



Livequestion

Documentations :

Fonctionnelle & Technique

Projet : *Livequestion*, réseau social basé sur l'avis de ses utilisateurs.

Date de création du document : 5/10/2020

Date de modification du document : 06/12/2020

Table des matières

Dossier fonctionnel.....	3
Les données	3
Dictionnaire de données	3
Modèle Entité Association (MEA).....	4
Schéma relationnel	5
Script SQL.....	6
Les fonctionnalités	7
Le Backlog	7
L'arborescence.....	8
Les cas d'utilisations.....	9
Les maquettes.....	10
Dossier technique	12
Focus sur deux fonctionnalités.....	12
Captures d'écran et code commenté	12
Conduite de projet	14
Répartition des tâches	14
Compte rendu de réunion	15

Dossier fonctionnel

Les données

Dictionnaire de données

Propriété	Type	Description
id_reponse	INT	Id de la réponse
contenu_reponse	VARCHAR (50)	Contenu de la réponse
date_reponse	DATE	Date de la réponse
is_pending	BOOL	
id_profil	INT	Id du profil
pseudo_profil	VARCHAR (50)	Pseudo du profil
mail_profil	VARCHAR (50)	Mail du profil
motDePasse_profil	VARCHAR (50)	Mot de passe du profil
genre_profil	VARCHAR (50)	Genre du profil
avatar	VARCHAR (50)	Avatar du profil
date	DATE	Date de chaque « J'aime »
id_role	INT	Id du rôle
libelle_role	VARCHAR (50)	Libellé du rôle
id_categorie	INT	Id de la catégorie
libelle_categorie	VARCHAR (50)	Libellé de la catégorie
id_question	ID	Id de la question
titre_question	VARCHAR (50)	Titre de la question
date_creation_question	DATE	Date de création de la question
unique_key	VARCHAR (50)	Clé unique
visible_question	VARCHAR (50)	Question visible

