

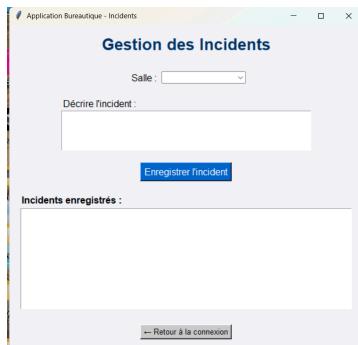


Rapport de projet

Création d'une Plateforme de Cinéma en Ligne :
Cinephoria 2025

ÉVALUATION EN COURS DE FORMATION

Bachelor Développeur 24/25



Abdou MOHAMED

22 juillet 2025

Table de matieres

Cinephoria

Introduction.....	3
Base de données et architecture.....	4
MCD.....	4
Architecture.....	8
Fonctionnement.....	8
Composants clés.....	8
Diagramme de cas d'utilisation - App Web.....	9
Diagrammes de séquences - App Web - Processus de Réservation de Films.....	11
Diagramme de séquences - App Mobile - Consultation des Réservations via Application Mobile.....	13
Base de données.....	15
MySQL.....	15
MongoDB.....	20
Wireframes.....	21
Application Web.....	21
Application Mobile.....	22
Application Bureautique.....	23

Cinephoria - Application Web

Introduction.....	24
Objectifs.....	24
Technologies.....	25
Front-end :.....	25
Back-end :.....	25
Fonctionnalités réalisés.....	27
Interface Utilisateur.....	27
Interface Employé.....	33
Interface Administrateur.....	34
Architecture technique (MVC).....	37
Modèles.....	37
Vues.....	38
Contrôleurs.....	38
Sécurité.....	39
Défis rencontrés et solutions.....	41
Gestion des réservations et des sièges en temps réel.....	41

Sécurisation des accès et protection des données.....	42
Optimisation de l'expérience utilisateur.....	42
Perspectives d'amélioration.....	43
Conclusion.....	44
Cinephoria - Application Mobile	
Introduction.....	45
Objectifs.....	45
Fonctionnalités principales.....	45
Technologies utilisées.....	46
Architecture du projet.....	46
Réalisation d'une fonction API.....	47
- Fonction : Récupération des séances réservées.....	47
Simulation.....	48
Sécurité.....	48
Défis rencontrés.....	49
Perspectives d'évolution.....	49
Conclusion.....	49
Cinephoria - Application Bureautique	
Introduction.....	50
Objectifs.....	50
Technologies utilisées.....	50
Fonctionnalités principales.....	51
Connexion sécurisée.....	51
Interface de gestion des incidents.....	51
Architecture de la base de données.....	51
Aspects techniques.....	52
Défis rencontrés.....	55
Perspectives d'amélioration.....	55
Conclusion.....	56



Cinephoria

Introduction

Le projet **Cinéphoria** s'inscrit dans une démarche de digitalisation de l'expérience cinématographique, en proposant un écosystème complet composé d'une **application web**, d'une **application mobile** et d'une **application bureautique**.

Dans un monde où la technologie redéfinit les usages, Cinéphoria répond aux attentes d'un public moderne, souhaitant réserver rapidement des billets, consulter les films à l'affiche, accéder à des informations détaillées, ou encore suivre ses réservations, le tout en quelques clics.

Ce projet vise également à **optimiser la gestion interne** des cinémas. À travers des interfaces dédiées aux **employés** et aux **administrateurs**, Cinéphoria permet de gérer efficacement les films, les séances, les salles, et même les incidents techniques, tout en exploitant des données analytiques en temps réel grâce à une base MySQL.

Développé avec une approche **multi-plateforme**, le projet met en œuvre plusieurs technologies modernes :

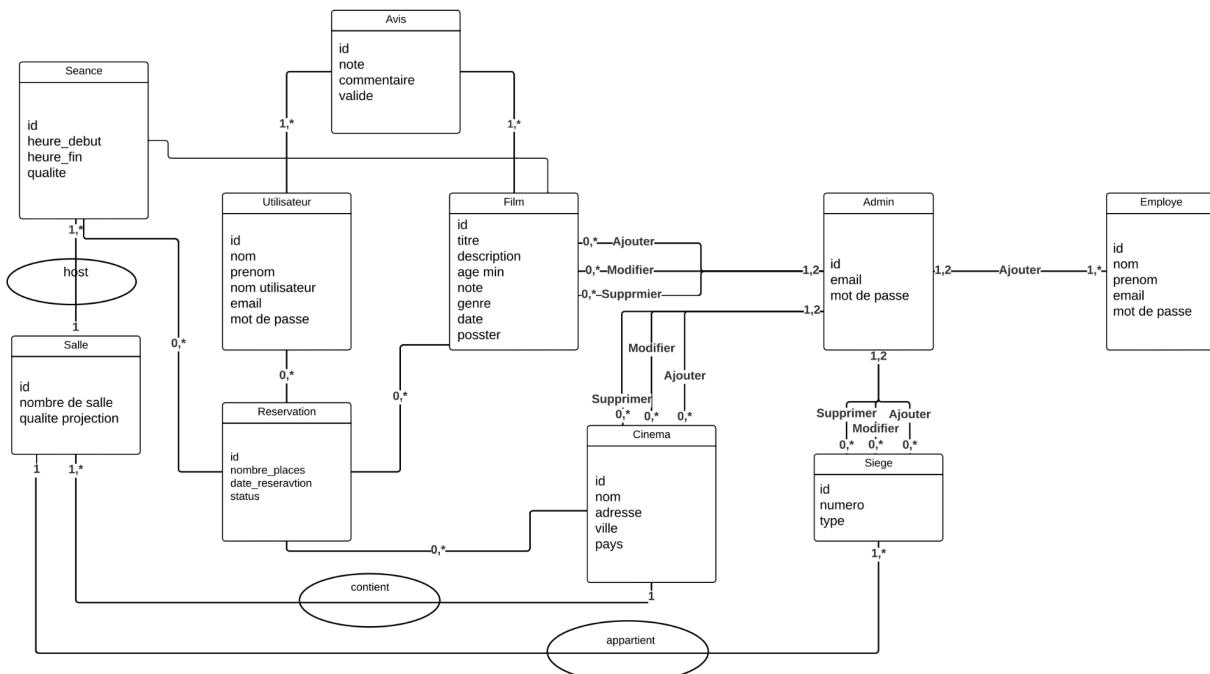
- **PHP / MVC** pour la logique serveur et l'API,
- **MySQL et MongoDB** pour les données,
- **Flutter** pour l'application mobile,
- **Python/Tkinter** pour l'outil bureautique.

Cinéphoria ne se limite pas à un outil de réservation : c'est une **solution numérique complète**, à la fois pratique pour les utilisateurs et puissante pour les gestionnaires de cinéma.

Base de données et architecture

MCD

Le Modèle Conceptuel de Données (MCD) de **Cinephoria** décrit les entités, les relations, et les cardinalités nécessaires à la gestion d'une plateforme de réservation et d'évaluation de films projetés dans des cinémas.



Les entités principales

1. Utilisateur

- Attributs : **id, nom, prenom, nom_utilisateur, email, mot_de_passe**
- Peut effectuer des réservations, donner des avis, et consulter les séances.

2. Film

- Attributs : **id, titre, description, age_min, note, genre, date, posster**
- Est projeté dans différentes séances. Il peut recevoir plusieurs **avis** d'utilisateurs.

3. Séance

- Attributs : `id, heure_debut, heure_fin, qualite`
- Correspond à une projection d'un film dans une **salle** à un horaire donné.

4. Salle

- Attributs : `id, nombre_de_salle, qualite_projection`
- Héberge plusieurs séances. Elle appartient à un **cinéma**.

5. Cinema

- Attributs : `id, nom, adresse, ville, pays`
- Contient plusieurs **salles** et **sièges**, et est lié à un ou plusieurs **films** projetés.

6. Reservation

- Attributs : `id, nombre_places, date_reservation, status`
- Fait par un **utilisateur** pour une **séance**, avec un certain nombre de places.

7. Avis

- Attributs : `id, note, commentaire, valide`
- Donné par un **utilisateur** sur un **film**.

8. Admin

- Attributs : `id, email, mot_de_passe`
- Peut **ajouter**, **modifier**, ou **supprimer** des films, cinémas et sièges.
- Peut également ajouter des employés.

9. Employé

- Attributs : `id, nom, prenom, email, mot_de_passe`
- Crée par un **admin**. Potentiellement utilisé pour des fonctionnalités de gestion.

10. Siège

- Attributs : **id**, **numero**, **type**
- Placé dans un **cinéma**. Peut être réservé, avec gestion par un **admin**.

Les relations

1. **host (Séance – Salle)**

- Une **salle** peut héberger plusieurs **séances**.
- Une **séance** se déroule dans une seule **salle**.

2. **contient (Cinema – Salle)**

- Un **cinéma** contient plusieurs **salles**.
- Une **salle** appartient à un seul **cinéma**.

3. **Reservation (Utilisateur – Séance)**

- Un **utilisateur** peut faire plusieurs **réservations**.
- Chaque **réservation** concerne une seule **séance**.

