

Dossier fonctionnel et technique



Date de début : 22/02/2020

Date de fin : 30/06/2020

I. Les données techniques

I.1	Le dictionnaire de données	page 3
I.2	Modèle d'entité-association	page 4
I.3	Schéma relationnel	page 4
I.4	Le Script SQL	page 5

II. Les fonctionnalités

II.1	Le backlog : la liste des fonctionnalités	page 6
II.2	L'arborescence du site	page 6
II.3	Les cas d'utilisations	page 7
II.4	Maquette final	page 7-8

III. Le dossier technique

III.1	Focus sur 2 fonctionnalités	page 9
III.2	Extrait de code commenté	page 10

IV. Veille et auto-formation contextualisée

	Mes veilles et résolution de mes problèmes	page 11
--	--	---------

V. Conduite de projet

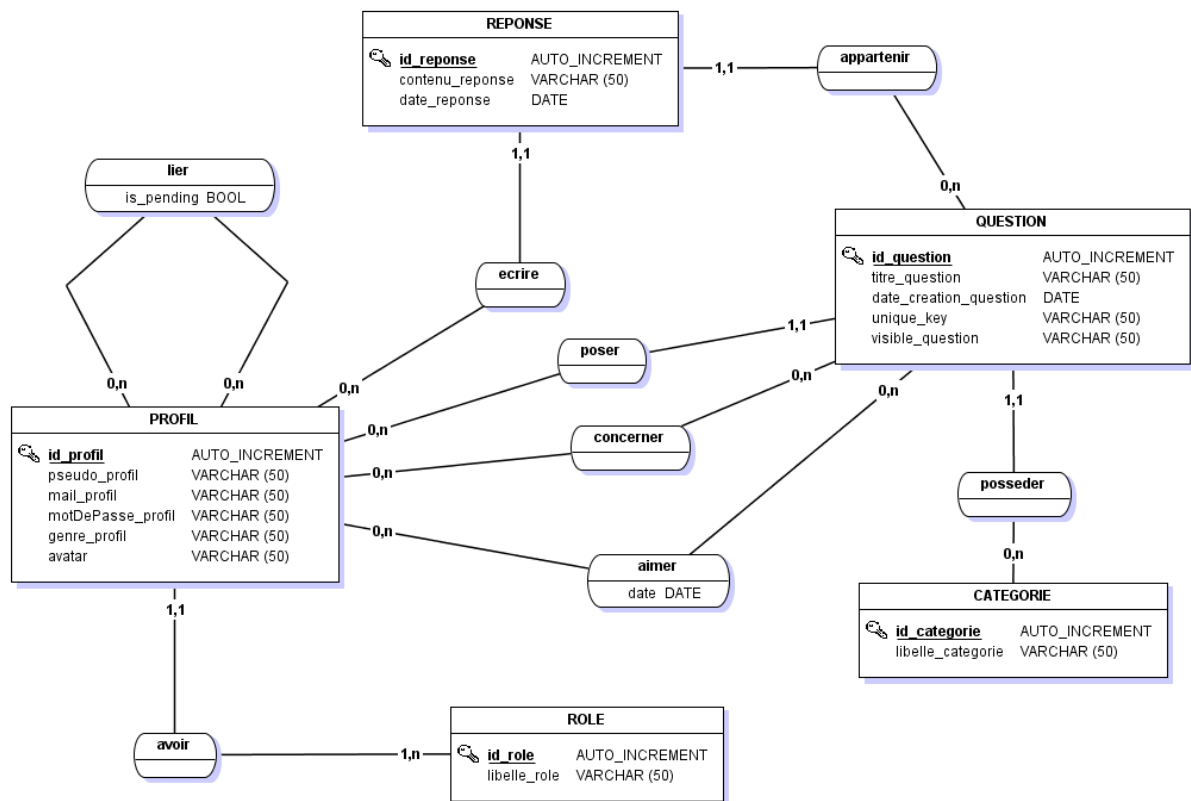
V.1	Répartition des taches et compte rendu de réunions	page 11
-----	--	---------

