

TP FINAL

Plateforme de Photos similaire à Instagram

Remise

- Sur Léa (code de l'application en fichier zip avec un fichier readme dans lequel vous spécifier le lien vers une vidéo illustrant les fonctionnalités de votre TP.
- **Date limite: 12 septembre 2023 avant minuit.**

1. Objectif

Développer une application web simplifiée permettant aux visiteurs d'uploader des photos, de les visualiser, d'y ajouter des commentaires et de rechercher des images par mots-clés.

2. Modélisation des données :

- **Photo** : Informations relatives à la photo comme la date de publication, la description, le chemin du fichier, et les mots-clés associés.
- **Commentaire** : ID de la photo associée, contenu du commentaire, date de publication.

3. Routes et Contrôleurs :

- **Photo** :
 - **Upload** : Offrir la possibilité d'ajouter une nouvelle photo et de l'associer à des mots-clés.
 - **Affichage** : Permettre la visualisation d'une photo individuelle.
 - **Mise à jour** : Donner la capacité de modifier les informations ou le contenu d'une photo déjà uploadée.
 - **Suppression** : Proposer une option pour retirer une photo de la plateforme.
 - **Recherche** : Intégrer une fonction de recherche pour retrouver des photos par mots-clés.
- **Commentaire** :
 - **Ajout** : Ajouter un commentaire sous une photo spécifique.
 - **Suppression** : Supprimer un commentaire existant.

4. Vues (Pug Templates) :

- **Photo :**
 - **Page d'upload** : Un formulaire permettant d'ajouter une nouvelle photo et de l'associer à des mots-clés.
 - **Page de galerie** : Un affichage des photos uploadées par les visiteurs, avec une barre de recherche pour retrouver des images par mots-clés.
 - **Page de détail** : Une visualisation détaillée de chaque photo.
- **Commentaire :**
 - **Section de commentaires** : Sous chaque photo, une section dédiée à l'affichage, l'ajout, et la suppression des commentaires.

5. Technologies et Middlewares recommandés

- **Backend** : Express.
- **Templating** : Pug.
- **Gestion des fichiers** : Utilisez `multer` pour gérer l'upload des photos.
- **Base de données** : MongoDB avec `mongoose` pour faciliter l'interaction et la modélisation des données.
- **Logging** : Utilisez `morgan` pour le journal des requêtes HTTP.
- **Gestion des erreurs** : Employez `errorhandler` pour une gestion efficace des erreurs pendant le développement.
- **Interactions client-serveur** : Intégrez `axios` pour faciliter les requêtes HTTP côté client qui pourrait être par exemple pour supprimer un commentaire sans avoir à recharger toute la page.

6. Les routes et les méthodes http associée

6.1. Routes pour les photos avec les méthodes http (POST, GET, PUT, DELETE)

- POST /photos/upload**
 - **Description** : Route pour téléverser une nouvelle photo.
- GET /photos/:photoId**
 - **Description** : Route pour visualiser une photo spécifique.
 - **Données requises** : ID de la photo dans l'URL :**photoId**.
- PUT /photos/:photoId**
 - **Description** : Route pour mettre à jour les informations d'une photo.
 - **Données requises** : ID de la photo dans l'URL.
- DELETE /photos/:photoId**
 - **Description** : Route pour supprimer une photo spécifique.
 - **Données requises** : ID de la photo dans l'URL.
- GET /photos/search**
 - **Description** : Route pour rechercher des photos par mots-clés.

- **Données envoyées** : Mots-clés via des paramètres de requête (query parameters).
- **Exemple** : Pour la route /photos/search, l'utilisateur pourrait envoyer une requête telle que **/photos/search?keyword=emploi** pour rechercher toutes les photos associées au mot-clé **"emploi"**.

6.2. Routes pour les commentaires

- f) **POST** /photos/:photoId/comments
 - **Description** : Route pour ajouter un commentaire à une photo spécifique.
 - **Données requises** : ID de la photo dans l'URL.
- g) **DELETE** /comments/:commentId
 - **Description** : Route pour supprimer un commentaire spécifique.

7. Questions

1. Créer des maquettes d'écran à l'aide d'un outil graphique tel que draw.io pour obtenir une prévisualisation des interfaces graphiques avant de les implémenter en JavaScript (**5 points**).
2. Créer la structure de vos dossiers et fichiers nécessaires en concevant votre application web en suivant une architecture MVC avancée (**5 points**).
3. Créer la base de données MongoDB Atlas nécessaire pour le stockage des informations requises à cette application web tel que décrites en haut (i.e., stockages des informations relatives aux photos et leurs commentaires) (**10 points**).
4. Implémenter une application Express, les routes, les contrôleurs, les requêtes et les vues nécessaires tel que décrit dans le point 6 (**70 points**).

Note : Chaque point dans 6. Est noté sur 10 points. Exemple : 6.1.a vaut 10 points.

5. Documenter votre application avec un fichier Readme et élaborer une vidéo illustrative de votre travail qui ne dépasse pas 5 minutes. Vous pouvez utiliser MTeams et partager le lien dans votre fichier Readme (**10 points**).