

Projet Développement Web : Gestion d'évènements de festivals et concerts



Réalisé par :
Meriam SLIMANE
Abdelhalim ZERARA

30/06/2024

Table des matières

I.	Introduction.....	3
II.	URL Github.....	3
III.	Diagramme de la base de données	3
	3	
IV.	Choix techniques.....	4
	5	
V.	Avis sur le cours.....	6
VI.	Ressources.....	6

I. Introduction

Notre projet vise à développer une page web dédiée à l'organisation d'événements. À l'approche de l'été, période propice aux rassemblements festifs, nous avons choisi de nous concentrer sur le thème des festivals et des concerts.

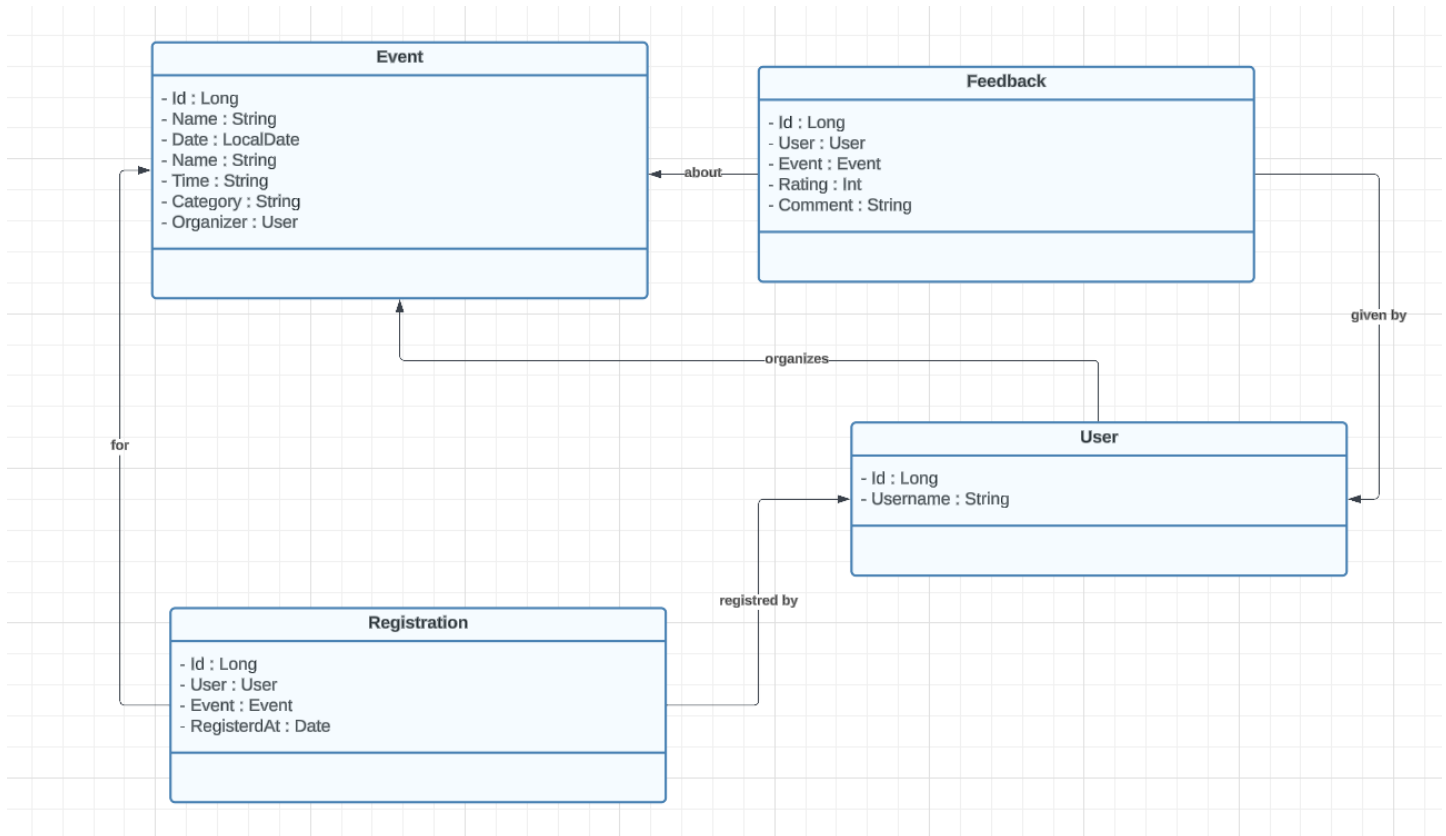
Pour ce faire, nous avons employé la pile technologique étudiée dans le cadre de notre cours, incluant Spring Boot pour le backend, Angular pour le frontend, et PostgreSQL pour la gestion des données.

