

# Rapport du Projet REST :

Application Web Cinémas  
Parisiens

Walid MAADI

Florien EPEE

Dines NAGULANATHAN

Soazic FOURNIER

Amine GAHBICHE

## Introduction :

Ce projet consiste en la création d'une application Web orientée services REST pour gérer la programmation et la projection de films dans les cinémas parisiens. L'objectif principal est de fournir des fonctionnalités pour les propriétaires de salles de cinéma, ainsi qu'un accès aux utilisateurs pour consulter les films et les horaires disponibles.


## Description des pages :

L'application "Cinémas Parisiens" offre une expérience conviviale et intuitive à travers plusieurs pages clés. La page d'accueil guide les utilisateurs vers les différentes sections de l'application. Elle propose des options pour rechercher les films à l'affiche dans une ville donnée ou pour accéder à l'espace réservé aux cinémas partenaires. Cette page constitue le point de départ de l'application et présente de manière claire les différentes possibilités offertes.

Cinémas ParisiensFilms à ParisConnexion Cinéma

Bienvenue sur Cinémas Parisiens


Découvrez tous les films à l'affiche dans les salles parisiennes



### Recherche par Ville


Trouvez tous les films à l'affiche dans votre ville en un seul clic.

[Voir les films à Paris](#)



### Programmation Détaillée

Consultez les horaires, les langues et les informations détaillées de chaque film.





### Espace Cinéma


Vous êtes propriétaire d'un cinéma ? Gérez votre programmation en ligne.

[Espace Cinéma](#)

Comment ça marche ?

 **Recherchez par ville**  
Sélectionnez votre ville pour voir tous les films disponibles

 **Consultez les détails**  
Accédez aux informations complètes sur chaque film

 **Vérifiez les horaires**  
Trouvez la séance qui vous convient le mieux

La page "Films à Paris" permet aux utilisateurs de consulter tous les films projetés à Paris. Les informations affichées incluent les titres des films, les horaires de projection et les lieux. En cliquant sur un film, l'utilisateur est redirigé vers une page dédiée qui présente des détails complémentaires. Cette page individuelle pour chaque film affiche des informations telles que la durée, la langue, l'âge minimum requis, le réalisateur et les acteurs principaux. Elle permet aux spectateurs de choisir les films qui les intéressent le plus.

## Films à l'affiche à Paris

**Gladiator**  
Cinéma Test - Paris  
 123 Avenue des Champs-Élysées  
 Du 2025-01-10 au 2025-01-31  
 Séance à 12:00

Voir les détails

**Mufasa : Le Roi Lion**  
Cinéma Test - Paris  
 123 Avenue des Champs-Élysées  
 Du 2025-01-13 au 2025-01-31  
 Séance à 15:00

Voir les détails

Les horaires sont susceptibles d'être modifiés. Veuillez vérifier auprès des cinémas.

### Gladiator

**Durée**  
155 minutes

**Langue**  
Français  
Sous-titres : None

**Réalisateur**  
Ridley Scott

**Acteurs principaux**  
Russel Crowe, Joaquin Phoenix

**Âge minimum requis**  
12 ans

#### Séances disponibles

**Cinéma Test - Paris**  
 123 Avenue des Champs-Élysées  
 Du 2025-01-10 au 2025-01-31  
Jours de projection : Lundi

12:00

[← Retour à la liste des films](#)

Pour les propriétaires de salles de cinéma, une page de connexion est mise à disposition. Cet espace réservé leur permet d'accéder à un tableau de bord où ils peuvent gérer la programmation des films. Une fois connectés, ils ont la possibilité d'ajouter de nouveaux films en renseignant les détails tels que le titre, la durée, la langue, les sous-titres, le réalisateur, les acteurs principaux et l'âge minimum requis. Ils peuvent également programmer les horaires de projection, y compris les dates de début et de fin ainsi que les jours et heures précis.

### Connexion Espace Cinéma


Email

test@cinema.com

Mot de passe

.....

Se connecter

 Cet espace est réservé aux cinémas partenaires.