4. **Avis (Utilisateur – Film)**

- Un **utilisateur** peut donner plusieurs **avis**.
- Chaque **avis** est sur un seul **film**.

5. **appartient (Siège – Cinema)**

- Un **cinéma** possède plusieurs **sièges**.
- Un **siège** appartient à un seul **cinéma**.

6. **Admin – Employé**

- Un **admin** peut créer plusieurs **employés**.

- 
- Chaque **employé** est lié à un seul **admin**.

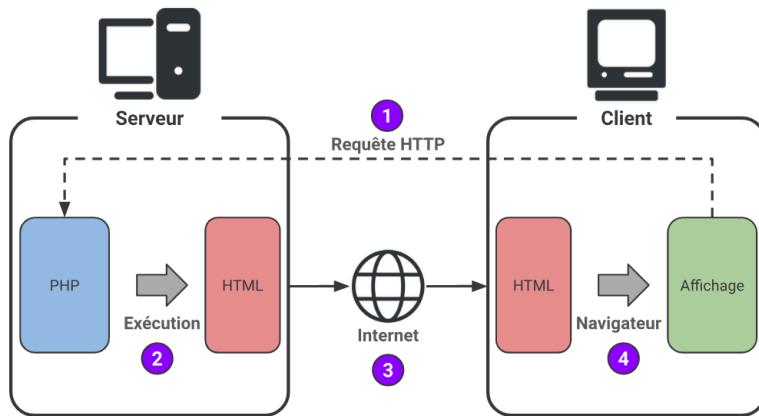
7. Admin - Film/Cinema/Siège

- L'admin peut **ajouter**, **modifier**, et **supprimer** des entités comme **film**, **cinéma**, ou **siège**.

Cardinalités importantes

- **Utilisateur — Avis** : (1,) pour Utilisateur, (1,) pour Film.
- **Utilisateur — Reservation** : (0,*) pour chaque côté.
- **Film — Séance** : (1,*) pour chaque côté.
- **Salle — Séance** : (1,*) pour Salle, (1,1) pour Séance.
- **Cinema — Salle / Siège** : (1,*) pour Salle et Siège, (1,1) pour Cinéma.
- **Admin — Employé** : (1,*) pour Employé, (1,2) pour Admin.
- **Admin** peut avoir des droits multiples sur plusieurs entités.

Architecture



Ce diagramme représente une architecture web classique client-serveur utilisant PHP pour le développement d'applications web dynamiques.

Fonctionnement

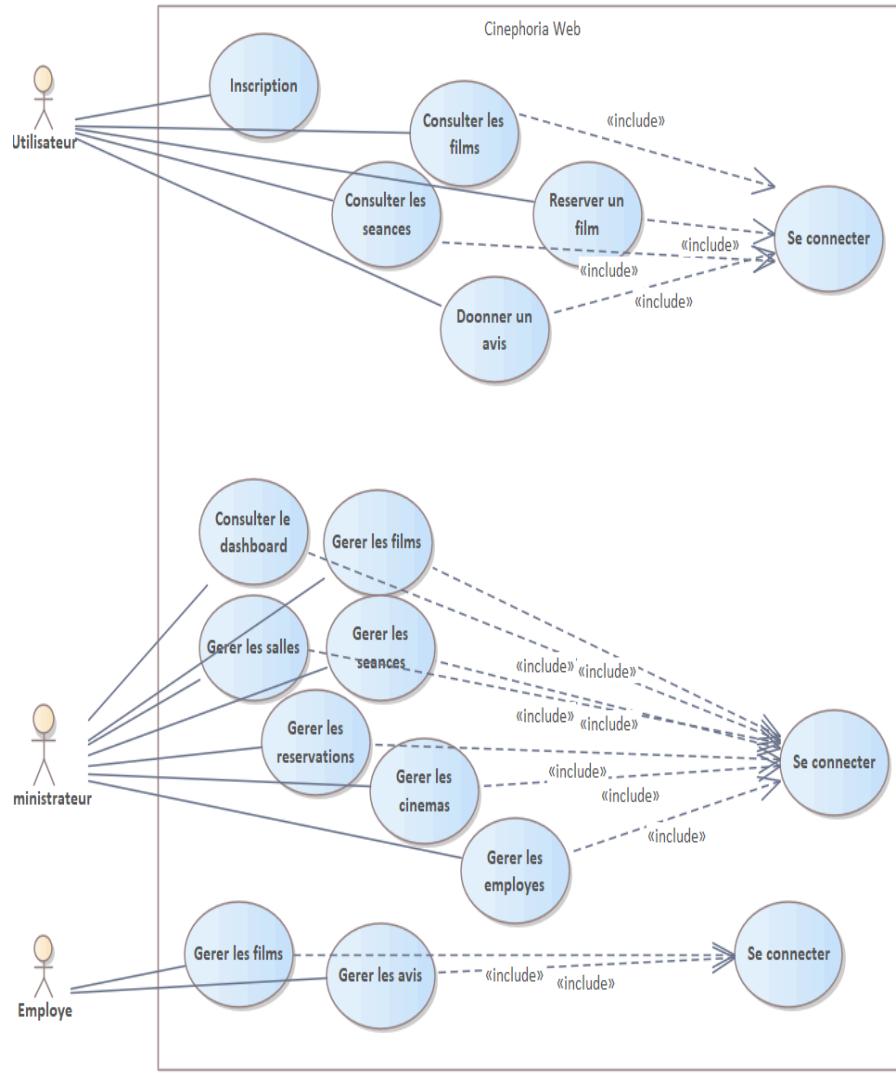
L'architecture illustre le cycle complet d'une requête web : le client envoie une requête HTTP au serveur, qui exécute le code PHP pour générer dynamiquement du contenu HTML. Cette réponse HTML est ensuite transmise via Internet vers le navigateur du client qui l'affiche à l'utilisateur final.

Composants clés

- **Côté serveur :** PHP traite la logique métier et génère le code HTML de manière dynamique selon les données et les besoins de l'application.
- **Côté client :** Le navigateur reçoit le code HTML généré et procède à son rendu pour afficher la page web à l'utilisateur.

Cette architecture garantit une séparation claire entre le traitement des données (serveur) et la présentation (client), tout en centralisant la logique applicative côté serveur pour une meilleure sécurité et maintenance.

Diagramme de cas d'utilisation - App Web



Ce diagramme représente les interactions possibles entre les **acteurs** du système et les **fonctionnalités** offertes par la plateforme **Cinephoria Web**. On distingue trois types d'acteurs :

👤 Utilisateur

L'utilisateur standard peut :

- **S'inscrire** pour créer un compte.
- **Se connecter** pour accéder aux fonctionnalités réservées.

- **Consulter les films** disponibles à l'affiche.
- **Consulter les séances** proposées par les différents cinémas.
- **Réserver un film**, ce qui inclut obligatoirement la connexion («**include**» Se connecter).
- **Donner un avis** sur un film, également conditionné par la connexion.

Remarque : L'usage des relations «**include**» pour forcer l'authentification avant certaines actions est justifié et reflète bien la logique de sécurité d'un site de réservation.

Administrateur

L'administrateur a des privilèges avancés :

- **Consulter un tableau de bord** de gestion.
- **Gérer les films**, les séances, les salles, les réservations, les cinémas, et les employés.
- Chacune de ces actions est précédée d'une connexion obligatoire («**include**» Se connecter).

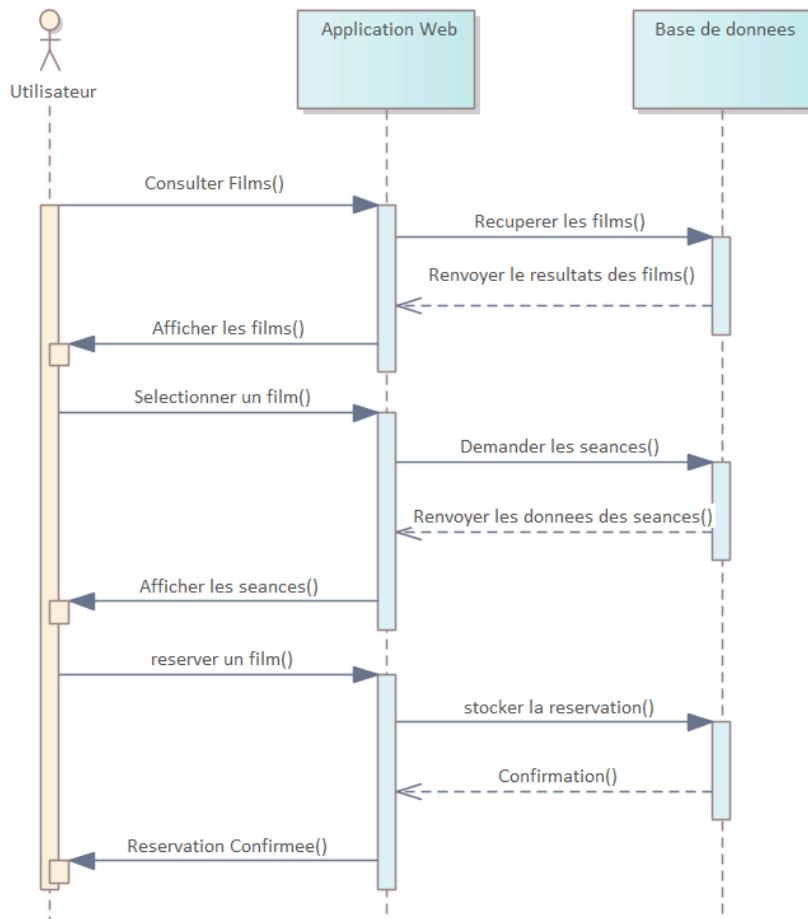
Point fort : Les cas d'usage sont bien segmentés par domaine de gestion, ce qui facilite l'implémentation modulaire côté backend.

