

2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs
	Date de rédaction : 29/09/2025

## Projet site web R3st0.fr - Itération n°2 – résolution de tickets

Bourgeois / SiteResto2025 / Tickets / #5

### Ticket n°5 - amélioration du chiffrement des mots de passe

Modifier :

Ouvert Incident créé Il y a 7 minutes par **Bourgeois**

Résumé Chronologie

Par souci de renforcer la sécurité de l'application, les maîtres d'œuvre souhaitent faire évoluer le traitement des mots de passe des utilisateurs (authentification, nouvelle inscription) pour respecter les préconisations de PHP en la matière. Actuellement, le chiffrement des mots de passe utilise la fonction crypt() avec un sel simpliste (= "sel"). PHP conseille l'usage du couple de fonctions password\_hash / password\_verify avec l'algorithme de hachage par défaut BCrypt.

Références :

- password\_hash <https://www.php.net/manual/fr/function.password-hash.php>
- crypt <https://www.php.net/manual/fr/function.crypt.php>

**Avantages de l'utilisation de password\_hash :** • meilleur algorithme de hachage par défaut (BCRYPT) • évolutive (adaptation automatique aux améliorations des algorithmes) • salage efficace • compatibilité avec crypt, donc les anciens mots de passe resteront utilisables, même s'il sera préférable de les modifier pour générer une meilleure empreinte.

**Lexique** fonction de hachage : calcule une empreinte numérique non réversible. salage : renforce la sécurité du hachage en y ajoutant une donnée supplémentaire (le sel) afin d'empêcher que deux informations identiques conduisent à la même empreinte => protège des attaques par force brute et par table arc-en-ciel. coût : rend l'algorithme arbitrairement lent et contribue à dissuader les attaques par table arc-en-ciel et par force brute. table arc-en-ciel : table comportant un grand nombre d'empreintes connues, permettant de retrouver un mot de passe à partir de son empreinte.

Modification effectuée à l'instant par **Bourgeois**



Créer une requête de fusion

Éléments enfants 0

Ajouter

Aucun élément enfant n'est actuellement assigné. Utilisez des éléments enfants pour diviser ce ticket en parties plus petites.

Incidents ou tickets liés 0

Ajouter

Références :

- password\_hash <https://www.php.net/manual/fr/function.password-hash.php>
- crypt <https://www.php.net/manual/fr/function.crypt.php>

Ajouter une tâche à faire >>

Personne assignée Modifier  
Aucun(e) - assigner à vous-même

Gravité Modifier  
Élevée - S2

Paging status Modifier  
Déclenché

Labels Modifier  
complexe x prioritaire x

Jalon Modifier  
Itération n°2

Date d'échéance Modifier  
29 sept. - supprimer la date d'échéance 2025

Suivi du temps +  
Aucune estimation ou décompte de temps

1 participant



Déplacer le ticket

2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs Date de rédaction : 29/09/2025

## Table des matières

<b>Modification du fichier UtilisateurDao.class.php :</b>	<b>3</b>
Modification de la méthode insert()	3
Modification de la méthode update()	5
Modification de la méthode updateMdp()	6
<b>Modification du fichier authentication.inc.php</b>	<b>7</b>
Modification de la fonction login()	7
<b>Modification de la classe Client.class.php :</b>	<b>9</b>
Modification de la méthode magique __toString()	9
<b>Modification du fichier connexion.php</b>	<b>9</b>
Reformatage du code	9
<b>Tests d'inscription</b>	<b>10</b>
<b>Test de connexion</b>	<b>12</b>

2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs
	Date de rédaction : 29/09/2025

