

FICHE TECHNIQUE

PRÉSENTATION GÉNÉRALE

NOM DE L'APPLICATION: CT-Comments

OBJECTIF DE L'APPLICATION : Permettre aux utilisateurs de commenter n'importe quel post en ligne via son URL pour centralisé les discussions sur une seule plateforme.

LANGAGES UTILISÉS :

- PHP et Laravel pour le Back-end;
- Vue js et tailwind css pour le Front-end;
- MySQL pour gérer la base de données(nom de la base de données : CT-database);
- Serveur : localhost:127.0.0.1;

IDE : Visual Studio Code

ARCHITECTURE TECHNIQUE

- Utilisation d'un modèle MVC grace à Laravel;
- Usage d'une API REST pour la communication entre front-end et back-end;
- Base de données relationnelle possédant des table s pour:
 - . Utilisateurs
 - . URLs de posts
 - . Commentaires
- Usage de Vue js pour une interface dynamique et réactive
- .

PRINCIPALES FONCTIONNALITÉS

- Authentification [Inscription, connexion, mot de passe oublié]
- Ajout d'URL [L'utilisateur colle l'URL d'un post externe]
- Commentaires [Les utilisateurs peuvent commenter chaque URL ajoutée]
- Modération(admin) [Suppression de commentaires, suppression de compte d'utilisateurs]

STRUCTURE DE LA BASE DE DONNÉES

La base de données contient les tables :

- users : (id, name, email, password)
- posts : (id, url, date, user_id)
- comments :(id, user_id, post_id, comment, date)

INTERFACE UTILISATEUR:

- Formulaire d'ajout d'url et de commentaires;
- Liste de commentaire;
- Filtrage (possibilité de n'afficher que les commentaires postés par l'utilisateur;
- Design responsive s'adaptant au appareil mobile et aux desktop;

SÉCURITÉ

- Hashage du mot de passe avec bcrypt;
- Protection CSRF avec laravel pour éviter les injections;

API

Routes:

*Pour les posts:

- ~ api/posts (affichage de tous les posts)
- ~ api/posts/create (ajouter un post)
- ~ api/posts/edit/{posts} (mise a jour de post)
- ~ api/posts/{posts} (effacer un post)

*Pour les commentaires:

- ~ api/comments (affichage de tous les commentaires)
- ~ api/comments/add (ajouter un commentaire)
- ~ api/comments/edit/{id} (mise a jour de commentaires)
- ~ api/comments/{id} (effacer un commentaire)

*Pour les utilisateurs:

- ~ api/register (inscription)
- ~ api/login (connection)
- ~ api/users (liste des utilisateurs)
- ~ api/user/edit/{id} (modification)
- ~ api/user/{id} (supprimer un utilisateur)

Format : JSON

•