Employé

L'employé dispose d'un accès restreint :

- Il peut **gérer les films** (ajouter, modifier, supprimer).
- Il peut également **gérer les avis** (probablement pour les valider ou les supprimer).
- Là encore, les deux actions incluent une **connexion obligatoire**

Diagrammes de séquences - App Web - Processus de Réservation de Films



Ce diagramme illustre le processus complet de réservation d'un film dans une application web de cinéma. Il met en évidence l'interaction entre trois entités principales :

Acteurs impliqués :

- **Utilisateur** : L'utilisateur final de l'application
- **Application Web** : L'interface utilisateur frontale
- **Base de données** : Le système de stockage des données

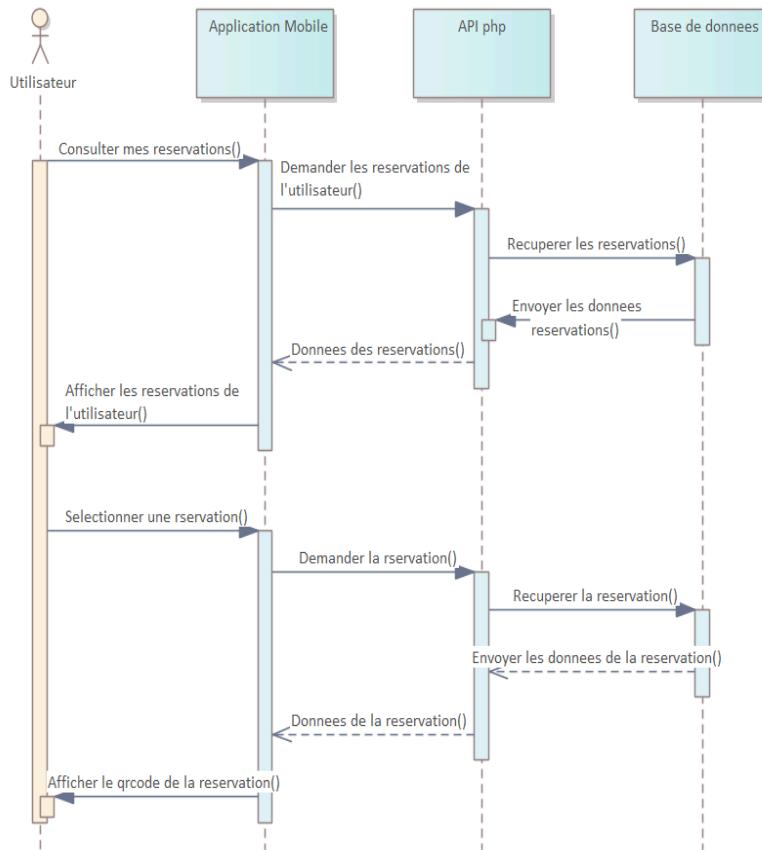
Flux de processus :

1. **Phase de consultation** : L'utilisateur consulte la liste des films disponibles. L'application web interroge la base de données pour récupérer et afficher tous les films programmés.

- 
2. **Phase de sélection** : Une fois qu'un film spécifique est sélectionné, l'application demande les séances disponibles pour ce film à la base de données, puis les affiche à l'utilisateur.
 3. **Phase de réservation** : Lorsque l'utilisateur choisit de réserver une séance, l'application enregistre cette réservation dans la base de données et renvoie une confirmation à l'utilisateur.

Ce diagramme démontre un flux séquentiel typique d'une application de réservation, avec des interactions bidirectionnelles entre le frontend et la base de données pour chaque étape du processus.

Diagramme de séquences - App Mobile - Consultation des Réservations via Application Mobile



Ce diagramme présente le processus de consultation des réservations existantes à travers une architecture mobile avec API.

Acteurs impliqués :

- **Utilisateur** : L'utilisateur de l'application mobile
- **Application Mobile** : L'interface mobile cliente
- **API PHP** : La couche d'interface de programmation d'application
- **Base de données** : Le système de stockage des données

Flux de processus :

1. **Consultation globale** : L'utilisateur demande à voir toutes ses réservations. L'application mobile fait appel à l'API PHP qui interroge la base de données pour

récupérer toutes les réservations associées à l'utilisateur, puis retourne ces données via l'API vers l'application mobile pour affichage.

2. **Consultation détaillée** : Lorsque l'utilisateur sélectionne une réservation spécifique, l'application mobile demande les détails via l'API PHP, qui récupère les informations complètes de cette réservation dans la base de données et génère un QR code pour l'affichage final.

Architecture mise en évidence : Ce diagramme illustre une architecture à trois niveaux (3-tier) avec séparation claire entre :

- La couche présentation (Application Mobile)
- La couche logique métier (API PHP)
- La couche données (Base de données)

Cette architecture permet une meilleure modularité et facilite la maintenance et l'évolutivité du système.

Base de données

MySQL

Creation des tables

-- Table des administrateurs

```
CREATE TABLE admin (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(150) NOT NULL UNIQUE,
    mot_de_passe VARCHAR(255) NOT NULL
);
```

-- Table des utilisateurs

```
CREATE TABLE utilisateurs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(50) NOT NULL,
    prenom VARCHAR(50) NOT NULL,
    nom_utilisateur VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL,
    mot_de_passe VARCHAR(50) NOT NULL
);
```

-- Table des employés

```
CREATE TABLE employer (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(30) NOT NULL,
    prenom VARCHAR(30) NOT NULL,
    email VARCHAR(100) NOT NULL,
    mot_de_passe VARCHAR(32) NOT NULL
);
```

-- Table des cinémas

```
CREATE TABLE cinemas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nom VARCHAR(50) NOT NULL,
    adresse VARCHAR(250) NOT NULL,
    ville VARCHAR(50) NOT NULL,
    pays VARCHAR(50) NOT NULL
);
```

-- Table des salles

```
CREATE TABLE salles (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nbr_places INT NOT NULL,
    qualite_projection VARCHAR(50) NOT NULL,
    cinema_id INT NOT NULL
);
```

-- Table des films

```
CREATE TABLE films (
    id INT AUTO_INCREMENT PRIMARY KEY,
    titre VARCHAR(50) NOT NULL,
    description VARCHAR(250) NOT NULL,
    age_min INT NOT NULL,
    note INT NOT NULL,
    genre VARCHAR(50) NOT NULL,
    date DATE NOT NULL,
    poster VARCHAR(5000) NOT NULL
);
```

-- Table des séances

```
CREATE TABLE seances (
    id INT AUTO_INCREMENT PRIMARY KEY,
    heure_debut TIME NOT NULL,
    heure_fin TIME NOT NULL,
    qualité VARCHAR(50) NOT NULL,
    film_id INT NOT NULL,
    salle_id INT NOT NULL
);
```

);

-- Table des sièges

```
CREATE TABLE siege (
    id INT AUTO_INCREMENT PRIMARY KEY,
    numero INT NOT NULL,
    type VARCHAR(50) NOT NULL,
    salle_id INT NOT NULL
);
```

-- Table des avis

```
CREATE TABLE avis (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    film_id INT NOT NULL,
    note INT NOT NULL,
    commentaire VARCHAR(255) NOT NULL,
    valide VARCHAR(100) NOT NULL
);
```

-- Table des réservations

```
CREATE TABLE reservations (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    seance_id INT NOT NULL,
    film_id INT NOT NULL,
    nombre_places INT NOT NULL,
    date_reservation DATE NOT NULL,
```

```
statut VARCHAR(50) NOT NULL );
```

-- Table des incidents (optionnelle)

```
CREATE TABLE incidents (
    id INT AUTO_INCREMENT PRIMARY KEY,
    salle VARCHAR(255) NOT NULL,
    description TEXT NOT NULL,
    date_incident VARCHAR(255) NOT NULL
);
```

Remplissage des tables

-- Ajout d'un administrateur

```
INSERT INTO admin (email, mot_de_passe) VALUES  
('admin@gmail.com', 'admin');
```

-- Ajout d'un utilisateur

```
INSERT INTO utilisateurs (nom, prenom, nom_utilisateur, email, mot_de_passe) VALUES  
('Mohammed', 'Adam', 'MoAdam', 'adam@gmail.com',  
'098f6bcd4621d373cade4e832627b4f6');
```

-- Ajout d'un film

```
INSERT INTO films (titre, description, age_min, note, genre, date, poster) VALUES  
('Inception', 'Un film sur les rêves et la réalité.', 12, 4, 'Science-fiction', '2010-07-16',  
'https://exemple.com/inception.jpg');
```