Modèle Entité Association

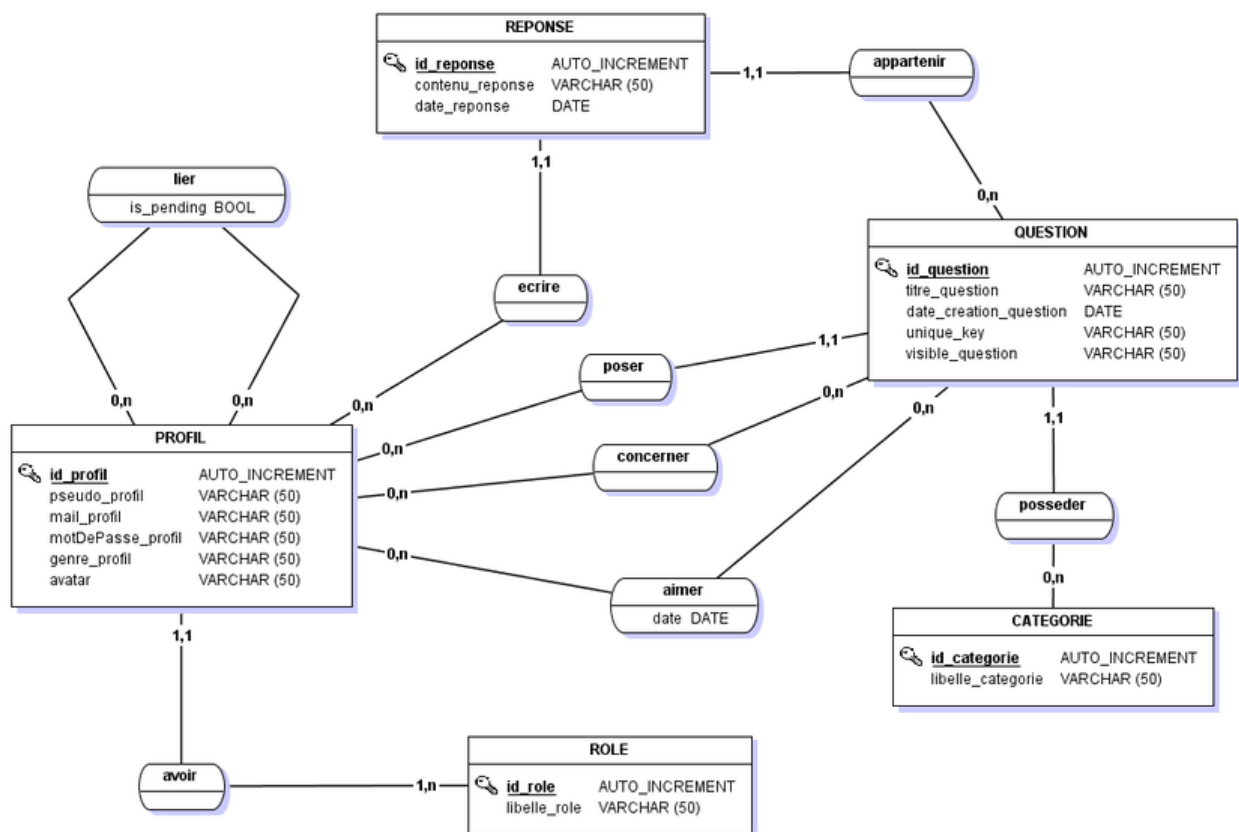


Schéma relationnel

AMIS (id_profil, id_profil_lien, is_pending)

Clé primaire : (id_profil, id_profil_lien).

Clé étrangère : id_profil en référence à id_profil de la table PROFIL.

CATEGORIE (id_categorie, libelle_categorie)

Clé primaire : id_categorie.

PROFIL (id_profil, pseudo_profil, mail_profil, motDePasse_Profil, genre_profil, avatar, id_role)

Clé primaire : id_profil.

Clé étrangère : id_role en référence à id_role de la table ROLE.

ROLE (id_role, libelle_role)

Clé primaire : id_role.

QUESTION (id_question, titre_question, date_creation_question, visible_key, id_categorie, id_profil)

Clé primaire : id_question.

Clé étrangère : id_categorie en référence à id_categorie de la table CATEGORIE.

Clé étrangère : id_profil en référence à id_profil de la table PROFIL.

REPONSE (id_reponse, contenu_reponse, date_reponse, id_question, id_profil)

Clé primaire : id_reponse.

Clé étrangère : id_question en référence à id_question de la table QUESTION.

Clé étrangère : id_profil en référence à id_profil de la table PROFIL.

VOTE (id_vote, action_vote, id_question, id_profil)

Clé primaire : id_vote.

Clé étrangère : id_question en référence à id_question de la table QUESTION.

Clé étrangère : id_profil en référence à id_profil de la table PROFIL.

CONCERNER (id_profil, id_question)

Clé primaire : (id_profil, id_question).

Clé étrangère : id_profil en référence à la table PROFIL.

Clé étrangère : id_question en référence à la table QUESTION.

Scripts SQL

```
CREATE TABLE CATEGORIE (id_categorie Int Auto_increment NOT NULL, libelle_categorie Varchar (50) NOT NULL, CONSTRAINT CATEGORIE_PK PRIMARY KEY (id_categorie)) ENGINE=InnoDB;
```

```
CREATE TABLE ROLE (id_role Int Auto_increment NOT NULL, libelle_role Varchar (50) NOT NULL, CONSTRAINT ROLE_PK PRIMARY KEY (id_role)) ENGINE=InnoDB;
```

```
CREATE TABLE PROFIL (id_profil Int Auto_increment NOT NULL, pseudo_profil Varchar (50) NOT NULL, mail_profil Varchar (50) NOT NULL, motDePasse_profil Varchar (50) NOT NULL, genre_profil Varchar (50) NOT NULL, avatar Varchar (50) NOT NULL, id_role Int NOT NULL, CONSTRAINT PROFIL_PK PRIMARY KEY (id_profil), CONSTRAINT PROFIL_ROLE_FK FOREIGN KEY (id_role) REFERENCES ROLE (id_role)) ENGINE=InnoDB;
```

```
CREATE TABLE QUESTION( id_question Int Auto_increment NOT NULL , titre_question Varchar (50) NOT NULL , date_creation_question Date NOT NULL , unique_key Varchar (50) NOT NULL , visible_question Varchar (50) NOT NULL , id_categorie Int NOT NULL , id_profil Int NOT NULL ,CONSTRAINT QUESTION_PK PRIMARY KEY (id_question) ,CONSTRAINT QUESTION_CATEGORIE_FK FOREIGN KEY (id_categorie) REFERENCES CATEGORIE(id_categorie) ,CONSTRAINT QUESTION_PROFILO_FK FOREIGN KEY (id_profil) REFERENCES PROFIL(id_profil) )ENGINE=InnoDB;
```

```
CREATE TABLE VOTE (id_vote Int Auto_increment NOT NULL, action_vote Int NOT NULL, id_question Int NOT NULL, id_profil Int NOT NULL, CONSTRAINT VOTE_PK PRIMARY KEY (id_vote), CONSTRAINT VOTE_QUESTION_FK FOREIGN KEY (id_question) REFERENCES QUESTION (id_question), CONSTRAINT VOTE_PROFILO_FK FOREIGN KEY (id_profil) REFERENCES PROFIL (id_profil)) ENGINE=InnoDB;
```

```
CREATE TABLE REPONSE (id_reponse Int Auto_increment NOT NULL, contenu_reponse Varchar (50) NOT NULL, date_reponse Date NOT NULL, id_profil Int NOT NULL, id_question Int NOT NULL, CONSTRAINT REPONSE_PK PRIMARY KEY (id_reponse), CONSTRAINT REPONSE_PROFIL_FK FOREIGN KEY (id_profil) REFERENCES PROFIL (id_profil), CONSTRAINT REPONSE_QUESTION_FK FOREIGN KEY (id_question) REFERENCES QUESTION (id_question)) ENGINE=InnoDB;
```

```
CREATE TABLE concerner (id_profil Int NOT NULL, id_question Int NOT NULL, CONSTRAINT concerner_PK PRIMARY KEY (id_profil,id_question) ,CONSTRAINT concerner_PROFIL_FK FOREIGN KEY (id_profil) REFERENCES PROFIL(id_profil) ,CONSTRAINT concerner_QUESTION_FK FOREIGN KEY (id_question) REFERENCES QUESTION(id_question) )ENGINE=InnoDB;
```

