

Talks Manager

Une application de gestion de conférences et présentations avec persistance des données dans MongoDB.

Fonctionnalités

- 🌓 Thème sombre/clair
- 📝 Gestion complète des talks (CRUD)
- 🔍 Recherche et filtrage
- 📱 Interface responsive
- 💾 Persistance des données avec MongoDB

Technologies Utilisées

Frontend

- React avec TypeScript
- Zustand pour la gestion d'état
- Tailwind CSS pour le styling
- Axios pour les requêtes HTTP

Backend

- Node.js avec Express
- MongoDB avec Mongoose
- CORS pour la communication frontend/backend

Structure du Projet

```
talks-app/  
├── public/  
│   └── images/  
│       └── logo.png  
├── src/  
│   ├── components/  
│   │   ├── App.tsx  
│   │   ├── TalkForm.tsx  
│   │   ├── TalkList.tsx  
│   │   └── ThemeToggle.tsx  
│   ├── store/  
│   │   └── useStore.ts  
│   └── index.tsx  
└── talks-app-api/  
    ├── models/  
    │   └── Talk.js  
    ├── routes/  
    │   └── talks.js  
    ├── .env  
    └── index.js
```

Installation

1. Backend

```
cd talks-app-api  
npm install
```

2. Frontend

```
cd talks-app  
npm install
```

3. Configuration

- Créez un fichier `.env` dans le dossier `talks-app-api` :

```
PORT=5000  
MONGODB_URI=mongodb://localhost:27017/talks-manager
```

Démarrage

1. Démarrer MongoDB

- Assurez-vous que MongoDB est installé et en cours d'exécution

2. Démarrer le Backend

```
cd talks-app-api  
node index.js
```

3. Démarrer le Frontend

```
cd talks-app  
npm start
```

Utilisation

Gestion des Talks

1. Créer un Talk
- Remplissez le formulaire avec :

■ Titre (obligatoire)

■ Présentateur (obligatoire)

■ Date (obligatoire)

■ Description (optionnel)

■ Statut (planifié par défaut)
2. Modifier un Talk
- Cliquez sur le bouton "Modifier"

Modifiez les champs souhaités

Cliquez sur "Enregistrer"
3. Supprimer un Talk
- Cliquez sur le bouton "Supprimer"

Confirmez la suppression

Fonctionnalités Avancées

1. Recherche
- Utilisez la barre de recherche pour filtrer les talks par titre, présentateur ou description
2. Tri
- Triez les talks par :

■ Date

■ Titre

■ Présentateur
3. Filtrage
- Filtrez les talks par statut :

■ Tous

■ Planifié

■ En cours

■ Terminé
4. Thème
- Basculez entre le thème clair et sombre avec le bouton en haut à droite

Structure des Données

Modèle Talk

```
interface Talk {
  _id?: string
  title: string
  speaker: string
  date: string
  description: string
  status: 'planned' | 'in-progress' | 'completed'
  createdAt?: string
  updatedAt?: string
}
```

API Endpoints

- GET /api/talks - Récupérer tous les talks
- GET /api/talks/:id - Récupérer un talk spécifique
- POST /api/talks - Créer un nouveau talk
- PUT /api/talks/:id - Mettre à jour un talk
- DELETE /api/talks/:id - Supprimer un talk

Dépendances

Frontend

```
{
  "dependencies": {
    "axios": "^1.6.7",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "zustand": "^4.5.0",
    "@heroicons/react": "^2.1.1",
    "tailwindcss": "^3.4.0"
  }
}
```

Backend

```
{
  "dependencies": {
    "express": "^4.18.2",
    "mongoose": "^8.0.0",
    "cors": "^2.8.5",
    "dotenv": "^16.3.1"
  }
}
```

Contribution

1. Fork le projet
2. Créez une branche pour votre fonctionnalité (`git checkout -b feature/AmazingFeature`)
3. Committez vos changements (`git commit -m 'Add some AmazingFeature'`)
4. Push vers la branche (`git push origin feature/AmazingFeature`)
5. Ouvrez une Pull Request

Licence

Ce projet est sous licence MIT. Voir le fichier `LICENSE` pour plus de détails.