



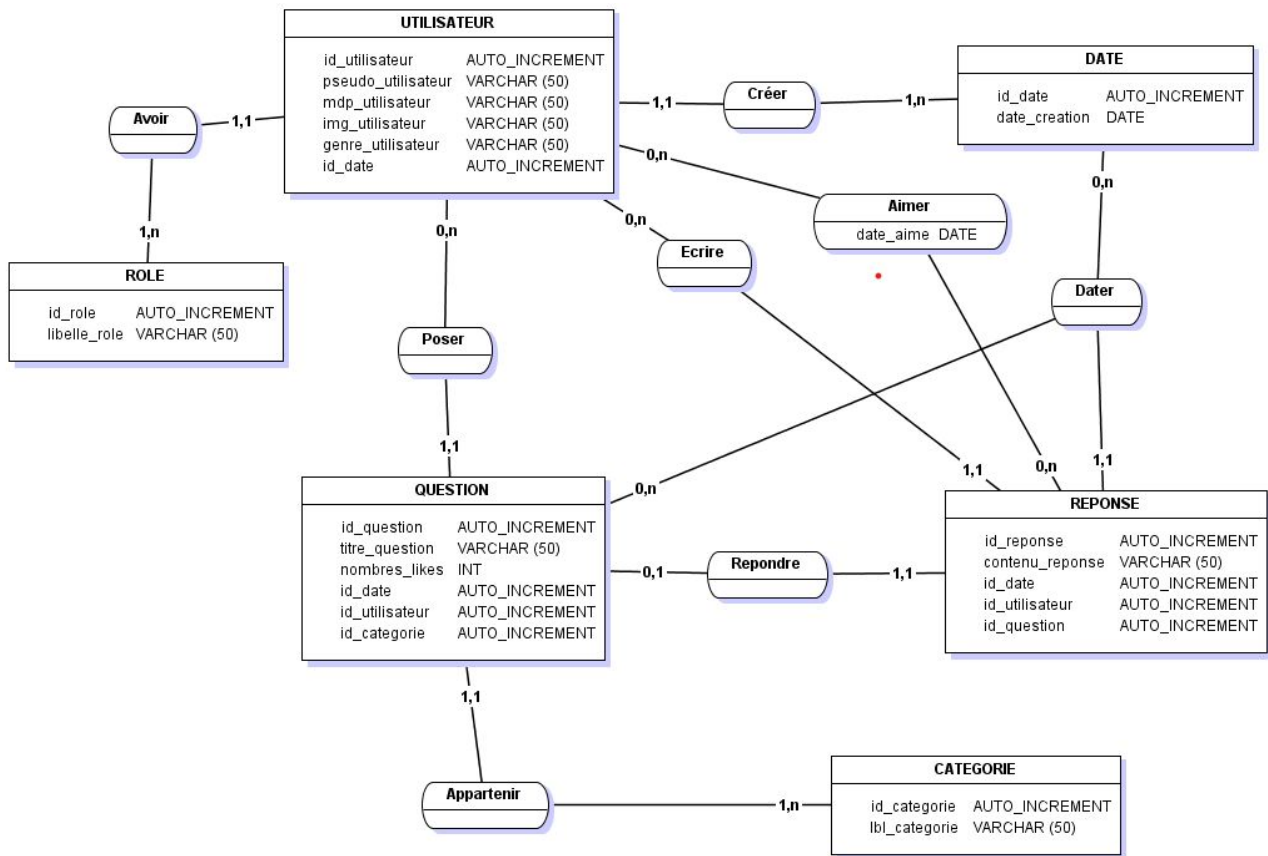
## Sommaire

- I. Les données
  - A. Dictionnaire de données
  - B. Modèle entité association
  - C. Schéma Relationnel
  - D. Scripts
- II. Les fonctionnalités
  - A. Backlog
  - B. Arborescence du projet
  - C. Maquettes
- III. Dossier technique
  - A. Focus sur une fonctionnalité
  - B. Les fonctionnalités
- IV. Veille et auto-formation
- V. Conduite de projet
  - A. Qui a fait quoi ?
  - B. Compte rendu de réunion