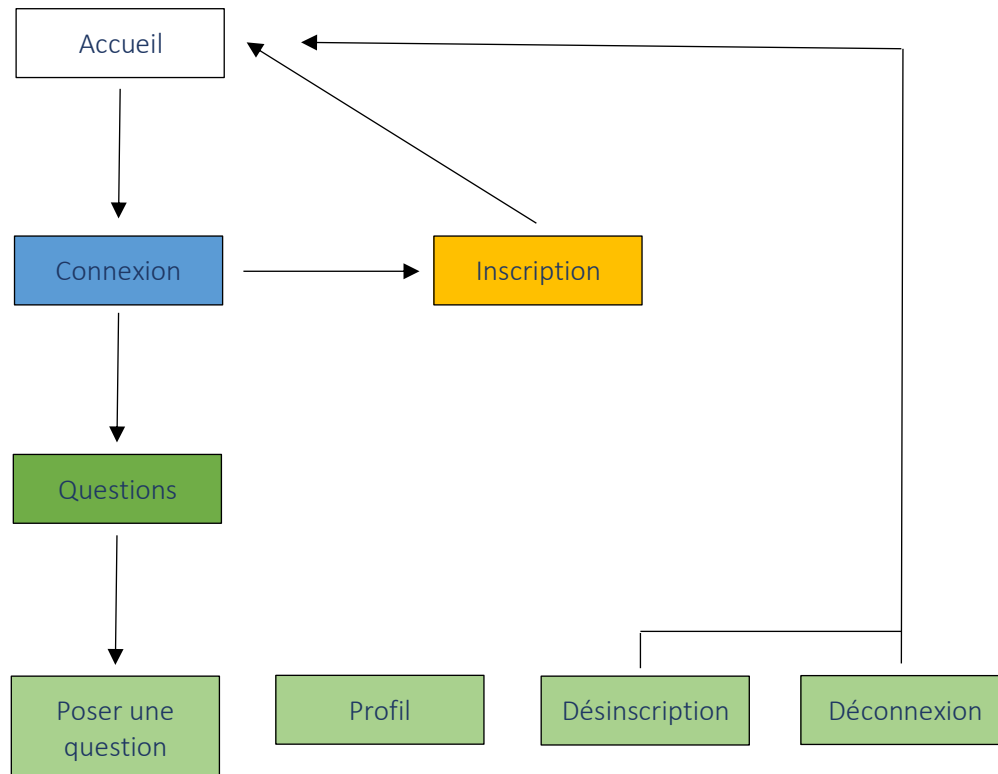
```
CREATE TABLE lier (id_profil Int NOT NULL, id_profil_liier Int NOT NULL, is_pending Bool NOT NULL, CONSTRAINT lier_PK PRIMARY KEY (id_profil,id_profil_liier) ,CONSTRAINT lier_PROFIL_FK FOREIGN KEY (id_profil) REFERENCES PROFIL(id_profil) ,CONSTRAINT lier_PROFILO_FK FOREIGN KEY (id_profil_liier) REFERENCES PROFIL(id_profil) )ENGINE=InnoDB;
```

