

Rapport

Projet Développement Web Full Stack

Introduction

Groupe composé de **Jeremy Allier**, chef de projet et responsable Back-end, **Quentin Augier**, responsable Front-end, et **Esteban Kropf**, responsable DevSecOps.

Ce projet consiste à développer une application web de gestion de projets en utilisant différents langages tel que HTML, CSS mais aussi des langages de programmation dynamique tel que PHP ou JavaScript. Le tout devra être lié à une base de données dans laquelle on stockera les informations.

Le développement a été réalisé principalement en PHP avec une base MySQL, en intégrant progressivement le front-end aux fonctionnalités backend. Ce dernier a été fait en utilisant du HTML et CSS.

Front-End - Quentin Augier.....	3
I. Déroulement du Projet.....	3
21/11 : Planification & Mise en place.....	3
24/11 : Apprentissage PHP.....	3
27/11 : Apprentissage JavaScript.....	4
29/11 : Création de la page d'accueil.....	5
01/12 : Création de la page de connexion.....	6
02/12 : Création de l'interface principale du site.....	8
II. Problèmes Rencontrés.....	14
1. Incompatibilité avec certains navigateurs.....	14
2. Problèmes avec Xampp.....	15
3. Adaptations aux différentes méthodes de travail.....	15
Back End.....	16
I. Déroulement du Projet.....	16
24/11: Planification & Mise en place.....	16
26/11: Lien SQL, php & Connexion à la BD.....	16
27/11: Inscription + Connexion dans login.php.....	17
28/11: Gestion des Projets et Tâches.....	17
01/12: Améliorations & Sécurité.....	18
02/12: Ajout de la Table Block & Réorganisation.....	19
03/12: Finalisation de la Gestion des Membres + Dossiers.....	20
II. Problèmes Rencontrés et Solutions.....	21
1. Problèmes de requêtes SQL.....	21
2. Problèmes de redirection PHP.....	22
3. Intégration du Front-End.....	22
DevSecOps.....	22
Conclusion.....	24

Front-End - Quentin Augier

I. Déroulement du Projet

21/11 : Planification & Mise en place

Lorsque nous avons commencé le projet nous nous sommes réunis pour répartir les différents rôles dans la création de celui-ci.

Ayant peu de connaissances en programmation je me suis proposé comme responsable front-end, domaine que j'ai eu l'occasion de découvrir lors de la semaine précédente, me permettant ainsi d'approfondir ces nouvelles connaissances.

Nous avons ensuite choisis de développer une plateforme de gestion de projets et avons défini un ordre de priorité des fonctionnalités de ce site.

Une fois notre vision du site claire, j'ai commencé par faire des recherches de plusieurs plateformes de gestion de projets afin de m'inspirer de leurs interfaces graphiques et de proposer des fonctionnalités à mes collègues.

24/11 : Apprentissage PHP

Pensant qu'il me serait essentiel pour le front-end, j'ai décidé de commencer par découvrir le langage PHP. Je me suis plus tard rendu compte que ce n'était pas forcément le cas, mais cela m'a néanmoins permis de mieux comprendre les explications de mon collègue responsable du back-end et à rendre mon travail plus simple lorsqu'est arrivé le moment de joindre nos codes.

Ainsi les deux premiers jours de cette semaine ont servi à apprendre le PHP en suivant principalement le cours d'OpenClassrooms avec lequel j'ai pu accomplir plusieurs exercices pour découvrir la syntaxe, les variables, tableaux et conditions.

```

<?php
    $recipes = [
        ['Canistrelli','[...]', 'miam', true, ],
        ['Frappes', '[...]', 'yum', false, ],
    ];
?>
<ul>
    <?php for ($miam = 0; $miam <= 1; $miam++): ?>
        <li><?php echo $recipes[$miam][0] . ' (' . $recipes[$miam][2] . ')'; ?></li>
    <?php endfor; ?>
<ul>

```

- Canistrelli (miam)
- Frappes (yum)

Exemple de code PHP que j'ai appris lors de ces deux jours, utilisant variables, tableaux et conditions, suivis de son affichage dans la page HTML.

27/11 : Apprentissage JavaScript

Après m'être rendu compte que le PHP n'était pas une priorité pour la partie front-end dont j'étais responsable, je suis passé à l'apprentissage du langage JavaScript, qui était plus proche de ce dont j'avais besoin.

Cette fois encore j'ai principalement utilisé le cours OpenClassrooms et leurs exercices afin de me familiariser avec la syntaxe, les variables, tableaux, conditions et fonctions qui, bien que je les avais déjà abordé avec le PHP, ne se présentaient pas exactement de la même façon.

Cet apprentissage me permettra plus tard, lors de la création de l'interface de notre plateforme de gestion de projet, de pouvoir, entre autres, créer un menu "sidebar" qu'il est possible de réduire afin de cacher son contenu.

```

function lancerJeu() {
    let choix = choisirPhrasesOuMots()
    let score = 0
    let nbMotsProposes = 0
    if (choix === "mots" ) {
        score = LancerBoucleDeJeu(listeMots)
        nbMotsProposes = listeMots.length
    } else {
        score = LancerBoucleDeJeu(listePhrases)
        nbMotsProposes = listePhrases.length
    }
    afficherResultat(score, nbMotsProposes)
}

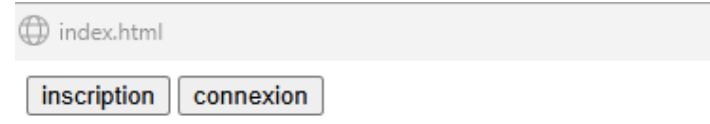
```

Extrait de code JavaScript que j'ai pu apprendre à réaliser lors de ces deux jours. Ce passage présente une fonction, exécutée lors du lancement de la page, qui démarre un jeu en définissant plusieurs variables, dont une faisant appelle à une autre fonction, le tout proposant à l'utilisateur de choisir entre une liste de mots ou de phrases qu'il devra parfaitement écrire afin d'augmenter son score.

29/11 : Création de la page d'accueil

Une fois assez familiarisé avec le JavaScript, et après avoir reçu de la part de mon collègue en charge du back-end les premiers débuts de codes pour la page d'accueil, j'ai commencé le travail sur la réalisation de celle-ci.

