**Compte Rendu du Projet : Application de Gestion d'Événements**

**Introduction**

L'application de gestion d'événements a été développée pour permettre aux utilisateurs de créer, gérer et suivre des événements en toute simplicité. Ce projet utilise **Spring Boot** pour le backend et **Angular** pour le frontend, avec une base de données **SQL Server**.

**Objectifs du Projet**

* Faciliter la création et la gestion des événements.
* Permettre aux utilisateurs de s'inscrire aux événements.
* Offrir une interface utilisateur intuitive et réactive.
* Assurer une communication fluide entre le frontend et le backend.

**Technologies Utilisées**

* **Frontend** : Angular
* **Backend** : Spring Boot
* **Base de données** : SQL Server
* **Outils de développement** : VS Code, XAMPP , phpMyAdmin

**Architecture du Projet**

L'application suit une architecture **RESTful**, où le frontend communique avec le backend via des API REST.

* **Backend (Spring Boot)** :
  + Gestion des événements
  + Connexion à la base de données
* **Frontend (Angular)** :
  + Interface utilisateur dynamique
  + Formulaires d'inscription aux événements
  + Affichage et gestion des événements

**Structure du Code**

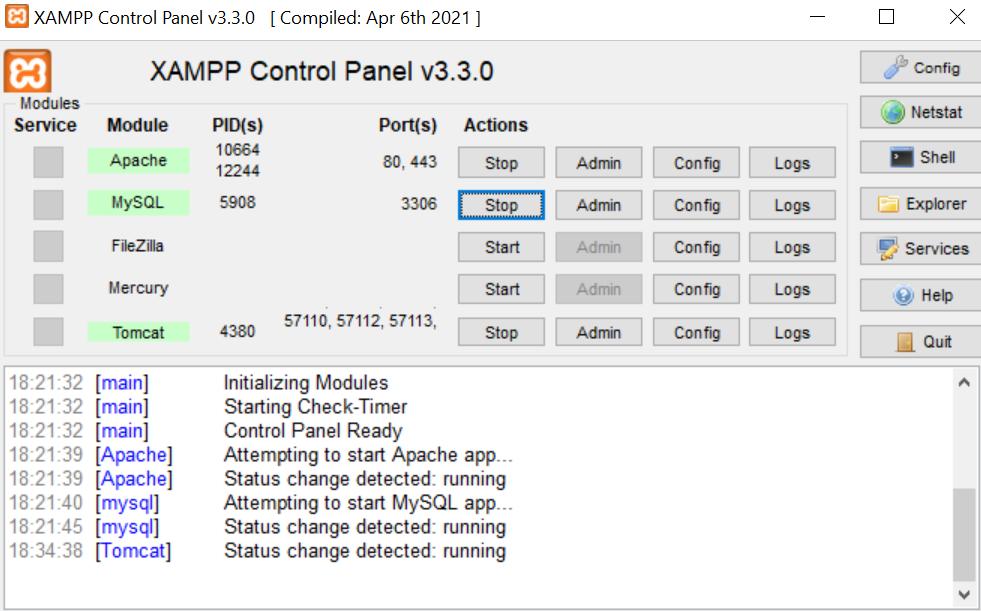
* **Backend (Spring Boot)** :
  + com.project.event\_management
    - controllers/
      * EventController.java : Contient les contrôleurs REST
    - services/
      * EventService.java : Implémente la logique métier
      * IEventService.java : Interface définissant la logique métier
    - repositories/
      * EventRepository.java : Gestion des accès à la base de données
    - entities/
      * Event.java : Définit les entités de l'application
* **Frontend (Angular)** :
* src/app/ → Contient le cœur de l'application
* components/event-list/ → Contient les composants UI pour l'affichage des événements
* services/event-service.service.ts → Service pour appeler l'API et gérer les requêtes HTTP liées aux événements
* models/event.ts → Définit la structure de données pour les événements
* app.module.ts → Configuration des modules Angular et importation des composants et services

