

# Compte rendu : *Projet Geoworld*

## *Sommaire :*

### 1. **Partie Steeven**

- a. Création d'une base de données
- b. Formulaire inscription

### 2. **Partie Devamadushan**

- a. Gestion des rôles utilisateurs
- b. La requête SQL

### 3. **Partie Adrien**

- a. Mise à jour des données manuellement pour l'enseignant et l'administrateur
- b. Exploitation de la barre de recherche

### 4. **Missions complémentaires**

- a. Email
- b. Cookies
- c. Mentions légales
- d. Geomap

### 5. **Conclusion**

## Introduction :

Nous avons établi une répartition des tâches pour réaliser au mieux le projet Geoworld. Les tâches ont été attribuées de manière logique afin que chacun puisse travailler individuellement sur les missions imposées. +

### IMPORTANT

Avant de lancer le projet, veuillez importer le script ci-dessous pour le bon fonctionnement du projet

### Script :

```
DROP TABLE IF EXISTS `utilisateur`;
CREATE TABLE `utilisateur` (
  `id` int(11) NOT NULL auto_increment,
  `nom` varchar(100) NOT NULL default '',
  `prenom` varchar(50) NOT NULL default '',
  `date_naissance` date NOT NULL ,
  `login` varchar(50) NOT NULL ,
  `password` varchar(50) NOT NULL ,
  `role` varchar(50) NOT NULL ,

  PRIMARY KEY (`id`)
) ;
INSERT INTO `utilisateur` VALUES (1,'Tandabany','Adrien','1999-05-15','Adrien','123456','admin'),
INSERT INTO `utilisateur` VALUES (2,'Toto','Deva','1999-12-22','Deva','1234','enseignant'),
INSERT INTO `utilisateur` VALUES (3,'Song','Steeven','1999-09-09','Steeven','12345','eleve');
```

### IMPORTANT

Lien GitLab :

## I. Partie Steeven

### a) Création d'une base de données

Nous avons créé une table dans la base de données geowrold qui nous a permis d'attribuer les rôles utilisateurs, les logins, les mots de passe, la date de naissance, le nom/prénom et l'id.

```
DROP TABLE IF EXISTS `utilisateur`;
CREATE TABLE `utilisateur` (
  `id` int(11) NOT NULL auto_increment,
  `nom` varchar(100) NOT NULL default '',
  `prenom` varchar(50) NOT NULL default '',
  `date_naissance` date NOT NULL ,
  `login` varchar(50) NOT NULL ,
  `password` varchar(50) NOT NULL ,
  `role` varchar(50) NOT NULL ,

  PRIMARY KEY (`id`)
) ;
INSERT INTO `utilisateur` VALUES (1,'Tandabany','Adrien','1999-05-15','Adrien','123456','admin'),
INSERT INTO `utilisateur` VALUES (2,'Toto','Deva','1999-12-22','Deva','1234','enseignant'),
INSERT INTO `utilisateur` VALUES (3,'Song','Steeven','1999-09-09','Steeven','12345','eleve');
```

### b) Formulaire inscription

Dans un premier temps, nous vérifions si les variables sont définies et remplies avec le code suivant :

```
if (isset($_POST['login']) && isset($_POST['pwd']) && !empty($_POST['login']) && !
empty($_POST['pwd'])) {
```

Puis nous appelons la fonction « getAuthentification » depuis la page manager-db en indiquant en paramètre le login et le mot de passe.

```
$result = getAuthentification($_POST['login'],$_POST['pwd']);
```

On donne une condition qui indique que si le login et le password est reconnu, le visiteur est redirigé vers la page d'accueil avec comme paramètre le login et le rôle.

