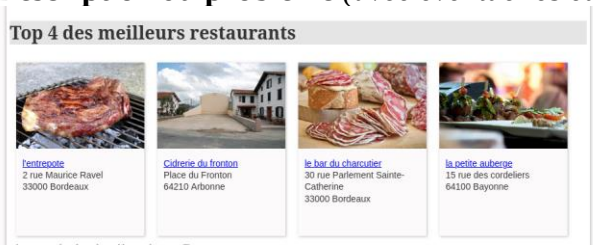
Ticket n°10
Titre du ticket : faille de sécurité sur la page d'authentification

Type du ticket : incident *(évolution/incident)*	**Niveau de gravité** : ☐Bloquant ☒Majeur ☐ Mineur
Émetteur : Nicolas BOURGEOIS *(nom de l'émetteur)*	**Date signalement** : 21/09/2023 *(jj/mm/aaaa)*
Assignation : Martin PLISSONNEAU *(nom du membre de l'équipe en charge du ticket)*	**Date de résolution souhaitée** : 02/10/2023 *(jj/mm/aaaa)*
Application concernée : *R3st0.fr*	**Version** : 1.0 initiale – septembre 2023

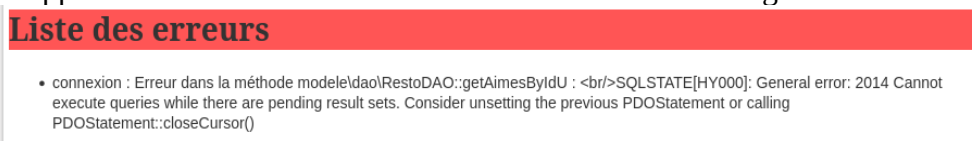
Description du problème *(avec éventuelles captures d'écran, messages d'erreurs)* :



On m'a signalé qu'une attaque par injection SQL est possible sur la page de connexion.
Scénario :
L'utilisateur saisit la chaîne de caractères suivante dans le champ de saisie de l'email :

zzz' OR 1 = 1 ; DELETE FROM photo WHERE '1' = '1

et une valeur quelconque dans le mot de passe.
L'application refuse l'authentification en affichant le message d'erreur suivant :



Mais, ensuite, on peut constater que l'attaque a réussi, car **les photos des restaurants ne sont plus affichées** sur la page d'accueil (ni ailleurs) : les données de la table photo ont été supprimées !

AVANT : ***APRÈS :***

On souhaite donc rendre impossibles les attaques par injection SQL sur ce formulaire.

Solution *(diagnostic, localisation, modification, test) :*

La page d'accueil après avoir l'injection SQL :

Le problème se trouve dans le fichier « UtilisateurDAO.class », il faut remplacer la ligne

```
$requete = "SELECT * FROM utilisateur WHERE mailU = '".$mailU."'";
```

Par la ligne :

```
$requete = "SELECT * FROM utilisateur WHERE mailU = :mailu";
```

Cette ligne de code est une requête préparée qui empêche les injections SQL.

Test :

Après avoir saisit une injection SQL dans le champ du mail, on observe qu'aucune photos n'a été supprimé de la base de données.