II. URL Github

- **URL du backend :**
<https://github.com/Meriam27/Event-Manager/tree/master/backend>
- **URL du Frontend :**
<https://github.com/Meriam27/Event-Manager/tree/master/frontend>

III. Diagramme de la base de données

https://lucid.app/lucidchart/bcca625e-47ba-4275-9205-3bac715b8663/edit?invitationId=inv_175680c5-a5e9-42c9-b1ec-4ec2f5241d35&page=0_0#



Ce diagramme représente le modèle de données pour notre système de gestion d'événements. Il inclut quatre entités principales : Event, User, Registration et Feedback.

Les relations entre les entités sont définies par des liens comme "organizes" entre User et Event, indiquant qu'un utilisateur organise un événement, "registered by" entre User et Registration, montrant que l'utilisateur s'est inscrit à un événement, et "about" entre Event et Feedback, indiquant que le retour concerne un événement spécifique.

IV. Choix techniques

1. Système d'Authentification

- **Fonctionnalité**

Le système d'authentification de l'application permet aux utilisateurs de se connecter en fournissant simplement un nom d'utilisateur. Ce système ne nécessite pas de mot de passe, simplifiant ainsi le processus d'authentification tout en garantissant que seuls les utilisateurs valides peuvent accéder à certaines fonctionnalités de l'application.

- **Processus de Connexion**

L'utilisateur saisit son nom d'utilisateur sur la page de connexion.

Lors de la soumission du formulaire de connexion, une requête est envoyée au backend.

Si l'utilisateur existe déjà dans la base de données, les informations de l'utilisateur sont renvoyées en réponse.

Si l'utilisateur n'existe pas, un nouveau compte utilisateur est créé avec le nom d'utilisateur fourni et les informations de ce nouvel utilisateur sont renvoyées.

Stockage des Informations de l'Utilisateur :

Les informations de l'utilisateur retournées par le backend sont stockées dans le localStorage du navigateur.

Cela permet de maintenir la session de l'utilisateur même après un rafraîchissement de la page ou la fermeture et la réouverture du navigateur.

- **AuthGuard :**

L'AuthGuard est utilisé pour protéger les routes de l'application.

Toutes les routes, à l'exception de la page de connexion, sont protégées par l'AuthGuard.

Si le localStorage ne contient pas d'informations sur l'utilisateur connecté, l'utilisateur est redirigé vers la page de connexion.

2. Page Principale

La page principale affiche des cartes pour chaque événement disponible. Chaque carte contient les éléments suivants :

Bouton S'inscrire : Permet à l'utilisateur de s'inscrire à l'événement. Une fois inscrit, ce bouton se transforme en bouton Quitter permettant à l'utilisateur de se désinscrire.

Bouton Avis : Affiche la page des avis pour l'événement, où l'utilisateur peut consulter et donner son

avis.

Étoiles de Note : Permet à l'utilisateur de donner une note à l'événement. La note moyenne des avis est affichée à côté des étoiles.

3. Gestion des Avis

Sur la page des avis pour un événement, les utilisateurs peuvent :

Consulter les Avis: Voir tous les avis donnés par d'autres utilisateurs pour cet événement.

