a conforté dans mon orientation professionnelle en me confirmant mon désir de devenir développeur fullstack. En travaillant sur ce projet concret de refonte de site web avec le framework Laravel, j'ai pu renforcer et approfondir mes compétences techniques en développement web, tant sur le front-end que sur le back-end.

2 Rapport technique

2.1 Introduction

Comme expliqué dans la partie note de synthèse, nous avons commencé le stage par une analyse de l'existant. Avant de passer au développement, il était essentiel de comprendre les subtilités du site actuel. Celui-ci comprend trois grandes pages principales :

- Activités

La page "Activités" permet aux administrateurs du site de publier des événements organisés par l'association. Cela pourrait être par exemple l'escape game "pollinisateurs" qu'ils ont organisé le 05 Juin 2024 Avec pour intrigue :

Je suis entomologiste : je m'occupe de l'étude et de la protection des insectes. Aujourd'hui, j'ai besoin de votre aide. Je vis actuellement à Lestrem où j'étudie la vie des abeilles sauvages. Dans mon jardin, j'ai une grande verrière où 951 espèces d'abeilles vivent leur vie tranquillement. Je peux ainsi les observer régulièrement et étudier leur comportement. Hier, lorsque je suis entrée dans la verrière pour les observer, une abeille s'est envolée !
Votre mission sera de retrouver de quelle abeille il s'agit. Mes abeilles sont très joueuses, celle qui s'est envolée aura probablement semé des énigmes un peu partout pour vous permettre de la retrouver.

- Sorties

La page "Sorties" permet quant à elle aux administrateurs de publier des événements d'observation collective. Par exemple, une sortie pour observer les orchidées du Bois des Bruyères a été organisée le 21 avril 2024.

- Observations

Pour finir, il reste la page "Observations" dans laquelle chaque membre pourra poster ses propres observations.

2.2 Refonte de la base de données

2.2.1 État de la base de données lors de la récupération du projet

Lorsque nous avons récupéré le projet, la base de données présentait plusieurs incohérences. Nous avons donc décidé, avec l'accord de M. Wallon, d'en faire la refonte.

Comme mentionné dans la note de synthèse, nous avons renommé les tables relatives aux activités et aux observations pour une meilleure compréhension.

Un autre problème était la gestion des administrateurs : il y avait une table `users` et une table `admins` contenant des champs similaires (name, email, password). Ces tables pouvaient en réalité être fusionnées.

2.2.2 Solutions apportées

Pour corriger ce problème, nous avons supprimé la table `admins` et créé une table nommé `roles` qui contient le champ `id_role` et `name`.

Après cela, nous avons pu ajouter différents rôles tels que "Admin", "Moderateur", "Adherent" (pour les membres de l'association), "En Attente" (pour les utilisateurs ayant créé un compte mais n'étant pas encore membres), et "Banni" qui représente un utilisateur banni pour mauvais comportement.

	id_role	name
1	1	Admin
2	2	Moderateur
3	3	Adherent
4	4	En Attente
5	5	Banni

Capture d'écran du résultat après la création des rôles

Nous avons ajouté le champ `id_role` comme clé étrangère dans la table `users`. Ainsi, en attribuant ce champ à 1, l'utilisateur devient un administrateur.

Il manquait également différentes tables tels que `tags`, `pieces_jointes`, `commentaires`, et `sorties`.

- Table `tags` :

Utile pour les activités, cette table contient une clé primaire et un nom, permettant de catégoriser les activités avec des tags tels que "Jeux", "Photographies", etc.

	id_tags	name_tags
1	1	Jeux
2	2	Photographie

Capture d'écran d'un exemple de tags

- Table `pieces_jointes` :

Contenant les champs `id_pj`, `file_name` (nom du fichier), `file_path` (chemin du fichier), `file_type` (format, ex. PDF), `uploaded_by` (id de l'utilisateur) et `id_activite` (lien avec l'activité), cette table permet aux administrateurs d'ajouter des pièces jointes aux activités.

	id_pj	file_name	file_path	file_type	uploaded_by	id_activite
1	1	20240626122315-Réglement intérieur au 09021…	storage/files/activite/20240626122315-Régle…	pdf	1	51
2	2	20240626122315-Statuts-ANG.pdf	storage/files/activite/20240626122315-Statu…	pdf	1	51
3	3	20240626122315-sorties-2024.pdf	storage/files/activite/20240626122315-sorti…	pdf	1	51

Capture d'écran d'un exemple de pièces jointes

- Table `commentaires` :

Cette table permet aux membres de commenter les activités. Elle comporte les champs `id_commentaire`, `commentaire` (texte du commentaire), `date_commentaire` (date de publication), `id_activite` (lien avec l'activité) et `id_user` (auteur du commentaire).

	id_commentaire	commentaire	date_commentaire	id_activite	id_user
1	1	C'est trop bien, j'ai adoré !	2020-03-08	6	9
2	2	Belles photos	1994-10-13	25	9
3	3	La nature est vraiment magnifique	1993-04-23	18	10
4	4	C'est ce que je préfère les noix !	2022-07-31	48	4
5	5	Les insectes sont très important pour les s…	2001-07-09	26	3

Capture d'écran d'un exemple de commentaires

- Table `sorties` :

Absente initialement, nous l'avons créée avec les champs `id_sortie`, `title`, `description`, `image`, `date_sortie`, `pour_membre` (booléen indiquant si la sortie est réservée aux membres), et `guide` qui contient le nom-prénom du guide de la sortie.

	id_sortie	title	description	image	date_sortie	pour_membre	guide
1	1	Ipsa quae magni doloremque.	Omnis reprehenderit et a…	6d0b945dd66b24e8ac0ef80a78827dd2.png	2024-04-24 02:23:14	0	Zoé Lecomte
2	2	Aut harum voluptatum sit.	Aspernatur fugiat volupt…	ba41d3272d61c949a8328258cdcbf951.png	2024-06-29 01:43:54	0	Alix du Daniel
3	3	Perspiciatis eum cum.	Voluptatem voluptatibus …	a52d3229e276990fe747fb3f3a7c2c1f.png	2023-12-25 22:25:34	1	Maryse de Leduc
4	4	Libero nobis cum.	Repellat quibusdam eos i…	136864131e5e2d82b07db2aa2cce781d.png	2024-03-25 15:06:41	1	Zacharie Ollivier-Schmitt
5	5	Fuga veniam.	Id id quaerat dolores. S…	329fc0940aed8b62ef2e26812a08da4b.png	2023-10-09 21:53:16	0	Pénélope Benard
6	6	Aut sit at.	Laudantium pariatur qui …	e98bf0c8cd012fca6dad943fd113aeda.png	2023-10-30 18:18:22	0	Gilles du Petit
7	7	Consequuntur qui ratione rep…	Culpa dicta recusandae o…	1a9485b7b3658d6be3415c4b2acb4e1f.png	2024-10-23 15:33:08	1	Juliette Maury
8	8	Dolore qui id.	Saepe alias beatae sequi…	b21e6a4deab7c20dc4ff3470fe4811c4.png	2025-06-25 05:49:43	1	Monique Loiseau-Laporte
9	9	Quis quidem voluptas.	Eius alias non quaerat e…	dbc5841d2765e746261277291cfca982.png	2024-06-12 05:15:52	0	Hélène Petitjean
10	10	Assumenda illum.	Animi aliquid dolor qui …	42743fce4f501b548c5e603f8399f725.png	2024-06-30 22:47:19	0	Robert Dupuy

Capture d'écran d'un exemple de sorties

2.2.3 Résultats

Après refonte de la base de données, nous obtenons les diagrammes suivants :