-- Ajout d'un cinéma

```
INSERT INTO cinemas (nom, adresse, ville, pays) VALUES  
('Cinéma Lumière', '10 rue des Étoiles', 'Paris', 'France');
```

-- Ajout d'une salle

```
INSERT INTO salles (nbr_places, qualite_projection, cinema_id) VALUES  
(200, 'IMAX', 1);
```

-- Ajout d'une séance

```
INSERT INTO seances (heure_debut, heure_fin, qualité, film_id, salle_id) VALUES  
('18:00:00', '20:30:00', 'IMAX', 1, 1);
```

-- Ajout d'un siège

```
INSERT INTO siege (numero, type, salle_id) VALUES (1, 'Standard', 1);
```

-- Ajout d'un avis

```
INSERT INTO avis (user_id, film_id, note, commentaire, valide) VALUES  
(1, 1, 5, 'Excellent film !', '1');
```

-- Ajout d'une réservation

```
INSERT INTO reservations (user_id, seance_id, film_id, nombre_places, date_reservation,  
statut) VALUES  
(1, 1, 1, 2, '2025-07-21', 'En attente');
```

MongoDB

The screenshot shows the MongoDB interface with the following details:

- Database:** Cinephoria
- Collection Summary:**
 - LOGICAL DATA SIZE: 430B
 - STORAGE SIZE: 44KB
 - INDEX SIZE: 44KB
 - TOTAL COLLECTIONS: 11
- Create Collection:** A green button in the top right corner.
- Collection List:**
 - Admin
 - Avis
 - Cinema
 - Employer
 - Films
 - Incidents
 - Reservations
 - Salles
 - Seances
 - Siege
 - Utilisateurs
- Sample Databases:**
 - sample_mflix
 - vacation_hub

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
Admin	2	0B	0B	4KB	1	4KB	4KB
Avis	21	0B	0B	4KB	1	4KB	4KB
Cinema	10	0B	0B	4KB	1	4KB	4KB
Employer	3	0B	0B	4KB	1	4KB	4KB
Films	23	0B	0B	4KB	1	4KB	4KB
Incidents	5	0B	0B	4KB	1	4KB	4KB
Reservations	9	0B	0B	4KB	1	4KB	4KB
Salles	12	0B	0B	4KB	1	4KB	4KB
Seances	7	0B	0B	4KB	1	4KB	4KB
Siege	32	0B	0B	4KB	1	4KB	4KB
Utilisateurs	9	0B	0B	4KB	1	4KB	4KB

Wireframes

Application Web

- Page de connexion

The wireframe shows a simple login form. At the top, there's a header with the text "S'inscrire" (Sign up) and "Se connecter" (Log in). Below the header are two input fields: one for "Email" and one for "Mot de passe" (Password). At the bottom of the form is a button labeled "Envoyer" (Send).

- Page d'accueil

The wireframe of the homepage features a dark grey header bar with a "LOGO | NAME" placeholder on the left and navigation links for "Accueil", "A propos", "Contact", "Se connecter", and a search icon on the right. The main content area has a white background. It includes a section titled "Selection du jour" (Selection of the day) which displays six placeholder cards, each labeled "Image Film". Below this is a "Voir tous les films" (View all movies) button. Another section titled "Films en tendance" (Trending movies) also displays six placeholder cards, each labeled "Image Film".

- Page admin

The screenshot shows the 'Administration Cinephoria' dashboard. At the top, there's a navigation bar with links: 'Gerer les films', 'Gerer les salles', 'Gerer les seances', 'Gerer les employes', 'Accueil', and 'Se connecter'. Below the navigation bar, a welcome message reads 'Bienvenue dans l'administration de Cinephoria'. Six summary boxes are displayed in a 2x3 grid:

Totales Reservatios	Totales Utilisateurs	Totales Employes
17	4	7
Go	Go	Go

Totales Films	Totales Salles	Totales Cinemas
5	25	23
Go	Go	Go

At the bottom of the page, there are two footer sections: 'Cinephoria Administration' on the left and 'Connecte en tant que : admin@gmail.com' on the right.

Application Mobile



Application Bureautique

Gestion des Incidents

Salle ▼

Décrire Incident :

Décrire

Enregistrer l'incident

Incidents enregistrés :

Retour à la page connexion

Cinephoria - Application Web

Introduction

L'application web Cinéphoria a été développée afin de répondre aux attentes d'une clientèle moderne et exigeante, en quête d'une expérience cinéma à la fois intuitive, rapide et agréable. Fondée sur une analyse approfondie des besoins utilisateurs, cette plateforme numérique se positionne comme un véritable levier d'innovation et de différenciation dans le secteur du divertissement cinématographique.

L'objectif principal est d'offrir une interface simple d'utilisation permettant aux clients de consulter facilement les films disponibles, les séances proposées par chaque cinéma de la chaîne et d'effectuer en toute simplicité leurs réservations en ligne, avec choix précis des sièges et visualisation claire des tarifs selon les différentes options disponibles.

En parallèle, l'application vise également à renforcer l'interaction et la fidélisation des utilisateurs en leur proposant des fonctionnalités telles que la notation des séances visionnées, la consultation d'avis sur les films, et un espace personnel où ils peuvent suivre l'historique de leurs réservations et gérer leurs préférences.

Enfin, Cinéphoria propose une gestion optimale des ressources internes grâce à une interface d'administration complète dédiée aux employés et administrateurs. Celle-ci leur permet de piloter efficacement l'offre cinéma (films, salles, séances), d'analyser les statistiques de fréquentation, et d'assurer un haut niveau de qualité des services proposés grâce à des outils de gestion des incidents et de modération des contenus utilisateurs.

Objectifs

Les objectifs principaux du projet Cinéphoria visent à répondre précisément aux attentes variées des utilisateurs, tout en optimisant la gestion interne des cinémas. Ces objectifs peuvent être détaillés ainsi :

- **Présentation claire des séances** : Proposer une interface permettant aux utilisateurs de facilement visualiser les séances disponibles dans chaque cinéma, avec indication claire des horaires, des salles, et des formats proposés (3D, 4K, etc.).

- **Réservation en ligne optimisée** : Offrir un processus de réservation fluide et rapide, incluant la sélection précise des sièges, une visibilité en temps réel des places disponibles, et une transparence totale des tarifs associés.
- **Expérience utilisateur interactive** : Permettre aux utilisateurs de consulter les fiches détaillées des films, visionner des bandes-annonces, lire et rédiger des avis, et noter les films après visionnage.
- **Espace utilisateur personnalisé** : Fournir aux utilisateurs un espace dédié où ils peuvent gérer leur profil, suivre l'historique de leurs réservations, consulter leurs billets électroniques (QR codes), et gérer leurs préférences personnelles.
- **Interface d'administration complète** : Développer une interface robuste pour les employés et administrateurs, permettant la gestion efficace des films, des séances, des salles et des employés, ainsi qu'un suivi analytique des performances grâce à un tableau de bord statistique.

Technologies

Front-end :

- **Bootstrap 5** : Framework CSS permettant de créer une interface responsive, adaptative à tous les écrans (ordinateur, tablette, smartphone), et offrant une expérience utilisateur cohérente et agréable.
- **CSS** : Styles avancés permettant une organisation claire du code et une maintenance facilitée grâce à la modularité offerte par SCSS.
- **JavaScript, jQuery** : Rendent l'application interactive en facilitant les animations, la gestion dynamique du contenu et des formulaires, ainsi que l'amélioration de l'expérience utilisateur en général.
- **Font-Awesome** : Bibliothèque d'icônes modernes et esthétiques permettant une interface utilisateur intuitive et attractive.

Back-end :

- **PHP 7+ (Architecture MVC)** : Offre une structuration claire et logique du code côté serveur, facilitant la maintenance, la réutilisation du code, et permettant une séparation efficace des responsabilités entre la logique métier, l'affichage et la gestion des données.
- **MySQL** : Gestion des données transactionnelles robustes et sécurisées, assurant la cohérence et l'intégrité des données essentielles comme les réservations, utilisateurs, films, et séances.
- **MongoDB** : Stockage de données NoSQL utilisé spécifiquement pour gérer les analyses statistiques et générer des tableaux de bord interactifs en temps réel, offrant ainsi une vision claire des performances et des tendances pour les administrateurs.

