

NOM	MOHAMED
Prénom	Abdou
Date de naissance	10/01/1979

Copie à rendre

TP Concepteur développeur d'applications

CINEPHORIA

Partie 1 : Livrables

1. Entrez le lien de votre outil de gestion de projet

VISUAL STUDIO CODE, Lucidchart, PYCHARM, PHPMyAdmin, TKINTER, SQLite, MongoDB

2. Entrez le lien de votre git (*chaque lien Repo github dispose d'un README file*)

- github App web **Cinephoria** avec le rapport mis à jour :

<https://github.com/abdoma-git/Cinephoria>

Lien du Rapport Globale : https://github.com/abdoma-git/Cinephoria/blob/master/Rapport_ECF_Cinephoria_MOHAMED_Abdou.pdf

- github de l'application bureautique (avec le fichier « **Cinephoria.exe** »)
- github de l'application mobile Android (avec « **app-release.apk** » de l'application)

https://github.com/abdoma-git/Cinephoria_bureautique

https://github.com/abdoma-git/Cinephoria_Mobile

- 3. Entrez le lien de vos applications déployées en ligne

- App Web Cinephoria : <https://abdou.website/Cinephoria/Vues/>
(--simple U (web et mobile) : abdou@gmail.com / test)
(-- admin U : admin@gmail.com / admin)

- App Bureautique Cinephoria (« fichier « **Cinephoria.exe** ») :
(--Employe U : falih@gmail.com / test)

https://github.com/abdoma-git/Cinephoria_bureautique/tree/master/output/Cinephoria_Bureau

- App Mobile Cinephoria (« fichier « **app-release.apk** ») :

https://github.com/abdoma-git/Cinephoria_Mobile/blob/master/app-release.apk

Partie 2 : Planification

1. Comment avez-vous effectué la planification de votre projet ?

La planification du projet Cinéphoria a été réalisée de manière structurée, en respectant les bonnes pratiques de gestion de projet logiciel. Elle s'est articulée autour des étapes suivantes :

1. Analyse des besoins

Une première phase a été dédiée à l'identification et à la formalisation des besoins exprimés par les parties prenantes (direction, équipe technique, utilisateurs). Cela a donné lieu à la rédaction d'un cahier des charges fonctionnel, incluant toutes les User Stories décrites dans le rapport.

2. Découpage du projet en modules

Le projet a été découpé en trois modules principaux :

- Application Web (clients, employés, administrateurs)
- Application Mobile (interface utilisateur)
- Application Bureautique (employés – gestion des incidents)

Chaque module a été ensuite subdivisé en fonctionnalités livrables indépendamment, ce qui a permis d'organiser le développement par priorités (MVP).

3. Choix de l'architecture et des technologies

L'architecture MVC (Modèle – Vue – Contrôleur) a été choisie pour l'application web afin de garantir une organisation claire et modulaire. Les technologies les plus adaptées ont été sélectionnées en fonction des besoins de performance, de sécurité, et de compatibilité (PHP, MySQL, MongoDB, Flutter...).

4. Élaboration d'un planning prévisionnel

Un planning détaillé a été élaboré, basé sur un cycle de développement itératif, inspiré des principes Agile :

- Phase 1 : Conception et modélisation UML
- Phase 2 : Mise en place de l'architecture et des bases de données
- Phase 3 : Développement progressif des fonctionnalités principales
- Phase 4 : Tests unitaires et fonctionnels
- Phase 5 : Intégration, débogage, sécurisation
- Phase 6 : Préparation des livrables et documentation

Chaque phase a été validée par des tests réguliers et des revues de code pour garantir la qualité et la conformité avec les exigences.

5. Suivi de l'avancement

Le suivi du projet a été effectué à l'aide d'un tableau de bord de tâches (Kanban ou Trello), permettant de visualiser l'état de chaque fonctionnalité (À faire – En cours – Terminé), facilitant la communication et la réactivité de l'équipe.