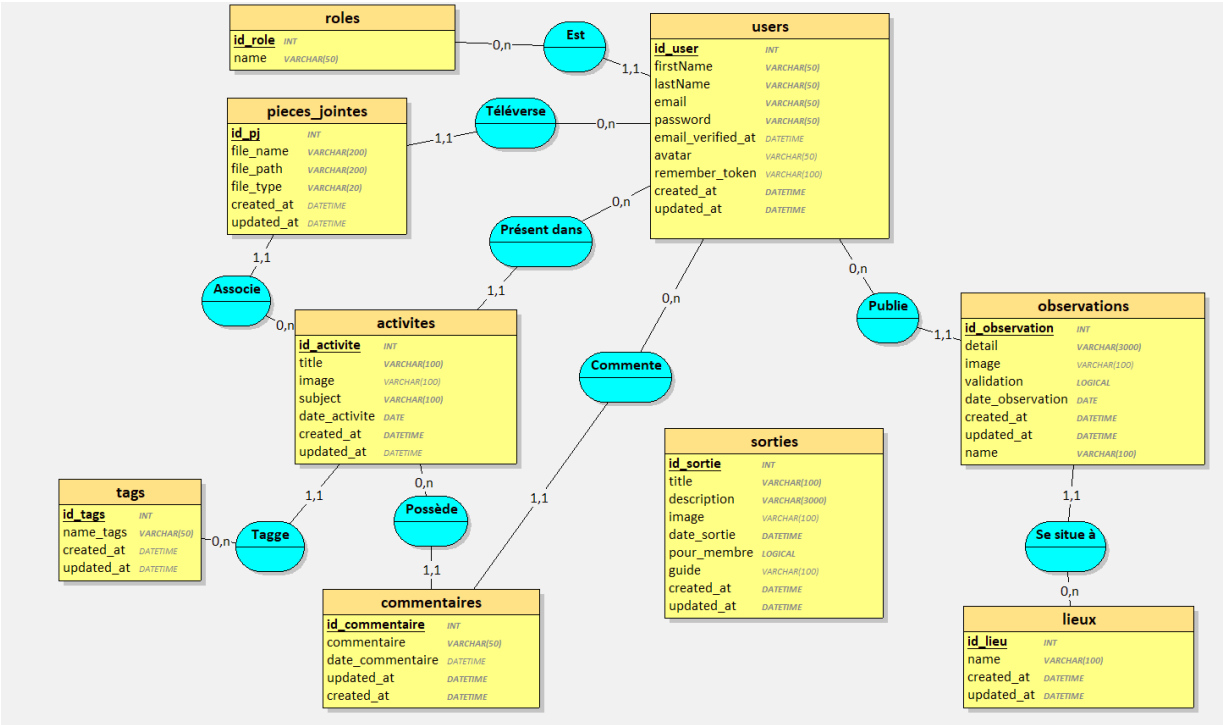


Diagramme entité-association de la base de données

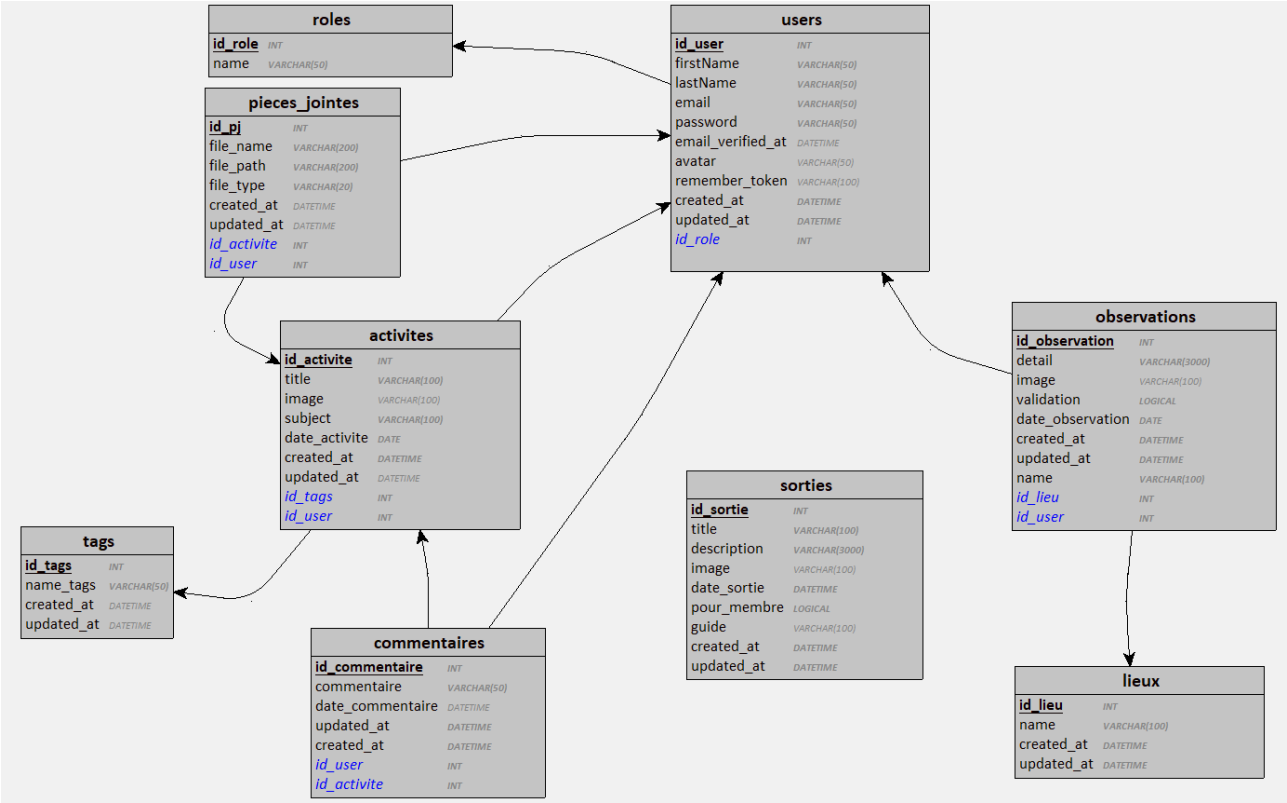


*Diagramme MLD (*Modèle Logique des Données*) de la base de données*

2.3 CRUD pour la page "Observations"

Je vais maintenant vous présenter le CRUD de la page "Observations". L'acronyme CRUD désigne les opérations CREATE (Créer), READ (Lire), UPDATE (Mettre à jour) et DELETE (Supprimer) sur les données.

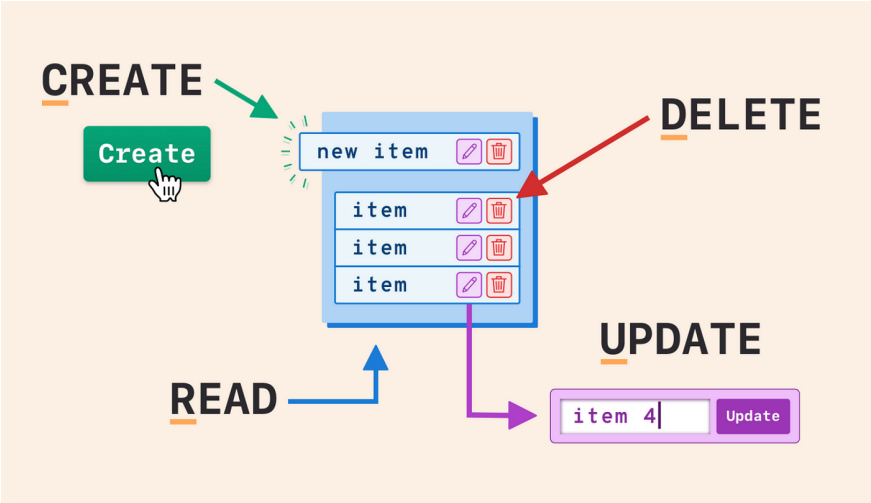


Schéma de ce que représente un CRUD

Source : João Neto Florencio sur LinkedIn

Le framework Laravel utilise l'architecture MVC (Model-View-Controller). Cette architecture permet de développer plus aisément une application en PHP en la séparant en trois grandes composantes :

- Model (Modèle) :

Le modèle s'occupe de récupérer les informations de la base de données, de les manipuler, et de les envoyer au contrôleur.

- Controller (Contrôleur) :

Le contrôleur est un intermédiaire entre les modèles et les vues. Il reçoit les demandes des utilisateurs, récupère les données auprès des modèles et les transmet aux vues.

- View (Vue) :

La vue présente les données à l'utilisateur. Elle prend les informations fournies par le contrôleur et les affiche sous forme de pages web.

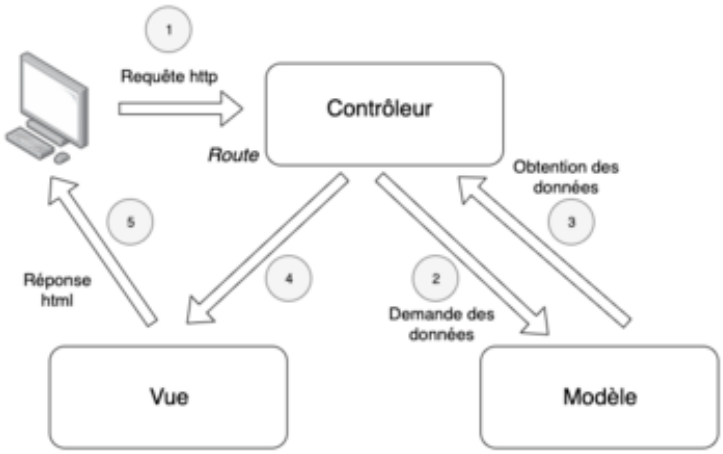


Schéma expliquant les relations d'une architecture MVC

Source : Benoît Prieur sur Wikipédia

Le CRUD est implémenté dans les contrôleurs. Par exemple, nous avons un contrôleur nommé `ObservationsController` dans lequel nous trouvons toutes les fonctions du CRUD que je vais vous présenter.