## Modification du fichier UtilisateurDao.class.php :

### Modification de la méthode insert()

Tout d'abord, j'ai commencé par analyser la méthode statique insert(). Afin d'améliorer la compréhension du code, j'ai ajouté des commentaires pour chaque grandes étapes.

```

89 public static function insert(Utilisateur $unUser): bool {
90     $ok = false;
91     try {
92         $requete = "INSERT INTO utilisateur (mailU, pseudoU) VALUES (:mailU,:pseudoU)";
93         $stmt = Bdd::getConnexion()->prepare($requete);
94         // $mdpUCrypt = crypt($unUser->getMdp(), "sel");
95         $stmt->bindValue(':mailU', $unUser->getMailU(), PDO::PARAM_STR);
96         // $stmt->bindValue(':mdpU', $mdpUCrypt, PDO::PARAM_STR);
97         $stmt->bindValue(':pseudoU', $unUser->getPseudoU(), PDO::PARAM_STR);
98         $ok = $stmt->execute();
99     } catch (PDOException $e) {
100         throw new Exception("Erreur dans la méthode " . get_called_class() . " :: insert : <br/>" . $e->getMessage());
101     }
102     return $ok;
103 }

```

Figure 1 : Méthode insert() : Ancienne version.

La nouvelle méthode débute par la définition d'une requête préparée SQL. Ensuite, le mot de passe est haché et stocké dans la variable \$mdpHash. La fonction password\_hash() crée une clé de hachage pour le mot de passe de l'utilisateur. Puis, j'ai remplacé les paramètres dynamiques de la requête par leur valeur. Enfin, la requête s'exécute et si elle rencontre une erreur, elle nous est renvoyée.

2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs
	Date de rédaction : 29/09/2025

```

89 public static function insert(Utilisateur $unUser): bool {
90     $ok = false;
91     try {
92         // Préparation de la requête avec la colonne mdpU incluse
93         $requete = "INSERT INTO utilisateur (mailU, mdpU, pseudoU)
94                     VALUES (:mailU, :mdpU, :pseudoU)";
95         $stmt = Bdd::getConnexion()->prepare($requete);
96
97         // Nouveau hachage sécurisé du mot de passe
98         $mdpHash = password_hash($unUser->getMdpU(), PASSWORD_DEFAULT);
99
100        // Liaison des paramètres
101        $stmt->bindValue(':mailU', $unUser->getMailU(), PDO::PARAM_STR);
102        $stmt->bindValue(':mdpU', $mdpHash, PDO::PARAM_STR);
103        $stmt->bindValue(':pseudoU', $unUser->getPseudoU(), PDO::PARAM_STR);
104
105        // Exécution
106        $ok = $stmt->execute();
107    } catch (PDOException $e) {
108        throw new Exception("Erreur dans la méthode " . get_called_class() . "::insert : <br/>" . $e->getMessage());
109    }
110    return $ok;
111 }

```

Figure 2 : Méthode insert() : Nouvelle version.

2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs
	Date de rédaction : 29/09/2025

## Modification de la méthode update()

Par la suite, j'ai opéré de la même façon pour les deux prochaines méthodes. C'est-à-dire compréhension par la lecture et la mise en place de commentaires.

```

120 public static function update(Utilisateur $unUser): bool {
121     $ok = false;
122     try {
123         $requete = "UPDATE utilisateur SET mailU = :mailU, pseudoU = :pseudoU WHERE idU = :idU";
124         $stmt = Bdd::getConnexion()->prepare($requete);
125         // $mdpUCrypt = crypt($unUser->getMdpU(), "sel");
126         $stmt->bindValue(':mailU', $unUser->getMailU(), PDO::PARAM_STR);
127         // $stmt->bindValue(':mdpU', $mdpUCrypt, PDO::PARAM_STR);
128         $stmt->bindValue(':pseudoU', $unUser->getPseudoU(), PDO::PARAM_STR);
129         $stmt->bindValue(':idU', $unUser->getIdU(), PDO::PARAM_INT);
130         $ok = $stmt->execute();
131     } catch (PDOException $e) {
132         throw new Exception("Erreur dans la méthode " . get_called_class() . "::update : <br/>" . $e->getMessage());
133     }
134     return $ok;
135 }

```

Figure 3 : Méthode update() : Ancienne version.