I. Les données  
A. Dictionnaire de données

Nom	Type	Table
id_categorie	INT	Catégorie
lbl_categorie	VARCHAR	Catégorie
id_date	INT	Date
date_creation	DATETIME	Date
id_utilisateur	INT	Utilisateur
pseudo_utilisateur	VARCHAR	Utilisateur
mdp_utilisateur	VARCHAR	Utilisateur
genre_utilisateur	VARCHAR	Utilisateur
img_utilisateur	VARCHAR	Utilisateur
id_question	INT	Question
titre_question	VARCHAR	Question
nombres_likes	INT	Question
id_reponse	INT	Réponse
contenu_reponse	VARCHAR	Réponse
id_categorie	INT	Catégorie
Demander_amis	BOOL	Amis
id_role	INT	Rôle
libelle_role	VARCHAR	Rôle
date_aime	DATETIME	Aime

## B. Modèle entité association



## C. Schéma relationnel

**QUESTION** (id\_question, titre\_question, nombres\_likes, id\_date, id\_utilisateur, id\_categorie)  
 Clé primaire = id\_question  
 Clé étrangère = id\_date en référence à id\_date de la table DATE  
 id\_categorie en référence à id\_categorie de la table

**CATEGORIE**  
 id\_utilisateur en référence à id\_utilisateur de la table  
**UTILISATEUR**

**CATÉGORIE** (id\_categorie, libelle\_categorie)  
 Clé primaire = id\_categorie

**RÔLE** (id\_role, libelle\_role)  
 Clé primaire = id\_role

**UTILISATEUR** (id\_utilisateur, pseudo\_utilisateur, mdp\_utilisateur, img\_utilisateur, genre\_utilisateur, id\_date, id\_role)  
 Clé primaire = id\_utilisateur  
 Clé étrangère = id\_date en référence à id\_date de la table DATE  
 id\_role en référence à id\_role de la table ROLE

REPONSE (id\_reponse, contenu\_reponse, id\_date, id\_utilisateur,  
id\_question)  
Clé primaire = id\_reponse  
Clé étrangère = id\_date en référence à id\_date de la table DATE  
id\_utilisateur en référence à id\_utilisateur de la table  
UTILISATEUR  
id\_question en référence à id\_question de la table  
QUESTION

Aimer (id\_reponse, id\_utilisateur, date\_aime)  
Clé primaire = id\_reponse, id\_utilisateur  
Clé étrangère = id\_reponse en référence à id\_utilisateur de la table  
REPONSE  
id\_utilisateur en référence à id\_utilisateur de la table  
UTILISATEUR

## D. Scripts

```
#-----
# Table: CATEGORIE
#-----
CREATE TABLE CATEGORIE(
    id_categorie Int Auto_increment NOT NULL ,
    lb_categorie Varchar (50) NOT NULL
    ,CONSTRAINT CATEGORIE_PK PRIMARY KEY (id_categorie)
)ENGINE=InnoDB;
#-----
# Table: DATE
#-----
CREATE TABLE DATE(
    id_date Int Auto_increment NOT NULL ,
    date_creation Date NOT NULL
    ,CONSTRAINT DATE_PK PRIMARY KEY (id_date)
)ENGINE=InnoDB;
#-----
# Table: ROLE
#-----
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;
#-----
# Table: UTILISATEUR
#-----
CREATE TABLE UTILISATEUR(
    id_utilisateur Int Auto_increment NOT NULL ,
    pseudo_utilisateur Varchar (50) NOT NULL ,
    mdp_utilisateur Varchar (50) NOT NULL ,
    img_utilisateur Varchar (50) NOT NULL ,
    genre_utilisateur Varchar (50) NOT NULL ,
```

```

        id_date          Int Auto_increment NOT NULL ,
        id_date_Creer    Int NOT NULL ,
        id_role          Int NOT NULL
        ,CONSTRAINT UTILISATEUR_PK PRIMARY KEY (id_utilisateur)

        ,CONSTRAINT UTILISATEUR_DATE_FK FOREIGN KEY
(id_date_Creer) REFERENCES DATE(id_date)
        ,CONSTRAINT UTILISATEUR_ROLE0_FK FOREIGN KEY (id_role)
REFERENCES ROLE(id_role)
)ENGINE=InnoDB;