2.3.1 Affichage des observations

Pour commencer, la fonction `index` permet d'afficher toutes les observations enregistrées.

Cette fonction débute par : `$lieux = Lieux::orderBy('name')->get();`

Cela permet de récupérer les lieux dans l'ordre alphabétique en se basant sur le champ `name`. Cette liste est utilisée dans la vue pour permettre à l'utilisateur de filtrer les observations par lieu de manière ordonnée.

Ensuite, nous avons : `$query = Observation::where('validation', true);`
Ce code récupère les observations dont le champ `validation` est à "true". Cela signifie qu'un administrateur a validé l'observation concernée. De cette manière, nous affichons seulement les observations qui sont validées.

J'ai également implémenté un système de filtrage permettant à l'utilisateur de choisir de trier les observations par date, de manière ascendante ou descendante, et par lieu :

```
if ($request->has('lieu_id')) {
    $query->where('id_lieu', $request->lieu_id);
}

if ($request->has('sort')) {
    $sort = $request->sort;
    $query->orderBy('date_observation', $sort === 'asc' ? 'asc' : 'desc');
} else {
    $query->latest('date_observation');
}
```

Le premier bloc de code vérifie si un identifiant concernant les lieux (`id_lieu`) est présent dans la requête et filtre les observations par ce lieu. Le deuxième bloc vérifie la présence d'un paramètre de tri (sort) et trie les observations par date d'observation en fonction de ce paramètre (asc ou desc). Si aucun paramètre de tri n'est fourni, les observations sont triées par date d'observation décroissante (latest).

Il a rapidement été nécessaire d'ajouter un système de pagination car il y avait beaucoup d'observations à afficher. C'est pourquoi j'ai ajouté le code :

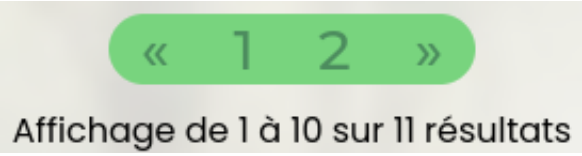
```
$perPage = 10;
$observations = $query->paginate($perPage)->appends($request->query());
```

Cela permet d'afficher 10 observations par page grâce à la variable `$perPage`.

À la fin de la fonction, le code retourne la vue correspondant à l'affichage des observations :
`return view('observations.index', compact('lieux', 'observations'));`

Capture d'écran de l'affichage de la première observation

Sur la capture d'écran ci-dessus, avec la vue que Lucas a créée, vous pouvez voir les filtres mentionnés dans la fonction `index`. La vue affiche également la liste des observations, dont un exemple est présenté ci-dessus. Vous trouverez également la pagination en bas de la page :



Capture d'écran de la pagination

2.3.2 Affichage des informations pour une observation donnée

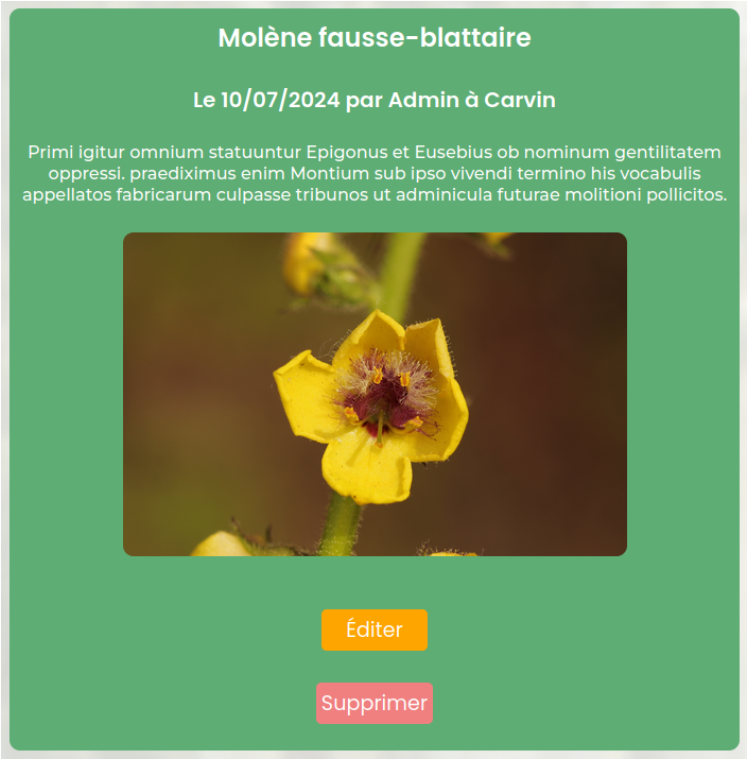
Pour afficher les détails d'une observation, on utilise la fonction `show` qui est la suivante :

```
public function show(int $id): View {
    $observation = Observation::findOrFail($id);
    return view('observations.show', compact('observation'));
}
```

Dans cette fonction, le paramètre `$id` correspond à l'identifiant de l'observation sur laquelle l'utilisateur a cliqué. Ce paramètre permet de recherche l'observation correspondante dans la base

de données avec la méthode `findOrFail($id)` qui renvoie l'observation si elle est trouvée, ou déclenche une erreur si elle ne l'est pas.

Pour finir, la fonction retourne la vue `observations.show` avec l'observation récupérée, grâce à la fonction `compact('observation')` qui passe la variable `$observation` à la vue.



Capture d'écran de la vue `show`

Lorsque l'utilisateur clique sur une observation, il est redirigé vers la vue `show` où il peut voir la description complète. De plus, pour les utilisateurs disposant des droits appropriés, deux boutons apparaissent pour éditer et supprimer l'observation.

2.3.3 Création d'une observation

Pour créer une observation, nous avons besoin de deux fonctions.

La première fonction est `create`. Elle permet simplement d'afficher la vue qui contient le formulaire de création d'une observation.

Afin que l'utilisateur puisse choisir un lieu pour créer son observation, il faut d'abord envoyer à la vue tous les lieux présents dans la base de données. Pour ce faire, on utilise la requête suivante :

```
$lieux = Lieux::all();
```

Ensuite, nous avons une partie de code qui permet de récupérer les anciennes valeurs du formulaire dans les cookies. Avant l'utilisation des cookies, lorsqu'un utilisateur oubliait de remplir un champ et validait le formulaire, cela créait une erreur et supprimait tout le reste du formulaire aussi. Désormais, même après une erreur, cela sauvegarde les autres champs.

```
$name = Cookie::get('name');
$detail = Cookie::get('detail');
$date = Cookie::get('date');
$name_lieu = Cookie::get('name_lieu');
```

Et pour finir nous envoyons toutes ces informations à la vue :

```
return view('observations.create', [
   'lieux' => $lieux,
   'name' => $name,
   'detail' => $detail,
   'date' => $date,
   'name_lieu' => $name_lieu,
]);
```

La deuxième fonction est `store`. C'est cette fonction qui enregistre les données du formulaire. Les lignes suivantes suppriment les cookies utilisés pour sauvegarder les valeurs du formulaire en cas d'erreur de validation précédente, garantissant que les anciennes valeurs ne persistent pas après un enregistrement réussi.

```
Cookie::unqueue('name');
Cookie::unqueue('detail');
Cookie::unqueue('date');
Cookie::unqueue('name_lieu');
```

Le code ci-dessous définit les règles de validation pour chaque champ du formulaire :

- `required` qui signifie que le champ est requis pour créer l'observation.
- `nullable` qui signifie que le champ peut être vide pour créer l'observation.
- `image` permet de vérifier que le champ est une image.
- `mimes` permet de vérifier le type du fichier autorisé (ici jpeg, png, jpg, gif, svg).
- `max` permet de limiter la taille maximale du fichier téléchargé (ici 10 Mo).

```
$request->validate([
   'name' => 'required',
   'detail' => 'nullable',
   'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:10000',
   'date' => 'required',
   'name_lieu' => 'required',
   'nom_lieu' => 'nullable',
]);
```

Ensuite on récupère les valeurs du formulaire dans la variable `$input`. Puis, la requête vérifie si le lieu spécifié existe dans la base de données. Si c'est le cas, l'identifiant du lieu (`id_lieu`) est ajouté à `$input`. Sinon, l'utilisateur est redirigé vers la page précédente avec un message d'erreur indiquant que le lieu sélectionné est invalide.

```
$input = $request->all();

$lieu = Lieux::where('name', $request->name_lieu)->first();
if ($lieu) {
   $input['id_lieu'] = $lieu->id_lieu;
} else {
   return redirect()->back()->withErrors(['name_lieu' => 'Le lieu sélectionné est invalide.']);
}
```

Le code ajoute la date courante ainsi que l'utilisateur qui a publié l'observation et vérifie si l'utilisateur connecté est un administrateur ou un modérateur :

- Si oui, le champ validation est défini à "true" et un message de succès indiquant que l'observation a été validée est défini.
- Sinon, le champ validation est défini à "false" et un message indiquant que l'observation est en attente de validation est défini.