Les fonctionnalités

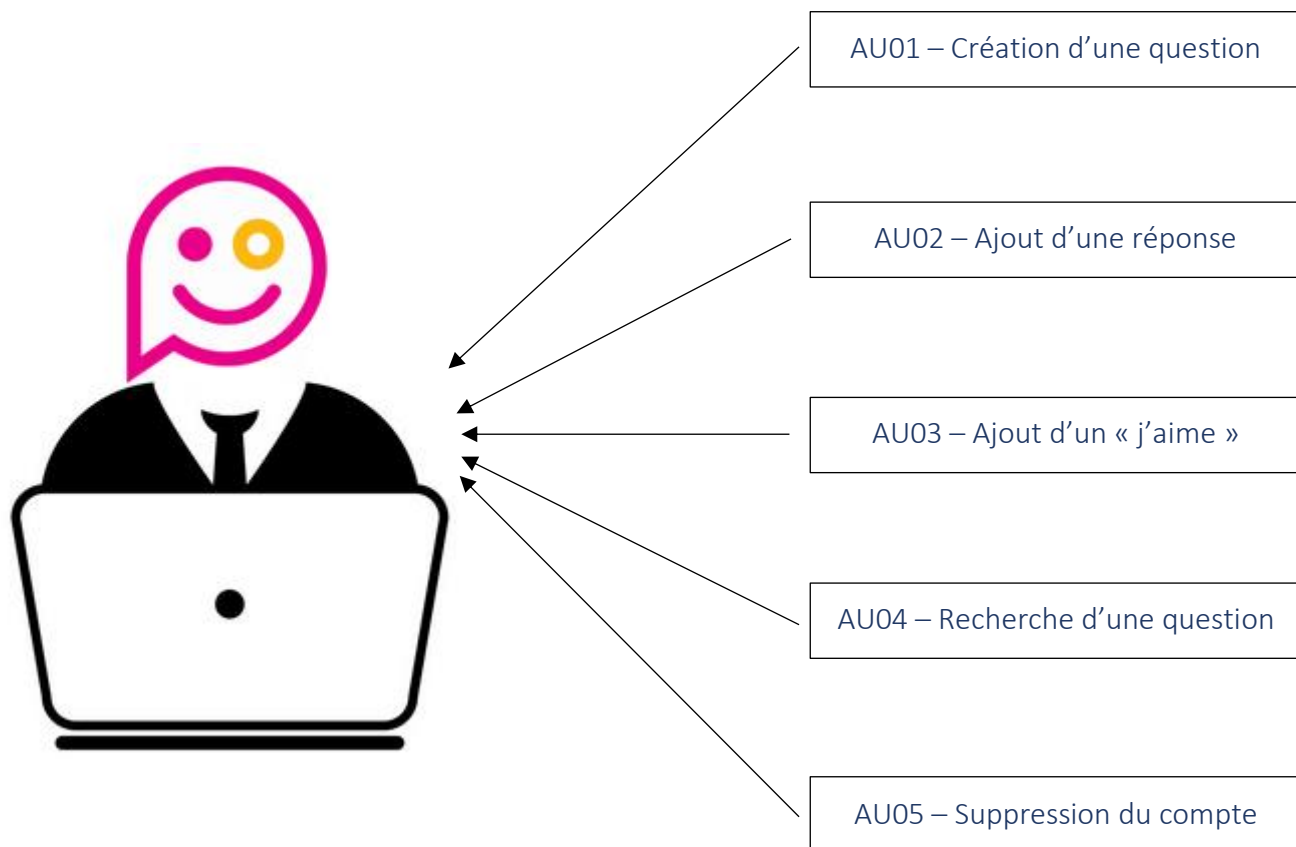
Le Backlog

Fonctionnalité	Descriptif
AS – Gestion des accès du site	Inscription d'un utilisateur
	Authentification d'un utilisateur inscrit
	Déconnexion d'un utilisateur connecté
AU – Gestion des actions utilisateur	Création d'une question publique ou privée
	Ajout d'une réponse à une question
	Ajout d'une mention « j'aime »
	Recherche d'une question
	Suppression du compte
EA – Espace administrateur	Suppression d'une question
	Suppression d'une catégorie
	Ajout d'une catégorie
EPU – Espace profil de l'utilisateur	Modification du profil
	Ajout d'un avatar
A – Gestion des amis	Envoyer une demande
	Supprimer une demande
	Accepter une demande
	Supprimer un ami