5. Quelle méthode de gestion de projet avez-vous utilisé, et pourquoi ?

J'ai choisi d'utiliser la méthode Agile pour gérer le projet *Cinéphoria* car elle est particulièrement adaptée aux projets de développement logiciel nécessitant de la flexibilité, une adaptation continue aux besoins des utilisateurs, et une livraison progressive des fonctionnalités.

La méthode Agile m'a permis de :

- Découper le projet en itérations courtes (sprints), ce qui facilite le suivi de l'avancement et permet d'obtenir rapidement des versions fonctionnelles de l'application.
- Prioriser les fonctionnalités essentielles pour livrer un premier produit minimum viable (MVP), utilisable et testable dès les premières phases.
- Recueillir des retours réguliers à chaque étape, notamment après chaque livraison intermédiaire, ce qui a permis d'ajuster le développement en fonction des besoins réels et de corriger les erreurs rapidement.
- Améliorer la communication et la réactivité, en gardant une vision claire des tâches en cours, à faire et terminées, grâce à un tableau de bord (type Kanban ou Trello).

En résumé, la méthode Agile m'a offert une meilleure maîtrise du projet, une plus grande qualité logicielle, et une capacité d'adaptation continue, tout en respectant les délais et les objectifs du projet.

Partie 3 : Technologies utilisés

1. **Quelles technologies avez-vous utilisées ? Précisez tous les éléments qui vous ont permis de faire ce choix. Ajoutez également un paragraphe expliquant en quoi ces choix sont le plus adaptés pour la demande spécifiée dans l'énoncé.**

Pour réaliser le projet Cinéphoria, j'ai choisi des technologies modernes, accessibles et adaptées à chaque type d'application (web, mobile, bureautique), en fonction des exigences fonctionnelles, de la rapidité de développement, de la maintenabilité et de la performance attendues.

1. Application Web – Technologies choisies :

- Langage serveur : PHP
 - Choisi pour sa simplicité, sa compatibilité native avec MySQL et sa large communauté.
 - L'architecture MVC a permis de bien structurer le projet (Modèle – Vue – Contrôleur).
- Base de données : MySQL
 - Parfaite pour gérer les données relationnelles et structurées (utilisateurs, films, réservations...).
 - Fiabilité, sécurité et outils d'administration (phpMyAdmin).
- MongoDB
 - Utilisé pour les statistiques et le tableau de bord grâce à sa capacité à manipuler des données non structurées rapidement.
- Front-End : HTML5 / CSS3 / Bootstrap / JavaScript / jQuery
 - Pour un rendu responsive, clair et agréable sur tous types d'écrans.
 - Bootstrap a permis un gain de temps dans le design et la compatibilité mobile.
 - jQuery a facilité l'implémentation rapide de l'interactivité (menus dynamiques, formulaires...).

2. Application Mobile – Technologies choisies :

- Flutter avec Dart
 - Choisi pour le développement multiplateforme (Android et iOS) à partir d'une seule base de code.
 - Permet un rendu fluide, moderne et rapide à déployer.
- API REST en PHP

Pour assurer la communication sécurisée avec le serveur via HTTP (ex : récupération des réservations d'un utilisateur).

- Permet de centraliser les données entre les différentes applications.
- qr_flutter
 - Génère les QR codes des réservations, nécessaires pour l'entrée en salle.
 -

3. Application Bureautique – Technologies choisies :

- Python + Tkinter
 - Choisi pour sa simplicité d'utilisation, son intégration facile à MySQL via mysql.connector, et sa capacité à créer une interface graphique légère.
 - Tkinter est natif à Python, ce qui facilite l'installation et réduit les dépendances.

