

Projet Web – Timeline Photo

Table des matières :

- Introduction
- Approche mise en place
- Rôles au sein de l'équipe
- Problématiques rencontrées & solutions apportées
- Conclusion

Introduction

Dans le cadre d'un projet web, nous avons été amenés à réfléchir à une nouvelle présentation d'une timeline photo d'évènements de l'ENSIIE, l'objectif étant de permettre à chaque utilisateur d'accéder rapidement et simplement aux différentes photos. Nous avons tenté d'améliorer les fonctionnalités de l'application déjà présente sur le portail iiens.net.

En effet celle-ci présente les photos de manière très basique, sous forme de dossiers. Nous apportons donc quelques fonctionnalités notamment visuelles pour améliorer et fluidifier la navigation.

Approche mise en place

Avant de commencer quoi que ce soit, il était très important de décider dans quelle direction nous allions nous diriger. C'est pourquoi nous avons en premier lieu discuté des nouvelles fonctionnalités qui devraient être présentes sur l'application.

Littéralement, une "timeline" est une ligne du temps. Il nous est ainsi naturellement venue l'idée de représenter les événements chronologiquement et de manière verticale. Pour organiser la navigation, nous avons décidé d'adopter un regroupement par "périodes" des photos. La timeline permet d'accéder à ces périodes sous forme de blocs. On pourrait ainsi accéder aux photos d'une période en cliquant dessus. Cette méthode permet de gagner du temps en termes de recherche de photo.

Avant de commencer à coder, nous avons listé l'ensemble des pages du site:

- Une page de connexion
- Une page menu de sélection d'évènement
- Une page de timeline photo
- Une page permettant de changer sa photo de profil
- Une page de changement de mot de passe

- Une page de proposition d'archive photo correspondant à un évènement
- Une page administrateur permettant la validation des propositions d'archives

Plusieurs modules sont alors ressortis de cette distribution, qui se reflètent dans l'attribution des rôles:

- La base de données et la gestion de connexion et de mots de passe
- Le téléchargement d'archives
- La timeline photo en elle-même

Puisque nous travaillons en équipe, nous avons opté pour le service Cloud9, nous permettant d'être connectés simultanément sur le projet.

Materialize a été utilisé pour le css des pages, à l'exception de la page de la timeline. On retrouve deux univers sur le site: la connexion/accès à la timeline avec des couleurs orangées, et la timeline photo basée sur le thème de l'espace.

Rôles au sein de l'équipe

Nous avons décidé de scinder le groupe en deux parties pour être plus efficaces, puisqu'on distingue deux grandes parties que sont la timeline, et la base de données et connexion/gestion. Voici les rôles ainsi attribués :

- Pierre-Olivier Gendraud et Mehdi Khadir : Connexion au site, modification du mot de passe, modification de la photo de profil, base de données, architecture des pages et design hors timeline photo.
- Qiu hao Qian et Damien Marchand : Création de la timeline photo et implémentation de la page relative au téléchargement d'archive.