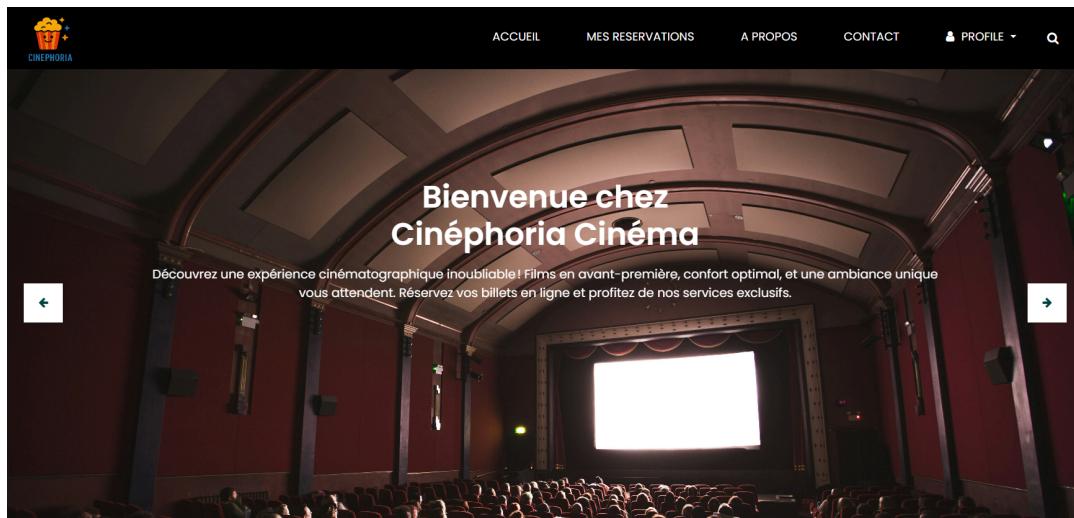
Ces technologies combinées assurent la fiabilité, la rapidité et la sécurité de l'application tout en fournissant une expérience utilisateur optimale.

Fonctionnalités réalisés

Interface Utilisateur

L'interface utilisateur de l'application web Cinéphoria est conçue pour offrir une expérience simple, fluide et agréable, tout en répondant efficacement aux besoins spécifiques des clients. Voici en détail les principales fonctionnalités :

- **Accueil dynamique** : Présentation claire et dynamique des derniers films ajoutés chaque mercredi, offrant ainsi aux utilisateurs une vue rapide sur les nouveautés disponibles.



Selection du jour



[Voir tous les films](#)

Films en tendances

Trailers



Inception - Trailer

Interstellar - Trailer

The Dark Knight - Trailer

Gladiator - Trailer

Cinéphoria CONTACT +33 123 45 67 89 contact@cinephoria.fr	Location HORAIRES Lundi-Vendredi: 10h-23h Samedi-Dimanche: 9h-00h	NOS CINÉMAS Paris Bordeaux Lille Liège Nantes Toulouse Charleroi	f t in @
--	--	--	--

- **Navigation intuitive** : Un menu clairement organisé et accessible sur chaque page, comprenant des liens rapides vers l'accueil, la connexion, la réservation, la consultation des films et un formulaire de contact direct.



- **Consultation détaillée des films** : Chaque film dispose d'une fiche complète comprenant une description approfondie, l'affiche officielle, des avis d'utilisateurs, ainsi que des options de filtrage avancées par cinéma, genre et jour de projection.

Inception

Genre : Science-fiction

Description:
ce film s'appelait inception

★★★★★

Seances disponibles pour ce film :

00:00:13 (fin à 00:00:15) Salle 1	12:15:16 (fin à 00:00:18) Salle 1	00:20:25 (fin à 00:20:25) Salle 1
--	--	--

< Retour

- **Réservation simplifiée et interactive :** Processus intuitif de réservation en ligne permettant la sélection précise des sièges avec une vue interactive en temps réel des disponibilités.

The screenshot shows a dark-themed website for 'CINEPHORIA'. At the top, there's a navigation bar with links for 'ACCUEIL', 'MES RESERVATIONS', 'A PROPOS', 'CONTACT', 'PROFILE', and a search icon. The main content area has a light gray background. On the left, a form for a movie reservation is displayed, asking for 'Nombre de places' (2) and 'Date de réservation' (14/07/2025), with a blue 'Réserver' button at the bottom. To the right, there are two movie posters: one for 'LAMB' showing a woman holding a lamb, and another for 'Inception' showing a woman in a dreamlike setting. Below the posters, the word 'Inception' is written.

- **Authentification sécurisée et gestion de compte :** Création facile d'un compte utilisateur avec système d'authentification robuste (login sécurisé, récupération de mot de passe), offrant un accès à un espace personnel où chaque utilisateur peut gérer ses réservations, consulter les historiques et préférences.

The screenshot displays two mobile application screens. The left screen is titled 'S'inscrire' (Sign Up) and contains fields for 'Nom', 'Prenom', 'Username', 'Email', and 'Mot de passe' (Password), with a 'Envoyer' (Send) button at the bottom. The right screen is also titled 'S'inscrire' (Sign Up) but features a large purple 'Se connecter' (Sign In) button at the top. It has fields for 'Email' and 'Password', and a 'Envoyer' (Send) button at the bottom right.

- **Système de notation et avis :** Permet aux utilisateurs de noter les films visionnés et de laisser des commentaires détaillés, favorisant ainsi l'interaction communautaire et aider les autres utilisateurs dans leurs choix.

- Si la date du film est passé, l'utilisateur est invité à noter le film :



Reservation pour le film **Avatar**

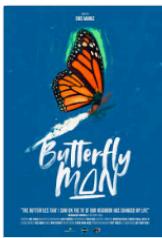
Nombre de places 1

Date : 2025-07-13

13:23:00 - 16:26:00

[Donner mon avis](#)

- Si la date du film 'est pas encore passée :



Reservation pour le film **Forrest Gump**

Nombre de places 1

Date : 2025-07-15

00:00:19 - 00:00:21

- Si la date du film est déjà passé, et l'utilisateur a noté ce film :



Reservation pour le film **Inception**

Nombre de places 2

Date : 2025-07-15

00:20:25 - 00:20:25

Vous avez déjà note ce film

- Page profile :

Accueil / Profile



Abdou Mohammed

Nom complet	Abdou Mohammed
Email	abdou@gmail.com
Mobile	(320) 380-4539

[Changer le mot de passe](#)

Mes Réervations

Titanic
Date : 2025-04-10 -- Qualité : IMAX

Inception
Date : 2025-04-18 -- Qualité : Dolby Atmos

Interface Employé

L'interface employé de Cinéphoria est pensée pour offrir aux équipes opérationnelles une gestion simplifiée et efficace des activités quotidiennes liées à la gestion des films, des séances et des incidents techniques. Voici les principales fonctionnalités :

- **Gestion des films et séances** : Permet aux employés d'ajouter, modifier ou supprimer les films à l'affiche ainsi que de gérer les horaires des séances et les formats de projection proposés (standard, 3D, 4DX).
- **Interface claire et ergonomique** : L'ensemble des outils proposés est facilement accessible via un tableau de bord ergonomique, favorisant une prise en main rapide et une utilisation quotidienne confortable pour les employés.



Séances du jour

Tous les films	Toutes les salles	
Inception ⌚ 00:00 - 00:00 📍 Salle 1 🎞️ IMAX	Joker ⌚ 00:00 - 00:00 📍 Salle 5 🎞️ IMAX	Parasite ⌚ 00:00 - 00:00 📍 Salle 6 🎞️ 4K

Interface Administrateur

L'interface administrateur de Cinéphoria est conçue pour offrir un contrôle complet et efficace sur l'ensemble des opérations liées à la gestion des cinémas. Elle permet de centraliser toutes les activités clés et fournit des outils analytiques performants. Voici les principales fonctionnalités détaillées :

- **Tableau de bord analytique** : Grâce à l'intégration de MongoDB, l'administrateur dispose d'un tableau de bord complet permettant une visualisation en temps réel des statistiques de réservation, de fréquentation des séances, et des tendances globales pour une prise de décision éclairée.

The screenshot shows the administrator dashboard for Cinephoria. At the top, there is a navigation bar with links for 'Administration Cinephoria', 'Gérer les films', 'Gérer les salles', 'Gérer les séances', 'Gérer les employés', and 'Déconnexion'. Below the navigation bar, the main title 'Bienvenue dans l'administration de Cinephoria' is displayed. A sub-instruction 'Utilisez le menu ci-dessous pour gérer les différentes sections du site.' is present. The dashboard features six cards arranged in a 2x3 grid, each containing a statistic and a 'Go' button:

Totales Réservations 7 Go	Totales Utilisateurs 1 Go	Totales Employés 4 Go
Totales Films 14 Go	Totales Salles 16 Go	Totales Séances 10 Go

- Gestion complète des films, séances et salles** : L'administrateur a un accès total pour ajouter, modifier ou supprimer des films, gérer précisément les horaires des séances, contrôler les formats de projection, et organiser les salles selon les besoins spécifiques de chaque cinéma.

Liste des Films

Titre	Description	Age Min	Note	Genre	Date	Poster	Actions
Inception	ce film s'appelait inception	13	4	Science-fiction	0000-00-00		
Interstellar	Voyage dans l'espace et le temps.	10	2	Science-fiction	0000-00-00		
Titanic	Une romance dramatique sur l'océan.	12	5	Drame	0000-00-00		

Liste des Salles

ID	nbr_places	qualite_projection	cinema_id	Actions
1	300	IMAX	1	
2	150	4K	2	
3	120	HD	3	
4	180	Dolby Cinema	4	
5	220	IMAX	5	

Liste des Séances

ID	heure_debut	heure_fin	qualité	film_id	salle_id	Actions
3	00:00:13	00:00:15	IMAX	1	1	
5	00:00:18	00:00:20	IMAX	5	5	
6	00:00:21	00:00:23	4K	6	6	
7	00:00:15	00:00:17	HD	7	7	

- **Gestion des employés** : Possibilité de créer, modifier ou supprimer les comptes employés, gérer leurs droits d'accès aux différentes parties du système, et superviser leur activité au quotidien.



