

Rapport du projet geoworld

BELKACEM Nassim

DARRAS Loïc

KINAMVUIDI Jonathan

Rapport établi le 14/03/2023

I) Présentation:

Dans le cadre de l'atelier professionnel, Nassim, Jonathan et moi-même avons fait, programmé et établi le projet "Geoworld" dans sa globalité.

Pour cela, nous avons suivi à la lettre les consignes données par nos enseignants, en nous aidant:

- du tableau "Agile - GEOWORLD" sur Trello :

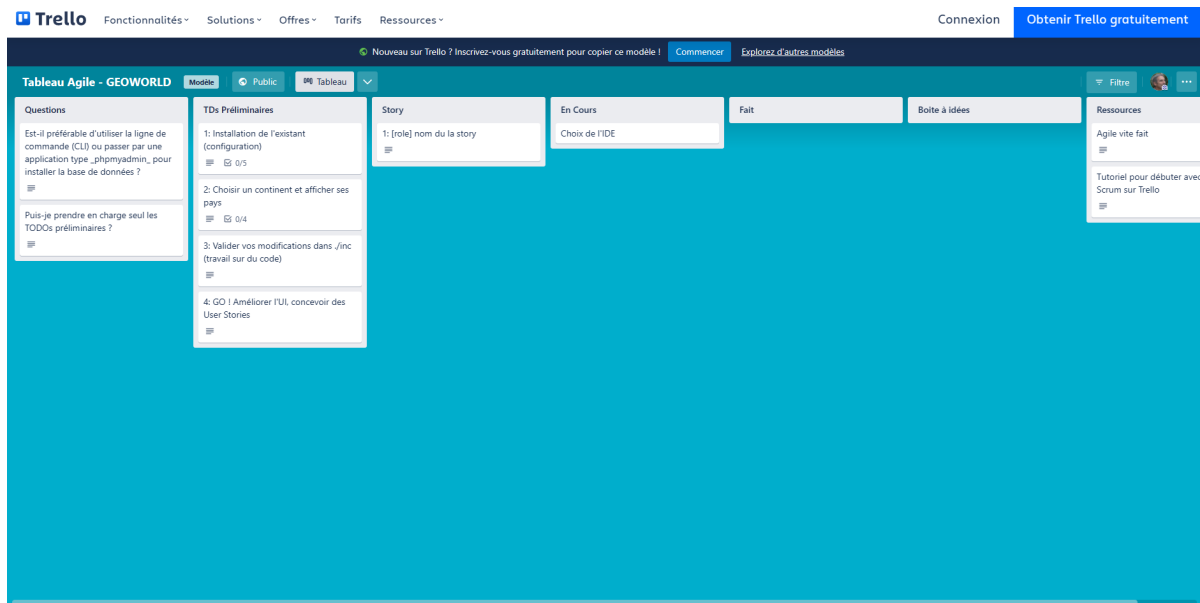


Figure 1. Image de l'outil concernant le **référentiel commun** de notre team : <https://trello.com/b/R4y72LIN/tableau-agile-geoworld>

- et du lien github du projet "geoworld" :