I/ Les données

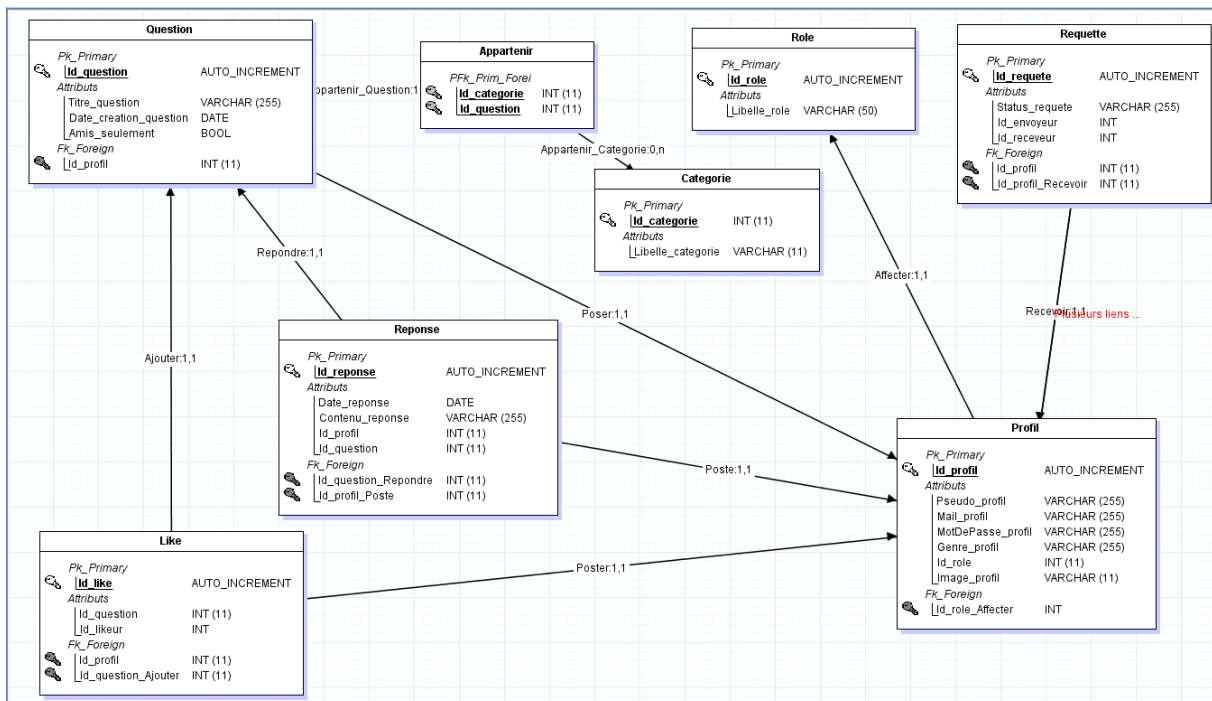
1.1dictionnaire de données

nom	type
id_role	int
libelle_role	varchar
id_profil	int
pseudo_profil	varchar
mail_profil	varchar
mot_de_passe_profil	varchar
genre_profil	varchar
id_role_profil	int
image_profil	varchar
id_categorie	int
libelle_categorie	varchar
id_question	int
titre_question	varchar
date_creation_question	date
amis_seulement	tinyint
id_reponse	int
date_reponse	date
contenu_reponse	varchar
id_profil_reponse	int
id_question_reponse	int
id_like	int
id_question_like	int
id_profil_like	int
id_requette	int
status_requette	varchar
id_envoyeur	int
id_receveur	int