```

```

#-----
# Table: QUESTION
#-----
CREATE TABLE QUESTION(
    id_question          Int Auto_increment NOT NULL ,
    titre_question       Varchar (50) NOT NULL ,
    nombres_likes        Int NOT NULL ,
    id_date              Int Auto_increment NOT NULL ,
    id_utilisateur       Int Auto_increment NOT NULL ,
    id_categorie         Int Auto_increment NOT NULL ,
    id_categorie_Appartenir Int NOT NULL ,
    id_utilisateur_Poser  Int NOT NULL
    ,CONSTRAINT QUESTION_PK PRIMARY KEY (id_question)

    ,CONSTRAINT QUESTION_CATEGORIE_FK FOREIGN KEY
(id_categorie_Appartenir) REFERENCES CATEGORIE(id_categorie)
    ,CONSTRAINT QUESTION_UTILISATEUR0_FK FOREIGN KEY
(id_utilisateur_Poser) REFERENCES UTILISATEUR(id_utilisateur)
)ENGINE=InnoDB;
#-----
# Table: REPONSE
#-----
CREATE TABLE REPONSE(
    id_reponse          Int Auto_increment NOT NULL ,
    contenu_reponse     Varchar (50) NOT NULL ,
    id_date             Int Auto_increment NOT NULL ,
    id_utilisateur       Int Auto_increment NOT NULL ,
    id_question         Int Auto_increment NOT NULL ,
    id_utilisateur_Ecrire Int NOT NULL ,
    id_question_Repondre Int NOT NULL
    ,CONSTRAINT REPONSE_PK PRIMARY KEY (id_reponse)

    ,CONSTRAINT REPONSE_UTILISATEUR_FK FOREIGN KEY
(id_utilisateur_Ecrire) REFERENCES UTILISATEUR(id_utilisateur)
    ,CONSTRAINT REPONSE_QUESTION0_FK FOREIGN KEY
(id_question_Repondre) REFERENCES QUESTION(id_question)
    ,CONSTRAINT REPONSE_QUESTION_AK UNIQUE
(id_question_Repondre)