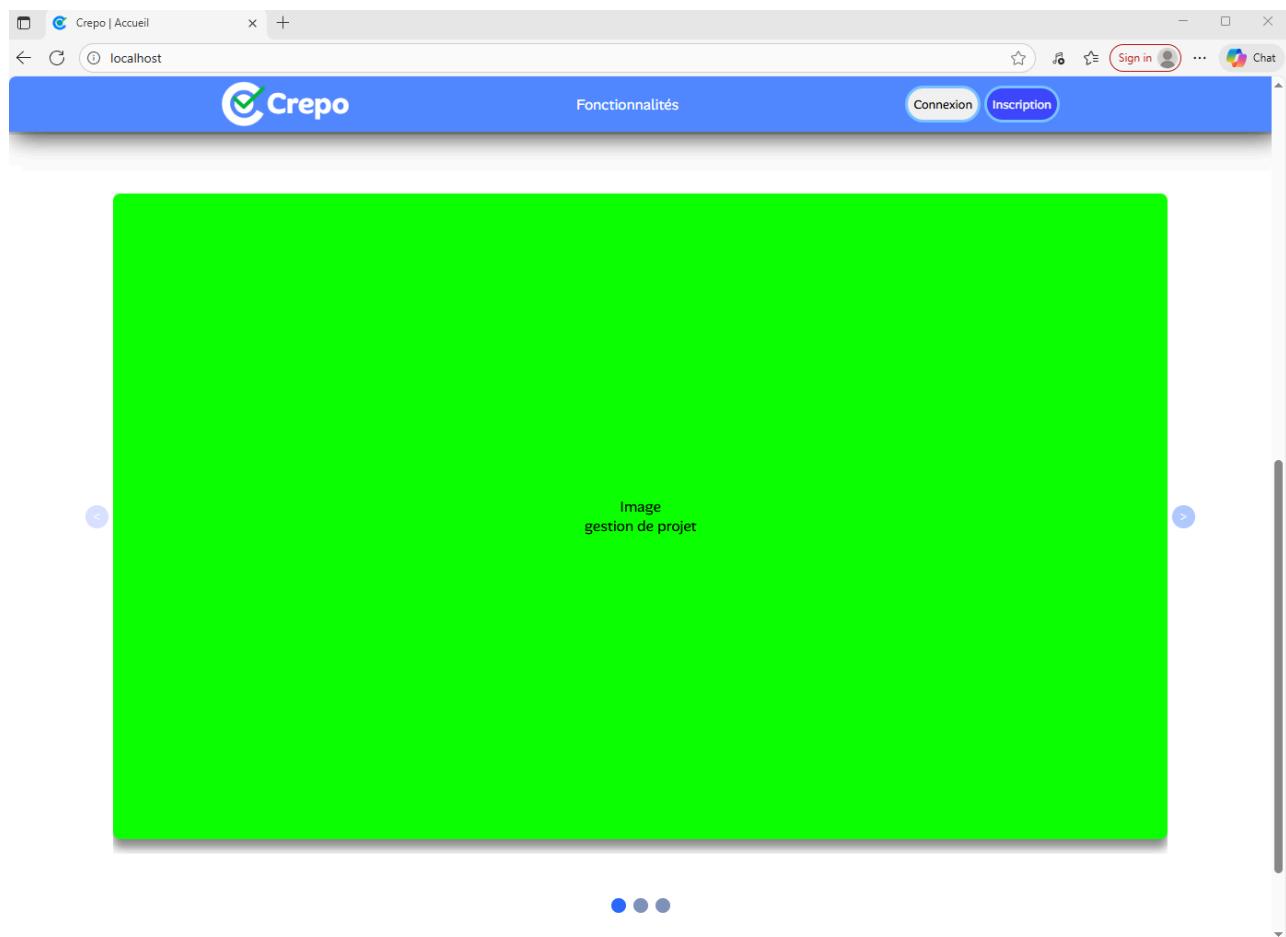
En appliquant les connaissances acquises la semaine précédente lors de la création de mon portfolio, j'ai ainsi pu faire passer l'apparence de la page d'accueil de son état initial, ici :



à cette première ébauche :

A screenshot of a web browser displaying the final design of the home page. The header features a blue navigation bar with the "Crepo" logo, a search bar, and links for "Fonctionnalités", "Connexion", and "Inscription". The main content area includes a "Bonjour et bienvenue !" message, two paragraphs of placeholder text (Lorem ipsum), and a decorative image of a calendar and a plant.

(haut de la page d'accueil)



(deuxième partie de la page)

01/12 : Création de la page de connexion

Ayant besoin de captures d'écran de l'apparence finale de notre site pour décorer la page d'accueil, j'ai alors porté mon attention sur la page de connexion.

Une fois de plus, après avoir reçu le fichier .php de la part du responsable back-end, j'ai pu commencer mon travail de responsable front-end.

Cependant, contrairement à la page d'accueil, le code de la page de connexion était rempli de code php un peu plus complexe que ce que j'ai eu l'occasion de voir lors de mon apprentissage sur OpenClassrooms, ainsi que des commandes SQL.

Ainsi, grâce à l'aide de mon collègue qui m'a expliqué le fonctionnement de ces commandes et comment modifier le fichier pour y incorporer mes lignes de codes HTML et CSS sans causer de problèmes, j'ai pu changer l'apparence de la page de connexion.

Nous sommes donc passé de ceci :

inscription connexion

pseudo:

email:

password:

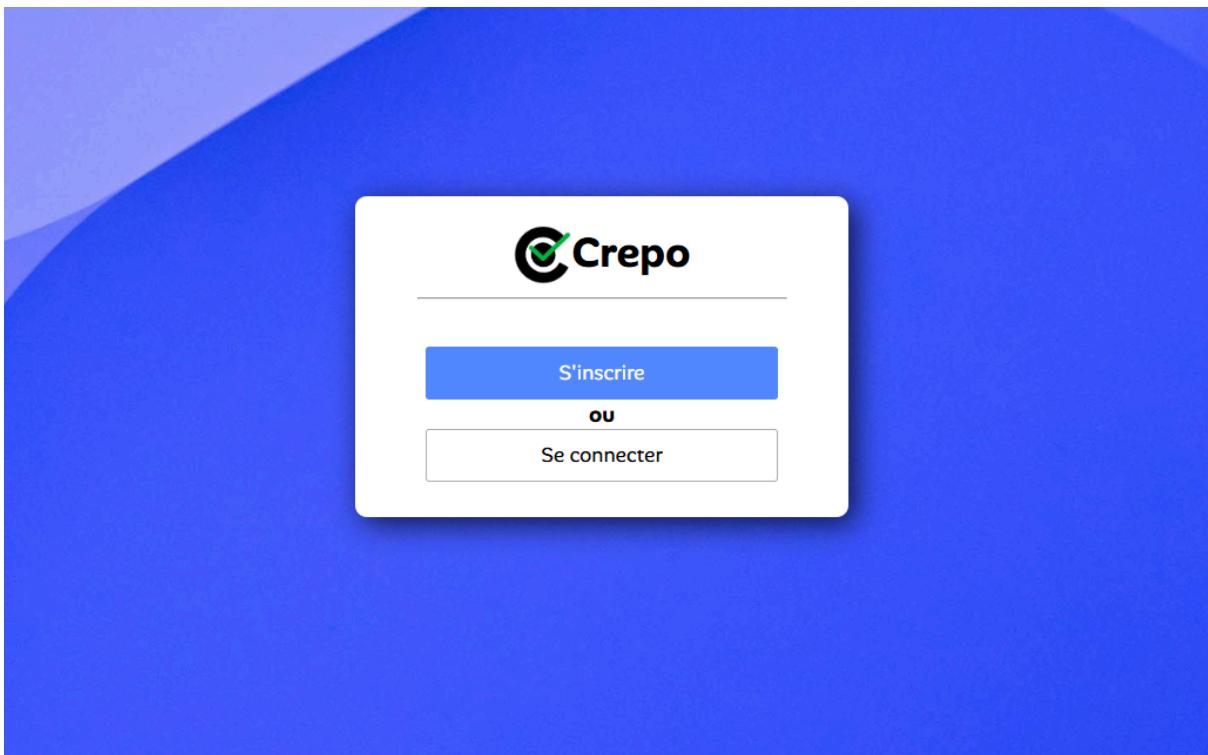
et :

inscription connexion

pseudo:

password:

à cette version finale :



(apparence de la page de connexion lors de l'ouverture de celle-ci)

The image displays two screenshots of the Crepo application's user interface. The left screenshot shows the sign-up process, featuring fields for 'Pseudo', 'Email', and 'Password', each with a placeholder text ('Choisissez un pseudo', 'Choisissez un email', 'Choisissez un password'). Below these fields is a blue 'Valider' button. Above the fields, the Crepo logo is visible. The right screenshot shows the sign-in process, featuring fields for 'Pseudo' and 'Password', each with a placeholder text ('Saisissez votre pseudo', 'Saisissez votre password'). Below these fields is a blue 'Continuer' button. Above the fields, the Crepo logo is visible. Both screenshots include an 'ou' (or) link below the respective forms, which links to a 'Se connecter' (Log in) or 'S'inscrire' (Sign up) button.

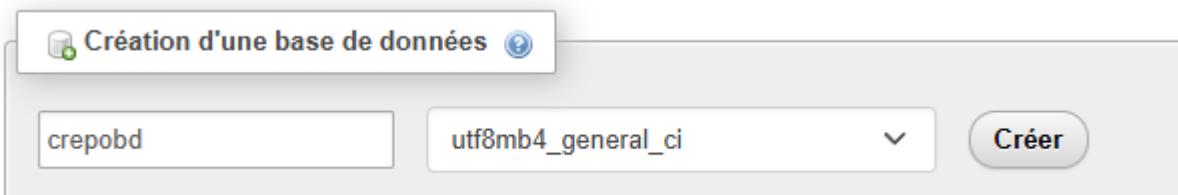
(interface d'inscription, à gauche, et de connexion, à droite)

02/12 : Création de l'interface principale du site

Il fallait maintenant que je me concentre sur l'apparence globale du site avec ses pages "Projets", "Tâches" et "Membres".

Mais avant de pouvoir commencer ce travail, il fallait que je puisse me connecter au site grâce à la base de données SQL créée encore une fois par mon collègue responsable du back-end.

Celui-ci m'a alors présenté le site "phpMyAdmin" où j'ai dû importer la base de données de notre site et toutes ses tables.



Informations pour la connexion

Nom d'utilisateur :	Saisir une valeur	phpclient
Nom d'hôte :	Local	localhost
Mot de passe :	Saisir une valeur	***** Force : Fort
Saisir à nouveau :	*****	
Extension d'authentification	Authentification MySQL native	
Générer un mot de passe:	<input type="button" value="Générer"/>	<input type="text"/>

Base de données pour ce compte d'utilisateur

- Créer une base portant son nom et donner à cet utilisateur tous les priviléges sur cette base.
- Accorder tous les priviléges à un nom passe-partout (utilisateur_%).

Priviléges globaux Tout cocher

NB : les noms de priviléges sont exprimés en anglais.

<input checked="" type="checkbox"/> Données	<input type="checkbox"/> Structure	<input type="checkbox"/> Administration	Limites de ressources
<input checked="" type="checkbox"/> SELECT <input checked="" type="checkbox"/> INSERT <input checked="" type="checkbox"/> UPDATE <input checked="" type="checkbox"/> DELETE <input type="checkbox"/> FILE	<input type="checkbox"/> CREATE <input type="checkbox"/> ALTER <input type="checkbox"/> INDEX <input type="checkbox"/> DROP <input type="checkbox"/> CREATE TEMPORARY TABLES <input type="checkbox"/> SHOW VIEW <input type="checkbox"/> CREATE ROUTINE	<input type="checkbox"/> GRANT <input type="checkbox"/> SUPER <input type="checkbox"/> PROCESS <input type="checkbox"/> RELOAD <input type="checkbox"/> SHUTDOWN <input type="checkbox"/> SHOW DATABASES <input type="checkbox"/> LOCK TABLES <input type="checkbox"/> REFERENCES	<small>NB : une valeur de 0 (zéro) lève la limite.</small> MAX QUERIES PER HOUR <input type="text" value="0"/> MAX UPDATES PER HOUR <input type="text" value="0"/> MAX CONNECTIONS PER HOUR <input type="text" value="0"/> MAX USER_CONNECTIONS <input type="text" value="0"/>
<input type="checkbox"/> Console de requêtes SQL <input type="checkbox"/> ALTER ROUTINE			

Ici il m'a fallu créer un nouvel utilisateur capable d'accéder à cette base de données depuis le code php de notre site, et lui donner les priviléges nécessaires pour son bon fonctionnement (Select, Insert, Update et Delete, que l'on peut retrouver à plusieurs reprises dans nos fichier .php).