Ensuite, nous créons un formulaire d'inscription.

```

<form action="GeoworldTraitement.php" class="traite" method="post">
<h1>Inscription</h1>

<label for="utilisateur">Utilisateur : </label>
<input type="text" id="utilisateur" name="login" required>

<label for="pwd">Mot de passe : </label>
<input type="password" id="password" name="pwd" required>

<label for="Nom">Nom : </label>
<input type="text" id="nom" name="nom" required>

<label for="prenom">Prenom : </label>
<input type="text" id="prenom" name="prenom" required></p>

<label for="calendrier">Date de naissance : </label>
|   <input type="date" name="calendrier" required></p>
|
<input type="submit" id="sub" value="S'inscrire">

</form>

```

Ce formulaire redirige vers la page « GeoworldTraitement.php ». Dans cette même page les données sont récupérées et renvoyées dans la base de données via la méthode PDO. Ainsi, l'utilisateur est redirigé vers une page qui l'informe qu'il est inscrit. (page : InscritSucces.php)

#### NOTE

Difficulté rencontrée : Nous avons eu des difficultés de compréhension sur la préparation des requêtes. De nombreuses erreurs ont été émises...

## II. Partie Devamadushan

### a) Gestion des rôles utilisateurs

Pour gérer les visiteurs, nous attribuons un droit d'accès à l'administrateur.

```
if ($_SESSION['role']=='admin'):
```

Ensuite, l'administrateur peut consulter la liste des utilisateurs qui ont un rôle d'admin, d'élève et d'enseignant comme indiqué ci dessous.

```
<?php foreach ($utilisateurs as $personne): ?>
<tr>
  <?php if ($personne->role=='admin' || $personne->role=='enseignant' || $personne->role=='eleve'): ?>
  <td><?php echo $personne -> id ?></td>
  <td><?php echo $personne -> nom ?></td>
  <td><?php echo $personne -> prenom ?></td>
  <td><?php echo $personne -> date_naissance ?></td>
  <td><?php echo $personne -> role ?></td>
  <td><a href="delete.php?id=<?php echo $personne ->id ?>">Supprimer</a></td>
  <td><a href="update.php?id=<?php echo $personne ->id ?>">Editer</a></td>
</tr>
<?php endif; ?>
<?php endforeach; ?>
```

Ainsi que supprimer l'utilisateur (renvoie sur la page delete.php) ou valider leur accès et leur attribuer un rôle (renvoie sur update.php)

```
<?php if ($personne->role=='visiteur'): ?>
<td><?php echo $personne -> id ?></td>
<td><?php echo $personne -> nom ?></td>
<td><?php echo $personne -> prenom ?></td>
<td><?php echo $personne -> date_naissance ?></td>
<td><?php echo $personne -> role ?></td>
<td><a href="delete.php?id=<?php echo $personne ->id ?>">Supprimer</a></td>
<td><a href="update.php?id=<?php echo $personne ->id ?>">Editer</a></td>
```

### b) La requête SQL

Les enseignants et les administrateurs pourront consulter, créer, modifier, supprimer. Les élèves ne pourront que saisir des requêtes.

On initialise chaque opération SQL (SELECT, INSERT, UPDATE, DELETE) à partir de la table country de la base de données.

```
<select id="selectOption1" class="option" >
<option value="">Sélectionnez une option</option>
<option value="Option 1.1">SELECT</option>
  <option value="Option 1.3">INSERT</option>
  <option value="Option 1.4">UPDATE</option>
  <option value="Option 1.5">DELETE</option>
</select>
```

On crée un formulaire pour entrer les informations sélectionnées.

```
<form class="RE" method="GET">
  <input class="REQUETE" type="search" name="q" id="inputField" placeholder="Taper votre requete.." /> <br>
  <input class="EX" type="submit" value="EXECUTER" />
</form>
```

Puis nous traitons les informations avec un programme JavaScript qui consiste à rentrer les éléments choisis dans le formulaire d'entrée.

En parallèle, nous créons une fonction qui permet de mettre à jour le champ saisi avec les valeurs sélectionnées dans les trois premières listes déroulantes.

```
// récupérer les références des éléments select et de l'élément input
var select1 = document.getElementById("selectOption1");
var select2 = document.getElementById("selectOption2");
var select3 = document.getElementById("selectOption3");
var select4 = document.getElementById("selectOption4");
var inputField = document.getElementById("inputField");

// ajouter un écouteur d'événement pour détecter le changement de sélection
select1.addEventListener("change", updateInputField);
select2.addEventListener("change", updateInputField);
select3.addEventListener("change", updateInputField);
select4.addEventListener("change", updateInputField);

// fonction pour mettre à jour le champ de saisie input avec les valeurs s
function updateInputField() {
  var option1 = select1.options[select1.selectedIndex].text;
  var option2 = select2.options[select2.selectedIndex].text;
  var option3 = select3.options[select3.selectedIndex].text;
  var option4 = select4.options[select4.selectedIndex].text;
  inputField.value = option1 + " " +option2+ " " +option3+ " " +option4 ;
}
```