**Nouveaux Ajouts et Améliorations**

1. **Bootstrap pour le Design** :
   * Bootstrap a été ajouté pour améliorer l'apparence et la réactivité de l'interface utilisateur. Ce framework CSS facilite la création de mises en page modernes et adaptatives, permettant une expérience utilisateur fluide sur différents appareils.
2. **HttpClient pour les Requêtes HTTP** :
   * L'**HttpClient** a été intégré dans le frontend Angular pour gérer les requêtes HTTP vers le backend. Cela permet une communication efficace entre le client et le serveur, notamment pour récupérer et envoyer les données des événements.
3. **FormModules pour la Gestion des Formulaires** :
   * **ReactiveFormsModule** et **FormsModule** ont été ajoutés pour simplifier la gestion des formulaires. Ces modules permettent une validation en temps réel et une gestion avancée des données de formulaire, notamment pour l'inscription et la création d'événements.
4. **Sécurisation avec CrossOrigin** :
   * Pour garantir la communication entre le frontend Angular et le backend Spring Boot, l'annotation **@CrossOrigin** a été ajoutée dans le backend. Cette annotation permet de spécifier les origines autorisées à accéder aux ressources de l'API

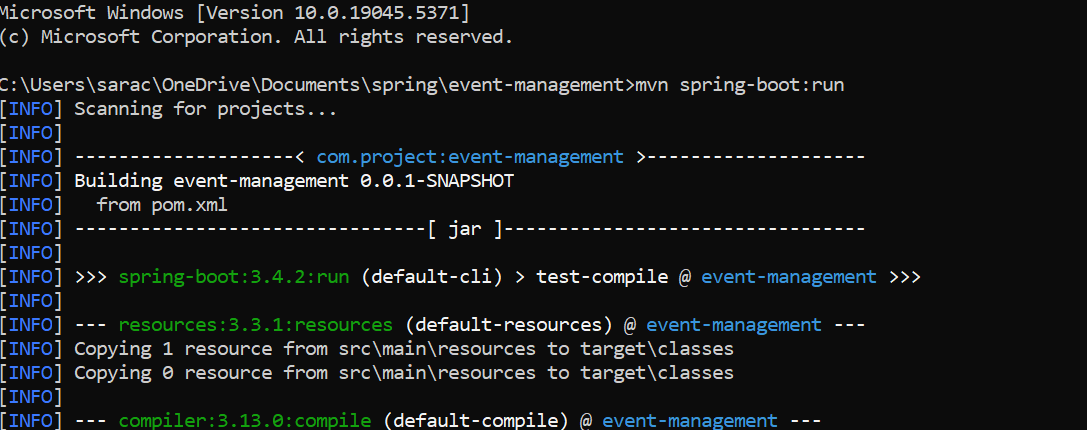
**Exécution du Projet**

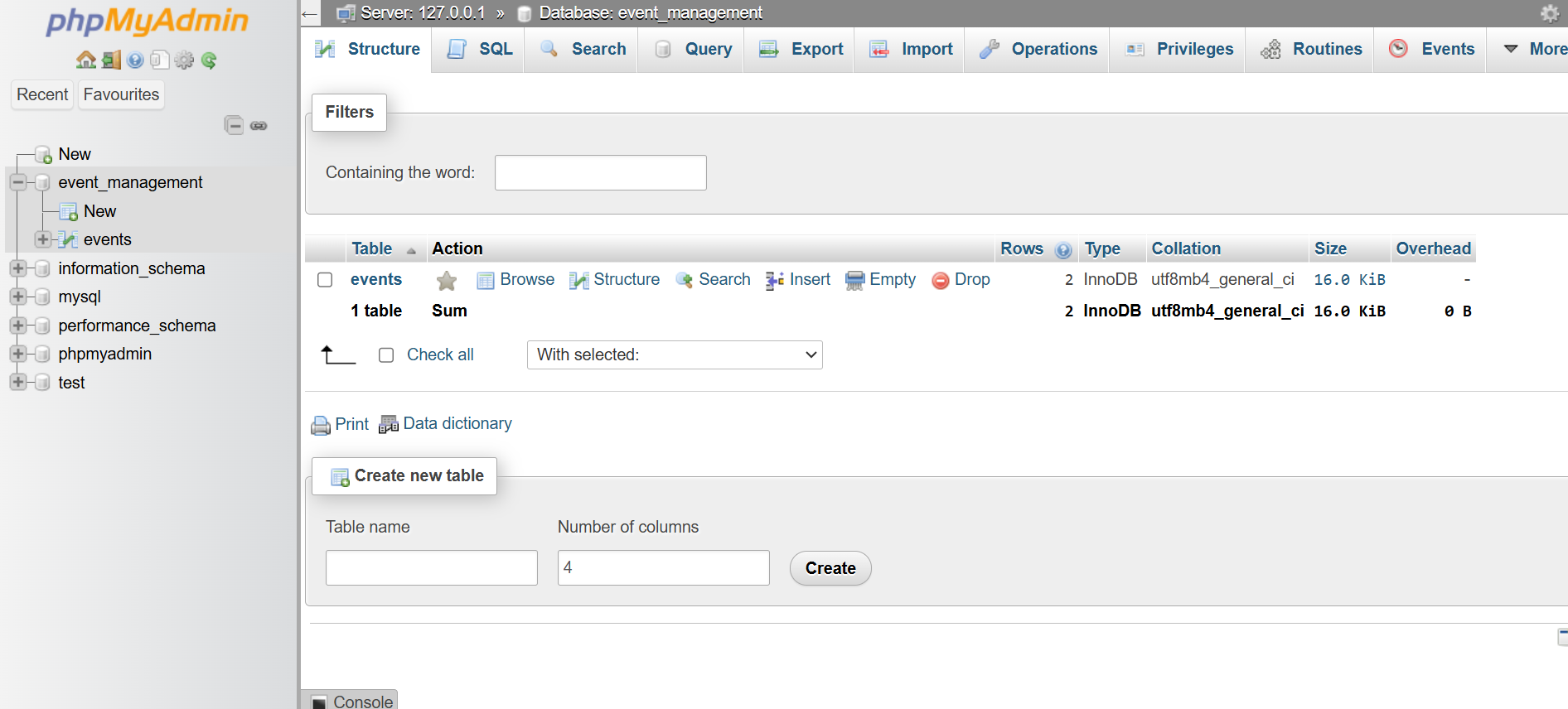
1. **Démarrer la base de données**
   * Ouvrir **XAMPP** et démarrer **Apache** et **MySQL**.