Une fois cette base de données reliée à mes fichiers .php, j'ai pu débuter mon travail sur l'interface principale du site.

J'ai d'abord commencé par délimiter les différentes parties de celle-ci, notamment la barre de menu, et l'espace d'affichage des projets.

Comme mentionné précédemment, j'ai pu utiliser mes nouvelles connaissances en JavaScript pour créer un menu "sidebar" pouvant se réduire grâce à un bouton :



```
const toggleButton = document.getElementById('toggleBtn');
const sidebar = document.getElementById('sidebar');

function toggleSidebar(){
    sidebar.classList.toggle('close');
    toggleButton.classList.toggle('rotate');
}
```

(Code JavaScript pour activer la "fermeture" du menu ainsi que la rotation de l'icône du bouton permettant celle-ci.)

```
#sidebar {
    box-sizing: border-box;
    height: 100vh;
    width: 200px;
    padding: 5px 1em;
    background-color: #rgb(80, 135, 255);
    border-right: 1px solid #rgb(0, 0, 0);
    position: sticky;
    top: 0;
    align-self: start;
    transition: 300ms ease-in-out;
    overflow: hidden;
    text-wrap: nowrap;
    border-radius: 10px;
}
```

(Extrait du code CSS s'occupant de l'apparence du menu, utilisant notamment une nouvelle propriété que j'ai pu découvrir lors de la création de celui-ci : "transition" qui permet d'effectuer une "animation" de fermeture en ralentissant la transition entre l'état fermé et ouvert du menu.)

En ce qui concerne l'apparence globale du site, nous sommes passé de la page php de liste des projets sans modifications CSS ici présente :

l'Arme

créer de quoi gentiment calmer tout le monde

Vol

fabriquer une MACHINE VOLANTE

OGear

Hello world

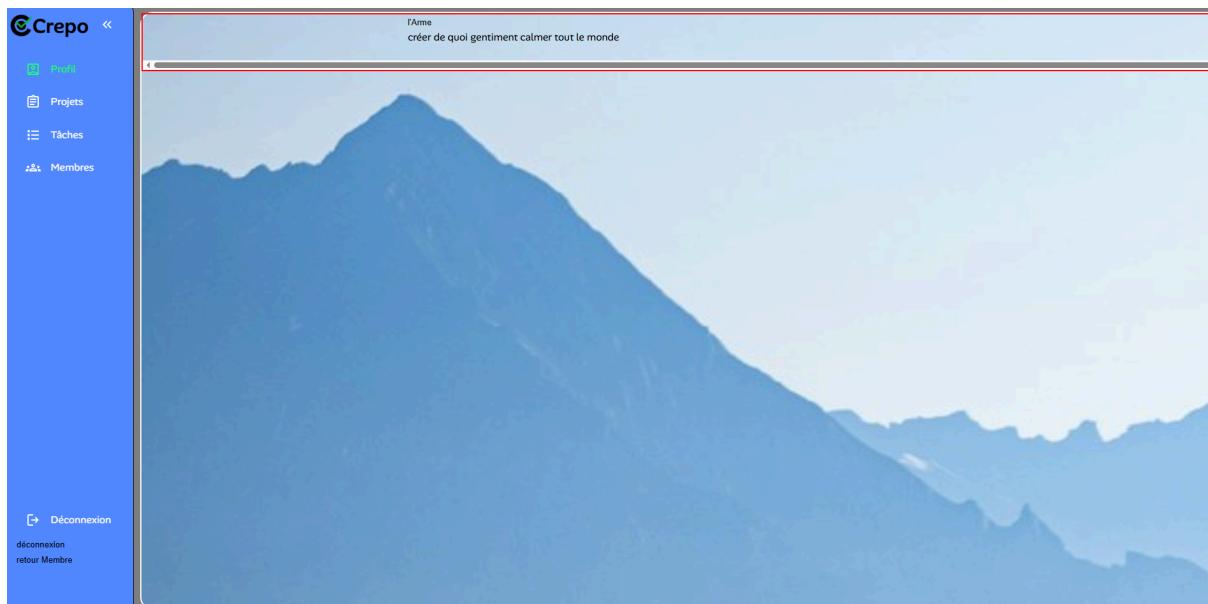
Nom du projet:

Description:

Créer

déconnexion

à celle-ci :



Ainsi que de la première page "Tâches" ici :

En cours

Nom de la tâche: remettre en ordre
Priorité: Extrême
Description: remettre en état le labo a 100% de son potentiel afin de l'utiliser
date de début: 2018-12-04
date de fin: 2025-12-16
Personne travaillant dessus:

Fini

Nom de la tâche: défense
Priorité: Haute
Description: création d'un système de défense pour protéger le laboratoire
date de début: 2025-11-24
date de fin: 2025-11-28
Personne travaillant dessus:
-Giro

Nom de la tâche: besoin primaire
Description: trouver une source de nourriture et d'eau potable ainsi qu'un lit décent pour dormir
date de début: 2025-11-10
date de fin: 2025-11-12
Personne travaillant dessus:
-Giro

Création de dossier:
Nom du dossier:

Création de tâche:
Nom:
Contenu:
Priorité:
Date début: mm/dd/yyyy
Date fin: mm/dd/yyyy

Ajout à une tâche:
Nom:
Tâche:

à :

The screenshot shows the Crepo application interface with a sidebar on the left containing navigation links: Profil, Projets, Tâches, and Membres. The main area is divided into three vertical columns:

- Futur**:
Nom de la tâche: capture
Priorité: Moyenne
Description: capture Ceres la source d'alimentation de l'Arme
date de début: 2017-06-12
date de fin: 2017-07-12
Personne travaillant dessus:
- En cours**:
Nom de la tâche: remettre en ordre
Priorité: Extrême
Description: remettre en état le labo a 100% de son potentiel afin de l'utiliser
date de début: 2018-12-04
date de fin: 2025-12-16
Personne travaillant dessus:
- Finis**:
Nom de la tâche: défense
Priorité: Haute
Description: création d'un système de défense pour protéger le laboratoire
date de début: 2025-11-24
date de fin: 2025-11-28
Personne travaillant dessus:
-Giro