On vérifie si la variable \$\_GET['q'] est définie et non vide.

```
if(isset($_GET['q']) AND !empty($_GET['q'])) {  
    $q = htmlspecialchars($_GET['q']);  
    $recherche = $pdo->query($q);  
    //print_r($recherche);  
    echo $q;  
    //print_r($recherche);  
}
```

On vérifie les données saisie avec la fonction "strpos"

```
<?php if(strpos($q,$all)||strpos($q,$id)||strpos($q,$id2)): ?><th>Id</th> <?php endif ?>  
<?php if(strpos($q,$code)||strpos($q,$all)): ?><th>Code</th> <?php endif ?>  
<?php if(strpos($q,$name)||strpos($q,$all)): ?><th>Name</th> <?php endif ?>  
<?php if(strpos($q,$Continent)||strpos($q,$all)): ?><th>Continent</th> <?php endif ?>  
<?php if(strpos($q,$region)||strpos($q,$all)): ?><th>Region</th><?php endif ?>  
    <?php if(strpos($q,$surfacearea)||strpos($q,$all)): ?><th>SurfaceArea</th> <?php endif ?>  
<?php if(strpos($q,$indepyear)||strpos($q,$all)): ?><th>IndepYear</th><?php endif ?>  
<?php if(strpos($q,$population)||strpos($q,$all)): ?><th>Population</th> <?php endif ?>  
<?php if(strpos($q,$lifeexpectancy)||strpos($q,$all)): ?> <th>Life Expectancy</th><?php endif ?>  
    <?php if(strpos($q,$gnp)||strpos($q,$all)): ?><th>GNP</th> <?php endif ?>  
<?php if(strpos($q,$gnpold)||strpos($q,$all)): ?> <th>GNPOLD</th><?php endif ?>  
    <?php if(strpos($q,$localname)||strpos($q,$all)): ?><th>Local Name</th><?php endif ?>  
<?php if(strpos($q,$governmentform)||strpos($q,$all)): ?><th>Government Form</th><?php endif ?>  
<?php if(strpos($q,$headofstate)||strpos($q,$all)): ?><th>Head Of State</th><?php endif ?>  
    <?php if(strpos($q,$capital)||strpos($q,$all)): ?><th>Capital</th><?php endif ?>  
<?php if(strpos($q,$code2)||strpos($q,$all)): ?><th>Code2</th><?php endif ?>
```

Puis, on affiche les réponses de la requête SQL du pays souhaité.

```
<?php foreach($recherche as $b): ?>  
    <tr>  
        <td>  
            <?php if(strpos($q,$all)||strpos($q,$id)||strpos($q,$id2)): ?> <td><?php echo $b->id ?></td> <?php endif ?>  
            <?php if(strpos($q,$code)||strpos($q,$all)): ?> <td><?php echo $b->Code ?></td> <?php endif ?>  
            <?php if(strpos($q,$name)||strpos($q,$all)): ?> <td><?php echo $b->Name ?></td> <?php endif ?>  
            <?php if(strpos($q,$Continent)||strpos($q,$all)): ?> <td><?php echo $b->Continent ?></td> <?php endif ?>  
            <?php if(strpos($q,$region)||strpos($q,$all)): ?> <td><?php echo $b->Region ?></td> <?php endif ?>  
            <?php if(strpos($q,$surfacearea)||strpos($q,$all)): ?> <td><?php echo $b->SurfaceArea ?></td> <?php endif ?>  
            <?php if(strpos($q,$indepyear)||strpos($q,$all)): ?> <td><?php echo $b->IndepYear ?></td> <?php endif ?>  
            <?php if(strpos($q,$population)||strpos($q,$all)): ?> <td><?php echo $b->Population ?></td> <?php endif ?>  
            <?php if(strpos($q,$lifeexpectancy)||strpos($q,$all)): ?> <td><?php echo $b->LifeExpectancy ?></td> <?php endif ?>  
            <?php if(strpos($q,$gnp)||strpos($q,$all)): ?> <td><?php echo $b->GNP ?></td> <?php endif ?>  
            <?php if(strpos($q,$gnpold)||strpos($q,$all)): ?> <td><?php echo $b->GNPOLD ?></td> <?php endif ?>  
            <?php if(strpos($q,$localname)||strpos($q,$all)): ?> <td><?php echo $b->LocalName ?></td> <?php endif ?>  
            <?php if(strpos($q,$governmentform)||strpos($q,$all)): ?> <td><?php echo $b->GovernmentForm ?></td> <?php endif ?>  
            <?php if(strpos($q,$headofstate)||strpos($q,$all)): ?> <td><?php echo $b->HeadOfState ?></td> <?php endif ?>  
            <?php if(strpos($q,$capital)||strpos($q,$all)): ?> <td><?php echo $b->Capital ?></td> <?php endif ?>  
            <?php if(strpos($q,$code2)||strpos($q,$all)): ?> <td><?php echo $b->Code2 ?></td> <?php endif ?>  
        </td>  
    </tr>  
<?php endforeach ?>
```