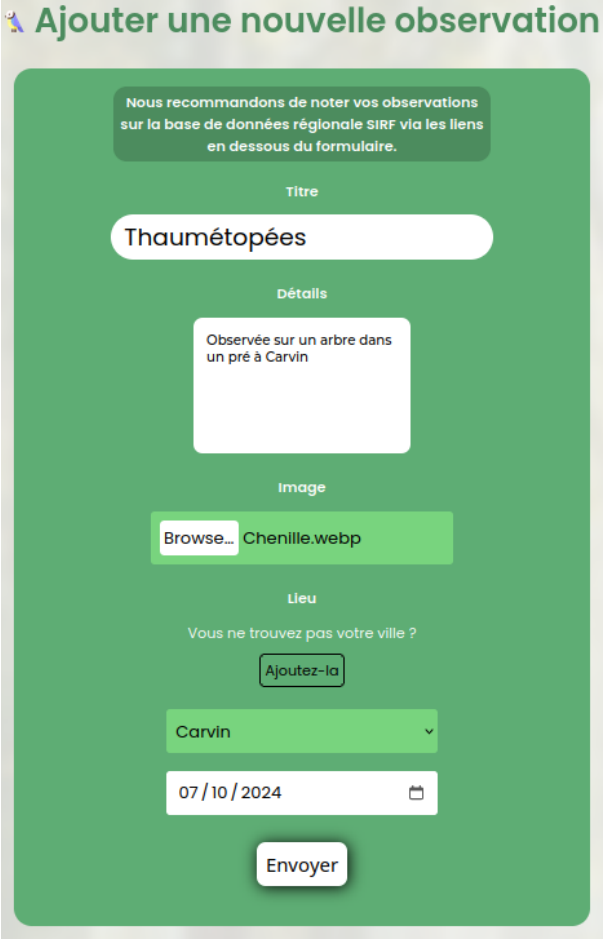
```
$input['id_user'] = Auth::id();
$input['date_observation'] = $request->date;

if (Auth::user()->isAdmin() || Auth::user()->isModerateur()) {
   $input['validation'] = true;
   $message = 'L\'observation a été envoyée et validée avec succès.';
} else {
   $input['validation'] = false;
   $message = 'L\'observation a été envoyée et est en attente de validation.';
}
```

Ensuite, l'observation est créée et l'image associée est ajoutée. Chaque image est nommée avec l'identifiant de l'observation, garantissant que chaque nom d'image est unique. L'image est stockée sur le serveur dans le répertoire "storage/img/observations".

```
$observation = Observation::create($input);

if ($image = $request->file('image')) {
   $destinationPath = 'storage/img/observations';
   $profileImage = $observation->id_observation . "_observation." . $image-
>getClientOriginalExtension();
   $image->move($destinationPath, $profileImage);
```

```
        $observation->image = $profileImage;
        $observation->save();
    }
```

Pour finir, la vue `observations.index` est retournée, affichant l'observation publiée avec un message de succès grâce à : `return redirect()->route('observations.index')->with('success', $message);`



Capture d'écran du formulaire de création d'observations

Ci-dessous, vous pouvez voir le résultat de la vue `create` qui affiche le formulaire permettant de saisir les données. Lorsque l'utilisateur clique sur le bouton, cela appelle la fonction `store` qui enregistre les données.

Capture d'écran suite à la création

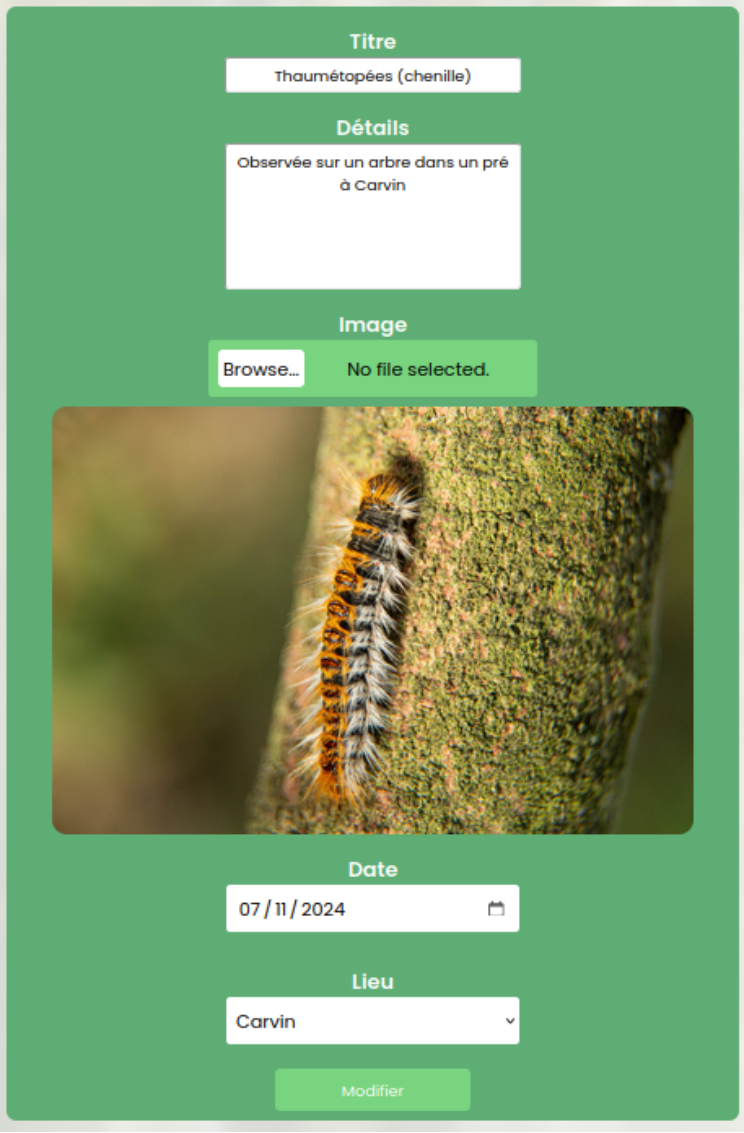
Suite à la création, on remarque que celle-ci s'est bien enregistrée et s'affiche dans la vue `index`.

2.3.4 Modification d'une observation

Pour modifier une observation, nous avons deux fonctions comme lorsque l'on crée une observation. La fonction `edit` permet d'afficher le formulaire tandis que `update` permet de mettre à jour les champs de la table. Ici je vais seulement évoquer la différence majeure car les fonctions `store` et `update` se ressemblent.

La principale différence concerne l'image. Dans `store`, l'image est requise alors qu'elle est optionnelle pour `update`. De cette manière, si une nouvelle image est uploadée, elle remplace l'ancienne sinon le champ image est retiré des données à mettre à jour, laissant l'image existante inchangée.

```
if ($image = $request->file('image')) {
   $destinationPath = 'storage/img/observations';
   $profileImage = $observation->id_observation . "_observation." . $image-
>getClientOriginalExtension();
   $image->move($destinationPath, $profileImage);
   $input['image'] = $profileImage;
} else {
   unset($input['image']);
}
```



Capture d'écran du formulaire de modification

Tout comme pour `create`, nous avons un formulaire de modification généré par la fonction `edit`. Lorsque l'utilisateur appuie sur le bouton, cela appelle la fonction `update` qui met à jour les données.

Capture d'écran après mise à jour de l'observation

Après la mise à jour, nous pouvons voir dans la vue `index` que le titre a bien été modifié.

2.3.5 Suppression d'une observation

Pour supprimer une observation, nous avons besoin que d'une seule fonction, `destroy`.