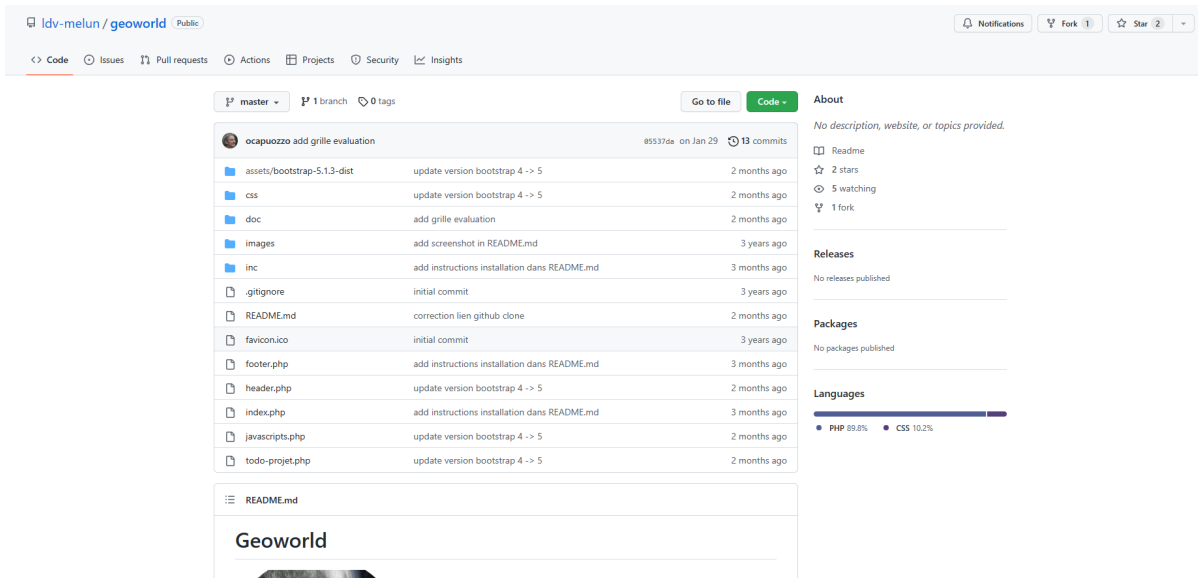


Figure 2. lien github du projet : <https://github.com/ldv-melun/geoworld>

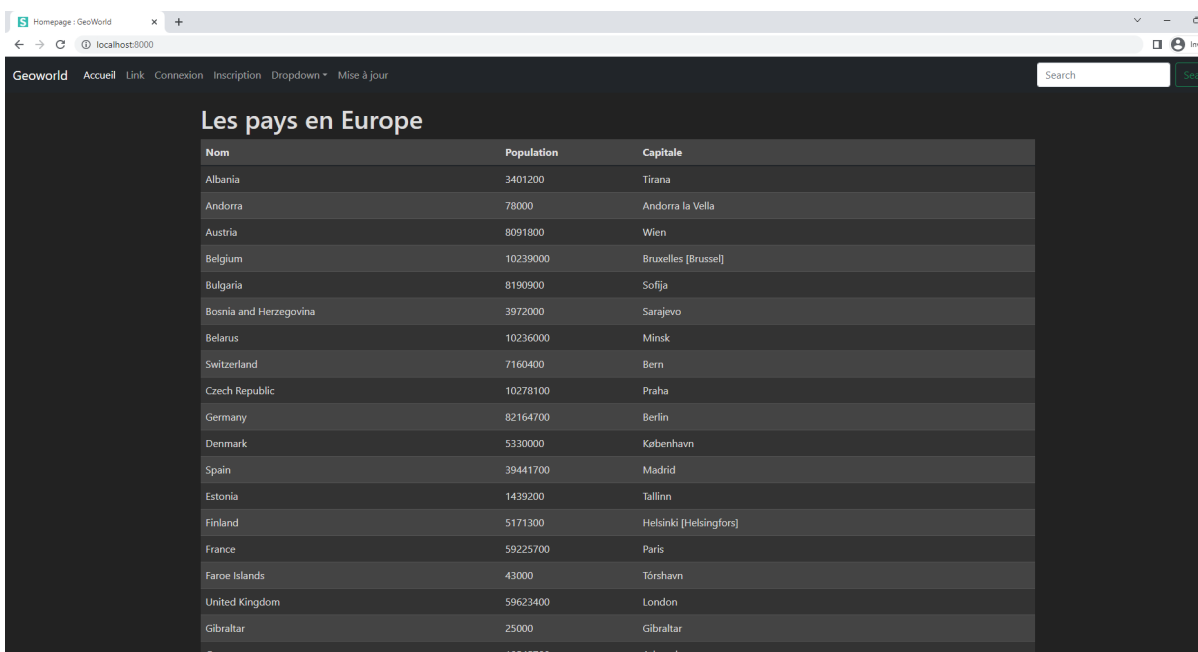
II) Ordre des étapes obligatoires du projet:

Première étape:

- Nous avons suivi la procédure indiquée par le **README.md** du "repository" du projet geoworld:

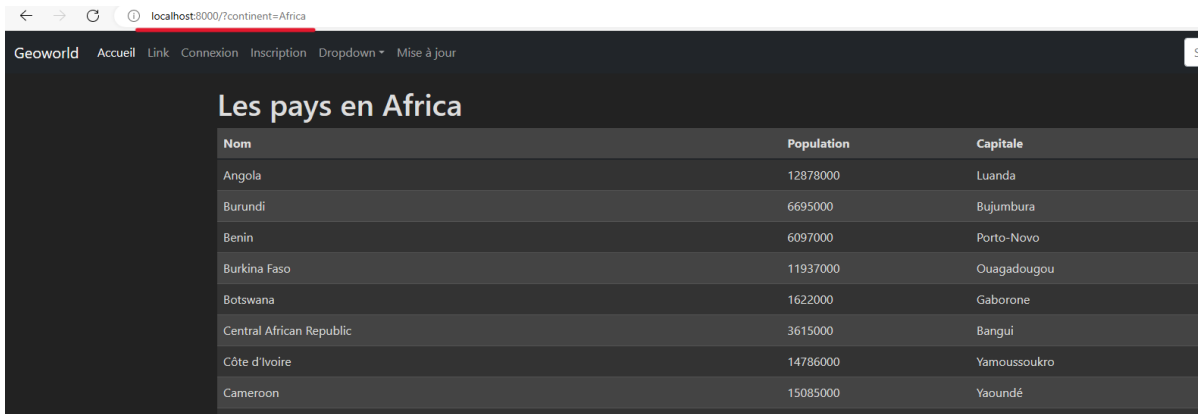
C'est à dire insérer le schéma de la base de données "geoworld" et celui de l'insertion des données.

- Nous avons créé la base de données **geoworld** à l'aide de : **"mysql > create database geoworld;"** et créé un utilisateur avec **"mysql> create user 'appuser'@%' identified by 'ThePassword;'"**.
- Ensuite nous avons lancé un service HTTP sachant interpréter du code PHP : **php -S localhost:8000** et à l'aide d'un navigateur (à l'occurrence pour nous "Google Chrome"), nous avons ouvert l'URL **localhost:8000** :



Deuxième étape:

- Il est tout à fait possible de spécifier un continent particulier à l'aide d'une valeur passée en argument de l'URL: `.../geoworld?continent="Nom_du_continent"`

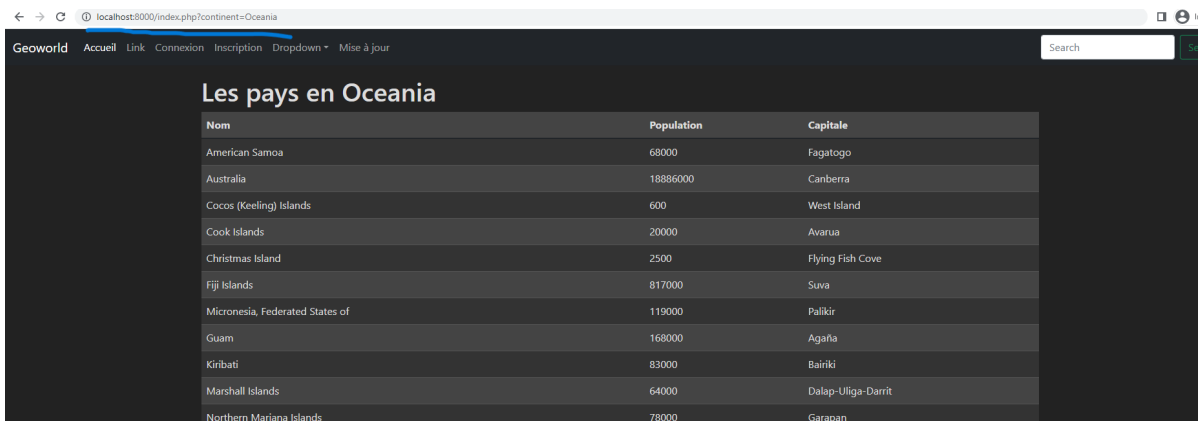


Geoworld Accueil Link Connexion Inscription Dropdown Mise à jour

Les pays en Africa

Nom	Population	Capitale
Angola	12878000	Luanda
Burundi	6695000	Bujumbura
Benin	6097000	Porto-Novo
Burkina Faso	11937000	Ouagadougou
Botswana	1622000	Gaborone
Central African Republic	3615000	Bangui
Côte d'Ivoire	14786000	Yamoussoukro
Cameroon	15085000	Yaoundé

Figure 3. liste des pays du continent "Africa" à l'aide de `.../geoworld?continent=Africa`



