1 La méthode de la double authentification

1.1) Explication :

La méthode de la double authentification permet de renforcer à la sécurité en rajoutant une couche de sécurité et en envoyant un code par mail ou message à l'utilisateur après qu'il ait rentré ses informations de connexion. Dans notre cas on va pas se servir de mail ou de message mais d'une application qui génére des mots de passe TOTP pour Time based One Time Password.

1.2) Tutoriel :

Etape 1 (Base du code) :

Avoir une base de code peut importe l'architecture utilisée pour développer l'application. C'est-à-dire :

- Avoir un formulaire d'inscription et de connexion

- Avoir les contrôleurs et le modèle qui gère l'inscription et la connexion si on utilise une architecture MVC

- Avoir une page d'accueil pour être redirigé après la connexion.

- Avoir un script de déconnexion.

Etape 2 (Implémentation) :

- Installer à l'aide de Composer la librairie TwoFactorAuth depuis github . Cette librairie contient déjà du code qui va nous aider à implémenter la double authentification dans notre application.

- Si l'installation a réussie, un dossier vendor doit apparaître sur VS Code.

-Installer Google Authenticator sur son téléphone IOS ou Android. Cette application génére des codes pour l'authentification à double facteurs.

Index.php:

```
session_start();
require('./vendor/autoload.php');
```

Ici on démarre la session dans l'index, car on va se servir de variables de session tout au long du code.

On inclus autoload.php du dossier vendor pour pouvoir utiliser la librairie TwoFactorAuth dans notre code.

v_connexion.php :

```
<form class="form-signin" id="connexion" method="POST"
action="index.php?uc=connexion&action=seCo">
      <div class="text-center">
      <div class="container-fluid">
      <div class="row">
            <div class="col-lg-3"></div>
            <div class="col-lg-6">
            </br></br>
            <h1 class="h3 mb-3 font-weight-normal">Connectez vous ! </h1>
            </br>
            <label class="sr-only">Login</label>
            <input type="text" name="login" class="form-control" placeholder="Login"
style="background-color: #fff0c1
            " required>
            </br>
            <label class="sr-only">Mot de passe</label>
            <input type="password" name="mdp" class="form-control" placeholder="Mot de passe"
style="background-color: #fff0c1
            " required>
            </br>
            <br><br><br>
            <button class="btn btn-lg btn-light btn-block"  type="submit">Se connecter</button>
            </div>
        <div class="col-lg-3"></div>
      </div>
      </div>
      </div>
</form>
```

Ici on crée le formulaire qui permet de se connecter.

contrôleur c_gererConnexion.php :

```php
use RobThreeAuthTwoFactorAuth;

  if (!empty($_POST['login']) && !empty($_POST['mdp'])) {
        var_dump($_POST);

        $login = $_POST['login'];
        $mdp = $_POST['mdp'];
        $tfaCode = $_POST['tfa_code'];

      $user=$pdo->recupUsersLogin($login);

      if ($user) {
            if (password_verify($mdp, $passwordHash)) {
                $tfa = new TwoFactorAuth();
                if (!$user['secret'] || $tfa->verifyCode($user['secret'], $tfaCode)) {
                   $_SESSION['user_id'] = $user['id'];
                   header('location: index.php?uc=accueil&action=accueil');
                   exit();
                } else {
                   echo "Code 2FA invalide";
                   }
            } else {
                echo "Identifiants invalides";
              }
       } else {
            echo "Identifiants invalides";
         }
      }
```

Ici on utilise la librairie, on vérifie que les identifiants de connexion ne sont pas vides et si c'est le cas, on les récupèrent, on récupère également le code généré.

On utilise la fonction du modèle qui nous permet de récupérer l'utilisateur en fonction du login rentré dans le formulaire.

Si cet utilisateur existe et que le mot de passe est le bon, on crée une instance de la classe TwoFactorAuth qui contient les méthodes pour mener à bien une double authentification.

Si l'utilisateur n'a pas de secret ou si le code qu'il a fourni correspond au secret, alors on le redirige vers l'accueil sinon on renvoie un message d'erreur si une des conditions précédemment dictées ne sont pas valides.

Modèle :

```php
public static function recupUsersLogin($login)
        {
            $req="SELECT * FROM users WHERE login= $login ";
            $res=PdoHeb::$monPdo->query($req);
            $user=$res->fetch();
            return $user;
        }
```

On crée une fonction qui nous permet de récupérer l'utilisateur en fonction du login rentré dans le formulaire.

```php
public static function recupUsersId($id)
        {
            $req="SELECT * FROM users WHERE id=$id";
            $res=PdoHeb::$monPdo->query($req);
            $user=$res->fetch();
            return $user;
        }
```

On crée une fonction qui va récupérer toutes les informations de l'utilisateur en fonction de son id.

```php
public static function modifOnUsers($id,$secret)
        {
            $req="UPDATE users SET secret =$secret WHERE id=$id";
            $res=PdoHeb::$monPdo->exec($req)
            return "Le secret est bien implémenté.";
        }
```

On crée une fonction qui modifie la table users en rajoutant un secret dans la ligne de l'utilisateur en fonction de son id.