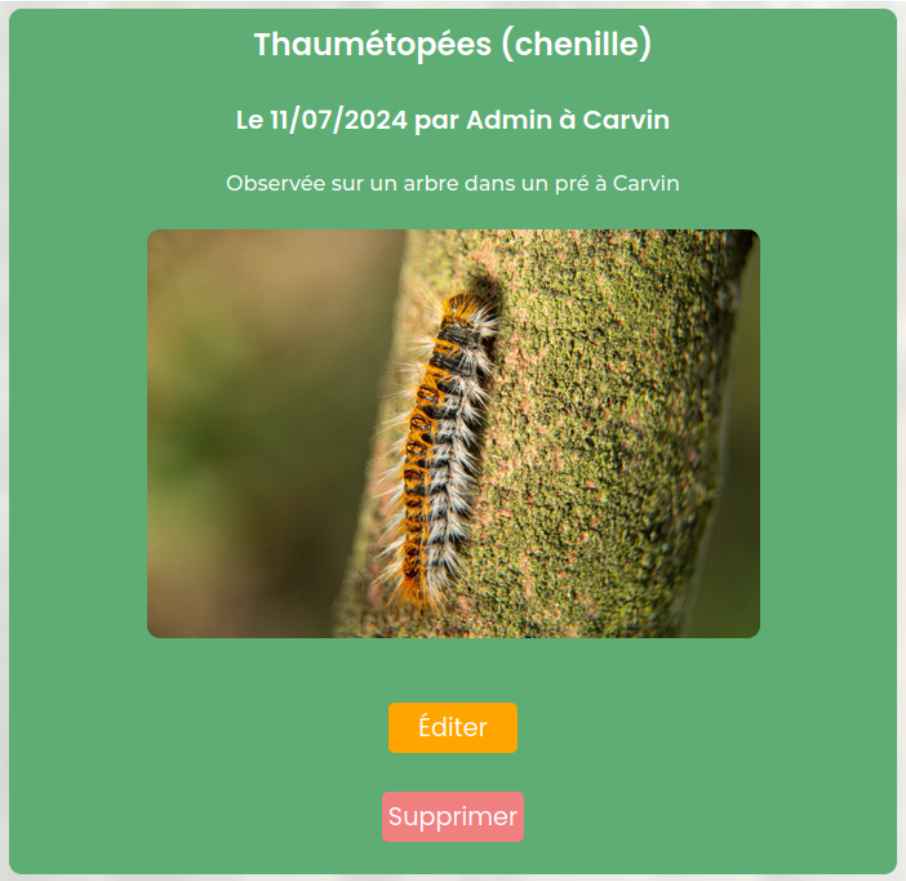
Cette fonction commence par rechercher l'observation que l'utilisateur souhaite supprimer et vérifie que l'utilisateur connecté est un administrateur. Seuls les administrateurs ont des droits définitifs comme la suppression par exemple. Ici, nous avons donc deux sécurités, une sur la route et une dans le contrôleur.

```
$observation = Observation::findOrFail($id);

if (!Auth::user()->isAdmin() && Auth::id() !== $observation->id_user) {
   return redirect()->route('observations.index')->with('error', 'Vous n\'êtes pas autorisé à
supprimer cette observation.');
}
```

La suite de la fonction s'occupe de la suppression de l'image stockée dans le serveur ainsi que de l'observation enregistrée au sein de la base de données. Pour finir, cela redirige sur la page principale d'observation.

```
$imagePath = 'public/img/observations/' . $observation->image;

if (Storage::exists($imagePath)) {
   Storage::delete($imagePath);
}

$observation->delete();

return redirect()->route('observations.index')->with('success', 'L\'observation a été
supprimée.');
```

Comme évoqué dans la partie `show`, l'utilisateur ayant les droits peut supprimer une observation dans la vue `show` en cliquant sur le bouton :



Capture d'écran de la vue `show` avec le bouton de suppression

Après la suppression, le contrôleur nous renvoie à la vue `index` avec un message indiquant que l'observation a bien été supprimée :

Capture d'écran de la vue `index` montrant le message de confirmation de suppression

Nous pouvons voir le message de confirmation de suppression ainsi que l'absence d'observations dans la liste, suite à la suppression de la seule observation de test que j'avais créée.

2.3.6 Routes

Pour lier les fonctions aux chemins URL tel que "https://www.assonaturegohelle.net/observations" par exemple, on utilise les routes. Ainsi pour toutes les fonctions réalisées au sein du contrôleur, nous avons les routes suivantes :

```
Route::get('observations', [ObservationsController::class, 'index'])->name('observations.index');

Route::get('observations/create', [ObservationsController::class, 'create'])-
>name('observations.create')->middleware('auth', 'adherent');

Route::post('observations', [ObservationsController::class, 'store'])->name('observations.store')-
>middleware('auth', 'adherent');

Route::get('/observations/edit/{id}', [ObservationsController::class, 'edit'])-
>name('observations.edit')->middleware('auth', 'adherent');

Route::put('/observations/{id}', [ObservationsController::class, 'update'])-
>name('observations.update')->middleware('auth', 'adherent');

Route::get('/observations/{id}', [ObservationsController::class, 'show'])-
```

```
>name('observations.show');

Route::delete('/observations/{id}', [ObservationsController::class, 'destroy'])-
>name('observations.destroy')->middleware('auth', 'administrateur');
```

Comme vous pouvez le voir, nous avons utilisé les middlewares `auth` et `moderateur`, ainsi que `auth` et `adherent` pour sécuriser les routes. Cela garantit que seules les personnes autorisées peuvent accéder à certaines parties de l'application. Je vais expliquer en détail comment nous avons implémenté cette sécurité dans la prochaine section du rapport.

2.4 Gestion des rôles

Pour sécuriser le site, nous avons défini des rôles pour chaque utilisateur comme expliqué dans la partie "Refonte de base de données".

Il nous suffisait ensuite de restreindre l'accès à certaines routes en fonction des rôles. Pour ce faire, nous utilisons des middlewares. Par exemple, nous avons les middlewares `auth`, `administrateur`, `moderateur`, et `adherent`.

- `auth` : Vérifie si l'utilisateur est connecté.
- `administrateur` : Vérifie si l'utilisateur est connecté en tant qu'administrateur.
- `moderateur` : Vérifie si l'utilisateur est connecté en tant qu'administrateur ou modérateur.
- `adherent` : Vérifie si l'utilisateur est connecté en tant qu'administrateur, modérateur ou adhérent.

Par exemple, une page ayant pour middleware `adherent` est disponible au minimum pour un utilisateur ayant ce rôle. Les modérateurs et administrateurs doivent pouvoir logiquement avoir accès à cette page également.

Pour chaque middleware, nous créons un fichier similaire à celui ci-dessous.

```php
class AdherentMiddleware {
    /**
     * Handle an incoming request.
     *
     * @param Request $request
     * @param Closure $next
     * @return mixed
     */
    public function handle(Request $request, Closure $next) {
        $user = $request->user();

        // Vérifie si l'utilisateur est authentifié
        if (!$user) {
            return redirect('login');
        }

        // Les rôles autorisés pour accéder à cette route
        $roles = [1, 2, 3]; // 1 pour Admin, 2 pour Moderateur, 3 pour Adherent

        // Vérifie si l'utilisateur a l'un des rôles autorisés
        if (!in_array($user->id_role, $roles)) {
            abort(403, 'Accès non autorisé.');
        }

        return $next($request);
    }
}
```

Ce code vérifie si l’utilisateur est connecté. Sinon, il le redirige vers la page de connexion. Si l’utilisateur est connecté, le code vérifie qu'il possède le rôle d'administrateur, de modérateur ou d'adhérent. Sinon, l’accès à la page où le middleware est implémenté est refusé.

2.5 Génération de fichiers sous format PDF

2.5.1 Explication et installation

Concernant les sorties, l'association souhaitait pouvoir distribuer des tracts pour une sortie donnée ainsi qu'une liste des sorties de l'année en cours.

Pour ce faire, j'ai implémenté un export des données en PDF avec la bibliothèque `dompdf`. Étant donné que cette bibliothèque n'est pas directement intégrée à Laravel, je l'ai installé avec la commande suivante :

```
composer require dompdf/dompdf
```

2.5.2 Fonctionnement

Pour vous décrire le fonctionnement de la génération de PDF, je vais vous présenter le code utilisé pour générer un tract d'une sortie donnée.

Premièrement, nous avons la route qui lie le chemin URL à une fonction au sein du contrôleur :

```
Route::post('/generate-pdf/{id}', [SortiesController::class, 'generatePdf'])->name('generate-pdf');
```

De ce fait, lorsqu'un utilisateur télécharge une sortie au format PDF, cela appelle la fonction `generatePdf` qui est la suivante :

```
public function generatePdf(int $id): Response {
    $sortie = Sortie::findOrFail($id);
    $pdf = Pdf::loadView('sorties.sortie-pdf', compact('sortie'));
    return $pdf->download('sortie-'.$id.'.pdf');
}
```

Cette fonction commence par récupérer la sortie choisie par l'utilisateur, puis appelle la vue `sortie-pdf` avec les informations concernant la sortie qui sont stockées dans la variable `$sortie`.

Par la suite, avec du code assez simple, nous pouvons récupérer les données de la sortie. Par exemple avec :

```
$sortie->title
$sortie->description
$sortie->date_sortie
$sortie->image
$sortie->guide
```

Vous trouverez en annexe le résultat du tract pour une sortie choisie.

2.6 Gestion des mails

2.6.1 Explication et problèmes rencontrées

Nous devions implémenter un système d'envoi de mails pour le formulaire de contact et la fonctionnalité de réinitialisation de mot de passe. Pour ce faire, il me fallait trouver un service capable de gérer les mails gratuitement.