```

```

)ENGINE=InnoDB;

#-----
# Table: Dater
#-----
CREATE TABLE Dater(
    id_question Int NOT NULL ,
    id_reponse Int NOT NULL ,
    id_date Int NOT NULL
    ,CONSTRAINT Dater_PK PRIMARY KEY
(id_question,id_reponse,id_date)

    ,CONSTRAINT Dater_QUESTION_FK FOREIGN KEY (id_question)
REFERENCES QUESTION(id_question)
    ,CONSTRAINT Dater_REPONSE0_FK FOREIGN KEY (id_reponse)
REFERENCES REPONSE(id_reponse)
    ,CONSTRAINT Dater_DATE1_FK FOREIGN KEY (id_date)
REFERENCES DATE(id_date)
)ENGINE=InnoDB;

#-----
# Table: Aimer
#-----
CREATE TABLE Aimer(
    id_reponse Int NOT NULL ,
    id_utilisateur Int NOT NULL ,
    date_aime Date NOT NULL
    ,CONSTRAINT Aimer_PK PRIMARY KEY (id_reponse,id_utilisateur)

    ,CONSTRAINT Aimer_REPONSE_FK FOREIGN KEY (id_reponse)
REFERENCES REPONSE(id_reponse)
    ,CONSTRAINT Aimer_UTILISATEUR0_FK FOREIGN KEY
(id_utilisateur) REFERENCES UTILISATEUR(id_utilisateur)
)ENGINE=InnoDB;

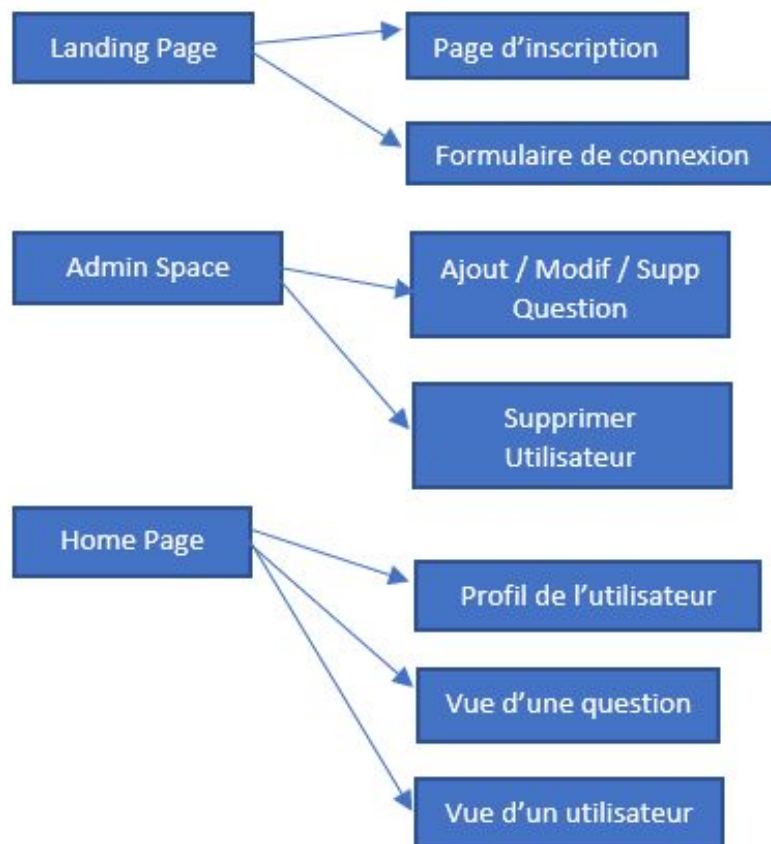
```

## II. Les fonctionnalités

### A. Backlog

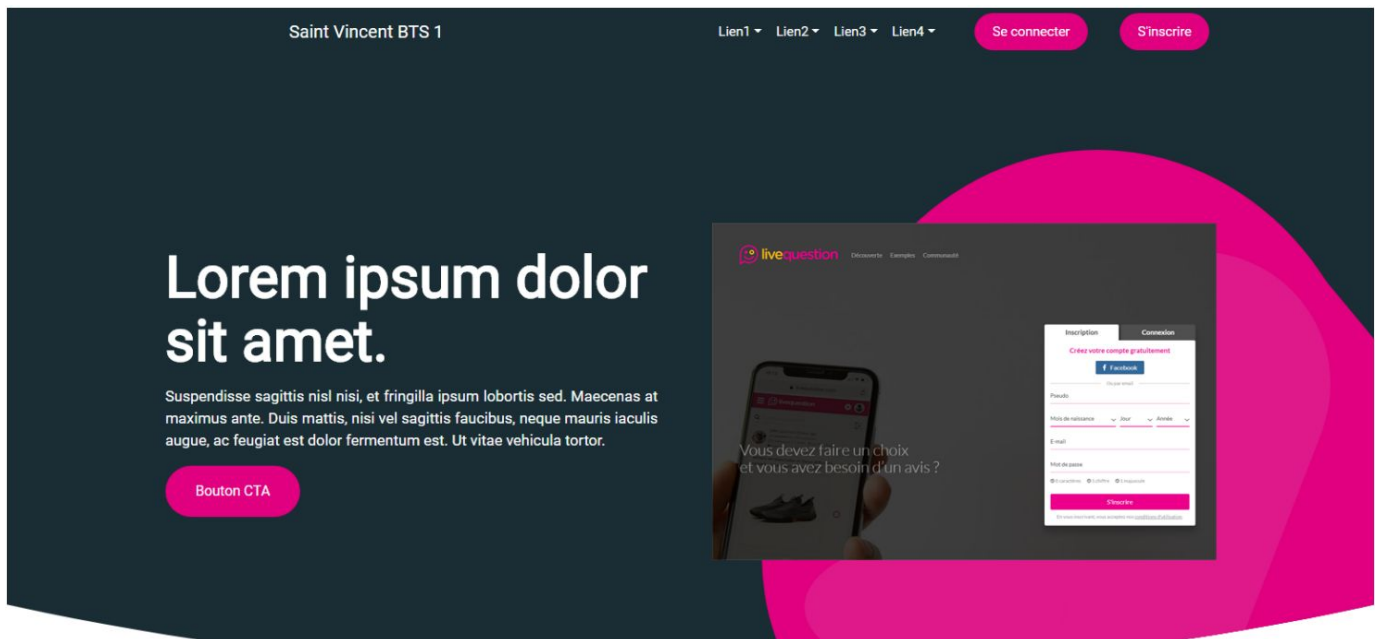
Fonctionnalité	Description
Inscription Utilisateur	Formulaire d'inscription
Connexion Utilisateur	Formulaire de connexion
Déconnexion	Bouton de déconnexion
Désinscription	Formulaire de désinscription
Flux question	Affiche l'ensemble des questions
Ajout Question	Formulaire d'ajout de questions
Supprimer Question	Supprimer une question
Modification Question	Formulaire modification question
Répondre Question	Formulaire ajout d'une réponse
Supprimer Réponse	Supprimer une Réponse
Aimer	Aimer une Question
Espace Administrateur	Back End pour gérer le site