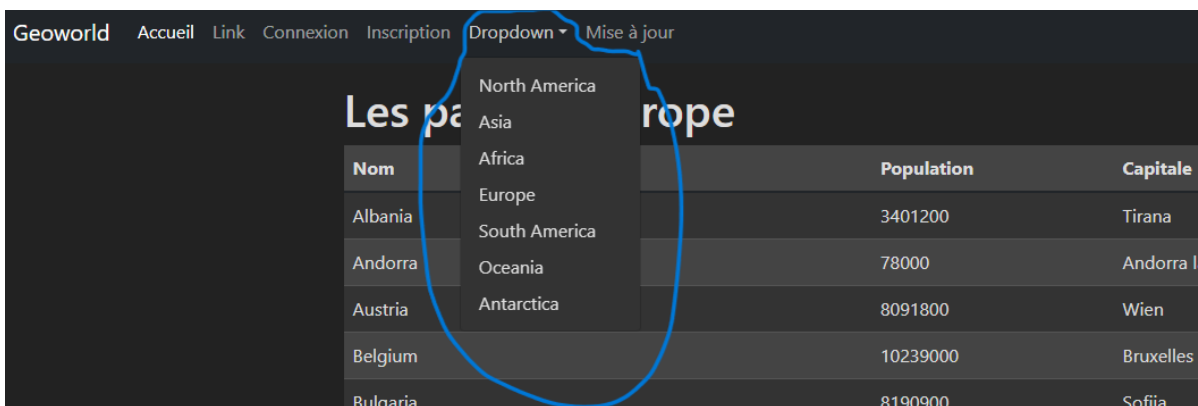
Geoworld Accueil Link Connexion Inscription Dropdown Mise à jour

Les pays en Oceania

Nom	Population	Capitale
American Samoa	68000	Fagatogo
Australia	18886000	Canberra
Cocos (Keeling) Islands	600	West Island
Cook Islands	20000	Avarua
Christmas Island	2500	Flying Fish Cove
Fiji Islands	817000	Suva
Micronesia, Federated States of	119000	Palikir
Guam	168000	Agaña
Kiribati	83000	Bairiki
Marshall Islands	64000	Dalap-Uliga-Darrit
Northern Mariana Islands	78000	Garapan

Figure 4. liste des pays du continent "Oceania" à l'aide de `.../geoworld?continent=Oceania`

- Nous avons bien adapté le "**Dropdown**" de la barre de menu afin de lister le nom de tous les continents que l'utilisateur aura le privilège de sélectionner (*North America, Asia, Africa, Europe, South America, Oceania, et Antarctica*):