Liste des Employés

ID	Nom	Prenom	Email	Actions
1	ALI	Djoumoi	ali@gmail.com	
2	Mohammed	Abdou	mohamed@gmail.com	
3	Employe 1	Falih	falih@gmail.com	
4	remi	dupont	remi@gmail.com	

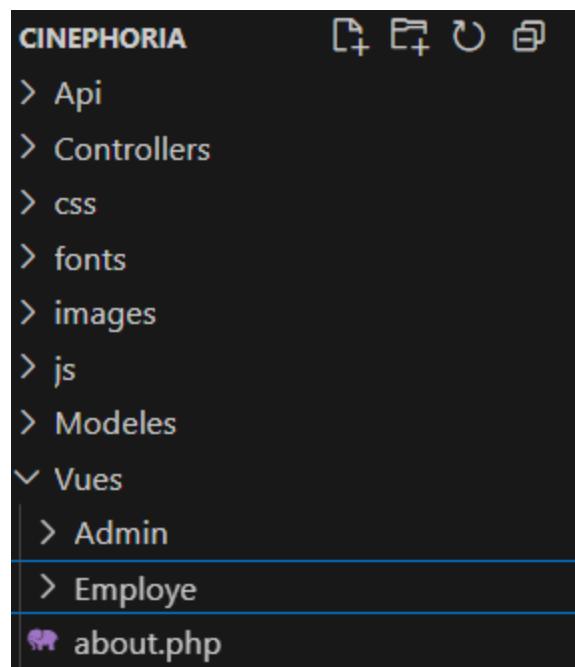


Liste des Utilisateurs

ID	Nom	Prenom	Nom_utilisateur	Email	Actions
5	Abdou	Mohammed	abdou_mohammed	abdou@gmail.com	

Architecture technique (MVC)

L'application web Cinéphoria est construite selon le modèle architectural MVC (Modèle-Vue-Contrôleur), assurant ainsi une organisation claire du code, une séparation efficace des responsabilités et une maintenance facilitée. Voici le détail des différentes couches :



Modèles

Les modèles constituent la couche responsable de la gestion des données et de la logique métier de l'application. Ils assurent l'interaction directe avec les bases de données :

- **MySQL** pour la gestion des données transactionnelles telles que les utilisateurs, films, séances et réservations.

Chaque modèle inclut des méthodes métier spécifiques, permettant une interaction sécurisée, rapide et cohérente avec les données de l'application.

Vues

La couche des vues se charge de présenter les données aux utilisateurs à travers une interface intuitive et ergonomique. Grâce à Bootstrap et jQuery, les vues offrent :

- Un design adaptatif (responsive) garantissant une expérience utilisateur optimale sur tous les appareils.
- Une interactivité poussée permettant une expérience fluide et agréable (menus dynamiques, formulaires interactifs, affichage en temps réel des disponibilités des sièges).

Contrôleurs

Les contrôleurs orchestrent les interactions entre les modèles et les vues. Ils traitent les requêtes reçues des utilisateurs, appliquent les règles métier définies dans les modèles, puis transmettent les données nécessaires aux vues :

- Gestion et validation des données entrantes pour garantir leur sécurité et leur conformité.
- Interaction précise avec les modèles pour récupérer, traiter ou enregistrer des données selon les requêtes utilisateur.
- Sélection et chargement dynamique des vues appropriées en fonction du contexte de chaque requête.

Cette structuration MVC permet une évolutivité aisée du projet, une gestion optimisée du code source et une simplification notable des opérations de maintenance.

Sécurité

La sécurité est une priorité absolue dans l'application Cinéphoria afin de garantir la protection des données personnelles des utilisateurs ainsi que l'intégrité globale du système. Plusieurs mécanismes robustes sont mis en place :

- **Authentification sécurisée :**

- Utilisation de la fonction PHP `password_hash()` avec l'algorithme bcrypt pour chiffrer les mots de passe, assurant une protection contre les attaques de type brute force.
- Mécanisme sécurisé de récupération de mot de passe via envoi sécurisé par courriel.

- **Gestion des sessions utilisateur :**

- Sessions PHP sécurisées avec des cookies paramétrés en HttpOnly et Secure afin d'empêcher les attaques par interception.
- Mise en place de mesures préventives contre les attaques de fixation de session (session fixation).

- **Validation des données entrantes :**

- Validation rigoureuse des données utilisateur côté client (JavaScript) et côté serveur (PHP) afin d'éviter les injections SQL, XSS, et autres attaques communes.
- Filtrage systématique des données reçues, garantissant leur conformité et leur sécurité.

- **Protection des données sensibles :**

- Stockage sécurisé des données personnelles en conformité avec les normes RGPD.
- Utilisation de connexions sécurisées et de requêtes préparées pour éviter toute injection SQL dans les interactions avec la base de données.

- **Sécurité du système d'administration :**

- Renforcement de l'accès administrateur par un système d'autorisation précis avec contrôle rigoureux des permissions.
- Suivi et journalisation des activités importantes pour permettre une traçabilité complète des actions sensibles.

Ces mesures cumulées assurent un niveau élevé de sécurité et renforcent la confiance des utilisateurs dans l'application web Cinéphoria.

Défis rencontrés et solutions

Lors du développement de l'application Cinéphoria, plusieurs défis techniques et fonctionnels ont été identifiés et efficacement résolus, permettant ainsi de garantir la stabilité et la performance de l'application. Voici les principaux défis relevés et les solutions mises en œuvre :

Gestion des réservations et des sièges en temps réel

- **Défi :** Assurer une synchronisation parfaite et une disponibilité en temps réel des sièges lors des réservations, afin d'éviter tout conflit, double réservation ou confusion pour les utilisateurs, particulièrement en période de forte affluence.
- **Solution détaillée :**
 - **Vérification instantanée des disponibilités :**

Chaque demande de réservation déclenche une requête en temps réel vers la base de données pour vérifier immédiatement la disponibilité des sièges sélectionnés.
 - **Système de verrouillage temporaire :**

Lorsqu'un utilisateur sélectionne un siège, celui-ci est temporairement verrouillé pendant quelques minutes pour éviter qu'un autre utilisateur puisse le réserver simultanément.
 - **Transactions sécurisées :**

Utilisation de transactions SQL pour assurer la cohérence et l'intégrité des données lors des opérations de réservation, avec un mécanisme d'annulation automatique en cas de détection d'anomalie.
 - **Mise à jour dynamique des sièges :**

Implémentation d'une interface dynamique en JavaScript (AJAX) permettant une mise à jour immédiate de l'affichage des sièges disponibles, réservés ou bloqués, assurant ainsi une expérience

utilisateur fluide et transparente.

- **Optimisation des performances :**

Techniques avancées de mise en cache, notamment pour les informations fréquemment consultées, réduisant ainsi le temps de réponse et augmentant la fluidité générale du processus de réservation.

Ces stratégies cumulées garantissent une gestion fiable, efficace et performante des réservations et des sièges en temps réel, améliorant significativement la satisfaction utilisateur.

Sécurisation des accès et protection des données

- **Défi :** Protéger efficacement l'application contre les menaces courantes telles que les attaques par injection SQL.
- **Solution :** Adoption de pratiques rigoureuses telles que la validation approfondie des données côté client et serveur, l'utilisation systématique de requêtes préparées, la sécurisation renforcée des sessions utilisateurs, et l'implémentation de techniques de hashing avancées pour protéger les mots de passe.

Optimisation de l'expérience utilisateur

- **Défi :** Fournir une expérience utilisateur fluide et réactive malgré une quantité importante de données dynamiques (séances, sièges, tarifs).
- **Solution :** Mise en place de techniques avancées d'optimisation front-end telles que la mise en cache intelligente, le chargement asynchrone des données (AJAX) et l'utilisation efficace de JavaScript et jQuery pour offrir une interface réactive et agréable.

Ces solutions techniques et méthodologiques ont permis de surmonter efficacement les défis rencontrés, contribuant ainsi à la réussite du projet Cinéphoria.

Perspectives d'amélioration

Afin de poursuivre l'amélioration continue de l'application web Cinéphoria et d'assurer son évolution vers une solution toujours plus performante et adaptée aux besoins des utilisateurs, plusieurs axes de développement sont identifiés :

- **Système de notifications en temps réel** : Mise en place d'un système avancé de notifications en temps réel afin d'améliorer l'interaction avec les utilisateurs. Ce système permettrait de notifier immédiatement les utilisateurs lors des confirmations de réservations, modifications de séances, disponibilités de places supplémentaires ou encore lors de la publication de nouveaux films ou promotions.
- **Développement d'une API REST complète** : Création d'une API REST robuste et documentée pour faciliter l'intégration avec d'autres services tiers tels que des applications partenaires, systèmes externes de billetterie ou encore des services de recommandation de films personnalisés. Cela améliorera également la flexibilité et la scalabilité de l'application.
- **Amélioration continue de l'expérience utilisateur (UX/UI)** : Renforcement des tests utilisateurs réguliers afin d'identifier et d'intégrer constamment les améliorations ergonomiques, visuelles et fonctionnelles nécessaires, garantissant ainsi une application toujours intuitive et agréable à utiliser.
- **Sécurité avancée et conformité RGPD** : Renforcement continu des protocoles de sécurité pour assurer une protection optimale des données personnelles et une conformité stricte aux exigences RGPD, incluant notamment des audits réguliers et l'intégration de nouvelles technologies de sécurité.