#### NOTE

Difficulté rencontrée : Nous avons eu des difficultés de compréhension sur l'usage de la fonction "strpos" et provoquait une confusion sur l'affichage du tableau. De plus, nous n'avons pas compris pourquoi les ( ' ') et les ( " ") dans la requête SQL n'étaient pas reconnus à cause de la fonction "htmlspecialchars".

### III. Partie Adrien

#### a) Mise à jour des données manuellement pour l'enseignant et l'administrateur

Après avoir attribué les droits pour chaque cession des utilisateurs, nous pouvons exploiter la mise à jour des données des pays pour l'administrateur et l'enseignant.

Avant toute chose, nous créons une fonction « updategeoworld » dans la page « manager-db » pour créer une requête et récupérer l'id de la table country.

```
function updategeoworld($id){
    global $pdo;
    $requete = "SELECT * FROM country where id = :id";
    try{
        $prep = $pdo->prepare($requete);
        $prep->bindParam(':id', $id, PDO::PARAM_INT);
        $prep->execute();
        $result = $prep->fetch();
        return $result;
    }
    catch ( Exception $e ) {
        die ("erreur dans la requete ".$e->getMessage());
    }
}
```

Puis nous créons un lien « Update » sur la page « continent.php » qui nous renvoie sur une page « updatepays.php » grâce à la balise <a href="updatepays.php?id=<?php echo \$pays->id ?>" afin de modifier les informations d'un pays en récupérant l'id du pays.

Ensuite, dans la page updatepays.php on initialise une variable pour récupérer l'id de la base de données \$id = \$\_GET['id'] ; dans la page «updatepays.php » et on initialise une variable pour appelé la fonction \$updatePays = updategeoworld(\$id);

Sur cette même page, nous créons un formulaire qui va récupérer et vérifier les données reçues de la page « continent.php » et qui enverra ces données sur la page «updatepaysmaj.php ».



```

if ( isset($_GET['id']) && !empty($_GET['id'])) {
    $id = $_GET['id'];
}
else {
    $id = $_GET['id'];
}
echo $id;

```

Les variables récupérées vont être insérées avec la méthode \$\_GET.

```
$code = $_GET['code'];
```

Puis on rédige la requête pour mettre à jour les données.

```

$sql = "update country set Code=:code, Name=:name, Continent=:continent,
Region=:region, SurfaceArea=:surfacearea, IndepYear=:indepyear,
Population=:population, LifeExpectancy=:lifeexpectancy, GNP=:gnp, GNPOld=:gnpold,
LocalName=:localname, GovernmentForm=:governmentform, HeadOfState=:headofstate,
Capital=:capital, Code2=:code2 where id=:id";