Geoworld Accueil Link Connexion Inscription Dropdown Mise à jour

Les pays en Europe

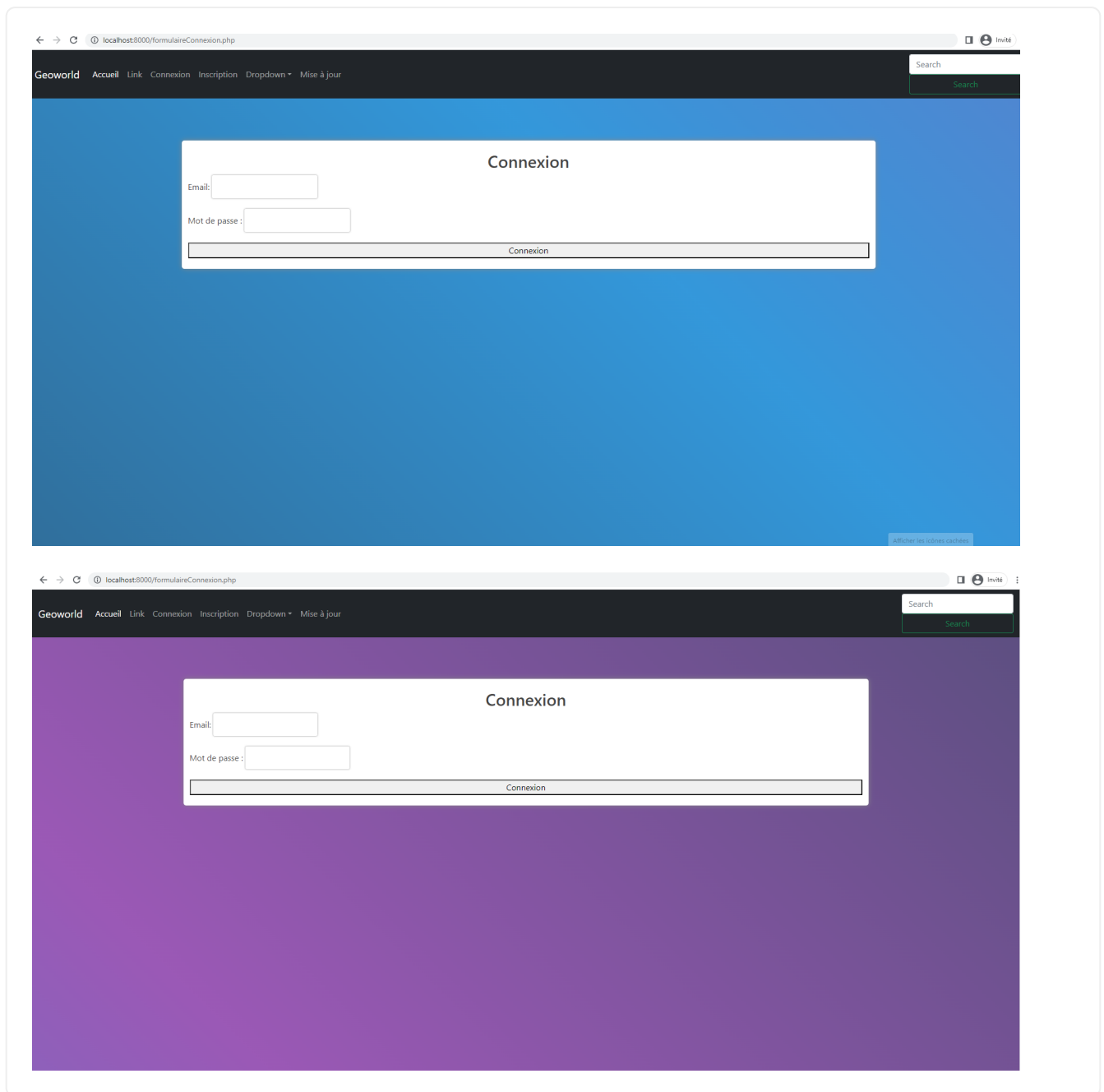
Nom	Population	Capitale
Albania	3401200	Tirana
Andorra	78000	Andorra la
Austria	8091800	Wien
Belgium	10239000	Bruxelles
Bulgaria	8190900	Sofia

- Nous pouvons aussi noter que nous avons bien **les noms des capitales** qui s'affichent à la place de l'id des capitales, ceci est permis à l'aide la fonction suivante:

```
function getNomCapitale($capital)
{
    global $pdo;
    $query = 'SELECT Name from city WHERE id =:Capital;';
    $prep= $pdo->prepare($query);
    $prep->bindValue(':Capital', $capital, PDO::PARAM_STR);
    $prep->execute();
    return $prep->fetchAll();
}
```

III) Actions secondaires du projet:

- Au sein de la programmation du site nous avons intégrer un formulaire d'inscription et un formulaire de connexion.
 - Le "background" du formulaire de connexion est caractérisé par une variation progressive de couleur entre le bleu et le violet :



Voici le code css permettant ce rendu illustratif:

```

# styleConnexion.css > 🚀 body
1  body {
2      background: linear-gradient(45deg, ■ #2c3e50, ■ #3498db, ■ #9b59b6, ■ #34495e); /* définit le dégradé de couleur */
3      background-size: 400% 400%; /* taille du dégradé */
4      animation: gradient 15s ease infinite; /* animation du dégradé */
5  }
6
7  @keyframes gradient {
8      0% {
9          background-position: 0% 50%; /* position initiale du dégradé */
10     }
11     50% {
12         background-position: 100% 50%; /* position intermédiaire du dégradé */
13     }
14     100% {
15         background-position: 0% 50%; /* position finale du dégradé */
16     }
17 }
18
19 .container {
20     width: 400px;
21     margin: auto;
22     margin-top: 100px;
23     padding: 20px;
24     background-color: ■ #fff;
25     border-radius: 5px;
26     box-shadow: 0px 0px 10px ■ #888888;
27 }
28
29 h2 {
30     text-align: center;
31     color: ■ #444444;
32     margin-bottom: 30px;
33 }
34
35 form {
36     display: flex;
37     flex-direction: column;
38 }
39