Ces choix technologiques sont parfaitement alignés avec les besoins définis dans l'énoncé :

- PHP + MySQL sont des technologies éprouvées, très utilisées dans les systèmes de réservation, faciles à héberger et bien documentées.
- Flutter permet de proposer une expérience mobile cohérente, essentielle aujourd'hui pour la réservation de billets en mobilité.
- Tkinter s'adapte bien aux applications bureautiques internes, ne nécessite pas de déploiement web, et permet aux employés de travailler localement sans connexion constante.
- L'usage combiné de MongoDB pour les données analytiques et de MySQL pour les données transactionnelles permet d'avoir à la fois performance et robustesse.

Ce choix de stack permet aussi une modularité : chaque application (web, mobile, desktop) peut évoluer indépendamment tout en partageant le même cœur de données.

2. Quelles mesures avez-vous prises afin de protéger vos applications des potentielles vulnérabilités de sécurité ? (Injection SQL, failles XSS, ...)

Pour protéger mes applications contre les potentielles vulnérabilités de sécurité, j'ai mis en place plusieurs mesures de sécurité, notamment en utilisant des fonctions et des bonnes pratiques de développement sécurisé en PHP.

Pour prévenir les attaques d'injection SQL, j'ai utilisé des requêtes préparées avec des déclarations PDO (PHP Data Objects) au lieu de concaténer directement des données utilisateur dans mes requêtes SQL. Par exemple :

```
$stmt = $pdo->prepare("SELECT * FROM utilisateurs WHERE nom = :nom");
$stmt->bindParam(':nom', $nom, PDO::PARAM_STR);
$stmt->execute();
```

Cela garantit que les données utilisateur sont correctement échappées et que les attaquants ne peuvent pas injecter du code SQL malveillant.

Pour lutter contre les failles XSS j'ai veillé à échapper correctement les données affichées dans le HTML en utilisant des fonctions telles que « htmlspecialchars() » pour convertir les caractères spéciaux en entités HTML. Par exemple :

```
echo "<p>" . htmlspecialchars($texte) . "</p>";
```

Cela empêche l'exécution de scripts malveillants dans le navigateur de l'utilisateur. En plus de ces mesures, j'ai également mis en place une gestion appropriée des sessions

utilisateur, des filtres de validation des données entrantes, et j'ai régulièrement mis à jour les bibliothèques et les composants tiers utilisés dans mes applications pour corriger les éventuelles vulnérabilités connues. En adoptant ces meilleures pratiques de sécurité, j'ai contribué à rendre mes applications plus résistantes aux attaques potentielles.

3. Décrivez les tests applicatifs que vous effectuez et expliquez dans quel but.

Objectif des tests applicatifs

Les tests applicatifs ont pour but de **vérifier que chaque fonctionnalité de l'application fonctionne comme prévu**, sans bug, et de garantir une **expérience utilisateur fluide et sécurisée**. Ils ont été réalisés tout au long du développement pour :

- Valider la **conformité des fonctionnalités** par rapport au cahier des charges
- **Détecter les erreurs** ou comportements inattendus
- S'assurer de la **sécurité** (résistance aux attaques)
- Garantir une **cohérence des données**
- Offrir une **navigation intuitive et sans blocage**

Types de tests effectués

• 1. Tests fonctionnels

Pour s'assurer que chaque fonctionnalité répond aux besoins de l'utilisateur.

- Vérification du bon fonctionnement de la **connexion / déconnexion**
- Réservation de séance : test du **choix des sièges**, de la **disponibilité** et de la **confirmation**
- Affichage des films, filtres, détails
- Gestion des incidents par les employés (interface bureautique)
- Affichage des réservations et QR code dans l'application mobile

Exemple : tester qu'un utilisateur connecté peut réserver une séance et retrouver sa réservation dans l'historique.

• 2. Tests de validation de formulaire

Pour éviter les données erronées ou malveillantes.

- Champs requis non remplis
- Formats invalides (e-mail, numéro, dates)
- Tentatives d'injection de scripts ou de caractères spéciaux

Exemple : test d'un champ "nom" contenant un script XSS (<script>alert(1)</script>)
→ le script ne doit pas s'exécuter.