```

Enfin, on prépare la requête avec les données reçues.

```

$stmt = $pdo->prepare($sql);
$stmt->bindParam(':code', $code, PDO::PARAM_STR);
$stmt->bindParam(':name', $pays, PDO::PARAM_STR);
$stmt->bindParam(':continent', $continent, PDO::PARAM_STR);
$stmt->bindParam(':region', $region, PDO::PARAM_STR);
$stmt->bindParam(':surfacearea', $surface, PDO::PARAM_STR);
$stmt->bindParam(':indepyear', $indep, PDO::PARAM_STR);
$stmt->bindParam(':population', $population, PDO::PARAM_STR);
$stmt->bindParam(':lifeexpectancy', $lifeexpectancy, PDO::PARAM_STR);
$stmt->bindParam(':gnp', $gnp, PDO::PARAM_STR);
$stmt->bindParam(':gnpold', $gnpold, PDO::PARAM_STR);
$stmt->bindParam(':localname', $localname, PDO::PARAM_STR);
$stmt->bindParam(':governmentform', $gouvernement, PDO::PARAM_STR);
$stmt->bindParam(':headofstate', $headofstate, PDO::PARAM_STR);
$stmt->bindParam(':capital', $capital, PDO::PARAM_STR);
$stmt->bindParam(':code2', $code2, PDO::PARAM_STR);
$stmt->bindParam(':id', $id, PDO::PARAM_STR);
$stmt->execute();

```

Et, on renvoie vers la liste des continents.

```
header("Location:continent.php");
```

## b) Exploitation de la barre de recherche

Nous avons établi une barre de recherche qui permet de lister les pays et leurs informations avec le nom du pays.

Pour ce faire on réalise la requête SQL qui permet de récupérer les données de la table country dans la base de données et qui les affichera.

```
$recherche = $pdo->query('SELECT * FROM country');
```

Puis nous réalisons une seconde requête qui affichera les pays qui comporteront le paramètre « q » entré par l'utilisateur. (Exemple : « J » = Japan, Jordan, Jamaica)

```
if(isset($_GET['q']) AND !empty($_GET['q'])) {  
    $q = htmlspecialchars($_GET['q']);  
    $recherche = $pdo->query('SELECT * FROM country WHERE Name LIKE "'. $q. '%" ORDER BY  
id DESC');  
}
```

Puis nous affichons les données que nous souhaitons faire apparaître lorsque l'utilisateur recherche un pays dans une ligne de tableau avec une boucle foreach.

```
<?php foreach($recherche as $b): ?>  
    <tr>  
        <td><?php echo $b->Name ?></td>  
        <td><?php echo $b->Continent ?></td>  
        <td><?php echo $b->Region?></td>  
        <td><?php echo $b->Population ?></td>  
        <td><?php echo $b->LifeExpectancy ?></td>  
        <td><?php echo $b->LocalName ?></td>  
        <td><?php echo $b->GovernmentForm ?></td>  
        <td><?php echo $b->HeadOfState ?></td>  
    </tr>  
<?php endforeach ?>
```