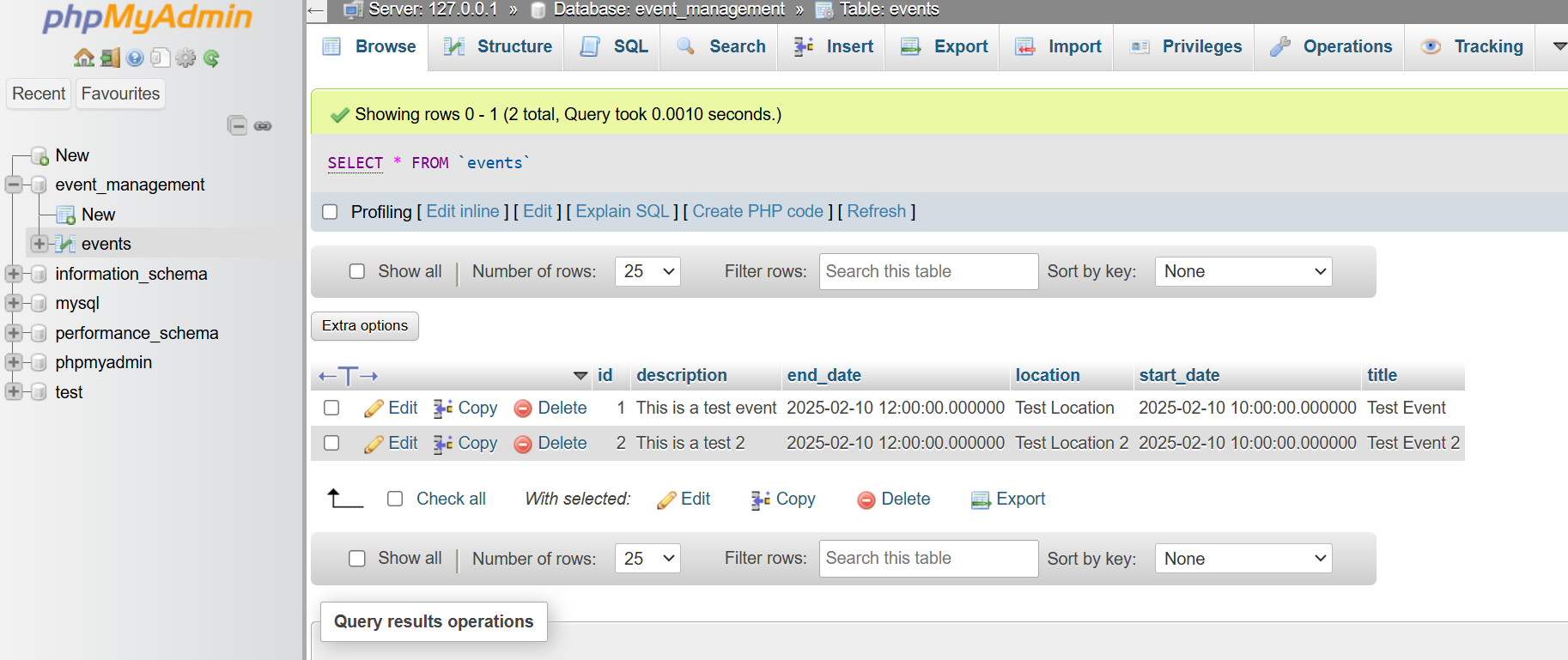


* + Accéder à **phpMyAdmin** via http://localhost/phpmyadmin.
  + Importer la base de données SQL nécessaire.

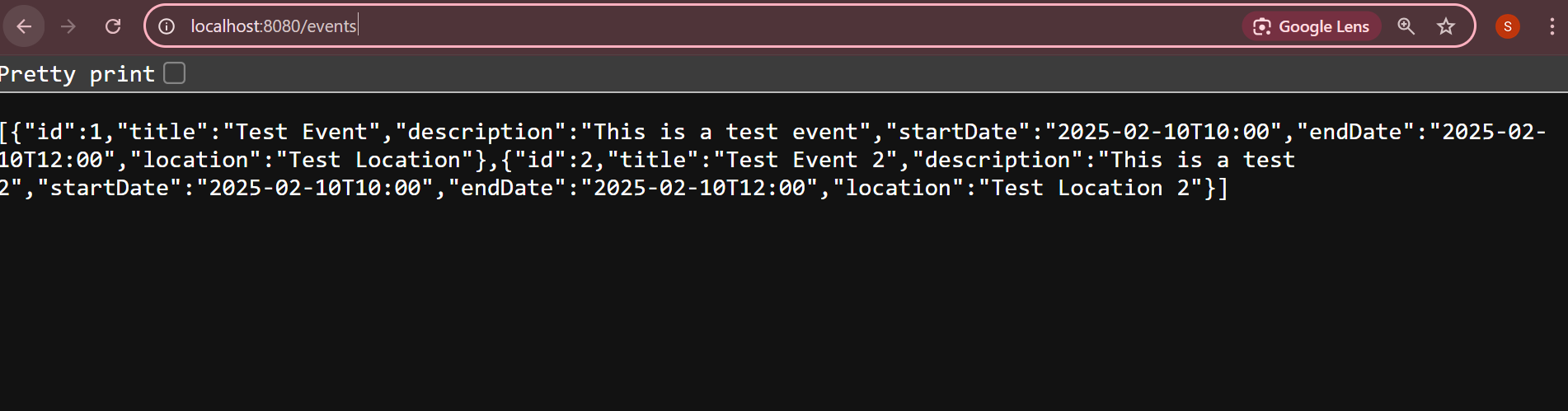
1. **Lancer le backend (Spring Boot)**
   * Ouvrir un terminal dans le dossier du projet backend.
   * Exécuter la commande suivante : mvn spring-boot:run