```

```

35  form {
36  |    display: flex;
37  |    flex-direction: column;
38  |  }
39
40  label {
41  |    margin-bottom: 10px;
42  |    color: #444444;
43  |  }
44
45  input[type="text"], input[type="password"] {
46  |    padding: 10px;
47  |    margin-bottom: 20px;
48  |    border: none;
49  |    border-radius: 3px;
50  |    box-shadow: 0px 0px 3px #888888;
51  |  }
52
53  input[type="submit"] {
54  |    background-color: #4CAF50;
55  |    color: white;
56  |    padding: 10px 20px;
57  |    border: none;
58  |    border-radius: 3px;
59  |    cursor: pointer;
60  |  }
61
62  input[type="submit"]:hover {
63  |    background-color: #3e8e41;
64  |  }
65

```

- Rentrant plus dans les détails du code php avec la partie "**back-end**":

La vérification des données transmises par l'utilisateur se fait grâce à la liaison avec *login.php* qui récupère les données envoyées par l'utilisateur avec la méthode "POST":

```
<form action="login.php" method="post">
```

La fonction "getAuthentification" permet de sélectionner l'email et le mot de passe des utilisateurs via la base de données spécifiée:

```
//
function getAuthentification($email,$mdp)
{
    global $pdo;
    $query = "SELECT * FROM utilisateurs where email=:email and mdp=:mdp";
    $prep = $pdo->prepare($query);
    $prep->bindValue(':email', $email);
    $prep->bindValue(':mdp', $mdp);
    $prep->execute();
    // on vérifie que la requête ne retourne qu'une seule ligne
    if($prep->rowCount() == 1){
        $result = $prep->fetch();
        return $result;
    }
    else
        return false;
}
}
```

Avec cette fonction, on envoie, on vérifie et on récupère les saisies. La récupération se fait à l'aide "\$résultat":

```
$result = getAuthentification($_POST['email'],$_POST['mdp']);
print_r($result);
```

Nous avons mis en place, dans le projet, un fichier php consacré au succès des saisies (**succes.php**):

```
suces.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Connexion réussie</title>
5  </head>
6  <body>
7  |   <h1>Félicitations, vous vous êtes connecté avec succès!</h1>
8  |   <a href="index.php">Retour à l'accueil</a>
9  </body>
10 </html>
11
12
```

et un autre pour son opposé, en cas d'erreur de saisies:

```
echec.php
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Connexion échouée</title>
5  </head>
6  <body>
7  |   <h1>Désolé, la connexion a échoué. Veuillez vérifier vos identifiants et réessayer.</h1>
8  </body>
9  </html>
10
```

La liaison entre **login.php** et ces fichiers s'expliquent par ce bout de programme:


```
// on redirige notre visiteur vers une page de notre section membre
header ('location: succes.php');

}

//si le résultat est false on redirige vers la page d'authentification
else{
header ('location: echec.php');
}
}

//si nos variables ne sont pas renseignées on redirige vers la page d'authentification
else {
header ('location: echec.php');
}
}
```

- Pour la page d'inscription, nous avons établi un formulaire classique avec le nom, le prénom, l'email, le mot de passe et un rôle interchangeable entre élève et enseignant:

Le code "back-end" de cette page fonctionne avec la même procédure que la page connexion:

Nous récupérons les données avec "**inscription.php**" qui permet d'intégrer le nom et l'email qui sont saisis.

```
<form action="inscription.php" action="#" method="post">
```

Pour éviter les confusions, nous avons instauré une vérification par requête SQL de l'email saisie par l'utilisateur:

```
// Vérifie si l'adresse e-mail existe déjà dans la base de données
$requete = $pdo->prepare('SELECT COUNT(*) FROM utilisateurs WHERE email = ?');
$requete->execute([$email]);
$resultat = $requete->fetchColumn();
```

Enfin, nous avons créé une rubrique "mise à jour" ou cas où si nous voulons améliorer notre site d'un point de vue personel pour nous entraîner...

FIN DU COMPTE RENDU

NOTE

Si vous voulez contacter un des développeurs de ce projet, vous pouvez vous aider du tableau de coordonnées suivant:

NOM/Prénom	email
BELKACEM Nassim	emailpro1@gmail.com
DARRAS Loïc	emailpro2@gmail.com
KINAMVUIDI Jonathan	emailpro3@gmail.com