**Dossier d’études**

Liste des utilisateurs avec leur rôle :

gerant = admin et client = utilisateur

**Bundle :**

Notre projet a été développé en utilisant les technologies suivantes :

- Symfony : Nous avons utilisé Symfony comme framework PHP pour faire le projet

- Twig : Twig a été utilisé comme moteur de template pour générer des vues claires et structurées. Son intégration native avec Symfony a simplifié le processus de rendu de nos pages.

- Bootstrap : Nous avons intégré Bootstrap, un framework CSS, pour assurer une mise en page responsive et un design moderne. Cela a facilité le développement d'une interface utilisateur esthétique et conviviale.

-PhpMyAdmin : PhpMyAdmin a été employé comme interface d'administration pour la gestion de notre base de données MySQL. Cela nous a permis de manipuler facilement les données dans les différentes tables.

**Cas d’utilisation :**

**Controller :**

Explication du MainController

`showAllCategoryNames`

La méthode `showAllCategoryNames` du `MainController` est dédiée à l'affichage de toutes les catégories disponibles. Elle utilise le `CategorieRepository` du MainBundle pour récupérer les noms et les identifiants des catégories, puis elle rend une vue Twig afin d'afficher ces informations. Cette action est cruciale pour la présentation des différentes catégories disponibles dans l'application.

`articlesParCategorie`

La méthode `articlesParCategorie` gère l'affichage des articles associés à une catégorie spécifique. En utilisant l'`ArticleRepository` du MainBundle, elle récupère les articles liés à la catégorie spécifiée. La vue Twig associée est ensuite rendue, présentant ainsi une liste d'articles pertinente à la catégorie sélectionnée. Cette action joue un rôle central dans la visualisation des articles en fonction des catégories.

`login` et `verification`

Les actions `login` et `verification` sont responsables, respectivement, de l'affichage de la page de connexion et de la validation des identifiants lors d'une tentative de connexion. L'action `login` rend la vue Twig associée à la page de connexion, tandis que l'action `verification` récupère les informations du formulaire (email et mot de passe), vérifie leur validité en interagissant avec la base de données via l'`EntityManager`, et redirige l'utilisateur vers la page principale en cas de succès. Ces actions jouent un rôle crucial dans le processus d'authentification, assurant la sécurité et l'accès approprié aux fonctionnalités de l'application.

**Les routings :**

**LA BDD :**

**Une image contenant texte, Police, nombre, capture d’écran

Description générée automatiquement**

Pour la bdd nous utilisons donc PhpMyAdmin , le nom de la bdd est ProjetBts, à l’intérieur de celle-ci nous avons 7 tables dont 6 principales avec article,categorie, commande, panier, rôle et utilisateur et une autre table doctrine\_migartions\_version qui est utilisée pour stocker des informations sur les migrations de schéma qui ont été appliquées à la base de données.

Dans la table commande

**Explication MCD :**

1. Table Categorie:
   * Stocke des informations sur les catégories de produits.
   * Champs :
     + ID\_categorie: Identifiant unique de la catégorie.
     + Nom\_Categorie: Nom de la catégorie.
   * La clé primaire est définie sur ID\_categorie.
2. Table Article:
   * Stocke des informations sur les articles (produits).
   * Champs :
     + ID\_article: Identifiant unique de l'article.
     + Nom: Nom de l'article.
     + Description: Description de l'article.
     + Image: Chemin de l'image de l'article.
     + Prix: Prix de l'article.
     + ID\_categorie: Clé étrangère faisant référence à Categorie(ID\_categorie).
   * La clé primaire est définie sur ID\_article.
3. Table Role:
   * Stocke des rôles ou des types d'utilisateurs.
   * Champs :
     + ID\_role: Identifiant unique du rôle.
     + Type: Type de rôle (par exemple, administrateur, utilisateur).
   * La clé primaire est définie sur ID\_role.
4. Table Utilisateur:
   * Stocke des informations sur les utilisateurs.
   * Champs :
     + ID\_utilisateur: Identifiant unique de l'utilisateur.
     + Nom\_: Nom de l'utilisateur.
     + Prenom: Prénom de l'utilisateur.
     + Adresse\_Mail: Adresse e-mail de l'utilisateur.
     + Mot\_de\_passe\_: Mot de passe de l'utilisateur.
     + Adresse\_Postale: Adresse postale de l'utilisateur.
     + Numero\_de\_Telephone: Numéro de téléphone de l'utilisateur.
     + ID\_role: Clé étrangère faisant référence à Role(ID\_role).
   * La clé primaire est définie sur ID\_utilisateur.
5. Table Commande:
   * Stocke des informations sur les commandes passées par les utilisateurs.
   * Champs :
     + ID\_commande: Identifiant unique de la commande.
     + Date\_Commande: Date de la commande.
     + Statut: Statut de la commande.
     + ID\_utilisateur: Clé étrangère faisant référence à Utilisateur(ID\_utilisateur).
     + list\_nom: Liste des noms des articles commandés (type simple\_array).
     + list\_prix: Liste des prix des articles commandés (type simple\_array).
     + list\_quantite: Liste des quantités des articles commandés (type simple\_array).
   * La clé primaire est définie sur ID\_commande.
6. Table Panier:
   * Stocke des informations sur les articles ajoutés au panier par les utilisateurs.
   * Champs :
     + id: Identifiant unique du panier.
     + id\_article: Clé étrangère faisant référence à Article(ID\_article).
     + nom: Nom de l'article dans le panier.
     + prix: Prix de l'article dans le panier.
     + quantite: Quantité de l'article dans le panier.
   * La clé primaire est définie sur id et id\_article.

**UML :**

Classe Categorie

-----------------

Attributs :

- ID\_categorie: INT

- Nom\_Categorie: VARCHAR(50)

Classe Article

-----------------

Attributs :

- ID\_article: INT

- Nom: VARCHAR(50)

- Description: VARCHAR(50)

- Image: VARCHAR(255)

- Prix: VARCHAR(50)

Associations :

- ID\_categorie: INT (Association vers la classe Categorie)

Classe Role

-----------------

Attributs :

- ID\_role: INT

- Type: VARCHAR(50)

Classe Utilisateur

-----------------

Attributs :

- ID\_utilisateur: INT

- Nom\_: VARCHAR(50)

- Prenom: VARCHAR(50)

- Adresse\_Mail: VARCHAR(50)

- Mot\_de\_passe\_: VARCHAR(50)

- Adresse\_Postale: VARCHAR(50)

- Numero\_de\_Telephone: VARCHAR(50)

Associations :

- ID\_role: INT (Association vers la classe Role)

Classe Commande

-----------------

Attributs :

- ID\_commande: INT

- Date\_Commande: DATETIME

- Statut: VARCHAR(50)

Associations :

- ID\_utilisateur: INT (Association vers la classe Utilisateur)

Listes (type simple\_array) :

- list\_nom: Liste des noms des articles commandés.

- list\_prix: Liste des prix des articles commandés.

- list\_quantite: Liste des quantités des articles commandés.

Classe Panier

-----------------

Attributs :

- id: INT

- id\_article: INT

- nom: VARCHAR(50)

- prix: VARCHAR(50)

- quantite: INT

Associations :

- id\_article: INT (Association vers la classe Article)