Après recherche, j'ai trouvé `Mailjet` qui permet d'envoyer jusqu'à 6000 mails par mois gratuitement, ce qui était suffisant pour l'association. J'ai donc par la suite ajouté le code permettant d'envoyer un mail mais j'obtenais toujours un problème de connexion avec le service `Mailjet` lors des différents tests.

Je supposais que le problème venait de la configuration du pare-feu du réseau de l'IUT en raison des différents tests effectués au sein du projet. J'avais raison puisqu'un jour au soir, j'ai effectué le test sur mon ordinateur sur le réseau de chez moi et cela a fonctionné.

2.6.2 Solution finale

Finalement, nous n'avons pas gardé la solution `Mailjet` car l'hébergeur (OVH) permettait également de créer des adresses mails personnalisées et était plus simple de gestion pour les membres de l'association.

Je vais vous présenter la fonctionnalité de mail pour la réinitialisation de mot de passe. Nous avons donc une vue dans laquelle l'utilisateur renseigne l'adresse mail.

Capture d'écran de la vue réinitialisation du mot de passe

Laravel avec la dépendance `Fortify` s'occupe de créer ce qu'on appelle un `token` pour chaque demande de réinitialisation. Laravel s'occupe d'appeler la fonction `sendPasswordResetNotification` présente dans le fichier modèle `User.php`.

```
public function sendPasswordResetNotification($token): void {
   $notification = new ResetPasswordNotification($token);
   $notification->toCustomMail($this);
}
```

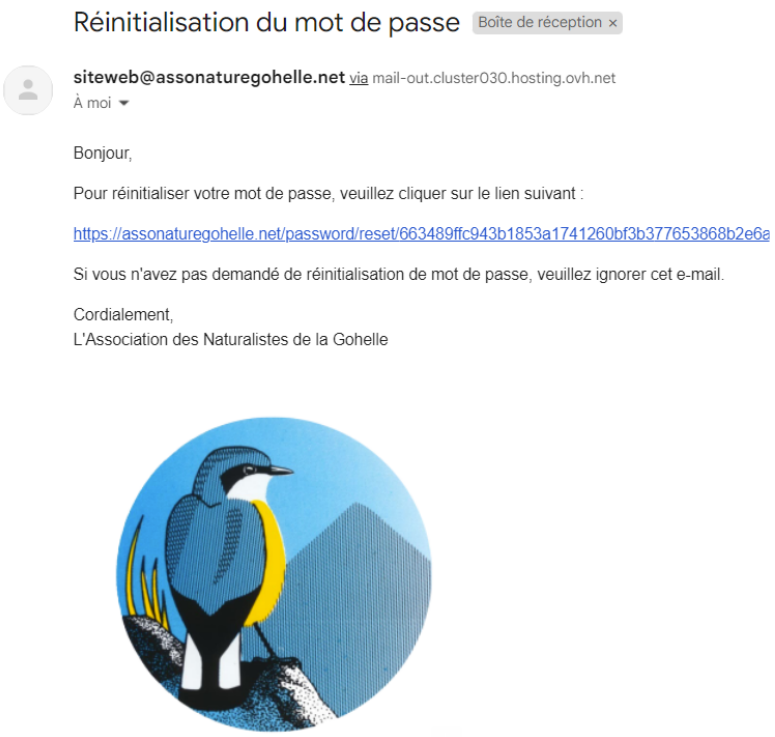
Cette fonction crée un objet contenant le token de réinitialisation pour l'utilisateur et appelle la fonction `toCustomMail` avec cet objet en paramètre.

```
$url = url(route('password.reset', $this->token, false));
$to = $notifiable->email;
$subject = 'Réinitialisation du mot de passe';

$logoUrl = asset('storage/img/logo.png');

$message = "
<html>
<head>
   <title>Réinitialisation du mot de passe</title>
</head>
<body>
   <p>Bonjour,</p>
   <p>Pour réinitialiser votre mot de passe, veuillez cliquer sur le lien suivant :</p>
   <p><a href='$url'>$url</a></p>
   <p>Si vous n'avez pas demandé de réinitialisation de mot de passe, veuillez ignorer cet e-
mail.</p>
   <p>Cordialement,<br>L'Association des Naturalistes de la Gohelle</p>
   <img src='$logoUrl' alt='Logo'>
</body>
</html>
";
```

Avec le code ci-dessus, nous récupérons les informations importantes pour créer le mail. Nous commençons par créer l'URL qui sera envoyée à l'utilisateur, lui permettant de réinitialiser son mot de passe. Ensuite, nous récupérons l'adresse mail de l'utilisateur, le sujet du mail, l'image qui sera affiché dans le mail et pour finir le contenu du message qui utilise le langage HTML et intègre les différentes valeurs.

```
$headers = "From: siteweb@assonaturegohelle.net\r\n";
$headers .= "Reply-To: siteweb@assonaturegohelle.net\r\n";
$headers .= "MIME-Version: 1.0\r\n";
$headers .= "Content-Type: text/html; charset=UTF-8\r\n";

mail($to, $subject, $message, $headers);
```

Pour finir, nous définissons les en-têtes pour préciser l'adresse mail de l'expéditeur et indiquer au système de gestion des mails qu'il doit interpréter le message comme du HTML afin de l'afficher correctement.



Capture d'écran du mail de réinitialisation reçu

Vous pouvez voir ci-dessus le résultat du mail envoyé après une demande de réinitialisation de mot de passe.

Capture d'écran après avoir cliqué sur le lien

Après avoir cliqué sur le lien et entré les informations valides, l'utilisateur peut choisir un nouveau mot de passe.

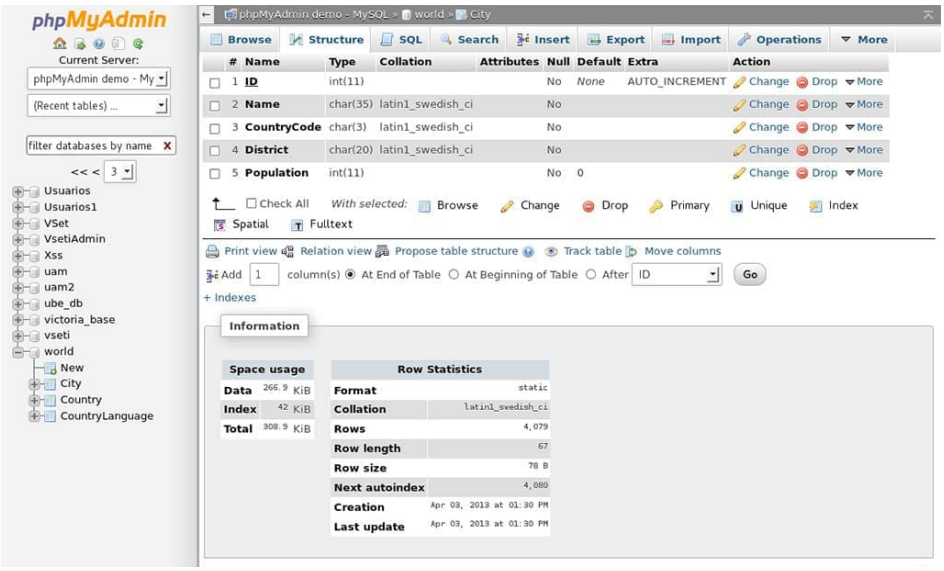
2.7 Déploiement du site sur l'hébergeur

Nous avons réalisé le déploiement du site à trois avec mon collègue Lucas et supervisé par M. Wallon car c'était notre première expérience en termes de déploiement. Nous avons commencé par importer la base de données car le site ne fonctionnerait pas sans elle.

2.7.1 Importation de la base de données

Pour importer la base de données, nous avons dû changer de système de gestion de base de données (SGBD). Pendant le développement, nous utilisions "SQLite". Cependant, sur OVH, nous devons utiliser "MySQL".

Ne parvenant pas à nous connecter directement à la base de données distante, nous avons créé une base de données localement via "phpMyAdmin".