Ces perspectives garantiront non seulement la pérennité de l'application mais aussi son adaptabilité aux évolutions technologiques et aux nouvelles attentes des utilisateurs.

Conclusion

Le développement de l'application web Cinéphoria s'inscrit dans une démarche de modernisation de l'expérience cinéma, en intégrant des outils numériques intuitifs, sécurisés et accessibles. Ce projet a permis de répondre à des besoins clairement identifiés : offrir aux clients une plateforme de réservation simple et agréable, tout en proposant aux équipes internes des outils performants pour la gestion quotidienne des films, séances et ressources matérielles.

Grâce à une architecture solide basée sur le modèle MVC, à une intégration efficace de technologies modernes (PHP, MySQL, MongoDB, Bootstrap), et à un effort constant de sécurisation et d'optimisation, l'application offre une solution complète et fiable à la fois pour les utilisateurs, les employés et les administrateurs.

Le projet Cinéphoria se distingue également par sa capacité à anticiper les évolutions futures grâce à des perspectives d'amélioration bien définies : déploiement d'API, système de notifications, renforcement de la sécurité et amélioration continue de l'expérience utilisateur.

Cinephoria - Application Mobile

Introduction

L'application mobile **Cinéphoria** s'inscrit dans la continuité de l'application web en proposant aux utilisateurs une solution mobile, simple et efficace, pour accéder à leurs réservations de cinéma. Développée avec **Flutter**, elle permet un déploiement multiplateforme (Android et iOS) tout en s'appuyant sur la même logique métier que la version web grâce à une **API REST** partagée.

L'interface a été pensée pour être **unique, fluide et intuitive**, concentrée uniquement sur les besoins essentiels de l'utilisateur : se connecter, consulter ses séances à venir et accéder à ses billets sous forme de QR code.

Objectifs

- Permettre aux utilisateurs d'accéder à leur compte depuis un smartphone.
- Visualiser rapidement les séances réservées à venir.
- Générer un **QR code de billet** pour chaque réservation.
- Fournir une expérience fluide, rapide et accessible, même en situation de mobilité.

Fonctionnalités principales

L'application mobile repose sur une **interface utilisateur unique** organisée selon un parcours clair et linéaire :

1. Connexion utilisateur

L'utilisateur saisit ses identifiants pour accéder à son espace personnel.
L'authentification est sécurisée via token.

2. Consultation des réservations à venir

Une fois connecté, l'utilisateur voit la liste de ses séances à venir, avec les

informations suivantes :

- Titre du film
- Affiche
- Date et heure de projection
- Salle et numéro(s) de siège(s)

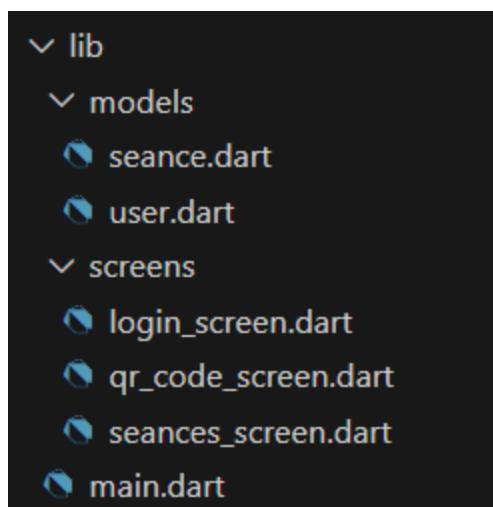
3. Affichage du QR code

En cliquant sur une réservation, un **QR code personnel** s'affiche, représentant le billet électronique. Ce code est scanné à l'entrée du cinéma pour valider l'accès.

Technologies utilisées

- **Flutter** : framework UI multiplateforme basé sur Dart
- **Dart** : langage de programmation moderne et optimisé
- **API REST** : pour interagir avec le backend PHP/MySQL
- **qr_flutter** : pour générer des QR codes en local

Architecture du projet



Réalisation d'une fonction API

Pour permettre la communication entre l'application mobile Flutter et le serveur backend PHP/MySQL, une API REST a été mise en place. L'URL principale de cette API est :

<http://10.0.2.2/Cinephoria/Api>

https://abdou.website/Cinephoria/Api/get_reservation.php?user_id=X

(utilisée pour l'émulateur Android ; sur appareil physique, utiliser l'IP locale de la machine)

- Fonction : Récupération des séances réservées

L'application mobile utilise la fonction `_fetchSeances()` pour interroger l'API :

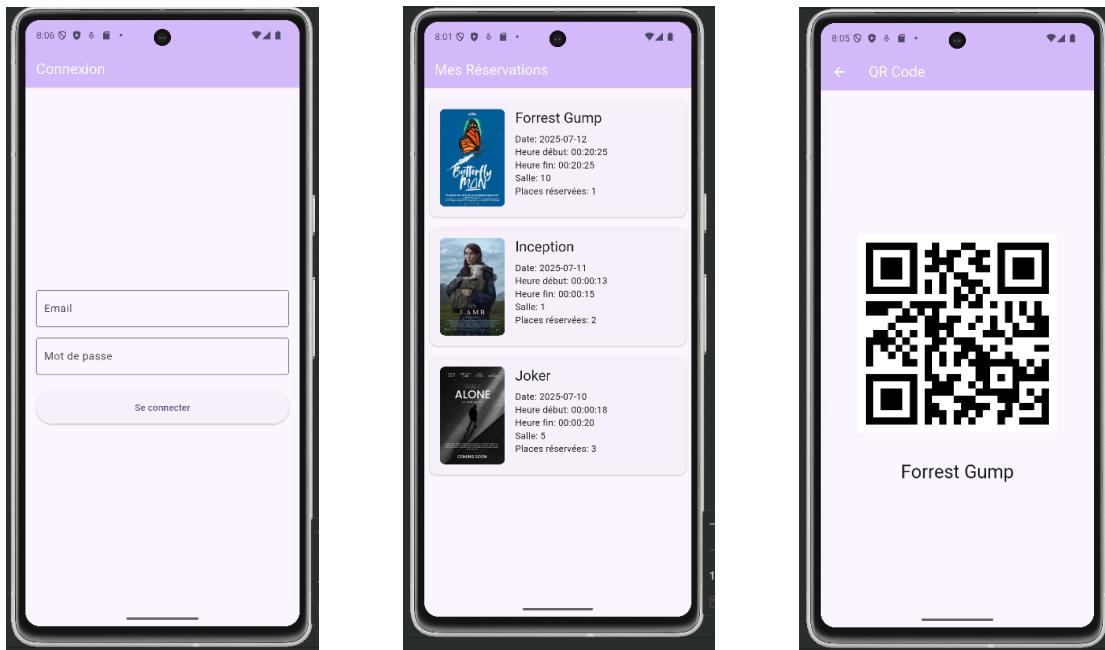
```
// Base URL de l'API
static const String baseUrl = 'http://10.0.2.2/Cinephoria/Api';
@Override
void initState() {
    super.initState();
    _fetchSeances();
}

Future<void> _fetchSeances() async {
    try {
        final response = await http.get(
            Uri.parse('$baseUrl/get_reservation.php?user_id=${widget.userId}'),
        );
        if (response.statusCode == 200) {
            final data = json.decode(response.body);
            if (data['reservations'] != null) {
                setState(() {
                    _seances = (data['reservations'] as List)
                        .map((json) => Seance.fromJson(json)).toList();
                    _isLoading = false;
                });
            }
        }
    } catch (e) {
        print(e);
    }
}
```

Explication :

- L'application envoie une requête HTTP GET avec le `user_id` à `get_reservation.php`.
- Le backend PHP retourne un tableau JSON des réservations.
- Flutter convertit ces données en objets `Seance` et les affiche dynamiquement.

Simulation



Sécurité

- Authentification via **token sécurisé**
- Communication via **HTTPS**
- Données stockées localement de manière chiffrée

- 
- Accès restreint aux fonctionnalités sans connexion

Défis rencontrés

- **Adaptation du backend web** pour une communication fluide avec l'application mobile
- **Gestion du QR code** en local tout en maintenant une bonne performance
- **Synchronisation en temps réel** avec les données du serveur

Perspectives d'évolution

- Ajout de **notifications push** pour rappeler les séances
- Gestion **hors ligne** avec synchronisation différée
- Personnalisation de l'espace utilisateur

Conclusion

L'application mobile Cinéphoria constitue une extension naturelle et efficace de la plateforme web, centrée sur **l'accessibilité et la praticité**. Elle répond aux besoins spécifiques des utilisateurs en déplacement tout en restant cohérente avec l'écosystème global Cinéphoria.