1.2 Modèle d'entité-association



1.3 Schéma relationnel



1.4 Script SQL

```
CREATE TABLE IF NOT EXISTS `categorie` ( `Id_categorie` int(11) NOT NULL AUTO_INCREMENT,  
`Libelle_categorie` varchar(255) CHARACTER SET latin1 NOT NULL, PRIMARY KEY (`Id_categorie`)  
) ENGINE=InnoDB AUTO_INCREMENT=51 DEFAULT CHARSET=utf8  
INSERT INTO `categorie` (`Id_categorie`, `Libelle_categorie`) VALUES (1, 'automobile'), (2, 'pop  
culture'), (3, 'nature'), (4, 'cuisine'), (5, 'sciences'), (6, 'art')
```

```
CREATE TABLE IF NOT EXISTS `likes` ( `Id_like` int(11) NOT NULL AUTO_INCREMENT,  
`Id_question` int(11) NOT NULL, `Id_likeur` int(11) NOT NULL, PRIMARY KEY (`Id_like`), KEY  
`Id_question` (`Id_question`), KEY `Id_likeur` (`Id_likeur`) ) ENGINE=InnoDB AUTO_INCREMENT=6  
DEFAULT CHARSET=latin1
```

```
CREATE TABLE IF NOT EXISTS `profil` ( `Id_profil` int(11) NOT NULL AUTO_INCREMENT,  
`Pseudo_profil` varchar(255) CHARACTER SET latin1 NOT NULL, `Mail_profil` varchar(255)  
CHARACTER SET latin1 NOT NULL, `MotDePasse_profil` varchar(255) CHARACTER SET latin1 NOT  
NULL, `Genre_profil` varchar(50) CHARACTER SET latin1 NOT NULL, `Id_role` int(11) NOT NULL,  
`Image_profil` varchar(255) NOT NULL DEFAULT 'https://i2.wp.com/yellowsunsummary.com/wp-  
content/uploads/2019/02/lcone-profil.png?fit=512%2C512&ssl=1', PRIMARY KEY (`Id_profil`), KEY  
`#Id_role` (`Id_role`) ) ENGINE=InnoDB AUTO_INCREMENT=26 DEFAULT CHARSET=utf8
```

```
CREATE TABLE IF NOT EXISTS `question` ( `Id_question` int(11) NOT NULL AUTO_INCREMENT,  
`Titre_question` varchar(255) CHARACTER SET latin1 NOT NULL, `Date_creation_question` date NOT  
NULL, `Id_profil` int(11) NOT NULL, `Id_categorie` int(11) NOT NULL, `Amis_seulement` tinyint(1) NOT  
NULL DEFAULT '0', PRIMARY KEY (`Id_question`), KEY `#Id_profil` (`Id_profil`), KEY `Id_categorie`  
(`Id_categorie`) ) ENGINE=InnoDB AUTO_INCREMENT=177 DEFAULT CHARSET=utf8
```

```
CREATE TABLE IF NOT EXISTS `reponse` ( `Id_reponse` int(11) NOT NULL AUTO_INCREMENT,  
`Contenu_reponse` varchar(255) CHARACTER SET latin1 NOT NULL, `Date_reponse` date NOT  
NULL, `Id_profil` int(11) NOT NULL, `Id_question` int(11) NOT NULL, PRIMARY KEY (`Id_reponse`),  
KEY `#Id_profil` (`Id_profil`), KEY `#Id_question` (`Id_question`) ) ENGINE=InnoDB  
AUTO_INCREMENT=2 DEFAULT CHARSET=utf8
```

```
CREATE TABLE IF NOT EXISTS `requete` ( `Id_requete` int(11) NOT NULL AUTO_INCREMENT,  
`Status_requete` varchar(255) NOT NULL DEFAULT 'attente', `Id_envoyeur` int(11) NOT NULL,  
`Id_receveur` int(11) NOT NULL, PRIMARY KEY (`Id_requete`), KEY `Id_envoyeur` (`Id_envoyeur`),  
KEY `Id_receveur` (`Id_receveur`) ) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT  
CHARSET=latin1
```

```
CREATE TABLE IF NOT EXISTS `role` ( `Id_role` int(11) NOT NULL AUTO_INCREMENT,  
`Libelle_role` varchar(50) CHARACTER SET latin1 NOT NULL, PRIMARY KEY (`Id_role`) )  
ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8
```

```
INSERT INTO `role` (`Id_role`, `Libelle_role`) VALUES (1, 'utilisateur'), (2, 'administrateur')
```

```
ALTER TABLE `likes` ADD CONSTRAINT `likes_ibfk_1` FOREIGN KEY (`Id_likeur`) REFERENCES  
`profil` (`Id_profil`)
```

```
ALTER TABLE `profil` ADD CONSTRAINT `FK_#Id_role` FOREIGN KEY (`Id_role`) REFERENCES  
`role` (`Id_role`)
```

```
ALTER TABLE `reponse` ADD CONSTRAINT `FK_#Id_profil2` FOREIGN KEY (`Id_profil`)  
REFERENCES `profil` (`Id_profil`), ADD CONSTRAINT `FK_#Id_question` FOREIGN KEY  
(`Id_question`) REFERENCES `question` (`Id_question`)
```