De ce fait, j'ai juste eu besoin de rajouter les commentaires précédents et les deux nouvelles lignes de code à la place des deux commentaires existants. A la rigueur le seul changement notable est que j'ai ajouté le paramètre dynamique :mdpU et que j'ai modifié la requête d'origine pour l'insérer. Toutefois, je trouve cela « illogique » dans le sens où la fonction suivante est prévue pour modifier uniquement le mot de passe. Dans le doute je laisse la modification que j'ai faite mais peut-être est-ce incorrect ?

```

112 public static function update(Utilisateur $unUser): bool {
113     $ok = false;
114     try {
115         // Préparation de la requête
116         $requete = "UPDATE utilisateur SET mailU = :mailU, pseudoU = :pseudoU, mdpU = :mdpU "
117             . "WHERE idU = :idU";
118         $stmt = Bdd::getConnexion()->prepare($requete);
119
120         // Nouveau hachage sécurisé du mot de passe
121         $mdpHash = password_hash($unUser->getMdpU(), PASSWORD_DEFAULT);
122
123         // Liaison des paramètres
124         $stmt->bindValue(':mailU', $unUser->getMailU(), PDO::PARAM_STR);
125         $stmt->bindValue(':mdpU', $mdpHash, PDO::PARAM_STR);
126         $stmt->bindValue(':pseudoU', $unUser->getPseudoU(), PDO::PARAM_STR);
127         $stmt->bindValue(':idU', $unUser->getIdU(), PDO::PARAM_INT);
128
129         // Exécution
130         $ok = $stmt->execute();
131     } catch (PDOException $e) {
132         throw new Exception("Erreur dans la méthode " . get_called_class() . "::update : <br/>" . $e->getMessage());
133     }
134     return $ok;
135 }

```

Figure 4 : Méthode update() : Nouvelle version.

2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs
	Date de rédaction : 29/09/2025

## Modification de la méthode updateMdp()

Même chose encore pour la compréhension du code.

```

136 public static function updateMdp(int $idUser, string $mdpClair): bool {
137     $ok = false;
138     try {
139         $requete = "UPDATE utilisateur SET mdpU = :mdpU WHERE idU = :idUser";
140         $stmt = Bdd::getConnexion()->prepare($requete);
141         $mdpUCrypt = crypt($mdpClair, "sel");
142         $stmt->bindValue(':mdpU', $mdpUCrypt, PDO::PARAM_STR);
143         $stmt->bindValue(':idUser', $idUser, PDO::PARAM_INT);
144         $ok = $stmt->execute();
145     } catch (PDOException $e) {
146         throw new Exception("Erreur dans la méthode " . get_called_class() . " : updateMdp : <br/>" . $e->getMessage());
147     }
148     return $ok;
149 }
150
151 }

```

Figure 5 : Méthode updateMdp : Ancienne version.

J'ai simplement eu a modifié la variable \$mdpUCrypt par \$mdpHash dans laquelle j'appelle la méthode password\_hash) qui prend en paramètre le mot de passe en clair, et la constante PASSWORD\_BCRYPT. De ce fait, on ajoute cette variable dans la requête préparée qui permet de mettre à jour le mot de passe enregistré dans la BDD.

```

137 /**
138  * Mettre à jour le mot de passe d'un enregistrement à la table utilisateur
139  * @param int $idUser identifiant de l'utilisateur à mettre à jour
140  * @param string $mdpClair nouveau mot de passe non chiffré
141  * @return bool true si l'opération réussit, false sinon
142  * @throws Exception transmission des erreurs PDO éventuelles
143  */
144 public static function updateMdp(int $idUser, string $mdpClair): bool {
145     $ok = false;
146     try {
147         // Préparation de la requête avec la colonne mdpU incluse
148         $requete = "UPDATE utilisateur SET mdpU = :mdpU WHERE idU = :idUser";
149         $stmt = Bdd::getConnexion()->prepare($requete);
150
151         // Nouveau hachage sécurisé du mot de passe
152         $mdpHash = password_hash($mdpClair, PASSWORD_BCRYPT);
153
154         // Liaison des données
155         $stmt->bindValue(':mdpU', $mdpHash, PDO::PARAM_STR);
156         $stmt->bindValue(':idUser', $idUser, PDO::PARAM_INT);
157
158         // Exécution
159         $ok = $stmt->execute();
160     } catch (PDOException $e) {
161         throw new Exception("Erreur dans la méthode " . get_called_class() . " : updateMdp : <br/>" . $e->getMessage());
162     }
163     return $ok;
164 }