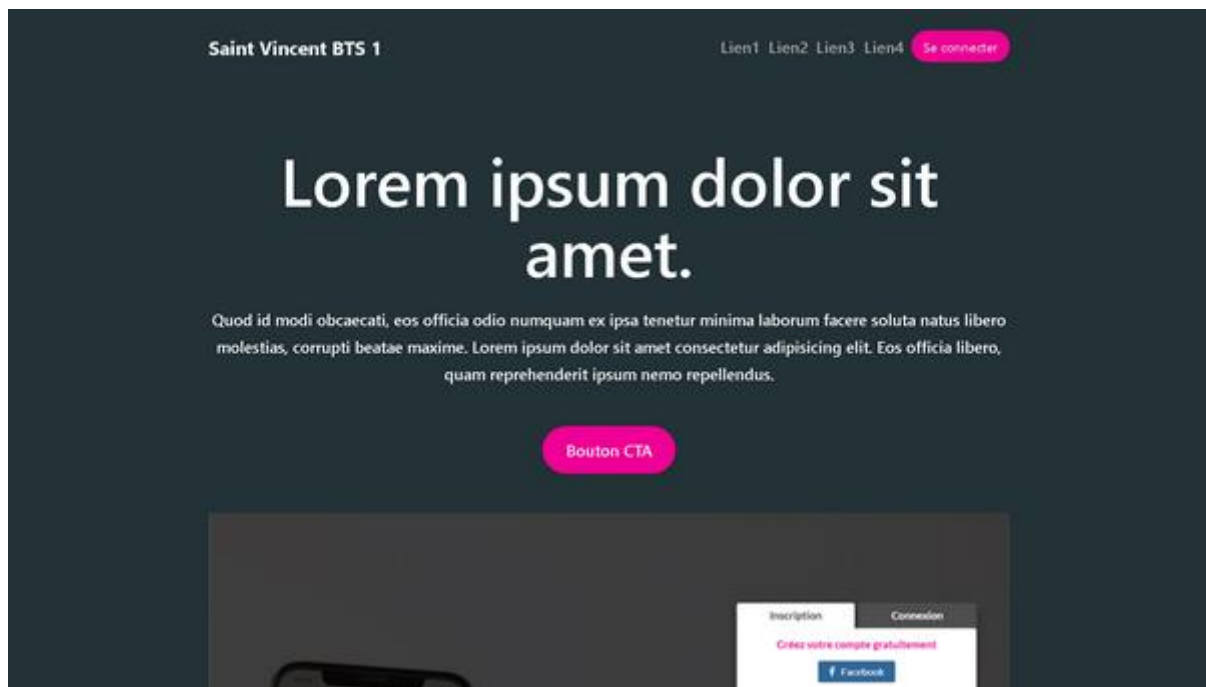
L'arborescence



Les cas d'utilisations



Les maquettes



Page de connexion

nom d'utilisateur
mot de passe
se connecter
Vous n'avez pas de compte ? Inscrivez vous ici

Inscription

nom d'utilisateur
mot de passe
genre
email
Finaliser l'inscription
lien vers l'accueil

Bonjour, root!

Demandes d'amis:

vous n'avez aucune requête d'ami

envoyer une demande d'ami:

Pseudo de l'utilisateur



pseudo : root

e-mail : djilfabien@yahoo.com

genre : femme

Questions posées par l'utilisateur :

 root | 0 avis | cuisine | 2020-12-06

Quel est votre plat préféré ?

Poser une question à la communauté

Quelle est votre question ?*

Catégorie de la question*

Poser cette question uniquement à vos amis ☐

Bonjour, root!

Êtes-vous sûr de vouloir vous désinscrire ? Ceci entraînera la suppression de toutes vos questions/réponses.