• 3. Tests de sécurité

Pour éviter les attaques courantes.

- **Injection SQL** : en tentant d'insérer ' OR 1=1 -- dans les champs de formulaire
- **Faible XSS** : injection de scripts dans des zones de texte
- **Accès non autorisé** à des pages d'administration ou de gestion sans authentification

Résultat : les requêtes préparées et l'échappement des caractères ont bloqué ces attaques.

• 4. Tests de compatibilité

Pour garantir un bon affichage sur tous les appareils.

- Affichage web sur différents navigateurs (Chrome, Firefox, Edge)
 - Version responsive sur smartphone / tablette
 - Application mobile testée sur **émulateur Android** et **appareil physique**
- **5. Tests d'ergonomie et d'expérience utilisateur (UX)**

Pour s'assurer que l'interface est simple, claire et agréable.

- Navigation intuitive
- Boutons et textes bien positionnés
- Réactivité et lisibilité sur différents formats d'écran

Conclusion

L'ensemble de ces tests a permis de livrer une application **stable, sécurisée et fonctionnelle**, répondant aux **attentes des utilisateurs finaux**, des employés et des administrateurs.

4. Comment avez-vous effectué le déploiement de votre application ?

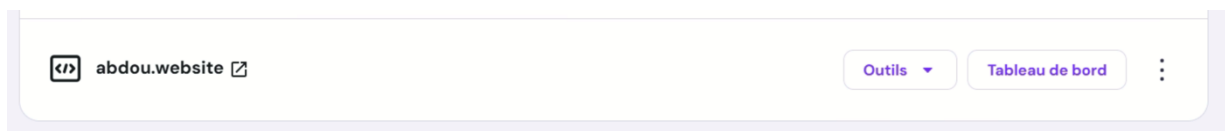
Déploiement de l'application Web CINEPHORIA sur Hostinger



Le nom de domaine pour mes applications web est : www.abdou.website/Cinephoria/Vues

Connexion à Hostinger

Je me suis connecté à mon compte Hostinger, puis j'ai accédé à la section « Hébergements » depuis le hPanel. Ensuite, j'ai sélectionné mon domaine principal : abdou.website.



abdou.website
Créé le : 2024-02-18

Nom de domaine
Actif

Hébergement
Actif

Email gratuit
En attente de configuration

Sauvegardes
Quotidiennement

Analyse de PageSpeed

99

Appareil de bureau
Dernière analyse le 2025-07-07

Gestionnaire de fichiers

Bases de données

Installateur automatique

Vider le cache
Visitez votre site web sans cache ou videz-le en un simple clic.

[Aperçu sans cache](#)
[Vider le cache](#)

Importation des fichiers dans public_html

À l'aide du gestionnaire de fichiers, j'ai ouvert le dossier public_html. J'y ai téléversé tout mon dossier Cinephoria. Si nécessaire, je peux aussi compresser le dossier en .zip, le téléverser et l'extraire directement dans l'interface.

Le chemin d'accès à mon application devient donc : `www.abdou.website/Cinephoria/Vues/`

Nom ↑	Taille	Dernière modification
public_html	-	il y a 15 jours drwxr-xr-x

Création de la base de données

Depuis la section « Base de données MySQL », j'ai créé une base de données avec :

- Un nom de base de données : `u330942333_CinephoriaA`
- Un nom d'utilisateur : `u330942333_MohamedAbdou`
- Un mot de passe sécurisé

Base de données MySQL ↕	Utilisateur MySQL ↕	Créé le ↕	Site web	
u330942333_CinephoriaA 1 MB	u330942333_Mohamed Abdou	2025-07-04	abdou.website	Accéder à phpMyAdmin

J'ai noté soigneusement ces informations, car elles seront utilisées dans mon fichier de connexion.