## Dashboard - Cinéma Test - Paris

#### Ajouter un nouveau film

Titre du film

Durée (minutes)

Langue

Sous-titres

Réalisateur

Acteurs principaux

Âge minimum requis

#### Programmation

Date de début

jj/mm/aaaa

Date de fin

jj/mm/aaaa

Jours de projection

Lundi

Mardi

Mercredi

Jeudi

Heure de projection

--:--

Maintenez Ctrl pour sélectionner plusieurs jours

Ajouter le film

#### Films programmés

Gladiator

Du 10/01/2025 au 31/01/2025

Séance à 12:00

Mufasa : Le Roi Lion

Du 13/01/2025 au 31/01/2025

Séance à 15:00

Le tableau de bord affiche également une liste des films déjà programmés dans leur cinéma, offrant une vue d'ensemble pratique de leur programmation actuelle. Enfin, l'application est également accompagnée d'un rendu console qui permet de suivre les requêtes HTTP effectuées, offrant une transparence sur les interactions des utilisateurs avec le système.

```
(venv) C:\Users\dines\OneDrive\Bureau\projet_rest-main>python run.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 131-954-758
127.0.0.1 - - [14/Jan/2025 23:14:28] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2025 23:14:29] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [14/Jan/2025 23:15:25] "GET /movies/Paris HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2025 23:16:06] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2025 23:16:07] "GET /movies/Paris HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2025 23:16:09] "GET /movie/1 HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2025 23:16:38] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2025 23:16:40] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2025 23:17:33] "POST /login HTTP/1.1" 302 -
127.0.0.1 - - [14/Jan/2025 23:17:33] "GET /cinema/dashboard HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2025 23:19:46] "GET /cinema/dashboard HTTP/1.1" 200 -
127.0.0.1 - - [14/Jan/2025 23:19:53] "GET / HTTP/1.1" 200 -
```

## Structure du Projet

L'architecture du projet est organisée comme suit :

- **app/** : Package principal de l'application contenant l'ensemble des fichiers et modules nécessaires pour la logique backend et l'interaction avec le frontend.
  - **init.py** : Ce fichier initialise l'application Flask et configure ses extensions telles que SQLAlchemy pour la base de données, et Flask-Login pour la gestion des sessions.
  - **models.py** : Contient les modèles de données ORM (Object-Relational Mapping) définissant les tables et leurs relations dans la base de données.
  - **routes.py** : Définit les routes et points d'entrée API pour les différents services REST, ainsi que les réponses associées.
  - **templates/** : Regroupe les fichiers HTML créés avec le moteur de templates Jinja2, utilisé pour générer dynamiquement les pages du frontend.

- **static/** : Contient les ressources statiques comme les feuilles de style CSS, les fichiers JavaScript et les images.
- **venv/** : Environnement virtuel Python qui isole les dépendances nécessaires à l'application.
- **requirements.txt** : Liste des dépendances Python utilisées par le projet, telles que Flask, SQLAlchemy, et Flask-Login.
- **.env** : Fichier contenant des variables d'environnement sensibles comme les clés secrètes et les configurations de la base de données.

## Composants Principaux

### Backend (Python/Flask)

- **Application Core (écrit dans init.py)** : Ce composant est au cœur de l'application. Il configure Flask en tant que framework principal et initialise les extensions utilisées pour gérer la base de données, les sessions utilisateurs, et la protection des routes. Il inclut également les paramètres globaux tels que le chemin d'accès à la base de données et les clés secrètes.
- **Modèles de Données (models.py)** : Ce fichier définit les modèles utilisés pour structurer les données. Par exemple :
  - **Cinema** : Ce modèle gère les informations des cinémas, notamment leur nom, leur adresse et leurs identifiants pour l'authentification.
  - **Movie** : Il stocke les détails des films tels que le titre, la durée, la langue, les sous-titres, et l'âge minimum requis.
  - **MovieSchedule** : Ce modèle gère la programmation des films, incluant les dates, les horaires, et les jours de projection.
- **Contrôleurs (routes.py)** : Les contrôleurs définissent les API REST utilisées par le frontend. Ils incluent des routes pour afficher les listes de films, récupérer les détails d'un film, ajouter un nouveau film ou

modifier des informations existantes. Ces contrôleurs sont conçus pour gérer les requêtes GET, POST, et DELETE.

## Frontend

- **Templates (Jinja2 + Bootstrap 5)** : Les fichiers templates sont organisés pour assurer une interface utilisateur fluide. Ils incluent :
  - **base.html** : Le fichier de base qui contient la structure HTML commune à toutes les pages, notamment la barre de navigation et le pied de page.
  - **index.html** : La page d'accueil qui dirige les utilisateurs vers les différents services offerts.
  - **cinema\_dashboard.html** : Une interface d'administration permettant aux cinémas de gérer leur programmation de films.
  - **movie\_list.html** : Une liste affichant tous les films disponibles par ville.
  - **movie\_details.html** : Une page détaillant les informations complètes d'un film.
  - **login.html** : Une page permettant aux cinémas de se connecter à leur compte.