```

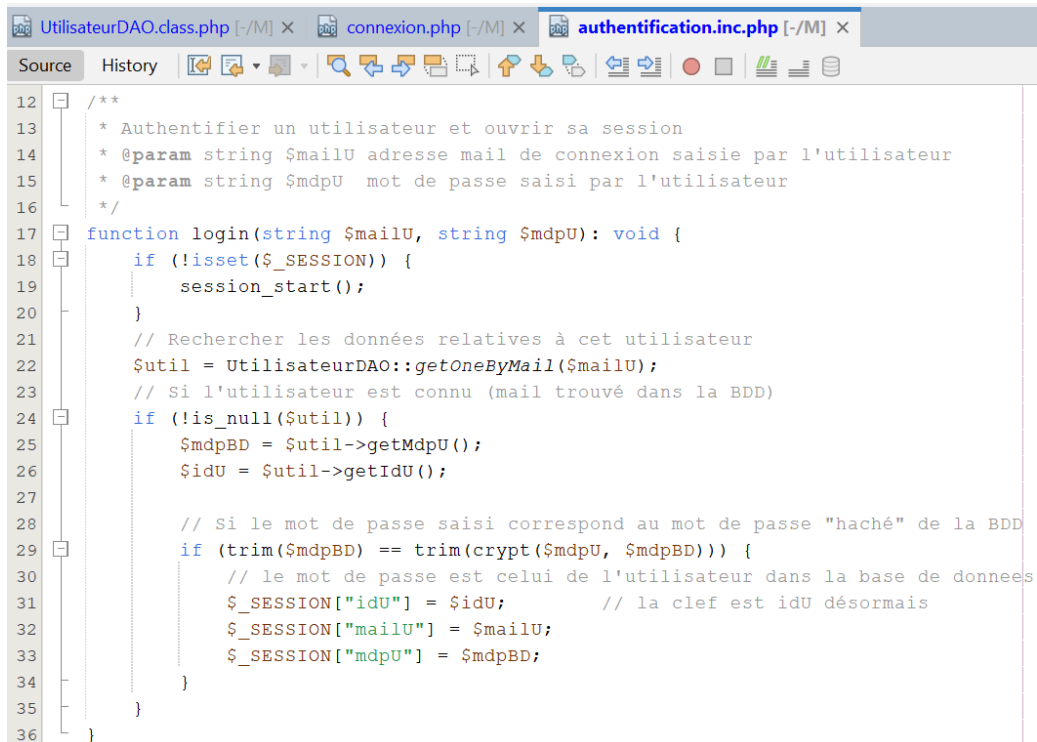
Figure 6 : Méthode updateMdp : Nouvelle version.

2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs
	Date de rédaction : 29/09/2025

## Modification du fichier authentication.inc.php

### Modification de la fonction login()

Par la suite, j'ai remarqué que la méthode login() du fichier d'authentification comportait la fonction crypt(). Hors pour que la connexion de l'utilisateur fonctionne correctement, il m'a fallut changer cela.



```

12  /**
13   * Authentifier un utilisateur et ouvrir sa session
14   * @param string $mailU adresse mail de connexion saisie par l'utilisateur
15   * @param string $mdpU mot de passe saisi par l'utilisateur
16   */
17  function login(string $mailU, string $mdpU): void {
18      if (!isset($_SESSION)) {
19          session_start();
20      }
21      // Rechercher les données relatives à cet utilisateur
22      $util = UtilisateurDAO::getOneByMail($mailU);
23      // Si l'utilisateur est connu (mail trouvé dans la BDD)
24      if (!is_null($util)) {
25          $mdpBD = $util->getMdpU();
26          $idU = $util->getIdU();
27
28          // Si le mot de passe saisi correspond au mot de passe "haché" de la BDD
29          if (trim($mdpBD) == trim(crypt($mdpU, $mdpBD))) {
30              // le mot de passe est celui de l'utilisateur dans la base de données
31              $_SESSION["idU"] = $idU;          // la clef est idU désormais
32              $_SESSION["mailU"] = $mailU;
33              $_SESSION["mdpU"] = $mdpBD;
34          }
35      }
36  }

```

Figure 7 : Méthode login() : Ancienne version.