Nom de la tâche: besoin primaire
Description: trouver une source de nourriture et d'eau potable ainsi qu'un lit décent pour dormir
date de début: 2025-11-10
date de fin: 2025-11-12
Personne travaillant dessus:
-Giro

Et enfin, de la première page "Membres" ici :

Liste des membres

Giro
9
patate
1

Ajouter un membre

Gérer les droits

Supprimer un membre

à la finale :

The screenshot shows the final state of the Crepo application. On the left, there is a sidebar with a blue background containing navigation links: 'Profil' (selected), 'Projets', 'Tâches', and 'Membres'. Below these are 'Déconnexion' and 'retour' buttons. The main content area has a light blue background with a mountain range silhouette. At the top right, the title 'Liste des membres' is displayed. Below it, two blue rectangular boxes show member details: 'Giro' (9) and 'patate' (1). At the bottom, three rounded rectangular boxes provide management functions: 'Ajouter un membre' (with a 'Nom' input field and 'Ajouter' button), 'Gérer les droits' (with dropdown menus for selecting a member and a right), and 'Supprimer un membre' (with a dropdown menu for selecting a member and a 'Supprimer' button).

II. Problèmes Rencontrés

1. Incompatibilité avec certains navigateurs

Lors de la création de la page d'accueil, inspirée par d'autres sites, j'ai voulu ajouter un "carrousel". Cependant la méthode que j'ai utilisé n'est pas compatible avec tout les navigateurs.



Cela signifiait que mon carrousel ressemblait à ceci sur Microsoft Edge :



(le défilement s'effectue avec les boutons de navigation à gauche et à droite, ainsi qu'avec les trois boutons en bas.)

Mais ressemblait à ceci sur Mozilla Firefox :



(les boutons de navigation ont disparu, seule une barre de navigation apparaît en bas, et les cartes ne sont pas centrées.)

Voulant tout de même continuer d'utiliser cette méthode d'affichage, j'ai donc décidé de me servir d'Edge comme navigateur principal pour avoir un aperçu de mon travail.

2. Problèmes avec Xampp

Pour coder en PHP il est nécessaire d'utiliser les services de l'application Xampp permettant d'accéder à un serveur Apache pour nous permettre de visualiser nos fichiers.

Malheureusement j'ai rencontré quelques problèmes lors de ma première utilisation de cette application, où j'ai notamment dû changer quelques lignes dans son fichier config afin de permettre son bon fonctionnement, et me faciliter la tâche en faisant en sorte que l'application puisse lire mes fichier directement depuis mon répertoire GitHub présent sur mon ordinateur (et ceux de l'école).

Un autre contretemps mineur en rapport avec Xampp a été son utilisation sur macOS qui n'est pas exactement la même que sur windows, me demandant un peu plus de recherche et de réflexion.

3. Adaptations aux différentes méthodes de travail

N'ayant jamais eu à travailler en groupe sur ce type de projet auparavant, je n'avais pas encore dû collaborer sur des fichiers de code avec d'autres personnes.

Il m'a donc fallu un certain temps d'adaptation à chaque fois que mes collègues m'ont partagé leurs fichiers pour qu'ils m'expliquent leurs façon de coder, et la signification de certaines lignes, notamment en PHP et SQL, que je n'avais jamais vu jusque là.

Il m'a aussi été difficile de trouver comment ajouter mes balises HTML au code PHP de mon collègue sans modifier celui-ci lorsque j'ai commencé à travailler dessus pour la première fois.

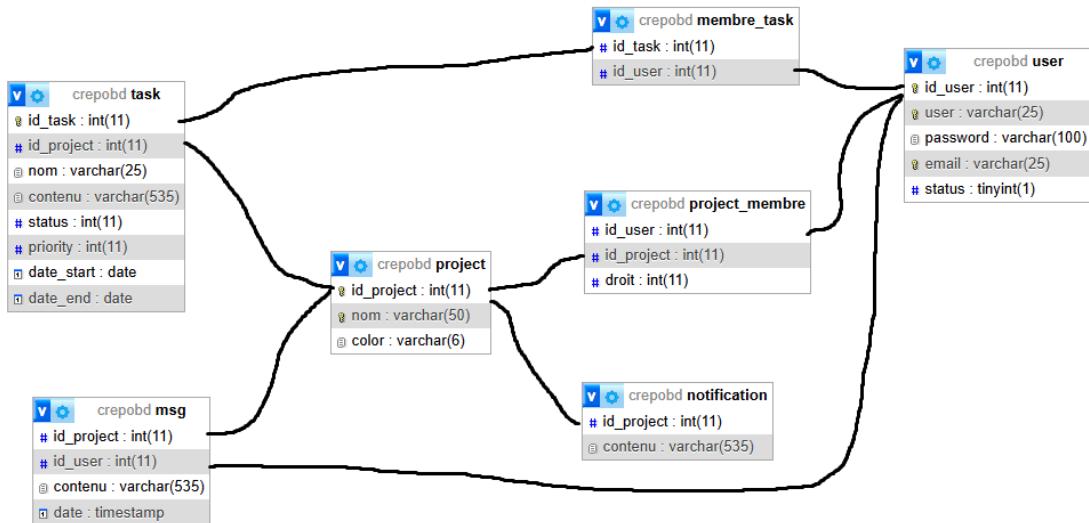
Finalement, grâce aux explications précises et détaillées des responsables Back-end et DevSecOps, j'ai réussi à m'adapter en comprenant les différentes parties de leurs lignes de code et cela m'a permis d'ajouter mes propres lignes sans causer de problèmes au fonctionnement du site.

Back End

I. Déroulement du Projet

24/11: Planification & Mise en place

- Définition de la liste des tâches à réaliser.
- Répartition des rôles dans le groupe (front end lead, back end lead et devsec ops lead.)
- Première réflexion sur l'apparence de l'interface.
- ébauche de la base de données.



26/11: Lien SQL, php & Connexion à la BD

- Rédaction des requêtes SQL nécessaires et conversion de ces dernières en fonctions PHP.
- Création du fichier de connexion et liaison du code php à la base de données.
- création dans la base de données d'un utilisateur avec des droits limités pour empêcher des problèmes futurs tel que drop table.