Dossier technique

Focus sur deux fonctionnalités

Formulaire d'ajout d'une question

```
<form method="POST">
  <div class="options">
    <p>Quelle est votre question ?<span class="required">*</span></p>
    <input type="text" class="form-text" name="libelle">
    <p>Catégorie de la question<span class="required">*</span></p>
    <select class="form-text" name="categories">
      <option value = "vide"></option>
      <!-- affichage des catégories -->
    </select>
    <input type="checkbox" name="amis"> Poser cette question uniquement à vos amis
    <input type="submit" class="button" name="valider"> Valider
  </div>
</form>
<?php
  $profil = $connexion->query('SELECT Pseudo_profil, Id_profil FROM profil WHERE Pseudo_profil = "'.$_SESSION['pseudo'].'"')->fetchALL();

  //vérification des champs et envoi des données

  if(isset($_POST['valider']) && (empty($_POST['libelle']) || $_POST['categories'] == "vide")){
    echo'Veuillez remplir tous les champs';
  }
  elseif(isset($_POST['valider']) && !empty($_POST['libelle'])){
    if(isset($_POST['amis']) && !empty($_POST['amis'])){
      $query = $connexion->prepare('INSERT INTO question (Titre_question, Date_creation_question, Id_profil, Id_categorie, Amis_seulement) VALUES (:Titre_question, :Date_creation_question, :Id_profil, :Id_categorie, :Amis_seulement)');
      $query->bindParam(':Amis_seulement', $Amis_seulement);
      $Amis_seulement = 1;
    }
    else{
      $query = $connexion->prepare('INSERT INTO question (Titre_question, Date_creation_question, Id_profil, Id_categorie) VALUES (:Titre_question, :Date_creation_question, :Id_profil, :Id_categorie)');
      $query->bindParam(':Titre_question', $Titre_question);
      $query->bindParam(':Id_profil', $Id_profil);
      $query->bindParam(':Id_categorie', $Id_categorie);
      $query->bindParam(':Date_creation_question', $date_creation_question);
      $date = new DateTime();
      $Titre_question = $_POST['libelle'];
      $Id_profil = $profil[0]['Id_profil'];
      $Id_categorie = $_POST['categories'];
      $date_creation_question = $date->format('Y-m-d');
      $query->execute();
      echo'<p class="text-center">Votre question a bien été envoyée</p>';
    }
  }
  <?>
```

Le formulaire utilise la méthode POST pour le traitement des données.

Récupération et affichage des catégories des questions de la base de données.

Insertion de la question posée dans la base de données.

Récupération et enregistrement de la question dans la base de données.

Ajout d'un ami

Le formulaire utilise la méthode POST pour le traitement des données.

Vérification de l'existence de l'utilisateur en question. On fait en sorte de ne pas se demander soi-même en ami.

On vérifie si l'on n'est pas déjà ami avec cet utilisateur.