Problématiques rencontrées & solutions apportées

1) Base de données

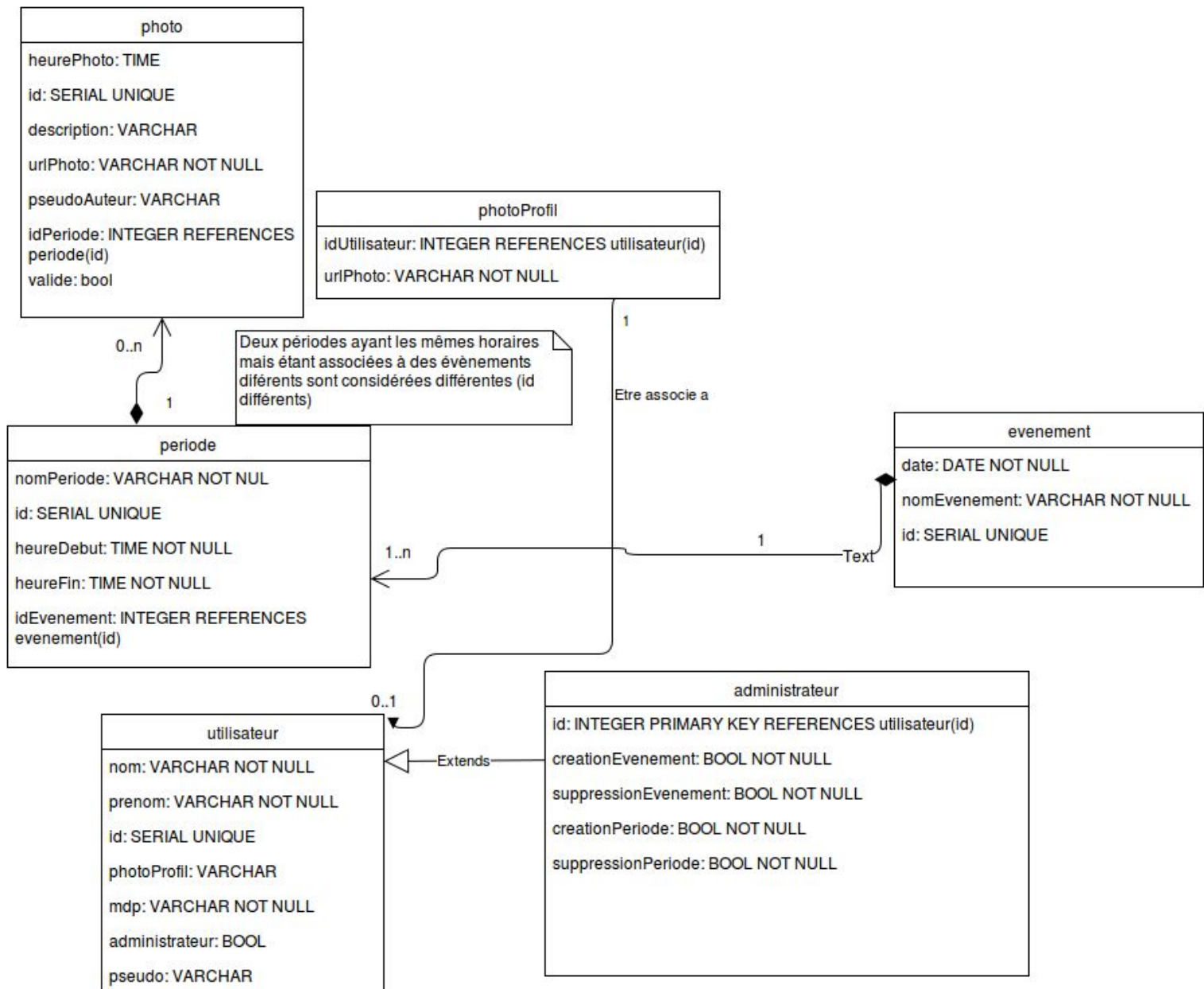
La base de données a été créée avant de commencer à coder. Elle se doit de refléter l'organisation du site.

Puisque la timeline photo est composée de photos, rangées dans des périodes elles-mêmes associés à un évènement, nous avons opté pour des compositions, traduites par des clés étrangères dans le modèle relationnel.

Pour que les utilisateurs puissent proposer des photos et modifier leur photo de profil, voire accepter des photos si ces utilisateurs sont des administrateurs, nous avons ajouté une classe administrateur héritant de la classe utilisateur.

Cloud9 ne proposant pas phppgadmin, toutes les tables ont dû être créées et mises à jour directement en console.

Le modèle conceptuel est résumé dans le schéma UML suivant :



2) Page d'accueil

La page d'accueil se veut la plus épurée possible, car l'utilisateur doit avant tout se connecter pour accéder aux événements et aux photos. C'est pourquoi cette page d'accueil ne contient principalement qu'un formulaire de connexion classique. Les actions de déplacement sur le site s'effectueront ensuite dans le menu des événements. Le menu des événements n'est affiché que si l'utilisateur s'est bien connecté.

On utilise `pg_escape_string` pour sécuriser nos requêtes SQL.

L'effet flottant du formulaire de connexion a été réalisé avec Parallax de Materialize.

3) Menu des événements

Cette page a pour but de permettre à l'utilisateur d'accéder à un événement qu'il choisit parmi une liste. Puisque notre site présente des photos de soirée, il fallait que le tout soit visuel. C'est pourquoi nous avons utilisé un avatar contente de Materialize, nous permettant d'avoir une photo représentative pour chaque événement.

Cette photo est obtenue en effectuant une double jointure. En effet, chaque photo est liée à une période qui est elle-même liée à un événement. Nous sélectionnons donc pour chaque événement les url des photos associées en liant ces trois tables. La photo affichée dans la liste est alors la première photo du résultat de la requête.

Nous retrouvons aussi dans la barre de navigation un menu déroulant utilisant Ajax. Il permet d'accéder aux différentes pages concernant l'utilisateur. Ce menu contient en plus les boutons d'administration si la personne connectée est un administrateur.

Ce menu déroulant est représenté par la photo de profil de l'utilisateur, récupérée en effectuant un `SELECT` dans la table utilisateur permettant de récupérer l'id de l'utilisateur actuellement connecté sous le nom "pseudo". On effectue alors un second `SELECT` dans la table photoprofil grâce à cet id pour obtenir l'URL de la photo de profil.

4) Changement de photo de profil

Les utilisateurs de l'application peuvent posséder une photo de profil, qui s'affichera dans la barre de navigation orange en haut de la page de sélection d'évènement.