<input type="checkbox"/> phpclient	localhost	Oui	SELECT, INSERT, UPDATE, DELETE	Non
------------------------------------	-----------	-----	--------------------------------	-----

Comme on peut le voir, l'utilisateur phpclient n'a que les droits de select (afficher), insert (insérer dans un table une ligne), update (mettre à jour certaines ligne) et de delete (suppression de certaines ligne). Ainsi en limitant c'est droit, on sécurise notre site.

```
$requete = "select user, email from user ;";
$resultat = $mysqli->query($requete);
```

Au début les requêtes sql qui était faite ressemblent à ceci ce qui est fonctionnel mais pas sécurisé. Ainsi on a changé les requêtes en ceci

```
$stmt = $mysqli->prepare("SELECT id_user, password FROM user WHERE user = ?");
$stmt->bind_param("s", $pseudo);
$stmt->execute();
```

La requête est déjà plus sécurisée et l'infection sql est bloquée.

27/11: Inscription + Connexion dans login.php

- Contrairement au plan initial, l'inscription et la connexion ont été réunies dans le fichier login.php pour réduire la multiplication des fichiers avec un faible nombre de lignes de code et permettre de plus aisément trouver les erreurs.
- Fonctionnalités réalisées :
 - Inscription avec vérification que l'adresse mail n'est pas déjà prise pour éviter les comptes avec le même email.
 - bouton permettant de passer de la connexion à l'inscription.
 - Gestion des erreurs d'authentification (pseudo ou email invalide pour la connexion ou déjà pris pour l'inscription, information invalide si les informations de connexion ne sont pas dans la base de données).

28/11: Gestion des Projets et Tâches

- Boutons connexion/inscription intégrés sur la page d'accueil permettant de rediriger l'utilisateur sur la page login.php et donc de lui permettre de créer un compte ou de se connecter.
- Création du formulaire d'ajout de tâche et d'ajout de cette dernière dans la base de données.
- Liaison automatique de la tâche et du projet dans lequel l'utilisateur se trouve.
- Formulaire de création de projet et Affichage des projets et des tâches associées.

```

$stmt = $mysqli->prepare("INSERT INTO `project` (`nom`, `contenu`) VALUES (?, ?);");
$stmt->bind_param("ss", $_POST["nomcrea"], $_POST["contenucrea"]);
$stmt->execute();

$stmt = $mysqli->prepare("select id_project from project where nom = ?;");
$stmt->bind_param("s", $_POST["nomcrea"]);
$stmt->execute();
$idPro = $stmt->get_result();

foreach ($idPro as $idPr) {

    $projectId = (int)$idPr["id_project"];
    $role = 9;

    $stmt = $mysqli->prepare("INSERT INTO project_membre VALUES (?, ?, ?)");
    $stmt->bind_param("iii", $id, $projectId, $role);
    $stmt->execute();

    $blocks = [
        [0, "Futur"],
        [1, "En cours"],
        [2, "Fini"]
    ];

    foreach ($blocks as $block) {
        $stmt = $mysqli->prepare("INSERT INTO block VALUES (?, ?, ?)");
        $stmt->bind_param("iis", $projectId, $block[0], $block[1]);
        $stmt->execute();
    }
}

```

voici ci dessus le code php permettant de la création d'un projet mais aussi des lignes par défaut. Le premier bloc permet simplement la création du projet mais comme l'id du projet s'incrémente toute seule par la base de données on a pas besoin de la donnée. Par contre on doit du coup dire quelle valeur vont être envoyés à la base de données.

Le second bloc ainsi que la boucle vont être là pour donner les droits de propriétaire au créateur du projet. Cependant on ne connaît pas l'id du projet car la base de donnée la choisit toute seule, du coup on va faire une requête pour récupérer l'id en se basant sur le nom du projet car il est censé être unique. Une fois l'id récupéré on peut donner les droits au créateur et avec une nouvelle boucle on peut créer 3 dossier par défaut assez classique, ces derniers étant, Futur, En cours , Fini.

La partie du code par rapport au bloc a été faite ultérieurement mais je trouvais aussi judicieux d'en parler.

