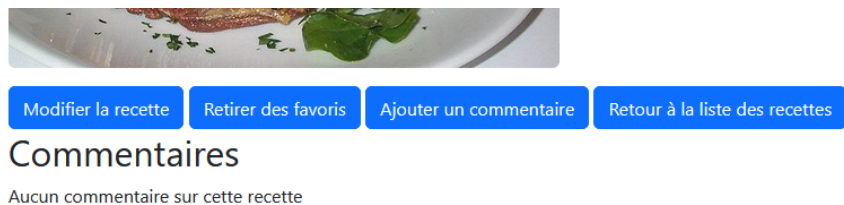


TP 7 : Proposer des commentaires sur une recette

Ce TP nous permet d'avoir la possibilité d'avoir des commentaires sur les recettes, le commentaire peut être anonyme ou lié à l'utilisateur actuellement connecté.

- Créer une table **comments** avec les colonnes :
 - id (INTEGER, AUTO_INCREMENT),
 - pseudo (VARCHAR 100),
 - commentaire (TEXT),
 - create_time (DATETIME)
- Créer le modèle **Commentaire** dans **Models** sur le modèle des précédents modèles
- Créer le contrôleur **CommentaireController.php**
- Dans la vue **detail.php**, prévoir :
 - De lister les commentaires en dessous de la recette grâce à une fonction contenu dans le contrôleur créé ci-dessus, s'il n'y a aucun commentaire, écrire le texte suivant : **Aucun commentaire sur cette recette**



Vue des commentaires

```
// fonction permettant de lister les commentaires d'une recette
function lister($id) {
    $commentaires = $this->commentaireModel->findBy(['recette_id' => $id]);
    return $commentaires;
}
```

Extrait de code

- D'avoir un bouton **Ajouter un commentaire** qui affiche un formulaire d'ajout de commentaire qui apparaîtra au-dessus des commentaires ou du texte par défaut. Le formulaire contient :
 - Une zone de texte (Commentaire)
 - Un bouton de validation (Valider le commentaire)
 - La méthode est **post**
 - L'action mène vers une action d'ajout de commentaire à créer dans le contrôleur **CommentaireController.php**
 - Le formulaire doit être ajouté par Javascript, au clic sur le bouton **Ajouter un commentaire**.

```

// Crée un élément <form>
let formComment = document.createElement('form');
formComment.method = 'post';
formComment.action = '?c=Commentaire&a=ajouter&id=' +
btAjoutCommentaire.dataset.id; // Action du formulaire

// Créer un textarea
let textarea = document.createElement('textarea');
textarea.name = 'commentaire';
textarea.placeholder = 'Saisir le commentaire';
textarea.rows = '4';
textarea.classList.add('form-control');
textarea.required = true; // Ajoute un attribut required pour vérifier la saisie

// Crée un bouton submit
let submitButton = document.createElement('button');
submitButton.type = 'submit'; // Type de bouton
submitButton.textContent = 'Valider le commentaire'; // Texte du bouton
submitButton.classList.add('btn', 'btn-primary'); // Ajoute une classe au bouton

// Ajoute un div de class mb-3
let divMessage = document.createElement('div');
divMessage.classList.add('mb-3');

divMessage.appendChild(textarea);
divMessage.appendChild(submitButton);

// Ajoute les éléments dans le formulaire
formComment.appendChild(divMessage);

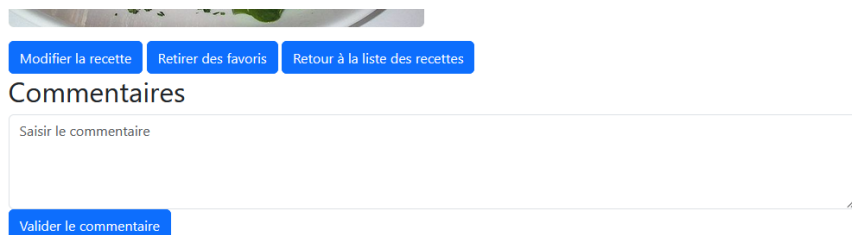
divCommentaire.prepend(formComment); // Ajoute le formulaire au div commentairev

btAjoutCommentaire.classList.add('d-none'); // Affiche le div commentaire