Accès à phpMyAdmin

Une fois la base créée, j'ai accédé à phpMyAdmin via le bouton dédié dans le hPanel. J'ai ensuite importé le fichier .sql de mon application pour créer les tables nécessaires.

Configuration du fichier connexion.php

Voici le contenu que j'ai utilisé dans le fichier includes/connexion.php :
J'ai bien remplacé les champs par mes informations réelles.

```
🏠 > public_html > Cinephoria > connexion.php
1 k?php
2 $servername = 'localhost';
3 $dbname = "u330942333_CinephoriaA";
4 $username = "u330942333_MohamedAbdou";
5 $password = "0330942333";
6
7 try {
8
9     $pdo = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
10    $pdo -> setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
11
12    // echo "Connexion reussite avec la base de donnees " . $dbname . "\n" ;
13
14 } catch (PDOException $erreur) {
15
16    echo "Erreur de connexion : ". $erreur->getMessage();
17 }
18 ?>
```

Conclusion

Grâce à ces étapes, j'ai pu héberger avec succès mon application PHP sur Hostinger et la rendre accessible depuis le lien : www.abdou.website/Cinephoria/Vues.

Deploiement de l'application Mobile Android Flutter CINEPHORIA



Mon application **CINEPHORIA** est une application mobile développée sur **Flutter** pour la plateforme **Android**. Elle permet de consulter la liste des reservation depuis la base de données de l'application Web associée, **CINEPHORIA**.

Étape 1 : Génération du fichier APK

Le fichier APK, nécessaire pour installer l'application sur un appareil Android, a été généré en mode *release* avec la commande suivante sur le terminal :

flutter build apk --release

```

37         if (response.statusCode == 200) {
38             final data = json.decode(response.body);
39             if (data['reservations'] != null) {
40                 setState(() {
41                     seances = (data['reservations'] as List)

```

PROBLEMS 6 OUTPUT **TERMINAL** PORTS

▼ **TERMINAL**

```

leman@abdou-pc MINGW64 ~/Desktop/cinephoria (master)
$ flutter build apk --release

```

Cette commande crée un fichier exécutable APK optimisé pour les utilisateurs finaux.

Étape 2 : Téléchargement de l'application

Le fichier APK généré a été mis en ligne sur mon profil GitHub, et vous pouvez le télécharger en suivant ce lien : [Cinephoria Mobile App APK](#).

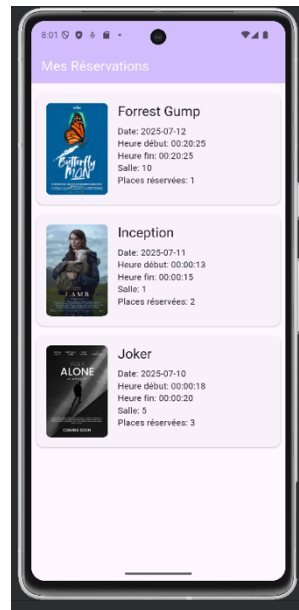
Étape 3 : Installation de l'application

Pour installer l'application sur un appareil Android :

1. **Téléchargez** le fichier APK à partir du lien GitHub ci-dessus.
2. **Ouvrez** le fichier APK téléchargé sur votre téléphone et autorisez l'installation d'applications provenant de sources inconnues si nécessaire.
3. Une fois l'installation terminée, vous verrez l'icône de l'application **Cinephoria** sur votre écran d'accueil

Étape 4 : Lancement de l'application

1. Appuyez sur l'icône **Cinephoria** pour lancer l'application.
2. À l'ouverture, l'application affiche une page d'accueil listant les réservations. Ces données sont récupérées directement à partir de la base de données de l'application Web **CINEPHORIA**.



Deploiment de l'application Bureautique Tkinter CINEPHORIA



L'application CINEPHORIA est une application bureautique développée en Python avec le module Tkinter pour le système d'exploitation Windows.