Si cet utilisateur nous a déjà demandé en ami, il suffit de l'accepter.

```
<h3>envoyer une demande d'ami:</h3>
<form method="POST">
  <input type="text" placeholder="Pseudo de l'utilisateur" name="pseudo">
  <input type="submit" value="envoyer" name="envoyer">
</form>

<?php
  $pseudo = $connexion->query('SELECT Id_profil from profil where Pseudo_profil = "'.$_SESSION['pseudo'].'"')->fetchAll();

  // vérification des champs
  if(!empty($_POST['pseudo']) && isset($_POST['envoyer'])){
    $utilisateur = $connexion->query('SELECT Pseudo_profil from profil where Pseudo_profil = "'.$_POST['pseudo'].'"')->fetchAll();

    // si l'utilisateur existe
    if(count($utilisateur) == 1){
      // si l'utilisateur n'est pas nous même
      if($utilisateur[0]['Pseudo_profil'] == $_SESSION['pseudo']){
        echo'Vous ne pouvez pas vous ajouter en ami';
      }
      else{
        // si une requête a déjà été envoyée à cet utilisateur
        $xiste = $connexion->query('SELECT Id_requete from requete where Id_envoyeur = (SELECT Id_profil from profil where Pseudo_profil = "'.$_SESSION['pseudo'].'"') and Id_receveur = (SELECT Id_profil from profil where Pseudo_profil = "'.$_POST['pseudo'].'"')->fetchAll();

        if(count($xiste) == 1){
          echo'Une demande a déjà été envoyée à cet utilisateur';
        }
        else{
          // si moi et cet utilisateur sommes déjà amis
          $xiste = $connexion->query('SELECT Id_requete from requete where Id_envoyeur = (SELECT Id_profil from profil where Pseudo_profil = "'.$_POST['pseudo'].'"') and Id_receveur = (SELECT Id_profil from profil where Pseudo_profil = "'.$_SESSION['pseudo'].'"') and Status_requete = "acceptee")->fetchAll();

          if(count($xiste) == 1){
            echo'Vous êtes déjà ami avec cet utilisateur';
          }
          else{
            // si cet utilisateur vous a déjà envoyé une requête
            $xiste = $connexion->query('SELECT Id_requete from requete where Id_envoyeur = (SELECT Id_profil from profil where Pseudo_profil = "'.$_POST['pseudo'].'"') and Id_receveur = (SELECT Id_profil from profil where Pseudo_profil = "'.$_SESSION['pseudo'].'"') and Status_requete = "attente")->fetchAll();

            if(count($xiste) == 1){
              $query = $connexion->prepare('UPDATE requete set Status_requete = "acceptee" where Id_envoyeur = (SELECT Id_profil from profil where Pseudo_profil = "'.$_POST['pseudo'].'"') and Id_receveur = (SELECT Id_profil from profil where Pseudo_profil = "'.$_SESSION['pseudo'].'"') and Status_requete = "attente"');
              $query->execute();
              echo'Vous êtes maintenant ami avec "'.$_POST['pseudo'].'"';
            }
            else{
              $xiste = $connexion->query('SELECT Id_requete from requete where Id_envoyeur = (SELECT Id_profil from profil where Pseudo_profil = "'.$_SESSION['pseudo'].'"') and Id_receveur = (SELECT Id_profil from profil where Pseudo_profil = "'.$_POST['pseudo'].'"') and Status_requete = "acceptee")->fetchAll();

              if(count($xiste) == 1){
                echo'Vous êtes déjà ami avec cet utilisateur';
              }
              else{
                $receveur = $connexion->query('SELECT Id_profil from profil where Pseudo_profil = "'.$_POST['pseudo'].'"')->fetchAll();

                $query = $connexion->prepare('INSERT INTO requete (Id_envoyeur, Id_receveur) VALUES (:Id_envoyeur, :Id_receveur)');
                $query->bindParam(':Id_envoyeur', $Id_envoyeur);
                $query->bindParam(':Id_receveur', $Id_receveur);

                $Id_envoyeur = $pseudo[0]['Id_profil'];
                $Id_receveur = $receveur[0]['Id_profil'];

                $query->execute();

                echo'La demande a bien été envoyée!';
              }
            }
          }
        }
      }
    }
  }
  else{
    echo'cet utilisateur n'existe pas';
  }
}
```

Si la demande est correcte, on récupère les données qu'on entre dans notre base de données.

Si l'utilisateur n'existe pas, ce message apparaîtra à l'écran de l'utilisateur.

Conduite de projet

Répartition des tâches

Fonctionnalité	Descriptif	Ressource
AS – Gestion des accès du site	Inscription d'un utilisateur	Fabien Djili
	Authentification d'un utilisateur inscrit	Léo Bruant
	Déconnexion d'un utilisateur connecté	
AU – Gestion des actions utilisateur	Création d'une question publique ou privée	Maxence Mlyncok
	Ajout d'une réponse à une question	Fabien Djili
	Ajout d'une mention « j'aime »	Léo Bruant
	Recherche d'une question	Tristan Evenot
	Suppression du compte	Fabien Djili
EA – Espace administrateur	Suppression d'une question	Maxence Mlyncok
	Suppression d'une catégorie	Tristan Evenot
	Ajout d'une catégorie	Léo Bruant
EPU – Espace profil de l'utilisateur	Modification du profil	Léo Bruant, Fabien Djili
	Ajout d'un avatar	Fabien Djili
A – Gestion des amis	Envoyer une demande	Léo Bruant
	Supprimer une demande	Maxence Mlyncok
	Accepter une demande	Léo Bruant
	Supprimer un ami	Tristan Evenot

Compte rendu de réunion

Appel à l'ordre

Compte rendu de la réunion de notre équipe de prestataires du 15 juin 2020 à Senlis.

Participants

Présents : Léo Bruant, Fabien Djili, Maxence Mlyncok, Tristan Evenot.

Approbation du compte rendu

Le site web développé pour la société Livequestion a été testé par l'ensemble de l'équipe, en physique.
La solution réalisée a alors été approuvée par l'ensemble de nos quatre développeurs.