```

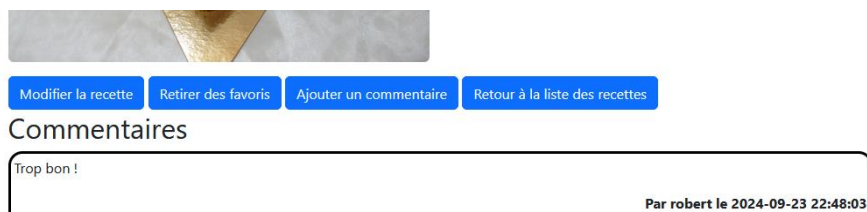
Extraits de code

- Une fois le formulaire affiché, le bouton **Ajouter un commentaire doit disparaître**



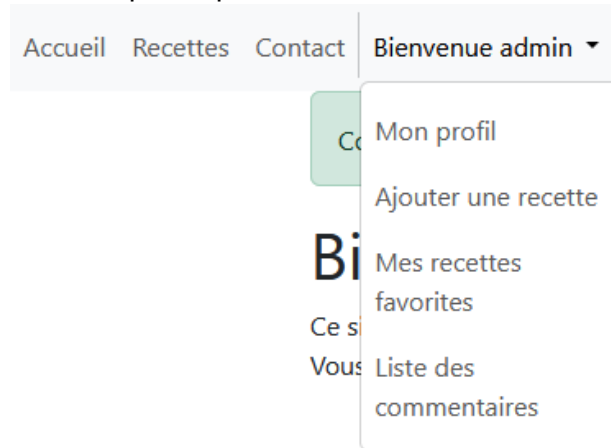
Le formulaire d'ajout de commentaire apparaît

- Une fois le formulaire validé, il faut revenir sur le détail de la recette avec le commentaire affiché :



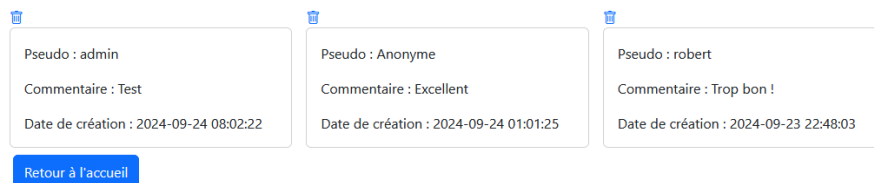
Vue globale

- Tester les commentaires, avec et sans connexion afin de vérifier que le commentaire Anonyme apparaisse
- Prévoir le fait que l'administrateur (**isAdmin** dans la base des utilisateurs) puisse supprimer un commentaire (modération)
 - Ajouter une variable de session **isAdmin** selon la valeur de la table **users** (champ isAdmin à 1 quand la personne est administrateur)
 - Créer un menu spécifique à l'administrateur : **Liste des commentaires**



- Lister les commentaires (par ordre décroissant de date – du plus récent au plus ancien) et créer un bouton (avec Bootstrap icon « trash ») **Supprimer le commentaire**

Liste des commentaires



Vue des commentaires

- Afficher un message de succès quand le commentaire est supprimé
- De la même manière, prévoir la suppression d'une recette (et donc des favoris et des commentaires associés) dans le fichier liste.php du dossier Views\Recette (sous la forme d'une poubelle Bootstrap)
- Enfin, pour finir, prévoir dans le fichier index, un message d'erreur pour chaque page non trouvée en remplacement du echo « page non trouvée » :

```
$_SESSION['message'] = ['danger' => 'Page non trouvée'];
header('Location: ?c=home');
break;
```