## Technologies Utilisées

### Backend

- **Flask (v3.1.0)** : Ce framework léger et modulaire offre des outils pour gérer les requêtes HTTP, la sécurisation des routes, et la gestion des sessions utilisateurs.
- **SQLAlchemy (v3.1.1)** : L'ORM permet de manipuler facilement les données de la base en Python. Il gère les tables, les relations entre entités, et les requêtes complexes.

- **Flask-Login (v0.6.3)** : Cette extension gère l'authentification des utilisateurs, la création et la gestion des sessions, ainsi que la protection des routes sensibles.

## Frontend

- **Bootstrap (v5.3.0)** : Framework CSS moderne pour créer des interfaces réactives et adaptées aux écrans de différentes tailles. Il propose des composants prêts à l'emploi tels que des boutons, des grilles et des formulaires.
- **Font Awesome (v6.0.0)** : Bibliothèque d'icônes vectorielles permettant d'améliorer la présentation visuelle des pages.

## Base de Données

- **SQLite** : Choisie pour sa légèreté et sa facilité d'utilisation, cette base de données est idéale pour un projet de cette taille. Elle ne nécessite pas de configuration serveur et offre des performances élevées pour des requêtes locales.

## Fonctionnalités Implémentées

### Espace Public

L'espace public permet aux utilisateurs non authentifiés de consulter les films disponibles :

- Recherche des films disponibles par ville.
- Affichage des informations détaillées pour chaque film, y compris les horaires et lieux de projection.
- Une interface adaptée à tout type d'appareil grâce à Bootstrap, garantissant une expérience utilisateur optimale.

### Espace Cinéma (Authentifié)

Les cinémas peuvent se connecter à un espace privé pour :

- Gérer leurs films et horaires.