2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs
	Date de rédaction : 29/09/2025

Ainsi, à la ligne 31, j'ai utilisé la fonction `password_verify()` qui prend en paramètre le mot de passe saisi par l'utilisateur (`$mdpU`) et le mot de passe de la BDD (`$mdpBD`).

```

12  /**
13   * Authentifier un utilisateur et ouvrir sa session
14   * @param string $mailU adresse mail de connexion saisie par l'utilisateur
15   * @param string $mdpU mot de passe saisi par l'utilisateur
16   */
17  function login(string $mailU, string $mdpU): void {
18      if (!isset($_SESSION)) {
19          session_start();
20      }
21
22      // Recherche de l'utilisateur avec son mail
23      $util = UtilisateurDAO::getOneByMail($mailU);
24
25      // S'il existe
26      if (!is_null($util)) {
27          $mdpBD = $util->getMdpU();
28          $idU   = $util->getIdU();
29
30          // Vérification avec password_verify()
31          if (password_verify($mdpU, $mdpBD)) {
32              // Connexion réussie
33              $_SESSION["idU"]   = $idU;
34              $_SESSION["mailU"] = $mailU;
35              $_SESSION["mdpU"]  = $mdpBD;
36          }
37      }
38  }

```

Figure 8 : Méthode `login()` : Nouvelle version.



2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs
	Date de rédaction : 29/09/2025

## Modification de la classe Client.class.php :

### Modification de la méthode magique \_\_toString()

Ici pour éviter de futures erreurs, j'ai ajouté « \$this » dans la méthode get\_class() comme c'est une méthode dite deprecated d'après les erreurs qu'elle provoquait durant l'exécution des fichiers de test.

```

34 // Le $this permet d'enlever l'erreur de la méthode get_class()
35 public function __toString() {
36     return get_class($this).\"{id=\".$this->idU.\", mail=\".$this->mailU.\", mdp=\".$this->mdpU.\", pseudo=\".$this->pseudoU.\", ... }\";
37 }

```

Figure 9 : Utilisateur.class.php : Modification de la méthode magique \_\_toString().

## Modification du fichier connexion.php

### Reformatage du code

J'ai effectué un simple reformatage du code pour le rendre plus lisible et j'ai inversé la condition IF/ELSE de façon à obtenir le résultat si l'utilisateur est connecté. Sinon une erreur apparaît sur son écran lors sa tentative de connexion.

```

25 if (!isset($_POST['mailU']) || !isset($_POST['mdpU'])) {
26     // Affichage du formulaire
27     $titre = "authentification";
28     require_once "$racine/vue/entete.html.php";
29     require_once "$racine/vue/vueAuthentification.php";
30     require_once "$racine/vue/pied.html.php";
31 } else {
32     $mailU = $_POST['mailU'];
33     $mdpU = $_POST['mdpU'];
34
35     // Tentative de connexion
36     login($mailU, $mdpU);
37
38     if (isLoggedIn()) {
39         header('Location: ./?action=profil');
40         exit;
41     } else {
42         ajouterMessage("Connexion : erreur de login ou de mot de passe");
43         $titre = "authentification";
44         require_once "$racine/vue/entete.html.php";
45         require_once "$racine/vue/vueAuthentification.php";
46         require_once "$racine/vue/pied.html.php";
47     }
}

```

Figure 10 : connexion.php : Reformatage du code et inversion de la condition IF/ELSE.