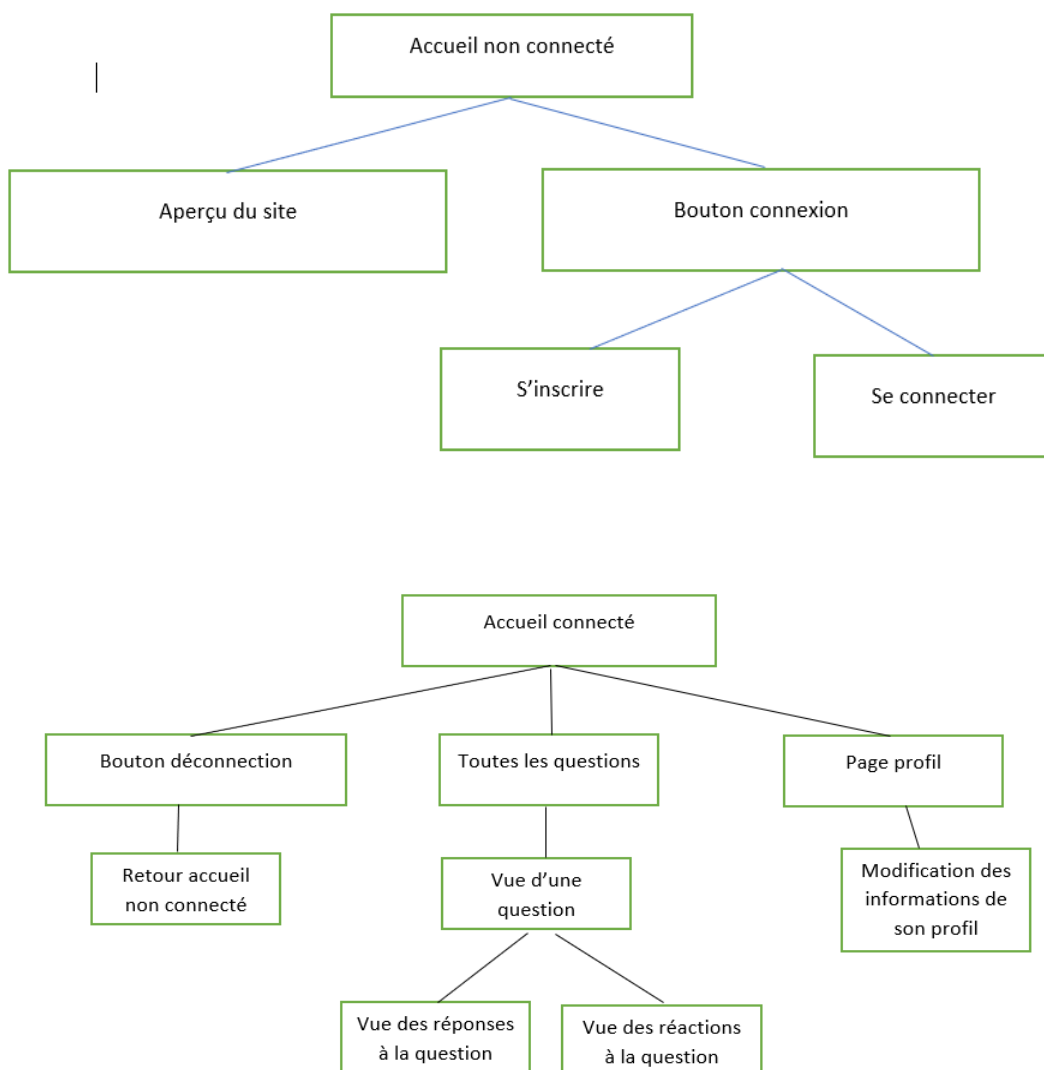
```
ALTER TABLE `requete` ADD CONSTRAINT `requete_ibfk_1` FOREIGN KEY (`Id_envoyeur`)  
REFERENCES `profil` (`Id_profil`), ADD CONSTRAINT `requete_ibfk_2` FOREIGN KEY (`Id_receveur`)  
REFERENCES `profil` (`Id_profil`)
```