Exemple de base de données sur phpMyAdmin

Source : locallhost.me

Nous avons ensuite lié le projet à la base de données MySQL locale en utilisant le fichier `.env` qui définit les principales variables du projet telles que les informations de connexion à la base de données. Voici un exemple de configuration :

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=nom_de_la_base_de_donnees
DB_USERNAME=nom_utilisateur
DB_PASSWORD=mot_de_passe
```

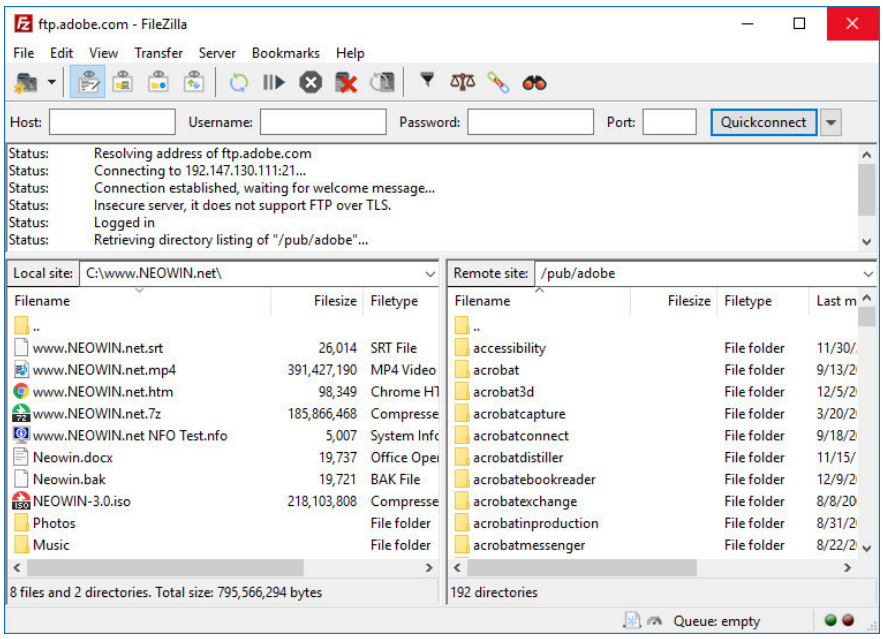
Ensuite, nous avons utilisé les commandes du projet pour créer la base de données et les tables. Voici les commandes :

```
php artisan migrate
php artisan db:seed
```

Après cela, nous avons pu voir la base de données MySQL dans "phpMyAdmin" en local. Nous l'avons donc exportée pour l'importer dans "phpMyAdmin" sur le serveur de l'hébergeur. Une fois exportée, la base de données était prête à fonctionner.

2.7.2 Transfert des fichiers

Après avoir importé la base de données, il ne restait plus qu'à transférer les fichiers du projet sur l'hébergement. Pour ce faire, nous avons utilisé le logiciel FileZilla, qui utilise le protocole FTP (File Transfer Protocol). FTP est un protocole de communication destiné au partage de fichiers sur un réseau. Ce logiciel affiche deux arborescences : celle de l'ordinateur et celle du serveur de l'hébergeur.



Exemple d'arborescences sur FileZilla

Source : neowin.net

Pour transférer les fichiers vers le serveur, il suffit de les glisser-déposer de la gauche (sur l'ordinateur) vers la droite (sur le serveur).

Après cela, en accédant au site "https://assonaturegohelle.net/", une erreur de version est apparue. Nous avions utilisé PHP 8.2 pour le développement, tandis que le serveur était en PHP 7.3. Ne pouvant pas mettre à jour PHP côté serveur, nous avons dû rétrograder la version PHP ainsi que toutes les dépendances nécessitant PHP 8.2. Nous avons utilisé la commande suivante :