2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs
	Date de rédaction : 29/09/2025

## Tests d'inscription

Après avoir coder toutes ces fonctionnalités, il est temps de les tester. Je commence par inscrire un nouvel utilisateur dont les données sont les suivantes :

- Mail : [test11@gmail.com](mailto:test11@gmail.com)
- Mot de passe : test
- Pseudo : test

Figure 11 : Test d'inscription: Remplissage des champs.

Une fois que l'utilisateur a cliqué sur le bouton « s'inscrire », le message suivant est apparu.

## Inscription

Inscription effectuée.  
Veuillez vous [connecter](#).

Figure 12 : Test d'inscription : Message de validation de l'inscription.

2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs
	Date de rédaction : 29/09/2025

Ainsi, en vérifiant la table UTILISATEUR de notre base de données, on peut voir que le 29 compté utilisateur créé est le nôtre. En plus, son mot de passe est même haché.

The screenshot shows the PhpMyAdmin interface for the 'utilisateur' table. The table has 14 rows, with the last row (id 29) being the newly added user. The table structure is as follows:

	idU	mailU	mdpU	pseudoU
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	alex.garat@gmail.com	\$1\$zvN5hYSQSFUIQSdufUQSDfZnHF5osT.	@lex
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	jj.soueix@gmail.com	\$1\$zvN5hYMI\$SDFGSDFGJqJSDJF.	drskott
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	3	mathieu.capliez@gmail.com	seSzpoUAQgll.	pich
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	4	michel.garay@gmail.com	\$1\$zvN5hYMI\$VSatLQ6SDFGdsfgznHF5osT.	Mitch
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	5	nicolas.harispe@gmail.com	\$1\$zvNDSFQSDfqsDfQsdfsT.	Nico40
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	6	test@bts.sio	seSzpoUAQgll.	testeur SIO
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	7	yann@lechambon.fr	sej6dETYI/ea.	yann
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	9	test2@gmail.com	NULL	test
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	10	le-mien@contact.fr	\$2y\$10\$BfnKaN1eZC4gOpXltHiZeVknP9YnGTHgk48QyRgh36...	pseudomodifie44
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	11	test5@gmail.com	NULL	test
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	12	test6@gmail.commm	\$2y\$10\$PVFo1UTeEX7QzW/g/KGqj.QTl5mB2ysq.4od137xB6E...	test
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	13	test7@gmail.com	\$2y\$10\$yoXGaWm6M.65jyRX9fwSTeqyMJpV/40bepk1PqxUGH...	test
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	14	test10@gmail.com	\$2y\$10\$CvqLCyY2R3.pUBSZpPAS5OAKBu9B3JHcjV4Tl1hwlj...	test
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	29	test11@gmail.com	\$2y\$10\$aPjL7q7R3hTnPKXKw/4ioOphvDEFVjSxgC9oW1QqXis...	test


Figure 13 : Test d'inscription - PhpMyAdmin : Visualisation de la nouvelle ligne ajoutée à la table UTILISATEUR.

2025/2026	AP – Itération 2
BTS SIO 2SLAM	Auteur : Taillé Jade & Delaunay-Guitton Benjamin & Portolleau Anaïs
	Date de rédaction : 29/09/2025

## Test de connexion

Maintenant que notre utilisateur « test11@gmail.com » s’est inscrit, il faut se connecter. Je rentre donc son mail et mot de passe (« test »), puis je clique sur le bouton « Envoyer ».

http://localhost/2SLAM/AP/PhpProjectSiteRestoInitial/?action=connexion

Accueil Recherche  CGU Connexion

### Connexion


test11@gmail.com

....

Envoyer

Figure 14 : Test de connexion : Remplissage des champs du formulaire.

http://localhost/2SLAM/AP/PhpProjectSiteRestoInitial/?action=profil

Accueil Recherche  CGU Mon Profil

### Mon profil

Mon adresse électronique : test11@gmail.com  
Mon pseudo : test

---

les restaurants que j'aime :

---

[se deconnecter](#)

Figure 15 : Test de connexion - Réussie : l'utilisateur a désormais accès à son profil.