contrôleur (après connexion, crée le secret si pas fait pour la prochaine fois) :

```php
use RobThreeAuthTwoFactorAuth;
    $tfa = new TwoFactorAuth();

    if (empty($_SESSION['tfa_secret'])) {
            $_SESSION['tfa_secret'] = $tfa->createSecret();
    }
    $secret = $_SESSION['tfa_secret'];

    if (empty($_SESSION['user_id'])) {
            header('location: index.php');
            exit();
    }

    if (!empty($_POST['tfa_code'])) {
            if ($tfa->verifyCode($secret, $_POST['tfa_code'])) {
                modifOnUsers($id,$secret);
            } else {
                echo "Code invalide";
            }
    }
```

Après la connexion, ce contrôleur crée un secret si l'utilisateur en question n'en avait pas. Si le code que l'utilsateur rentre est bien similaire au secret qui lui a été envoyé, alors un nouveau secret est implémenté dans la base de données et c'est ce secret qui lui sera fourni la prochaine fois qu'il se connectera.

v_accueil.php :

```php
<h1>Votre profil</h1>

    <?php var_dump($user) ?>

    <h2>Activation Double Authentification</h2>

    <?php if (!$user['secret']): ?>
            <p>Code secret : <?= $secret ?></p>
            <p>QR Code :</p>
            <img src="<?= $tfa->getQRCodeImageAsDataUri('Tuto', $secret) ?>">
            <form method="POST">
                <input type="text" placeholder="Vérification Code" name="tfa_code">
                <button type="submit">Valider</button>
            </form>
    <?php else: ?>
            <p>2FA activée</p>
    <?php endif ?>
```

Cette vue est la vue qui affiche la connexion pour la double authentification.

1.3.) Exemple de code source

Ce code est une base. Il est à réadapter selon l'architecture utilisée lors du développement.

- config.php

```php
<?php
session_start();
require('./vendor/autoload.php');

try {
  $db = new PDO("mysql:host=localhost;dbname=double_auth", 'root', 'root');
  $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
  echo "Connection failed: " . $e->getMessage();
}
```

- index.php

```php
<form action="register.php" method="POST">
    <input name="email" type="email" placeholder="Email" /><br />
    <input name="password" type="password" placeholder="Mot de passe" /><br />
    <button type="submit">Inscription</button>
</form>
```

- register.php

```php
<?php
require('./config.php');

if (!empty($_POST['email']) && !empty($_POST['password'])) {
    $email = $_POST['email'];
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT);

    var_dump($email);
    var_dump($password);

    $q = $db->prepare('INSERT INTO users (email, password) VALUES (:email, :password)');
    $q->bindValue('email', $email);
    $q->bindValue('password', $password);
    $res = $q->execute();

    if ($res) {
        echo "Inscription réussie";
    }
}
```

- login.php

```php
<?php
require('./config.php');

use RobThreeAuthTwoFactorAuth;

if (!empty($_POST['email']) && !empty($_POST['password'])) {
    var_dump($_POST);

    $email = $_POST['email'];
    $password = $_POST['password'];
    $tfaCode = $_POST['tfa_code'];

    $q = $db->prepare('SELECT * FROM users WHERE email = :email');
    $q->bindValue('email', $email);
    $q->execute();
    $user = $q->fetch(PDO::FETCH_ASSOC);

    var_dump($user);

    if ($user) {
        $passwordHash = $user['password'];
        if (password_verify($password, $passwordHash)) {
            $tfa = new TwoFactorAuth();
            if (!$user['secret'] || $tfa->verifyCode($user['secret'], $tfaCode)) {
                $_SESSION['user_id'] = $user['id'];
                header('location:/profile.php');
                exit();
            } else {
                echo "Code 2FA invalide";
            }
        } else {
            echo "Identifiants invalides";
        }
    } else {
        echo "Identifiants invalides";
    }
}
```

- profile.php

```php
<?php
require('./config.php');

use RobThreeAuthTwoFactorAuth;
$tfa = new TwoFactorAuth();

if (empty($_SESSION['tfa_secret'])) {
    $_SESSION['tfa_secret'] = $tfa->createSecret();
}
$secret = $_SESSION['tfa_secret'];

if (empty($_SESSION['user_id'])) {
    header('location:/');
    exit();
}

if (!empty($_POST['tfa_code'])) {
    if ($tfa->verifyCode($secret, $_POST['tfa_code'])) {
        $q = $db->prepare('UPDATE users SET secret = :secret WHERE id = :id');
        $q->bindValue('secret', $secret);
        $q->bindValue('id', $_SESSION['user_id']);
        $q->execute();
    } else {
        echo "Code invalide";
    }
}

$userReq = $db->prepare('SELECT * FROM users WHERE id = :id');
$userReq->bindValue('id', $_SESSION['user_id']);
$userReq->execute();
$user = $userReq->fetch(PDO::FETCH_ASSOC);

?>
<h1>Votre profil</h1>

<a href="/logout.php">Déconnexion</a>
<?php var_dump($user) ?>

<h2>Activation Double Authentification</h2>

<?php if (!$user['secret']): ?>
    <p>Code secret : <?= $secret ?></p>
    <p>QR Code :</p>
    <img src="<?= $tfa->getQRCodeImageAsDataUri('Tuto', $secret) ?>">
    <form method="POST">
        <input type="text" placeholder="Vérification Code" name="tfa_code">
        <button type="submit">Valider</button>
    </form>
<?php else: ?>
    <p>2FA activée</p>
<?php endif ?>
```

- logout.php

```php
<?php
require('./config.php');

$_SESSION = [];
session_destroy();
header('location:/');
exit();
```