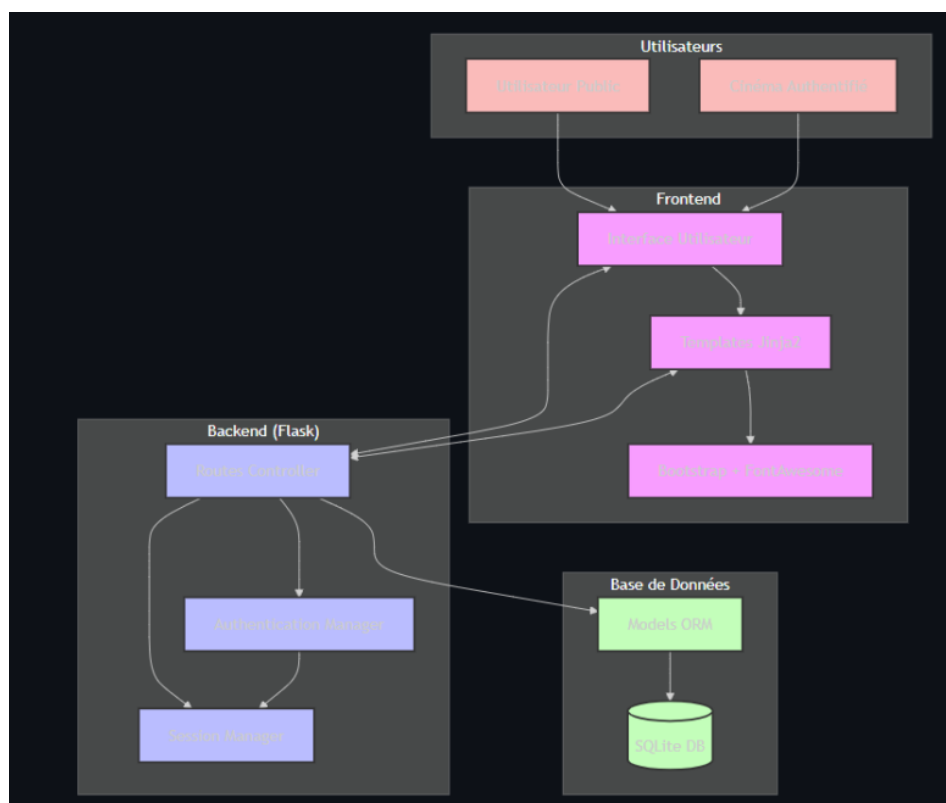


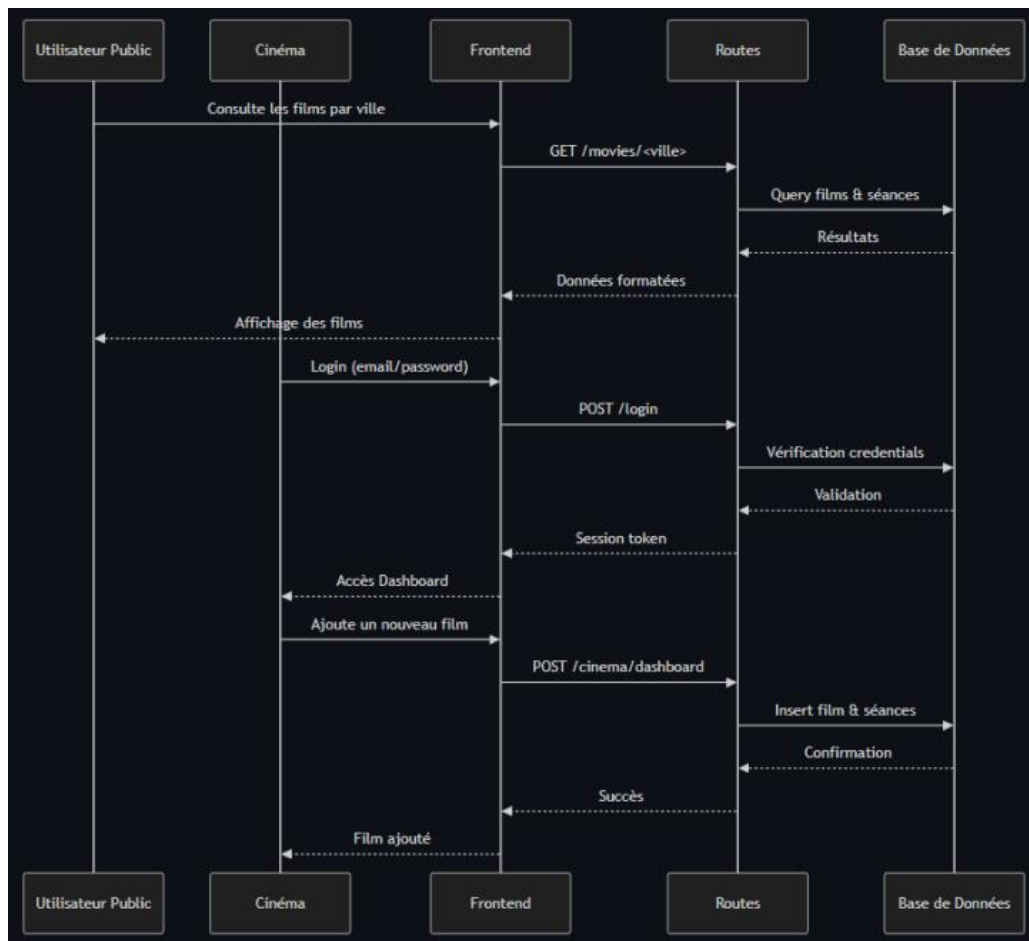
- Ajouter de nouvelles projections et éditer les existantes.
- Utiliser un tableau de bord personnalisé leur donnant une vue d'ensemble de leur programmation.
- Sécurité assurée grâce à un système d'authentification robuste et une gestion des sessions utilisateur.

## Sécurité

L'application intègre plusieurs mécanismes pour garantir la sécurité des utilisateurs et des données :

- **Hachage des mots de passe** : Les mots de passe des utilisateurs sont hachés avec Werkzeug, assurant qu'ils ne soient jamais stockés en clair dans la base de données.
- **Protection CSRF** : Les formulaires incluent des jetons CSRF pour empêcher les attaques de type Cross-Site Request Forgery.
- **Sessions sécurisées** : Les sessions utilisateurs sont gérées de manière à empêcher les accès non autorisés.
- **Variables d'environnement** : Les clés sensibles et les configurations critiques sont stockées dans des fichiers d'environnement et non dans le code source, réduisant les risques de fuite.





## Conclusion

En conclusion, "Cinémas Parisiens" est une application fonctionnelle et sécurisée qui facilite la gestion et la consultation des films pour les utilisateurs publics et les cinémas partenaires. Ce projet, basé sur une architecture REST claire, met en avant des compétences solides en développement web et en sécurité.