### B. Arborescence du projet

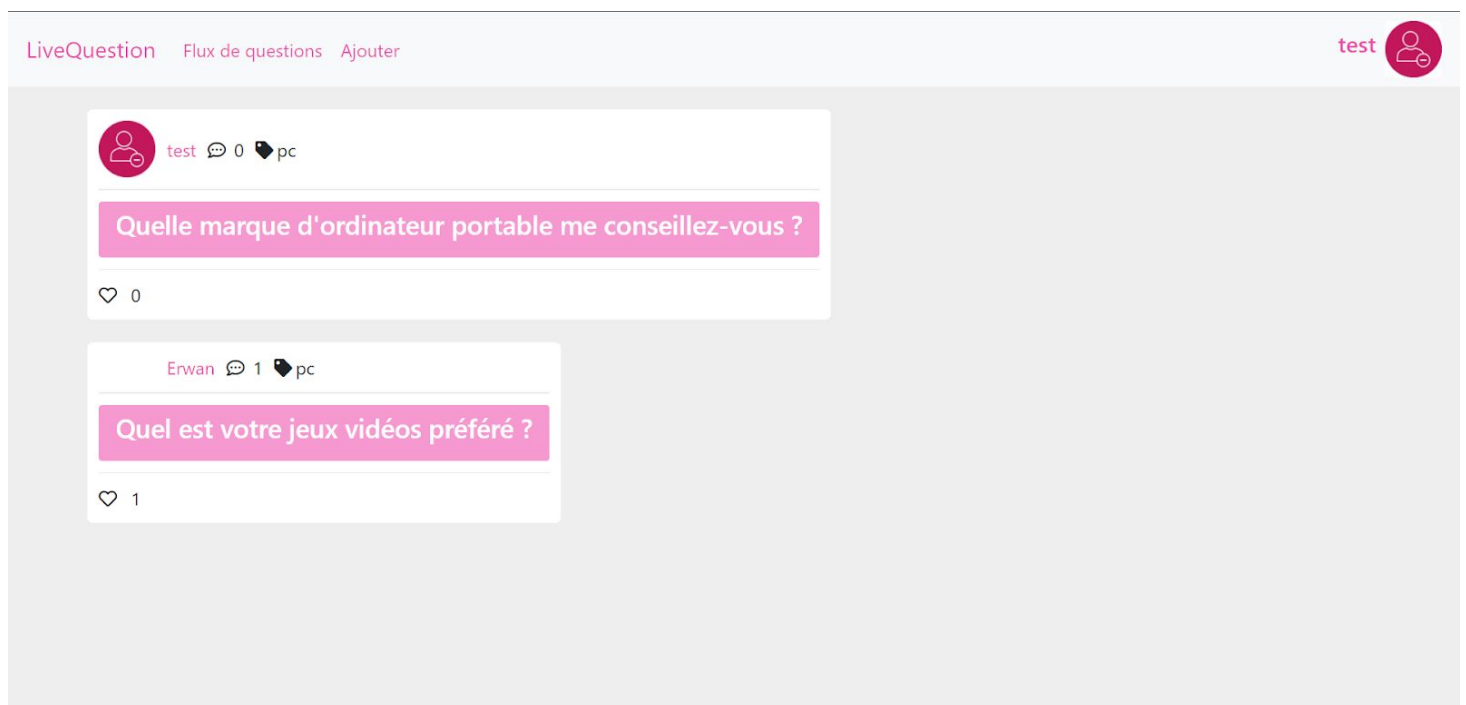


## C. Maquettes

### 1. Page d'accueil déconnecté



### 2. Page d'accueil connecté

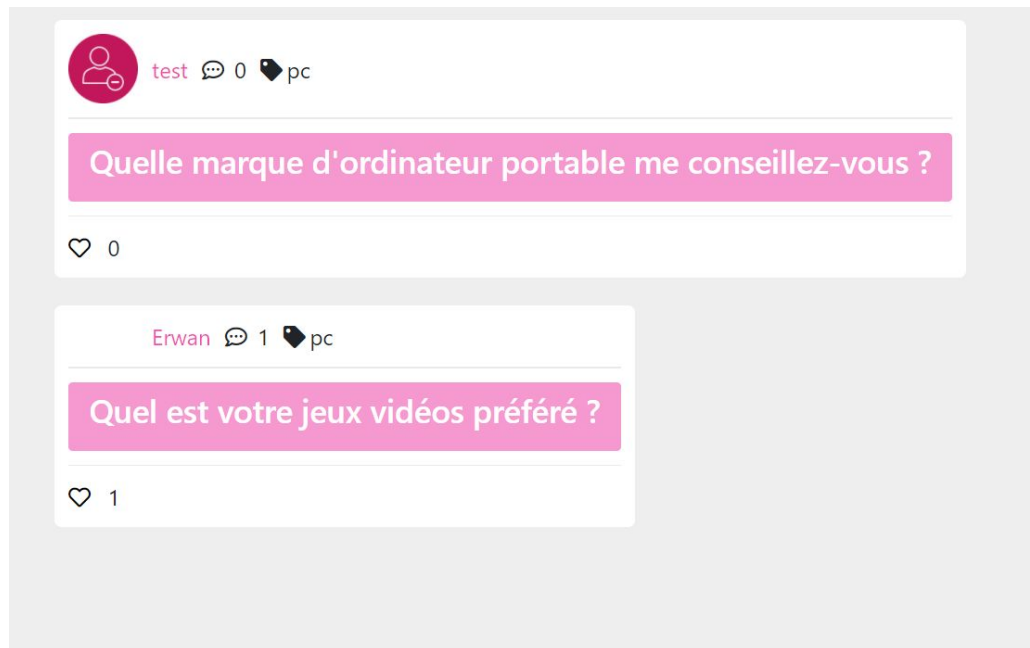




### III. Dossier technique

#### A. Focus sur certaines fonctionnalités

La fonctionnalité principale de ce site est le fait qu'on puisse poser des questions. Lorsqu'on arrive sur le flux de questions, on peut y voir toutes les questions posées.



Pour poser une question, il suffit d'aller dans le menu "ajouter". Un menu s'affiche et on peut choisir la catégorie de la question ainsi que le contenu de votre question

## Posez une question

Quelle est votre question ?

catégorie


BTS

valider

### B. Les fonctionnalités

Mais il y a également d'autres fonctionnalités :

- la réponse de question

 **test** ⌚ 2020-12-07 08:51:13 💬 0

Quelle marque d'ordinateur portable me conseillez-vous ?

♡ 0

Ajouter une réponse

valider

- l'inscription

A registration form titled "Inscrivez-vous" in pink. It contains five input fields: "Identifiant", "Adresse Mail", "Sexe" (a dropdown menu currently showing "Femme"), "Mot de passe", and "Vérification du Mot de passe". At the bottom is a pink button labeled "S'inscrire".