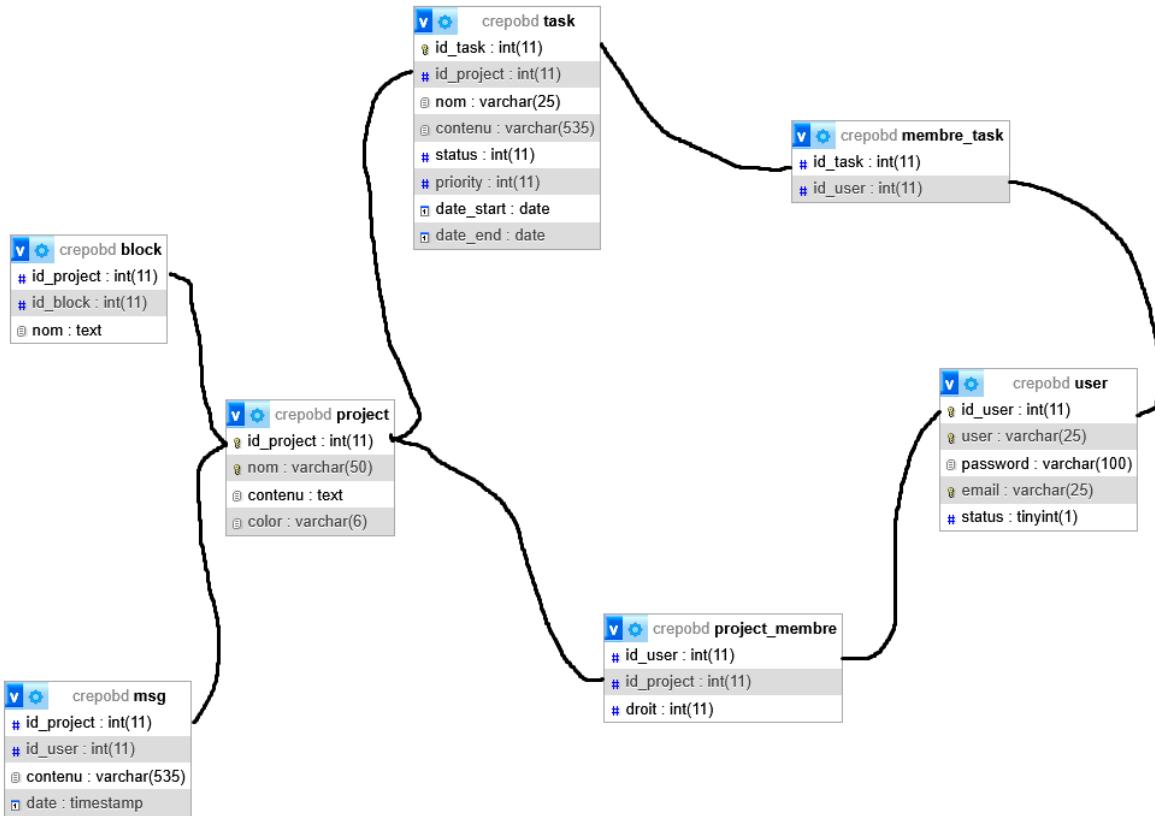
01/12: Améliorations & Sécurité

- Ajout de la colonne contenu dans la table project afin de permettre à l'utilisateur d'ajouter une description au projet qu'il créer.
- amélioration des tâches en leur ajoutant une date de début et de fin
- Création de membre.php : affichage liste des membres appartenant au projet et bouton pour retourner== aux tâches.

- Ajout d'une page plus détails des projets et des tâches
- Mise en place d'un système de sessions obligatoires :
 - accès refusé si l'utilisateur n'est pas connecté,
 - redirection vers login.php si non connecté.

02/12: Ajout de la Table Block & Réorganisation

- Modification de la BD : ajout de la table block dans lequel les utilisateurs pourront créer des dossiers où ils ajouteront leurs tâches.
- Affichage des tâches regroupées par dossier (valeur par static).
- Simplification du code:
 - suppression redondances
 - simplification de code trop long pour rien
- Début de la gestion des droits des membres.



03/12: Finalisation de la Gestion des Membres + Dossiers

- création de dossier défaut dans la table block (futur, en cours et fini)
- mise en place du système de dossier dynamique dans lequel on regroupe et affiche les tâches
- bouton pour modifier le dossier auquel appartient une tâche.
- Modification du système de priorité des tâches
- mise en place de boutons permettant de modifier la priorité des tâches et donc leur position dans le dossier (priorité plus élevée plus haute dans le dossier).
- Finalisation du backend de membre.php :
 - ajout et suppression de membres,
 - gestion des droits,
 - nettoyage du code.
- Nouvelle vague de corrections suite à des bugs rencontrés après la fusion avec le front end
- création d'un formulaire permettant l'ajout de membres à certaines tâches.

```
$stmt = $mysqli->prepare("select id_user, droit from project_membre where id_user = ?;");
$stmt->bind_param("i", $_SESSION["connect"]);
$stmt->execute();
$res_test = $stmt->get_result();

foreach($res_test as $res){
    $droit = $res["droit"];
}

echo ' Ajout à une tâche: <br><form action="" method="post">
<label for="nom-select">Nom:</label>
<select name="nomadd" id="nomadd">
<option value="">--Please choose an option--</option>';

$stmt = $mysqli->prepare("select DISTINCT U.id_user, U.user, droit from user U, project_membre M where U.id_user = M.id_user and
id_project = (select id_project from project where nom = ?);");
$stmt->bind_param("s", $_SESSION["projet"]);
$stmt->execute();
$res_test = $stmt->get_result();

foreach($res_test as $res){
    if($res["droit"] < $droit)
        echo '<option value="'. $res["id_user"].'">' . $res["user"] . '</option>';
    elseif($_SESSION["connect"] == $res["id_user"])
        echo '<option value="'. $res["id_user"].'">' . $res["user"] . '</option>';
}

echo '</select><br>';
```

Ce code et la première partie pour afficher le formulaire pour ajouter un membre à une tâche. Tout d'abord, on récupère le niveau de droit de l'utilisateur. Ensuite on affiche dans un select on récupère le nom des personnes participant au projet dans lequel la personne est. Ensuite dans une boucle on va vérifier 2 choses. si la personne a des droits supérieurs que la personne de la ligne qu'il vient de récupérer, ou si la ligne en question correspond à l'utilisateur lui-même. dans ces deux cas on ajoute le nom de la personne dans un select qui va servir de menu déroulant qu'elle utilisateur va être attribuer à quel tâche.

Pour ce qui est de la requête SQL, étant donné que l'on a besoin du nom de la personne ainsi que son niveau de droit on a besoin de récupérer à la fois une information de la table user et project membre. Pour rendre cela plus simple, on va donc faire une jointure entre les deux tables. Enfin, pour avoir seulement les personnes qui appartiennent au projet, on doit vérifier l'id project de la tables project_membre avec celui du projet dans lequel on est. Etant donné que l'on a que le nom du projet, on doit donc faire une requête imbriquées pour obtenir cette id.

```
echo '<label for="nom-select">Tâche:</label>
<select name="taskadd" id="taskadd">
<option value="">--Please choose an option--</option>';

$stmt = $mysqli->prepare("select * from task where id_project=(select id_project from project where nom = ?);");
$stmt->bind_param("s", $_SESSION["projet"]);
$stmt->execute();
$res_test = $stmt->get_result();

foreach($res_test as $res){
    echo '<option value="'.$res["id_task"].'">'.$res["nom"].'</option>';
}

echo '</select><br>
<input name="addtask" type="submit" value = change />
</form><br>';
```

La seconde partie consiste à afficher dans une seconde liste le nom des tâches en lien avec le projet dans lequel on se trouve.

```
if(isset($_POST["addtask"])){
    $stmt = $mysqli->prepare("INSERT INTO membre_task VALUES (?, ?);");
    $stmt->bind_param("ii", $_POST["taskadd"], $_POST["nomadd"]);
    $stmt->execute();
}
```

Etant donné que l'on a ajouté nous même les id dans les listes et que l'utilisateur ne peut pas rentrer lui même des informations on a juste besoin de savoir si ma personne a cliqué sur le bouton pour valider.

II. Problèmes Rencontrés et Solutions

1. Problèmes de requêtes SQL

Durant le développement, plusieurs requêtes SQL ont généré des erreurs : mauvais alias ou syntaxe oubliée, incohérences entre les anciens noms de colonnes et les nouvelles structures, erreurs liées à des requêtes complexes (jointure/imbrication).