Ajouter/Mettre à Jour un Avis: Chaque utilisateur peut ajouter un commentaire ou mettre à jour son commentaire existant. Chaque utilisateur ne peut avoir qu'un seul commentaire par événement.

4. Barre de Navigation

La barre de navigation contient les éléments suivants :

Logo: Redirige vers la page principale lorsqu'on clique dessus.

Bouton Déconnecter: Vide le « local Storage » et redirige l'utilisateur vers la page de login, effectuant ainsi la déconnexion.

Bouton Créer un Événement: Redirige vers la page de création d'événements.

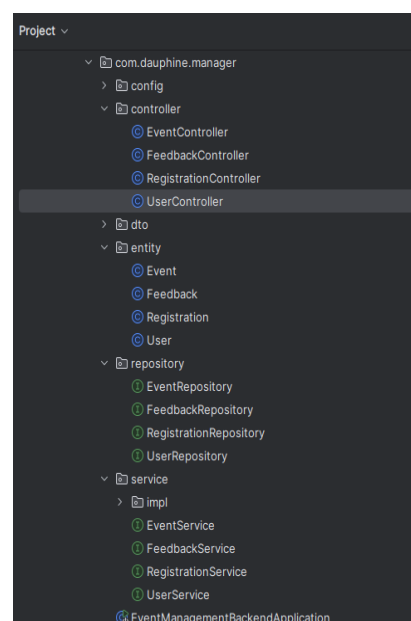
5. Page de Création d'Événement

Cette page permet aux utilisateurs de créer un nouvel événement en remplissant un formulaire. Des notifications SweetAlert sont utilisées pour informer l'utilisateur du succès ou de l'échec de l'envoi du formulaire.

6. Structure des Services

Nous n'avons pas utilisé de service abstrait hérité par d'autres services sur le frontend ou le backend, car chaque service avait des spécificités uniques.

7. Couches et structure du backend



L'architecture du modèle est basée sur le modèle MVC (Modèle-Vue-Contrôleur). Le code est structuré en plusieurs dossiers :

- **controller** : Contient les classes EventController, FeedbackController, RegistrationController, et UserController. Ces contrôleurs gèrent les interactions avec les utilisateurs en recevant les requêtes, en les traitant via les services appropriés, et en renvoyant les réponses.
- **dto (Data Transfer Object)** : Contient les classes Event, Feedback, Registration, et User qui définissent la structure des données échangées entre le client et le serveur.
- **entity** : Contient des classes similaires aux DTOs mais utilisées pour la définition des entités dans la base de données.
- **repository** : Contient les interfaces EventRepository, FeedbackRepository, RegistrationRepository, et UserRepository. Ces interfaces étendent généralement JpaRepository pour fournir des fonctionnalités CRUD (Créer, Lire, Mettre à jour, Supprimer) pour chaque type d'entité.
- **service et impl (implémentation)** : Contient des classes telles que EventService, FeedbackService, RegistrationService, et UserService, ainsi que leurs implémentations. Ces services encapsulent la logique métier de l'application.

V. Avis sur le cours

On tient à vous remercier pour votre engagement tout au long de ce semestre. Le cours était captivant et structuré de manière à favoriser une progression à notre propre rythme, ce qui a facilité l'apprentissage. L'ampleur des délais pour les rendus nous a permis de travailler de manière flexible. On a également apprécié votre gentillesse et votre disponibilité pour répondre à nos questions.

On a apprécié la liberté offerte par le projet, qui n'imposait pas de limites et encourageait l'exploration personnelle.

Cependant, on pense que le contenu du cours pourrait être mieux adapté aux profils des étudiants qui ne sont pas spécialisés en développement web. En effet, certains aspects du cours ont pu s'avérer complexes pour des personnes n'ayant jamais pratiqué le développement web auparavant.

VI. Ressources

<https://github.com/adisreyai/angular-star-rating/blob/master/projects/star-rating/src/lib/star-rating.component.ts>