La page de changement de photo de profil accessible via le menu déroulant en haut à droite de la page de menu permet de télécharger une nouvelle photo de profil.

Le principal risque était que le fichier téléchargé ne soit pas une photo. Nous avons effectué un contrôle sur sa nature dans la page `savePictureProfile.php` grâce à la fonction `getimagesize` qui renvoie 0 si le fichier téléchargé est vide ou qu'il ne correspond pas à une photo. Nous avons aussi défini la taille maximum de téléchargement à 500 ko par fichier pour ces photos de profil.

Automatiquement, le fichier importé est renommé avec l'identifiant de la personne connectée, et remplace la précédente photo de profil sur le serveur si existante.

La fonction Javascript `readURL` permet enfin de présenter un aperçu du fichier téléchargé à l'utilisateur, avant qu'il ne valide la modification.

5) Changement de mot de passe

Chaque utilisateur peut changer son mot de passe grâce à cette page, accessible depuis le menu déroulant de la barre de navigation.

Elle présente un formulaire demandant à l'utilisateur son ancien mot de passe, et deux fois son nouveau pour éviter une potentielle faute de frappe. La page de traitement des entrées du formulaire vérifie alors que l'ancien mot de passe de l'utilisateur est bien celui correspondant à la personne connectée en effectuant une requête `SELECT` du mot de passe dans la table `utilisateur`. Elle s'assure ensuite du fait que les deux nouveaux mots de passe soient les mêmes.

On utilise là encore `pg_escape_string` pour protéger la requête SQL.

Si tout est bon, la modification a lieu, sinon un message d'erreur est affiché.

6) Téléchargement d'archives

Dans cette partie l'objectif est d'upload un dossier zip contenant des photos, dans le but de compléter un évènement à partir de ces dernières.

Pour l'upload dans un premier temps nous avons souhaité ranger les photos grâce à leurs date de création, mais lors du téléchargement la date est modifiée par le serveur. Nous avons donc décidé que le dossier zip doit avoir une arborescence spécifique: il doit contenir des sous dossiers correspondant aux périodes de l'évènement.

De plus lors de l'upload on stocke une version plus petite des images pour permettre de réduire les temps de chargement de la page de la timeline.

7) Timeline photo

Cette page a pour but de montrer toutes les photos de l'évènement sélectionné par l'utilisateur dans la page "Menu". Elle affiche son pseudo en titre, possède deux container principaux disposés à gauche et à droite.

Le container de gauche présente le nom de l'évènement et les blocs de photos ordonnés sous forme de ligne du temps. Chacun de ces blocs représente une période et permet d'accéder aux photos associées en cliquant dessus. On peut ensuite faire défiler les photos grâce à une fancy box. L'heure de début et de fin sont indiquées en plus du nom. Pour obtenir ces périodes, on effectue une requête dans la table "periode" dont l'id correspond à l'évènement traité. Et pour chaque tuple, on observe l'heure et le nom.

On a besoin d'obtenir l'adresse (catégorie et nom) des photos pour les afficher. Pour se faire, on effectue une requête du type SELECT dans la table "photo" de l'élément dont l'id correspond à l'évènement traité.

Dans le conteneur de droite apparaissent quelques informations statistiques sur le nombre de périodes et de photos présentes dans l'évènement. On y retrouve aussi le profil de l'utilisateur et le nom de l'évènement.

Dans la barre de navigation en haut de la page, on retrouve les boutons permettant de revenir au menu et de se déconnecter.

Cette page a été réalisée sur le thème de la galaxie. Les blocs brillent lorsque la souris passe dessus. Pour réaliser cela, nous avons utilisé ":hover selector" dans le fichier CSS pour définir une nouvelle bordure des blocs de périodes quand la souris passe dessus.

De plus, elle a été ajustée à la dimension du navigateur: quand la taille est inférieure à 767px, la timeline change de forme pour ne montrer les blocs que d'un

côté de la ligne. On utilise @media dans le fichier CSS pour évaluer la largeur de l'écran de l'utilisateur.

8) Validation de photos

Dans cette partie un administrateur peut valider les photos proposées via des cases à cocher. Les photos qu'il a refusé de valider sont effacées.

Le booléen "valide" de photo est utilisé ici. Il passe à true lorsque les photos sont acceptées. Elles sont alors prêtes à être présentées sur la timeline.

Conclusion

Ce projet nous a avant tout permis de progresser en web, et de travailler en communauté. Nous venons tous d'horizons différents, avons des méthodes de travail différentes, nous avons donc appris à distribuer les tâches, à écouter les autres et à avancer ensemble.

Nous espérons avoir contribué au projet communautaire qu'est iiens.net, et voir pourquoi pas un jour notre timeline s'afficher sur ce portail !