II) Les fonctionnalités

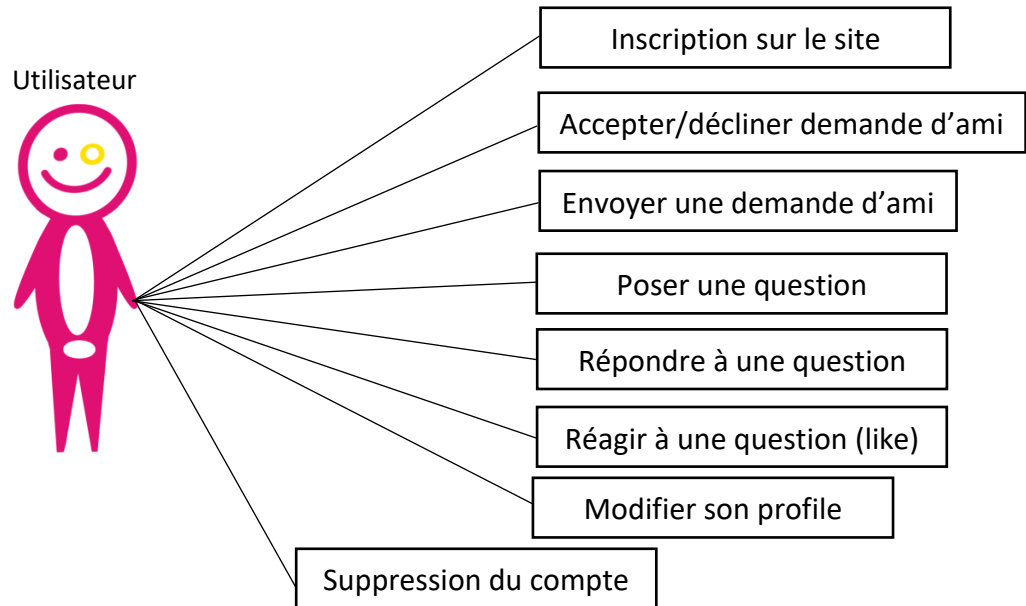
2.1 backlog : liste des fonctionnalités

- créer un compte
- se connecter/déconnecter
- poser des questions
- répondre à des questions
- mettre des "j'aimes" à des questions
- ajouter des amis
- poser des questions uniquement à nos amis
- modifier les informations du profil
- se désinscrire