```
composer require php:^7.3 -update-with-all-dependencies
```

Après modification de la version, le site s'affichait bien sans erreurs.

3 Annexes

3.1 Portfolio

Compétences mobilisées dans le cadre du stage de deuxième année

Compétence 1

Réaliser un développement d'application	Tâche	Pages
en respectant les besoins décrits par le client	CRUD de sorties	01-10
	Génération de PDF	01-10
	Envoi de mail	01-10
en veillant à la qualité du code et à sa documentation	Indentation du code	01-10
	Documentation du code	01-10
	Édition du README	01-10
en choisissant les ressources techniques appropriées	Utilisation de dépendances	02-10
	Authentification avec Fortify	02-10

Compétence 2

Optimiser des applications	Tâche	Pages
en analysant et en modélisant les attentes du client	Réalisation de diagrammes MLD	03-10
en adaptant des modèles de solution	Utilisation de l'architecture MVC	03-10
	Utilisation de middleware	03-10
en justifiant ses choix techniques et en validant les résultats	Création de captcha	03-10
	Gestionnaire de mails	03-10

Compétence 4

Gérer des données de l'information	Tâche	Pages
en définissant et exploitant une base de données adaptée aux besoins	Refonte de la base de données	05-10
en assurant la cohérence et la qualité	Attribution de nullable pour certains attributs	05-10
	Utilisation de clés étrangères	05-10
en assurant la protection des données	Sécurité des routes avec les middleware	05-10

Compétence 5

Conduire un projet	Tâche	Pages
en adoptant une démarche proactive, créative et critique	Proposition de fonctionnalités	07-10
	Création de labels GitLab	07-10
en communiquant efficacement avec les différents acteurs d'un projet	Utilisation de Mattermost	07-10
	Communication à l'oral (réunions)	07-10

Compétence 6

Collaborer au sein d'une équipe informatique	Tâche	Pages
en attribuant des rôles aux membres de son équipe	Attribution des tickets GitLab	09-10
en utilisant des outils facilitant la communication et le travail collaboratif	Utilisation de Mattermost	09-10
	Gestion des tickets GitLab	09-10
en utilisant des technologies de dernière génération	Version de PHP 7.3	09-10
	Version de Laravel 8.83	09-10
en veillant au respect des contraintes juridiques	Suppression du profil/des données	10-10

Mon Portfolio

2ème Année de BUT Informatique

Théo Vanbandon

2023-2024

Table des matières

Compétence 1 - Réaliser un développement d'application **1**
En respectant les besoins décrits par le client 1
En veillant à la qualité du code et à sa documentation 1
En choisissant les ressources techniques appropriées 2

Compétence 2 - Optimiser des applications **3**
En analysant et en modélisant les attentes du client 3
En adaptant des modèles de solution 3
En justifiant ses choix techniques et en validant les résultats 3

Compétence 4 - Gérer des données de l'information **5**
En définissant et exploitant une base de données adaptée aux besoins 5
En assurant la cohérence et la qualité 5
En assurant la protection des données 5

Compétence 5 - Conduire un projet **7**
En adoptant une démarche proactive, créative et critique 7
En communiquant efficacement avec les différents acteurs d'un projet 7

Compétence 6 - Collaborer au sein d'une équipe informatique **9**
En attribuant des rôles aux membres de son équipe 9
En utilisant des outils facilitant la communication et le travail collaboratif 9
En utilisant des technologies de dernière génération 9
En veillant au respect des contraintes juridiques 10

Compétence 1 - Réaliser un développement d'application

En respectant les besoins décrits par le client

Dans le cadre du projet de refonte du site de l'association, j'ai travaillé sur le CRUD pour les vues sorties. Ma contribution dans cette tâche peut être observée dans le contrôleur `SortiesController` et sur le site avec l'URL `https://www.assonaturegohelle.net/sorties`.

Cette tâche m'a permis de respecter les besoins décrits par le client car il était demandé de construire une page de sorties dans laquelle les administrateurs peuvent publier les prochaines sorties de l'association.

J'ai également respecté les besoins décrits par le client en réalisant la fonctionnalité de génération de PDF. En effet, chaque sortie devait pouvoir être exportée en fichier PDF.

De plus, il était nécessaire de réaliser l'envoi de mails au sein du site. Cela est utilisé pour réinitialiser le mot de passe et le formulaire de contact.

En veillant à la qualité du code et à sa documentation

Dans le cadre du projet de refonte du site de l'association, j'ai travaillé sur de nombreux fichiers en PHP pour lesquels j'ai ajouté de la documentation. Ma contribution dans cette tâche peut être observée surtout avant les fonctions présentes au sein des contrôlleurs.

Cette tâche m'a permis de veiller à la qualité du code et à sa documentation car en cas de lecture du code par une autre personne, elle comprendra comment il a été réalisé.

J'ai également porté attention à l'indentation qui permet de repérer facilement les structures telles que les classes, fonctions, blocs conditionnels (if…else).

Pour finir, nous avons ajouté les instructions permettant de lancer le projet dans le fichier README sous le format Markdown.

En choisissant les ressources techniques appropriées

Dans le cadre du projet de refonte du site de l'association, j'ai travaillé sur l'authentification. Ma contribution dans cette tâche peut être observée lors de la connexion et dans les fichiers `Fortify`.

Cette tâche m'a permis de choisir les ressources techniques appropriées car il a fallu trouver une solution permettant la connexion d'un utilisateur, et j'ai donc choisi `Fortify` qui permet cela.

De plus, j'ai dû réaliser un captcha pour la création de compte et le formulaire de contact. Pour ce faire, j'ai choisi la dépendance `gregwar`. De la même manière j'ai choisi la dépendance `dompdf` pour la création de PDF.

Compétence 2 - Optimiser des applications

En analysant et en modélisant les attentes du client

Dans le cadre du projet de refonte du site de l'association, j'ai travaillé sur la refonte de la base de données. Ma contribution dans cette tâche peut être observée dans les fichiers migrations, seeders, et factory qui se trouvent au sein du projet.

Cette tâche m'a permis d'analyser et modéliser les attentes du client car avant de réaliser la refonte, nous avons présenté un diagramme `MLD` (Modèle Logique de Données) qui représente un schéma de la future base de données.

En adaptant des modèles de solution

Dans le cadre du projet de refonte du site de l'association, j'ai travaillé sur le cadriciel Laravel.

Cette tâche m'a permis d'adapter des modèles de solution car j'ai utilisé l'architecture de Laravel (MVC - Modèle/Vue/Contrôleur) en utilisant les contrôleurs, modèles et vues au sein du projet.

J'ai également implémenter les `middleware` fournis par Laravel pour assurer la sécurité des routes.

En justifiant ses choix techniques et en validant les résultats

Dans le cadre du projet de refonte du site de l'association, j'ai travaillé sur la création de captchas. Ma contribution dans cette tâche peut être observée lors d'une demande de compte ou sur le formulaire de contact.

Cette tâche m'a permis de justifier mes choix techniques et d'en valider les résultats car au début j'avais importé et implémenté une dépendance comportant un captcha trop compliqué. Pour résoudre cela, j'en ai importé un autre, `gregwar` qui est plus simple en terme de résolution.

J'ai également travaillé sur l'envoi de mails au sein du site pour la réinitialisation des mots de passe ainsi que pour le formulaire de contact. Pour ce faire, j'ai commencé par utiliser le gestionnaire de

mails `mailjet` temporairement pour les tests. Ensuite, nous avons utilisé les adresses mails fournies par l'hébergeur `OVH` pour une solution à long terme.

Compétence 4 - Gérer des données de l’information

En définissant et exploitant une base de données adaptée aux besoins

Dans le cadre du projet de refonte du site de l’association, j’ai travaillé sur la refonte de la base de données. Ma contribution dans cette tâche peut être observée dans les fichiers migrations, seeders, et factory qui se trouvent au sein du projet.

Cette tâche m’a permis de définir et exploiter une base de données adaptée aux besoins car l’ancienne base de données ne répondait plus aux besoins actuels du site. Nous avons donc conservé les tables importantes, supprimé celles qui n’avaient plus lieu d’être, et effectué les modifications nécessaires sur les attributs. Nous exploitons ces tables dans les différents fichiers du site tels que les contrôleurs et vues.

En assurant la cohérence et la qualité

Dans le cadre du projet de refonte du site de l’association, j’ai travaillé sur les tables de la base de données. Ma contribution dans cette tâche peut être observée dans les fichiers migrations du projet.

Cette tâche m’a permis d’assurer la cohérence et la qualité car nous avons précisé quels champs pouvaient être nullables dans les différentes tables. Cela permet de clarifier quels champs de la table sont nécessaires pour pouvoir effectuer un enregistrement.

De plus pour désigner un utilisateur lors de la création d’une sortie par exemple, nous utilisons son identifiant sous forme de clé étrangère, ce qui est plus qualitatif que de créer un champ avec un nom.

En assurant la protection des données

Dans le cadre du projet de refonte du site de l’association, j’ai travaillé sur la sécurité du site. Ma contribution dans cette tâche peut être observée dans le fichier contenant les routes, où l’on retrouve les middleware.

Cette tâche m'a permis d'assurer la protection des données car lorsqu'une route est appelée, le middleware correspondant est invoqué pour vérifier les droits d'accès en fonction du rôle de l'utilisateur connecté.

Compétence 5 - Conduire un projet

En adoptant une démarche proactive, créative et critique

Dans le cadre du projet de refonte du site de l'association, j'ai travaillé sur la fonctionnalité de banissement. Ma contribution dans cette tâche peut être observée dans le panel administrateur.

Cette tâche m'a permis d'adopter une démarche proactive, créative et critique car elle n'était pas demandée mais je l'ai proposée suite à la création de la fonctionnalité de commentaires pour la page activités. Si un utilisateur entre un commentaire malhonnête, il pourra être banni par l'administrateur.

En communiquant efficacement avec les différents acteurs d'un projet

Dans le cadre du projet de refonte du site de l'association, j'ai créé des labels sur les tickets de GitLab.

Cette tâche m'a permis de communiquer efficacement avec les différents acteurs d'un projet car les labels comportent des couleurs, permettant de voir directement quelles tickets sont en cours de réalisation et quelles tickets sont à faire prochainement.

Pour communiquer, nous utilisions également le logiciel de messagerie instantanée `Mattermost` pour discuter à distance et demander des informations, par exemple.

Plus important encore, nous avions des réunions régulières où nous montrions nos avancées sur le site, ce qui permettait d'obtenir un retour client sur les améliorations ou modifications à effectuer.

Compétence 6 - Collaborer au sein d'une équipe informatique

En attribuant des rôles aux membres de son équipe

Dans le cadre du projet de refonte du site de l'association, nous avons utilisé le gestionnaire de projet GitLab dans lequel nous pouvons indiquer les tickets.

Cette tâche m'a permis d'attribuer des rôles aux membres de mon équipe car sur GitLab, nous pouvons nous attribuer des tâches à réaliser.

En utilisant des outils facilitant la communication et le travail collaboratif

Dans le cadre du projet de refonte du site de l'association, j'ai communiqué sur `Mattermost`.

Cela m'a permis d'utiliser des outils facilitant la communication et le travail collaboratif car M. Wallon n'était pas toujours présent sur place à l'IUT, il était donc plus rapide de communiquer sur Mattermost pour avoir un retour et faire les modifications au sein du projet.

Concernant le travail collaboratif, nous avions créé de nombreux tickets sur GitLab. De cette manière, nous voyions directement ce qu'il restait à faire et qui travaillait sur quoi.

En utilisant des technologies de dernière génération

Dans le cadre du projet de refonte du site de l'association, j'ai travaillé sur la version 7.3 de PHP et la version 8.83 de Laravel.

Cela m'a permis d'utiliser des technologies de dernière génération car chaque mise à jour contient de nouvelles fonctionnalités et l'hébergeur n'avait pas la même version PHP que celle installée sur nos machines. Il a donc fallu adapter les fichiers de configuration pour que cela fonctionne sur le serveur de l'hébergeur.

En veillant au respect des contraintes juridiques

Dans le cadre du projet de refonte du site de l'association, j'ai travaillé sur la suppression du profil utilisateur. Ma contribution dans cette tâche peut être observée dans la page du profil de l'utilisateur connecté.

Cette tâche m'a permis de veiller au respect des contraintes juridiques car un utilisateur connecté sur le site a la possibilité, dans la page profil utilisateur, de supprimer son compte et donc toutes données le concernant.

3.2 Lettre de mission

Théo Vanbandon
BUT informatique 2ème année

Année universitaire 2023-2024

Lettre de mission de stage

Le stage se déroulera du 21 Mai 2024 au 15 Juillet 2024 et la mission confiée est la suivante :

Contexte :

L'Association des Naturalistes de la Gohelle (ANG) est une association à but non lucratif œuvrant pour la préservation de l'environnement dans le bassin minier. Afin d'assurer sa visibilité, elle souhaiterait mettre en ligne une nouvelle version de son site web pour remplacer [http://assonaturegohelle.net/ | l'actuel] . Ce site web doit permettre à la fois de faire connaître ses actions dans le secteur, mais aussi de faire découvrir la biodiversité de la région au plus grand nombre avec des photographies prises par les membres de l'association.

Contenu de la mission :

Pour ce sujet de stage, il est proposé de travailler sur une évolution de l'application web dont le développement a débuté en PHP Laravel, afin de compléter les fonctionnalités qui y ont été implantées. Cette application devra permettre aux utilisateurs connectés (membres de l'association) de déposer des photographies dans une galerie d'images, et de saisir les actualités de l'association. L'interface doit être responsive pour s'adapter aux différents appareils (smartphones, tablettes, ordinateurs). Des fonctionnalités supplémentaires pourront être proposées suivant l'avancement du développement au cours du stage.

Moyens mis à disposition :

Les ordinateurs de l'IUT et la salle 05E

Lieu du stage :

L'IUT de Lens

Encadrement :

M. Romain Wallon

Théo Vanbandon

Vanbandon

Romain Wallon

Maître de conférences

3.3 Exemple de tract pour une sortie

Association des Naturalistes de la Gohelle

Une pharmacie dans mon jardin

Rejoignez notre sortie nature dédiée à la découverte des plantes sauvages, révélant les mystéres des herbes médicinales, explorez le monde secret de ces végétaux et apprenez à les identifier.

RDV à 14H 30 sur le parking des étangs du Petit-Sailly, rue de Bellenville prolongée à Sailly-Labourse.

(Tenue adaptée conseillée)

Date de sortie : **25/07/2024**

Commencement à **14:30**

Votre guide pour la sortie sera : **Vincent Eden62**

Sortie accessible pour tout le monde.

Plus d'informations sur notre Facebook ou notre site internet :