- la modification d'un profil

A profile modification form titled "Modifier votre avatar" in pink. It includes a file selection area with a button "Choisir un fichier" and the text "Aucun fichier choisi". Below are three input fields labeled "Modifier identifiant", "Nouvelle adresse mail", and "Nouveau mot de passe". A pink button labeled "valider" is at the bottom.

- un espace administrateurs (avec la possibilité pour eux de supprimer des questions)

Question	Date de publication	Auteur	
Quelle marque d'ordinateur portable me conseillez-vous ?	2020-12-07 08:51:13	test	×
Quel est votre jeux vidéos préféré ?	2020-04-29 16:18:36	Erwan	×

- la possibilité de se désinscrire



#### IV. Veille et auto-formation

Pour ce projet, nous avons dû parfaire nos connaissances en php car c'est le premier projet que l'on a fait qui demandait autant de connaissances dans ce langage. De plus, l'utilisation d'outils de versionning ou de collaboration tout au long du projet nous a permis de se familiariser avec ce genre d'outils.

## Etude code : Affichage Questions page d'accueil utilisateur connecté

```
1  <?php
2      require_once("../includes/db-connect.php");
3      function requetePagination(){
4          $questionParPage = 30;
5          $nbTotalQuestReq = connexionBdd()->query('SELECT Id_question FROM question');
6          $nbTotal = $nbTotalQuestReq->rowCount();
7          if(isset($_GET['page']) AND !empty($_GET['page']) AND $_GET['page']>0)
8          {
9              $_GET['page'] = intval($_GET['page']);
10             $pageCourante = $_GET['page'];
11         } else {
12             $pageCourante = 1;
13         }
14         $depart = ($pageCourante-1)*$questionParPage;
15         $page = 'LIMIT '.$depart.','.$questionParPage;
16         $nbPagesTotal = ceil($nbTotal/$questionParPage);
17         return $page;
18     }
19     function getQuestion(){
20         $query = connexionBdd()->prepare('
21             SELECT *
22             FROM question Q
23             LEFT JOIN restriction_question RQ ON Q.Id_question = RQ.`#id_question`
24             LEFT JOIN categorie C ON Q.`#Id_categorie` = C.Id_categorie
25             LEFT JOIN profil P ON Q.`#Id_profil` = P.Id_profil
26             LEFT JOIN role R ON P.`#Id_role` = R.Id_role
27             WHERE Q.type_question = 0 OR Q.type_question = 1 AND Q.`#Id_profil` = :id
28             UNION
29             SELECT *
30             FROM question Q
31             LEFT JOIN restriction_question RQ ON RQ.`#id_question` = Q.Id_question
32             LEFT JOIN categorie C ON Q.`#Id_categorie` = C.Id_categorie
33             LEFT JOIN profil P ON Q.`#Id_profil` = P.Id_profil
34             LEFT JOIN role R ON P.`#Id_role` = R.Id_role
35             WHERE RQ.`#id_profil` = :id
36             ORDER BY Date_creation_question DESC, Id_profil, Titre_question '
37             .requetePagination());
38         $query->execute([
39             "id" => $_SESSION['id'],
40         ]);
41         $data = $query->fetchAll(PDO::FETCH_ASSOC);
42         $query->closeCursor();
43         return $data;
44     }
45     ?>
```

## V. Conduite de projet

### A. Qui a fait quoi ?

Testeur	Fonctionnalité
Antoine	Page d'accueil avant la connexion
Erwan	Page d'accueil après la connexion
Antoine	La partie base de données
Erwan	Page de connexion/inscription + déconnexion
Antoine	menu de réponse aux questions
Erwan	modification profil + menu administrateurs
Antoine	Désinscription

### B. Compte rendu de la réunion

Tous les lundis à 13h30, nous mettons en commun ce qu'on avait fait ainsi que les différentes tâches à réaliser durant la semaine qui arrivait. Chacun expliquait ce qu'il avait fait et comment il avait fait pour que les membres du groupe sachent comment chaque fonctionnalité avait été effectuée.