2.2 l'arborescence

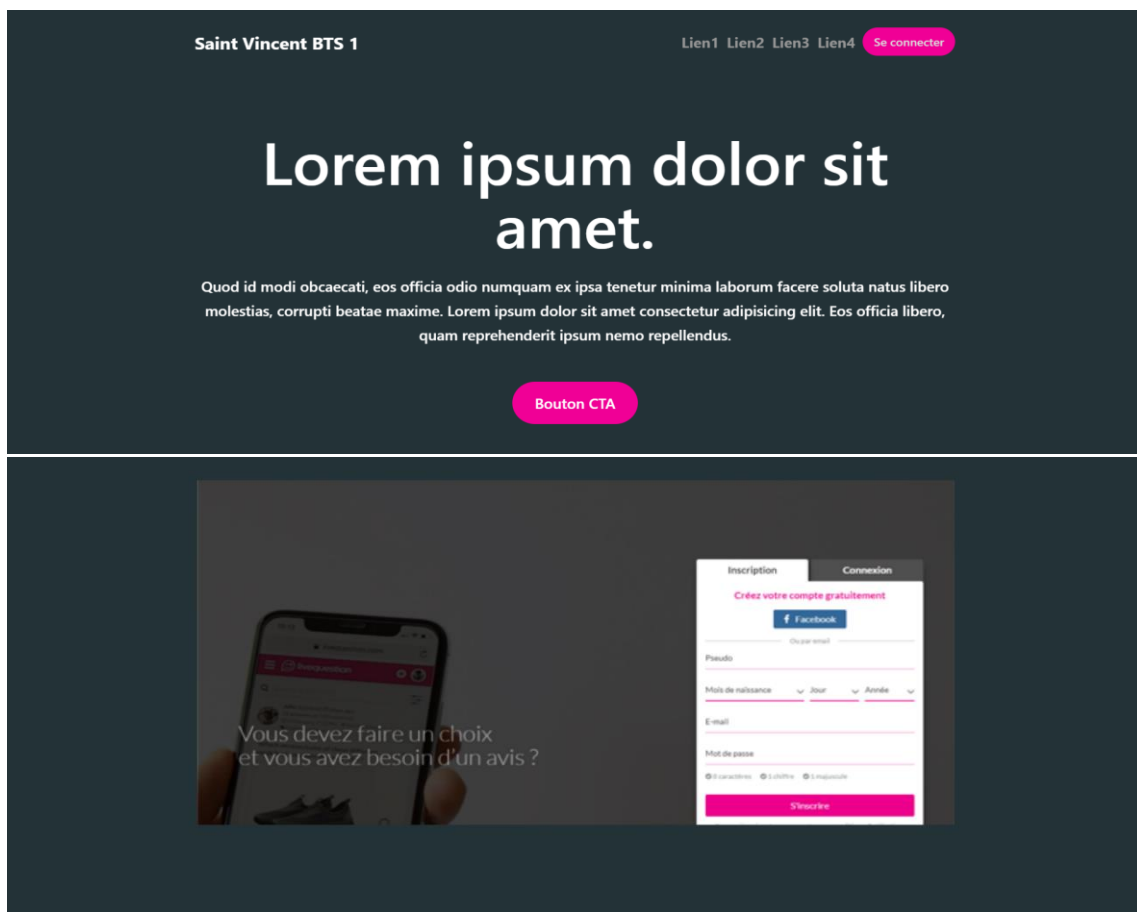


2.3 les cas d'utilisation



2.4 les maquettes du site final

Page d'accueil (imposé par la volonté du client)





Page d'affichage des questions



Page du profil

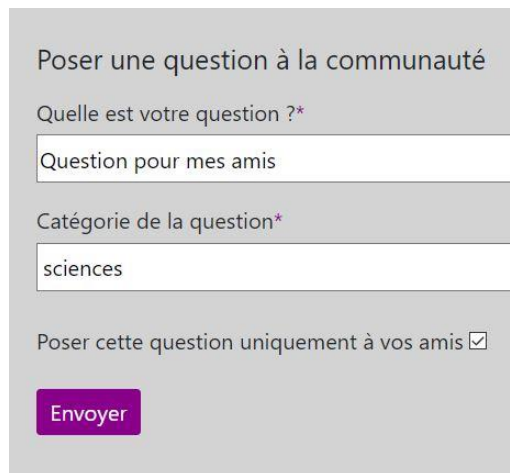


III) dossier technique

3.1 Focus sur 2 fonctionnalités

Fonctionnalité : poser une question à ses amis

En cochant une case, nous pouvons poser une question uniquement à nos amis.



The screenshot shows a web form titled "Poser une question à la communauté". It contains the following elements: a label "Quelle est votre question ?*" above a text input field containing "Question pour mes amis"; a label "Catégorie de la question*" above a text input field containing "sciences"; a checkbox labeled "Poser cette question uniquement à vos amis" which is checked; and a purple button labeled "Envoyer".