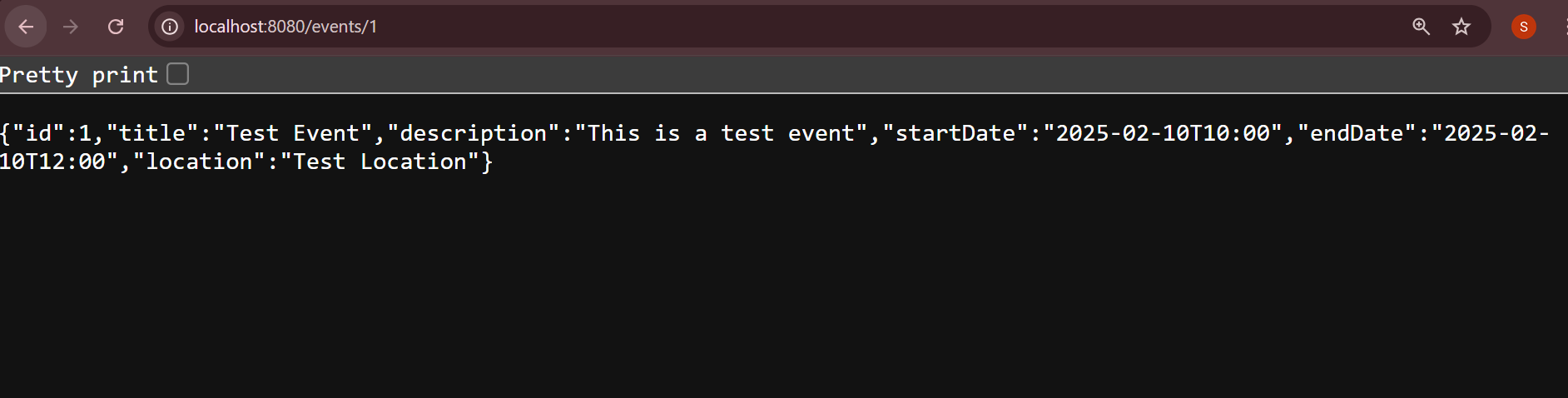




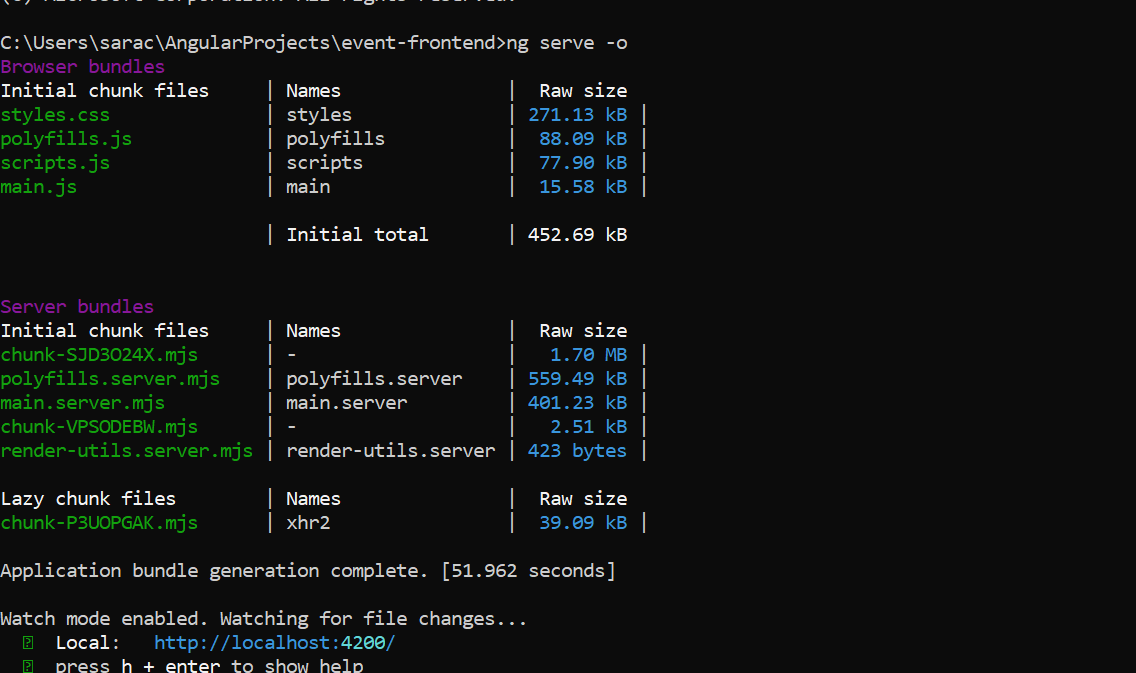
<http://localhost:8080/events>



<http://localhost:8080/events/1>

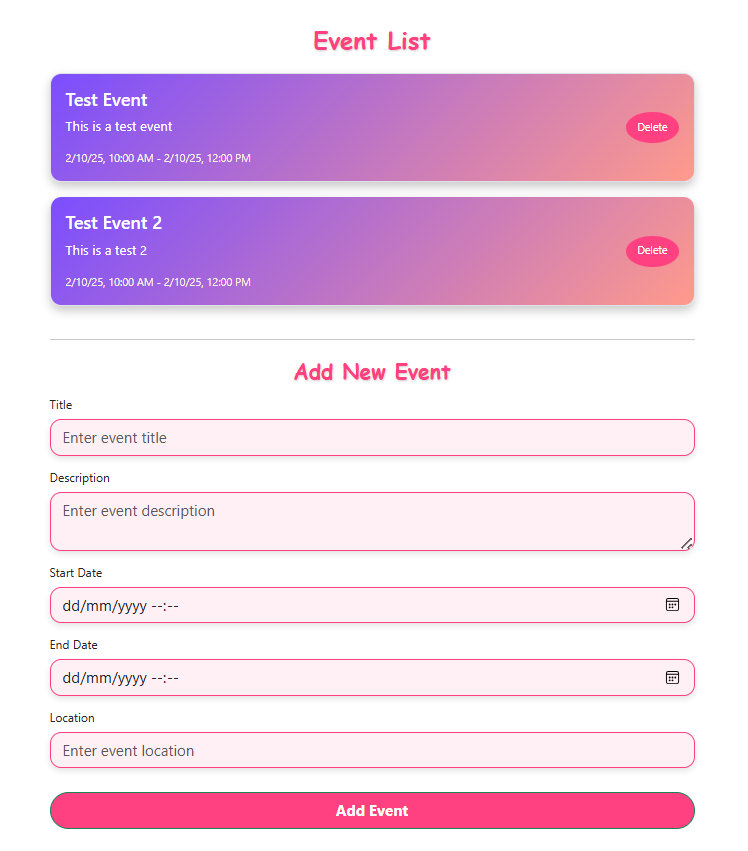


1. **Lancer le frontend (Angular)**
   * Ouvrir un terminal dans le dossier du projet frontend.
   * Exécuter la commande suivante : ng serve --open
   * L'application s'ouvrira automatiquement dans le navigateur sur <http://localhost:4200>.



***L’interface frontend :***

* **Détail d'un événement** : Affiche les informations complètes d'un événement avec possibilité d'inscription.
* **Formulaire de création d'événement** : Permet aux administrateurs d'ajouter de nouveaux événements.



**Conclusion**

Ce projet permet une gestion efficace des événements et offre une expérience utilisateur optimisée. Des améliorations futures incluront l'ajout de notifications et une meilleure gestion des inscriptions.