Son design épuré, son interface unique et son intégration avec le système existant en font un **outil simple, robuste et prêt à l'emploi**, qui pourra évoluer facilement vers une solution encore plus complète à l'avenir.

Cinephoria - Application Bureautique

Introduction

L'application bureautique **Cinéphoria** est un outil développé en Python avec la bibliothèque **Tkinter** pour l'interface graphique, et connecté à une base de données **MySQL**. Elle est spécifiquement destinée aux **employés** de la chaîne Cinéphoria pour leur permettre de **déclarer et visualiser les incidents techniques** survenus dans les salles de projection (ex. : siège cassé, panne d'équipement, problème de son, etc.).

Objectifs

- Permettre aux employés de **signaler en temps réel** tout incident technique.
- Centraliser l'ensemble des incidents dans une **base de données MySQL** pour faciliter le suivi et la maintenance.
- Afficher la **liste des incidents** enregistrés dans une salle donnée.
- Assurer la **tracabilité** avec enregistrement automatique de la date et de l'heure.

Technologies utilisées

- **Python 3.x** : langage principal de développement.
- **Tkinter** : pour la création de l'interface graphique.
- **MySQL** : pour la gestion et la sauvegarde des données.
- **mysql.connector** : bibliothèque permettant la communication entre Python et MySQL.

Fonctionnalités principales

1. Connexion sécurisée

- Interface de **connexion avec email et mot de passe**.
- Vérification des identifiants dans la table **employer** de la base de données.
- Message d'erreur affiché en cas d'identifiants incorrects.

2. Interface de gestion des incidents

- **Sélecteur de salle** (récupéré automatiquement depuis la base de données).
- Zone de **saisie de la description** de l'incident.
- **Bouton d'enregistrement** qui sauvegarde l'incident dans la table **incidents**.
- Zone d'**affichage de la liste des incidents** existants dans la salle sélectionnée.
- Affichage de la **date et l'heure** de chaque incident.

Architecture de la base de données

- **Table employer** : contient les informations de connexion des employés (email, mot de passe...).
- **Table salles** : liste des salles disponibles.
- **Table incidents** :
 - **id** : clé primaire
 - **salle** : identifiant ou nom de la salle

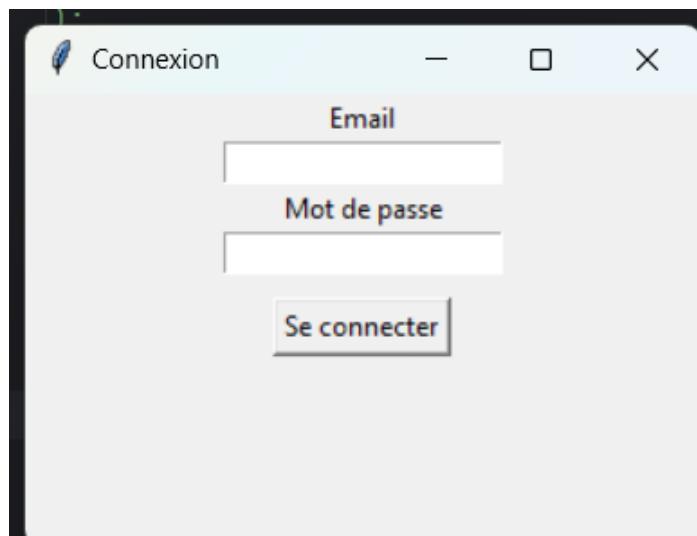
- **description** : texte de l'incident
- **date_incident** : date et heure d'enregistrement

Aspects techniques

- Création automatique de la table **incidents** si elle n'existe pas.
- Gestion des erreurs MySQL avec des messages d'erreur affichés dans l'interface.
- Interface ergonomique et facile à utiliser.
- Organisation du code séparant les fonctions de connexion et d'utilisation.

Captures d'écran (à insérer dans un document Word)

- Page de **connexion**



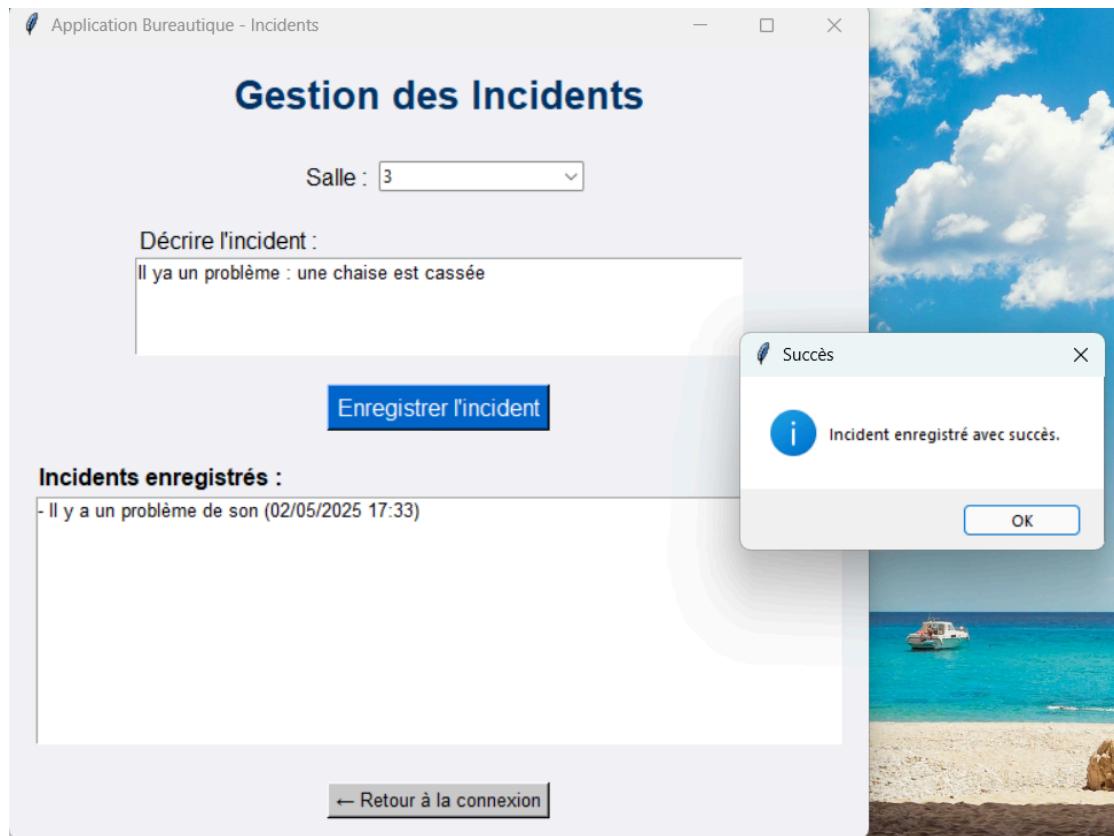
Cette capture représente l'interface de connexion de l'application bureautique. L'employé doit renseigner son adresse email et son mot de passe pour accéder à l'espace de gestion des incidents. Un système de vérification interroge la base de données pour authentifier l'utilisateur.

- Interface de déclaration d'incident

The screenshot shows a Windows application window titled "Application Bureautique - Incidents". The main title bar is light blue with the application name and standard window controls. Below the title bar, the window has a light gray header area containing the title "Gestion des Incidents" in bold blue font. In the top right corner of this header, there are three small icons: a minus sign for window size, a square for window position, and an X for close. The main content area starts with a dropdown menu labeled "Salle :". Below it is a text input field labeled "Décrire l'incident :" followed by a large empty text area for incident description. A blue button labeled "Enregistrer l'incident" is positioned below these fields. Further down, a section titled "Incidents enregistrés :" contains a large empty rectangular area. At the bottom of the window, there is a button labeled "← Retour à la connexion".

Ici, l'employé peut sélectionner une salle à partir d'un menu déroulant et saisir la description d'un incident technique observé. Une fois validée, la déclaration est automatiquement enregistrée dans la base de données MySQL avec la date et l'heure de l'incident.

- Affichage des **incidents enregistrés**



Cette section affiche l'historique des incidents enregistrés pour la salle sélectionnée. Chaque incident est listé avec sa description et la date de sa déclaration, ce qui facilite le suivi et l'intervention rapide par les techniciens.

Défis rencontrés

- **Connexion sécurisée** : assurer que les employés accèdent à leur propre espace.
- **Synchronisation en base** : récupérer dynamiquement les salles depuis MySQL.
- **Simplicité d'utilisation** : rendre l'interface intuitive même pour des utilisateurs peu expérimentés.

Perspectives d'amélioration

- Ajouter un système de **priorité d'incident** (faible, moyen, critique).
- Ajouter un **champ pour photos ou fichiers joints** lors de la déclaration.
- Permettre la **recherche avancée** par date, type ou salle.
- Générer des **rapports PDF** ou Excel avec les historiques d'incidents.
- Ajout d'un espace pour les **techniciens** qui reçoivent les alertes.

Conclusion

L'application bureautique Cinéphoria remplit efficacement sa mission de **gestion locale des incidents techniques**. Elle offre aux employés un outil simple, fiable, et connecté à une base centrale, améliorant ainsi la **réactivité et la coordination** interne pour la maintenance des salles de cinéma.