Étape 1 : Génération du fichier .exe

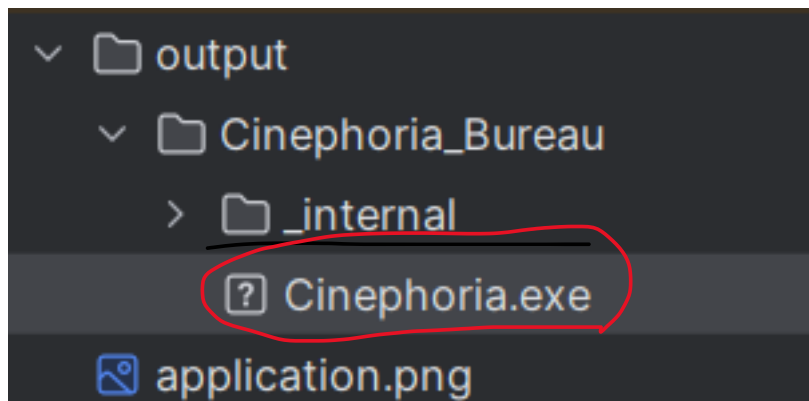
Pour permettre une utilisation directe sous Windows, le fichier exécutable .exe a été généré à l'aide de la commande **Auto-py-to-exe**. Cette commande permet de transformer le script Python en un exécutable autonome, optimisé pour les utilisateurs finaux.

Auto-py-to-exe

```
(.venv) PS C:\Users\leman\PycharmProjects\Mon_projet\pythonProject>  
(.venv) PS C:\Users\leman\PycharmProjects\Mon_projet\pythonProject>  
(.venv) PS C:\Users\leman\PycharmProjects\Mon_projet\pythonProject> auto-py-to-exe
```

Dans la fenêtre ci-dessous, configurez les éléments suivants pour préparer l'application :

- **Script Location** : indiquez le fichier principal de l'application, c'est-à-dire le fichier Python qui lancera l'application. Le nom de ce fichier deviendra le nom par défaut de l'application lors de sa création.



Cette étape génère un fichier exécutable .exe contenant tous les fichiers nécessaires au bon fonctionnement de l'application.

Étape 2 : Téléchargement de l'Application

Le fichier exécutable .exe a été compressé en archive .rar et mis à disposition sur mon profil GitHub. Pour télécharger l'application, cliquez sur le lien suivant : [CINEPHORIA - Application Bureau](#).

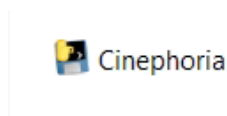
Étape 3 : Installation de l'Application

Pour installer l'application **CINEPHORIA** sur un ordinateur sous Windows :

1. **Téléchargez** l'archive CINEPHORIA.rar depuis le lien GitHub fourni.
2. Extrayez le contenu de l'archive à l'aide d'un logiciel de décompression (comme WinRAR ou 7-Zip).
3. Ouvrez le fichier CINEPHORIA.exe et suivez les instructions pour installer l'application sur votre PC.
4. Une fois l'installation terminée, une icône **CINEPHORIA** apparaîtra sur votre bureau, comme illustré ci-dessous.

Étape 4 : Lancement de l'Application

1. **Double-cliquez** sur l'icône **CINEPHORIA** pour démarrer l'application.



2. À l'ouverture, l'application affiche une page de connexion. Depuis cette page, les utilisateurs peuvent créer un compte (pour les nouveaux utilisateurs) ou se connecter directement s'ils possèdent déjà des identifiants.

Partie 4 : Informations complémentaires

1. Avez-vous quelque chose à partager avec votre correcteur ?

Oui, je reste entièrement disponible si vous avez besoin d'informations complémentaires ou si un document venait à manquer. N'hésitez pas à me contacter pour toute précision, je serai ravi de vous apporter les éléments nécessaires à la bonne évaluation de mon travail.