Fonctionnalité accepter ou refuser une demande d'ami

Il suffit de rentrer le nom d'utilisateur de notre ami pour lui envoyer une demande. Il a le choix ensuite sur la page de son profil de l'accepter ou de la décliner.

Demandes d'amis:

Utilisateur: leo

[Accepter la demande](#)

[Refuser la demande](#)

envoyer une demande d'ami:



The form consists of two input fields side-by-side. The first field is labeled "Pseudo de l'utilisateur" and the second field is labeled "envoyer".

3.2 Code commenté

Fonctionnalité : poser une question à ses amis

```
for($i = 0; $i < count($question) * $nb_questions * ($_GET['page']-1) && $i < $nb_questions; $i++){  
    // récupération des informations correspondant aux questions  
  
    $categorie = $connexion->query('SELECT Libelle_categorie FROM categorie WHERE Id_categorie = '.$question[$i  
+ $nb_questions * ($_GET['page']-1)]['Id_categorie']->fetchAll();  
    $profil = $connexion->query('SELECT Id_profil, Pseudo_profil, Image_profil FROM profil WHERE Id_profil = '.$question  
[ $i + $nb_questions * ($_GET['page']-1)]['Id_profil']->fetchAll();  
    $reponse = $connexion->query('SELECT COUNT(*) FROM reponse WHERE Id_question = '.$question[$i + $nb_questions * ($_GET  
['page']-1)]['Id_question']->fetchAll();  
  
    // vérification de si la question est en mode amis uniquement  
  
    $afficher = 0;  
  
    for ($i2 = 0; $i2 < count($amis); $i2++){  
        if($amis[$i2]['Id_envoyeur'] == $_SESSION['id']){  
            if($amis[$i2]['Id_receveur'] == $profil[0]['Id_profil'] || $_SESSION['id'] == $profil[0]['Id_profil']){  
                $afficher = 1;  
            }  
        }  
        elseif($amis[$i2]['Id_envoyeur'] == $profil[0]['Id_profil'] || $_SESSION['id'] == $profil[0]['Id_profil']){  
            $afficher = 1;  
        }  
    }  
}
```

Fonctionnalité accepter ou refuser une demande d'ami

Ici on récupère l'identifiant
de l'envoyeur directement
dans la session lancée

```
<?php  
session_start();  
require_once('connexion_bdd.php');  
  
$query = $connexion->prepare('UPDATE requete set Status_requete = "acceptee" where Id_envoyeur = '.$_GET['id'].'  
and Id_receveur = (SELECT Id_profil from profil where Pseudo_profil = "'.$_SESSION['pseudo'].'")');  
$query->execute();  
  
header( string: 'location: http://localhost/Livequestion-master/profil.php');  
?>
```

Ici on récupère l'identifiant
du receveur dans la table
profil

La seule différence est l'affectation à la
variable Status_requete à refusée ou acceptée

IV) Veille ou auto-formation contextualisée

Pour ce projet j'ai réalisé des veilles sur certaines fonctionnalités comme le hachage de mot de passe ainsi que des recherches sur le fonctionnement d'une session lorsqu'un utilisateur utilise un site. J'ai aussi sollicité de multiples fois l'aide de notre encadrant de projet, un CTO chez MentalWork.

V) Conduite de projet

5.1 Répartition des taches et compte rendu de réunions

La répartition des taches s'est faite par groupe, nous étions très souvent 2 à réaliser un commit et nous nous réunissions sur discord pour parler de nos éventuels problèmes. Les réunions se passaient en groupe de projet mais le reste du temps nous nous connectons aussi avec nos collègues pour faciliter la communication. Il n'y avait donc pas de réunions obligatoires hormis les comptes rendu a faire a notre tuteur projet Jeff Martins Jaquelot.

La répartition des taches du projet est disponible ici :

<https://github.com/LeoBruant/Livequestion/issues>