

DEV-WEB DS1

Objectif:

Réalisation d'un forum de discussion (questions/réponses).

Un utilisateur poste une question, les autres répondent à cette question.

Cette application se compose essentiellement de trois pages

Page	Description
<i>home.php</i>	Cette page affiche la liste des questions au public, en plus, si l'utilisateur est authentifié il peut poster une nouvelle question
<i>question.ph</i>	Cette page affiche une question et la listes des réponses à cette question au public, en plus, si l'utilisateur est authentifié il peut poster une nouvelle réponse à cette question
<i>login.php</i>	Cette page permet à un utilisateur de s'authentifier
<i>register.php</i>	Cette page permet à un visiteur de s'enregistrer
<i>db.php</i>	Des fonctions pour interagir avec la base de données

Partie 1:

1.1- Reproduire la structure ci contre et initialiser un dépôt git dans le dossier forum avec un premier commit "initialisation du projet".

1.2- Dans phpmyadmin, créer une base de donnée "forum" avec les tables suivantes:

Users(id, nom, email, password)

Questions(id, user_id, question, date)

Reponses(id, user_id, question_id, response, date)

1.3- Dans **db.php** écrire script pour établir une connexion à la base de données et une fonction

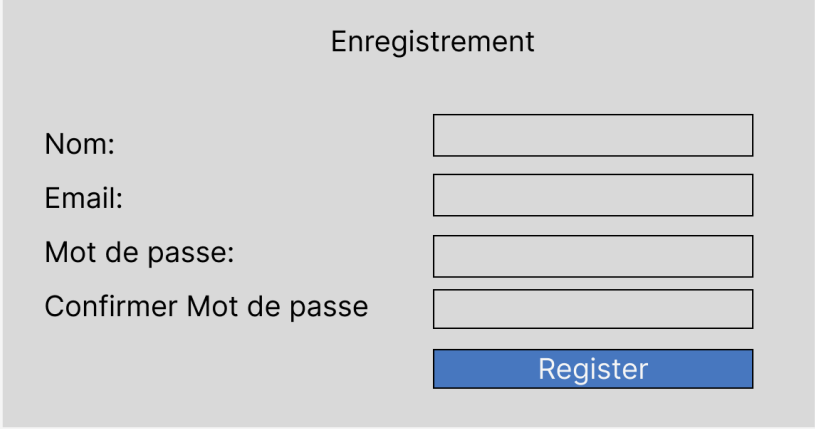
adduser(nom,email,password) pour insérer un nouvel utilisateur.

1.4- Dans **register.php** insérer un formulaire d'enregistrement d'utilisateur qui poste les données au script **adduser.php**.

```

forum
├── controllers
│   ├── adduser.php
│   └── auth.php
├── index.php
├── models
│   └── db.php
├── public
│   ├── questions.js
│   └── reponses.js
└── views
    ├── home.php
    ├── login.php
    ├── question.php
    └── register.php
  
```

1.5- dans le fichier **adduser.ph** vérifiez si les données sont valides puis utilisez la fonction **password_hash()** pour crypter le mot de passe et finalement appelez la fonction **adduser(nom,email,hash)**. Faire un commit avec le message "Partie 1 : Ajout d'un nouvel utilisateur".



Enregistrement

Nom:

Email:

Mot de passe:

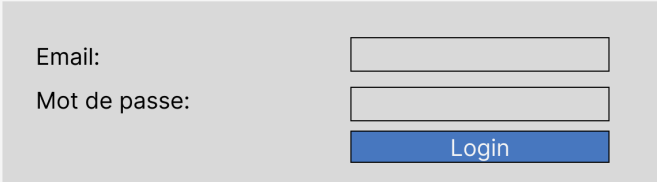
Confirmer Mot de passe:

Partie 2

2.1- Dans **login.php** insérer un formulaire d'authentification qui poste les données (email et password) au script **auth.php**.

2.2- Dans **db.php** écrire une fonction **getUserByEmail(email)** pour récupérer les données d'un utilisateur.

2.3- dans le fichier **auth.php** vérifiez si les données sont valides puis utiliser la fonction **password_verify()** pour comparer le mot de passe fourni avec celui de la base de donnée. S'il y a correspondance alors on crée une variable de session "**user**" contenant le nom et l'email de cette utilisateur puis en redirige l'utilisateur vers **home.php**. Sinon l'utilisateur est renvoyé vers login avec un message d'erreur en haut de page. Faire un commit avec le message "Partie 2: Login".



Email:

Mot de passe:

Partie 3

- 3.1- Dans **db.php** écrire une fonction **getQuestions()** pour récupérer toutes les questions.
- 3.2- Dans **db.php** écrire une fonction **getReponses(question_id)** pour récupérer toutes les réponses à une questions.
- 3.3- Dans **home.php** insérer la liste des questions (tableau), la plus récente en haut de page.

Date - nom_user - email_user - Question - réponses(lien)

The screenshot shows a web interface for a Q&A system. At the top right, there is a link "Logout/login". Below it is a text input field with the placeholder "Votre question ici". To the right of this field is a blue button labeled "Envoyer". Below the input field is a container for a list of questions. Each question entry consists of a date and user email (e.g., "20-11-2023 reda@est.ovh"), a text input field for the question (e.g., "une question"), and a blue link labeled "Réponses" at the bottom right of the entry.

Un clique sur réponse envoie le visiteur vers **question.php** qui affiche la liste des réponses à une question, la plus récente en haut de page. Si un utilisateur est authentifié, alors en affiche aussi en haut de page une zone de texte et un bouton Envoyer pour poster une nouvelle réponse.

- 3.4- faire un commit "Partie 3: Questions/réponses"

The screenshot shows a web interface for a forum. At the top right, there are links for 'Home' and 'Logout/login'. Below this, there is a section for asking a question, labeled 'Une question ...'. It contains a text input field with the placeholder 'Votre réponse ici' and a blue button labeled 'Envoyer'. Below the input field, there is a list of previous questions and answers. The first entry is dated '20-11-2023' and from 'reda@est.ovh', with a response field containing 'une reponse'. The second entry is dated '18-11-2023' and from 'yahya@est.ovh', with a response field containing 'une autre reponse'. The entire form is enclosed in a light gray border.

Partie 4

Pour voir les nouveaux posts, un utilisateur doit constamment rafraîchir la page.

4.1- Pour remédier à ce problème, écrire deux scripts javascript ***question.js*** et ***reponses.js*** qui récupèrent les nouvelles questions/réponses à une fréquence de 30s et mettent à jour les page questions et réponses.

4.2- faire un commit "Partie 4: js pulling"

Partie 5

5.1- A l'aide de phpmyadmin exportez le données de votre base de donnée dans des fichiers forum/users.sql, forum/questions.sql et forum/réponses.sql

5.2- Compresser le dossier forum et envoyer le sur ce lien gounane.ovh:5000