

582-21B-MA PROGRAMMATION WEB DYNAMIQUE

Travail Pratique 02:

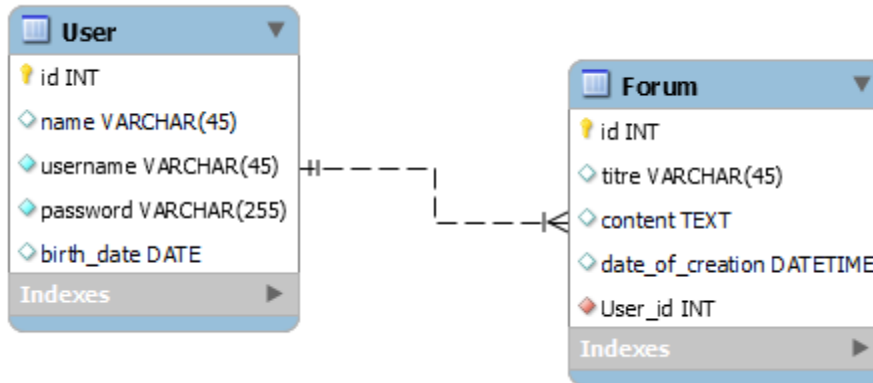
Forum maison neuve:

**[http://localhost/webDynamique/TP02-
base-de-donne/mvc/](http://localhost/webDynamique/TP02-base-de-donne/mvc/)**

Samaneh Mahboudi

E2495960

Modèle Logique de Données (MLD) :



Explication :

Le but de ce projet est de créer une base de données pour gérer un forum de discussion. Le système permet aux utilisateurs de s'inscrire, de se connecter et de publier des articles sur le forum. Chaque utilisateur peut publier plusieurs articles, mais chaque article est lié à un seul utilisateur.

Le modèle de données comprend deux tables principales :

1. **Utilisateur** : elle contient les informations des utilisateurs comme leur nom, nom d'utilisateur, mot de passe et date de naissance.
2. **Forum** : elle contient les articles du forum, avec un titre, un contenu, une date de publication et la référence à l'utilisateur qui a écrit l'article.

Le projet utilise des clés primaires et des clés étrangères pour garantir que les données restent cohérentes. Il est conçu pour gérer simplement les interactions entre utilisateurs et articles.

Ce projet peut servir de base pour un site web de forum où les utilisateurs ajoutent des articles, lisent ceux des autres et interagissent avec la communauté.

Modèle Physique de Données (MPD) :

```
1 • CREATE DATABASE IF NOT EXISTS ForumMaisonneuve;
2 • USE ForumMaisonneuve;
3
4 • CREATE TABLE User (
5     user_id INT AUTO_INCREMENT PRIMARY KEY,
6     full_name VARCHAR(45) NOT NULL,
7     username VARCHAR(45) NOT NULL UNIQUE,
8     password VARCHAR(255) NOT NULL,
9     birth_date DATE
10 );
11
12
13 • CREATE TABLE Forum (
14     forum_id INT AUTO_INCREMENT PRIMARY KEY,
15     title VARCHAR(45) NOT NULL,
16     content TEXT NOT NULL,
17     dateOfCreation DATETIME DEFAULT CURRENT_TIMESTAMP,
18     userForum_id INT,
19     CONSTRAINT fk_userForum FOREIGN KEY (userForum_id) REFERENCES User(user_id)
20 );
21
```

Rapport sur le travail réalisé :

Introduction :

Ce travail consiste à développer une application web basée sur l'architecture MVC (**Model-View-Controller**) pour gérer un forum en ligne. J'ai travaillé sur plusieurs aspects, notamment la création des vues, la mise en place du style CSS, et la connexion des différentes parties du modèle MVC.

Étapes du travail

1. Création de la vue principale : modification d'un article

J'ai d'abord créé une page HTML dynamique qui permet aux utilisateurs de modifier un article existant.

Structure de la page

- Une section <head> a été ajoutée pour définir les métadonnées, le titre de la page ("Forum MVC") et le lien vers un fichier CSS pour le style.
- Une barre de navigation a été créée avec des liens vers les différentes parties de l'application, tels que l'accueil, la création d'un utilisateur, et la connexion. Des liens supplémentaires, comme "Créer un article" et "Déconnexion", sont affichés uniquement si l'utilisateur est connecté.
- Dans la section principale, un formulaire a été intégré pour permettre la modification d'un article. Ce formulaire est pré-rempli avec les données existantes de l'article.

- Une gestion des erreurs a été mise en place pour afficher un message si l'article n'existe pas.
- Un pied de page contenant une mention de copyright a été ajouté.

Dynamisme avec PHP:

- Les liens de navigation et le formulaire utilisent des conditions PHP pour afficher ou masquer certains éléments en fonction de la situation.

2. Ajout des styles CSS :

Les éléments de la page ont été stylisés pour être clairs, attrayants et faciles à utiliser.

Titres et navigation:

- Les titres sont centrés et colorés pour être bien visibles.
- La barre de navigation est bien espacée, et les liens changent de style lorsqu'on les survole.

Tableaux:

- Les tableaux affichent la liste des articles avec un style professionnel :
 - Les en-têtes des colonnes ont un fond coloré.
 - Les lignes alternent avec des couleurs pour une meilleure lisibilité.
 - Les lignes changent légèrement de couleur lorsqu'on les survole.

Formulaires:

- Les champs de texte et les boutons du formulaire ont des bordures arrondies et des couleurs adaptées.
- Lorsqu'on interagit avec les champs, un effet visuel est appliqué pour indiquer qu'ils sont sélectionnés.

Pied de page

- Le pied de page a un fond coloré et du texte centré.

3. Compréhension et mise en œuvre du modèle MVC :

Pour connecter toutes les parties de l'application, j'ai suivi les principes du modèle MVC.

Le modèle (Model)

- Il est responsable de la gestion des données et de leur interaction avec la base de données. Par exemple, il récupère les articles existants ou enregistre les modifications apportées.

La vue (View)

- Elle est responsable de l'affichage des données sous forme de pages HTML. Dans ce travail, j'ai créé des vues pour afficher la liste des articles, un formulaire de modification, et des messages d'erreur.

Le contrôleur (Controller)

- Il fait le lien entre le modèle et la vue. Il transmet les données récupérées par le modèle aux vues et gère les actions demandées par l'utilisateur, comme la modification ou la suppression d'un article.

4. Liste des articles

Une table a été créée pour afficher tous les articles du forum. Les colonnes montrent les informations principales, comme le titre de l'article, l'auteur, la date de création, et des actions possibles (voir, modifier ou supprimer).

Conclusion

En suivant les principes du modèle MVC, j'ai organisé mon travail pour séparer la gestion des données, la logique métier et l'affichage. Grâce à CSS, j'ai rendu l'interface utilisateur plus attrayante et fonctionnelle.

Ce travail m'a permis de mieux comprendre :

1. Comment structurer une application MVC.
2. Comment rendre une application dynamique et interactive.
3. Comment organiser le code pour le rendre plus lisible et maintenable.