Solutions :

Relecture complète des structures de tables.

Réécriture de certaines requêtes pour correspondre au nouveau schéma BD.

Ajout de messages d'erreur détaillés pour identifier plus vite la source du problème.

2. Problèmes de redirection PHP

Quand l'équipe a intégré le frontend au backend, de nombreux fichiers faisaient déjà des affichages (HTML, echo...) avant les header("Location: ..."), ce qui a entraîné :

Des Avertissements comme par exemple:

"Warning: Cannot modify header information – headers already sent..."

Nous avons donc dû réorganiser le code, tout en gardant à l'esprit pourquoi nous avions créé les header à l'origine.

Les solutions que nous avons trouvées sont les suivantes.

- Réorganisation de l'ordre du code PHP.
- Suppression ou déplacement de certains echo.
- Simplification du code pour éviter les redirections inutiles.

Grâce à ces ajustements, le code est devenu plus propre et plus court.

3. Intégration du Front-End

Lorsqu'on a fusionné le front-end (HTML/CSS) avec le backend, certains conflits ont apparu :

- variables non initialisées,
- sessions démarrées trop tard,
- duplications de blocs de code.

DevSecOps

La partie DevSecOps a été réalisée par Esteban Kropp. L'objectif était d'installer Sonarqube et de le connecter à GitHub. Sonarqube est un logiciel qui analyse chaque code pushé dans un groupe GitHub, détecte les failles de sécurité, renseigne sur leur nombre et leur niveau de gravité, et localise précisément ces failles pour faciliter leur correction.

Cependant, Sonarqube n'a pas pu être utilisé correctement. Lors de l'installation, des problèmes mineurs sont survenus. Lors des tests, même si les pushes fonctionnaient, les fichiers analysés dataient de deux semaines. Malgré plusieurs tentatives pour résoudre ce problème, il est resté non résolu faute de temps. Par conséquent, j'ai cherché et corrigé manuellement le plus grand nombre possible de failles.

Exemples de sécurités mises en place manuellement :

Hachage des mots de passe avec **password_hash** et vérification avec **password_verify** :

```
php$hash = password_hash($passwordRaw, PASSWORD_DEFAULT); /* ... */
if ($stmt->fetch() && password_verify($passwordRaw, $hash)) { /* */
connexion OK* }
```

Validation de mot de passe avec au moins 8 caractères dont un caractère spécial :

```
phpelseif (!preg_match('/^(?=.*\W){8,}$/ ', $passwordRaw)) {
$error = "Mot de passe faible (8+ caractères, 1 spécial)."; }
```

Mise en place d'un anti-brute force limitant à 5 tentatives maximum, avec un blocage d'une minute en cas de dépassement :

```
php$maxAttempts = 5; $lockoutTime = 60; if
(!isset($_SESSION['login_attempts'])) { $_SESSION['login_attempts'] =
0; $_SESSION['last_attempt'] = time(); } if
(isset($_POST['login'])) { if ($_SESSION['login_attempts'] >=
$maxAttempts && (time() - $_SESSION['last_attempt']) < $lockoutTime)
{ $remaining = $lockoutTime - (time() - $_SESSION['last_attempt']);
$error = "Trop de tentatives. Réessayez dans {$remaining}s.";} else
{ if (échec) { $_SESSION['login_attempts']++;
$_SESSION['last_attempt'] = time(); } } }
```

Protection contre les injections SQL avec des requêtes préparées :

```
php$stmt = $mysqli->prepare("SELECT id_user, password FROM user
WHERE user = ?"); $stmt->bind_param("s", $pseudo); $stmt->execute();
```

Créations d'un fichier .php "crepo_db.php" qui contient juste

```
<?php return [ 'host' => 'localhost', 'user' => 'phpclient', 'pass' => 'php@code/1234',
'name' => 'crepobd', 'charset' => 'utf8mb4', ]; ?>
```

c'est un tableau PHP simple, utilisé pour stocker et partager les informations nécessaires à la connexion à une base de données MySQL.

```

// CONFIG COOKIES SÉCURISÉS

session_set_cookie_params([
    'httponly' => true, // Bloque JS
    'secure' => false, // HTTPS only
    'samesite' => 'Lax', // Anti-CSRF cross-site
]);

session_start();

// RÉGÉNÉRATION d'une nouvelle ID

session_regenerate_id(true); // Nouvelle ID session

```

Protection	Code	Attaque bloquée	Explication
httponly=true	'httponly' => true	XSS	<code>document.cookie</code> impossible, JS ne voit pas cookie session
samesite=Lax	'samesite' => 'Lax'	CSRF cross-site	Bloque POST/GET depuis site malveillant (iframe, lien)
session_regenerate_id(true)	<code>session_regenerate_id(true)</code>	Session fixation/hijacking	Nouvelle ID après login + user supprimé → attaquant perd session

Même si Sonarqube n'a pas été pleinement exploité, j'ai compris son fonctionnement et l'intérêt d'outils d'analyse automatique. De plus, j'ai découvert Vercel, une plateforme permettant de connecter un compte GitHub et d'effectuer des simulations d'attaques sur un site web, mais le manque de temps a empêché son utilisation.

Ce travail a permis de renforcer la sécurité manuellement.

Conclusion

Avec la participation de chaque membre et l'union de nos compétences nous avons réussi à réaliser un site web sécurisé de gestion de projet.

Le Front-end a eu quelques difficultés au départ, mais a pu finalement bien progresser grâce à l'entraide commune entre les membres du projet, permettant la réalisation de pages web incorporant les codes HTML, CSS, JS, PHP et SQL de tout le monde sans problèmes.

Le back-end a progressé malgré plusieurs difficultés techniques (requêtes SQL, redirections, intégration avec le front-end).

Les choix effectués, notamment l'unification de l'inscription/connexion dans login.php et la simplification progressive du code, ont permis d'obtenir une application : plus légère, plus logique, plus stable, plus facile à maintenir et avec une meilleure scalabilité (notamment la réduction au maximum de variable static dans l'affichage)

Toutes les fonctionnalités essentielles ont été finalisées dans les délais, et le code de base est maintenant suffisamment clair pour permettre l'ajout de nouvelle fonctionnalité dans le futur.