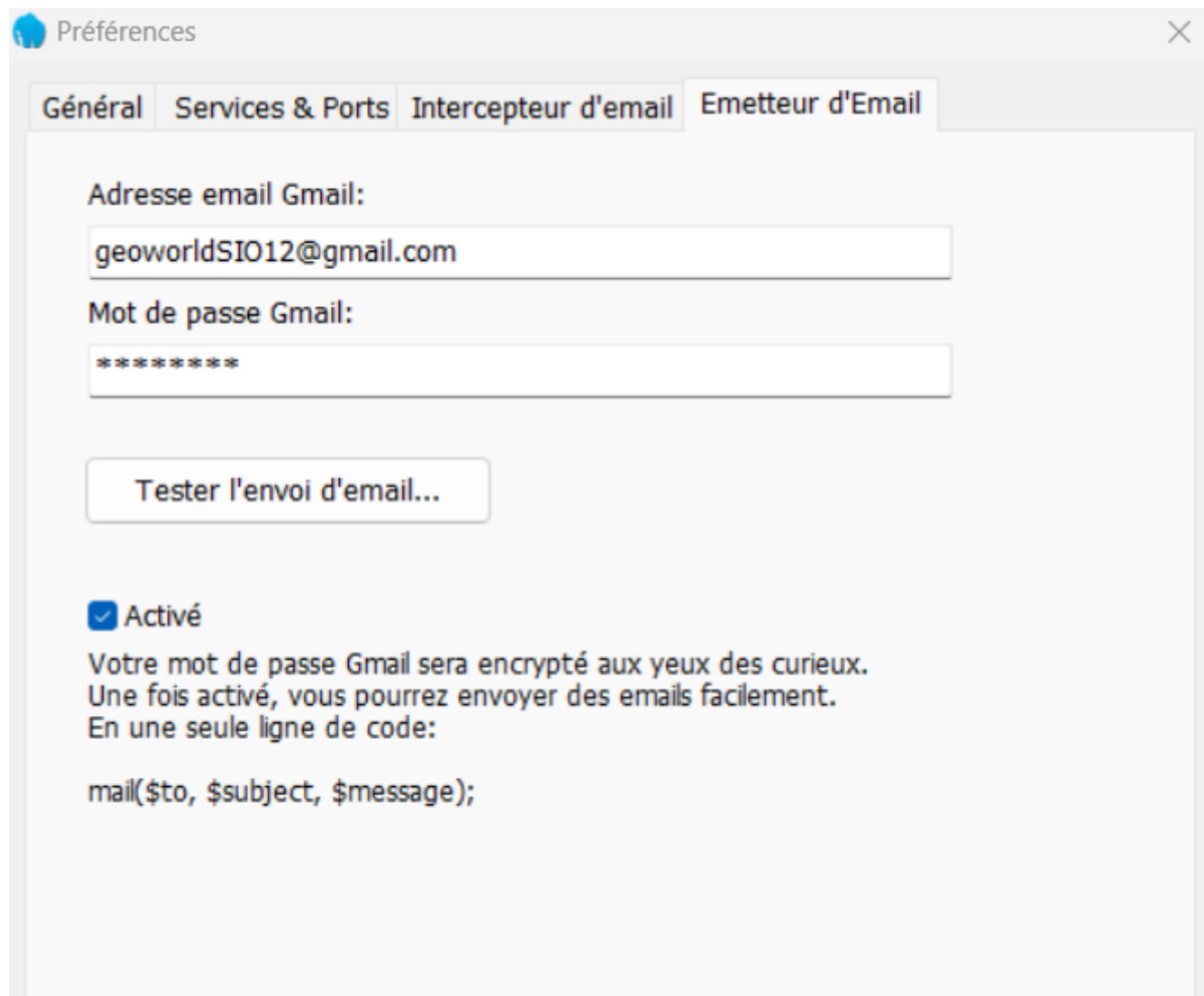
### NOTE

Difficulté rencontrée : Nous avons rencontré des difficultés sur l'envoi et la réception des données pour modifier les informations des pays. Les liens créés entre les différentes pages à été difficile à cerner.

## IV. Missions complémentaires

### a) Email

Nous avons créé un formulaire pour notre page contact qui est relié à une adresse Gmail et lié a Laragon afin de permettre à l'utilisateur d'envoyer son message et de permettre à l'admin de recevoir le message via l'email.



Préférences

Général Services & Ports Intercepteur d'email **Emetteur d'Email**

Adresse email Gmail:

geoworldSIO12@gmail.com

Mot de passe Gmail:

\*\*\*\*\*

Tester l'envoi d'email...

☒ Activé

Votre mot de passe Gmail sera encrypté aux yeux des curieux.  
Une fois activé, vous pourrez envoyer des emails facilement.  
En une seule ligne de code:

mail(\$to, \$subject, \$message);

### b) Cookies

Nous avons pris un script d'un cookie sur internet afin d'avoir notre barre de cookie.

### c) Mentions légales

Enfin nous avons introduit une page de mentions légale qui elle meme relié à la barre de cookie afin d'avoir des informations sur Geoworld et les regles RGPD.

### d) Geomap

Nous avons inséré un lien afin de visualiser une carte du monde.

## **V. Conclusion**

Le projet nous a soumis à différentes difficultés comme la répartition des tâches, la difficulté à comprendre les fonctions, à résoudre et déboguer des erreurs. De plus, nous avons eu des difficultés sur l'utilisation du logiciel Git/Github pour continuer sur l'avancé du projet.