

Documentation de la Gestion de Projet – Cinéphoria

Méthodologie choisie : Agile - Scrum

Le projet **Cinéphoria** a été mené en suivant la **méthodologie Agile Scrum**, idéale pour un développement itératif et incrémental. Cette approche a permis :

- Un découpage du projet en **sprints** courts et itératifs.
- Une adaptation rapide aux éventuels changements.
- Une communication claire avec des livrables intermédiaires.

Rôles dans Scrum :

- **Product Owner (PO)** : Définition des besoins et priorisation des fonctionnalités.
- **Scrum Master** : Garantie du bon déroulement de la méthode Agile.
- **Développeur** : Conception et implémentation technique.

Outil de gestion de projet : GitHub Projects

L'intégralité de la gestion de projet a été réalisée sur **GitHub Projects**, intégré au dépôt Git principal.

Lien vers le dépôt GitHub : [Cinéphoria - GitHub](#)

Structure du Kanban :

1. Backlog (Fonctionnalités prévues) :

- Authentification des utilisateurs
- Gestion des réservations
- Ajout et gestion des films
- Affichage des séances
- Gestion des utilisateurs et rôles
- Génération de QR Code pour les billets

2. À faire (Sprint actuel) :

- Connexion/Inscription utilisateur
- Intégration API backend avec frontend
- Création des pages React (Home, Films, Séances, Réservations)

3. En cours :

- Développement du frontend avec **React + Vite**
- Liaison API backend
- Sécurisation avec JWT

4. Terminé :

- Backend Express.js opérationnel
 - Base de données SQL (schema.sql et transactions.sql)
 - Déploiement du backend
 - Charte graphique et manuel d'utilisation
-

Gestion Git et déploiement

Structure des branches Git :

- main : Branche principale stable.
- develop : Branche de développement intégrant les nouvelles fonctionnalités.
- feature/* : Branches spécifiques à chaque fonctionnalité. Exemple : feature/auth, feature/films.

Cycle de développement Git :

- Création d'une branche feature/* pour chaque fonctionnalité.
- Développement et tests sur cette branche.
- **Merge** vers develop après validation.
- Tests finaux sur develop, puis **merge** vers main pour déploiement.

CI/CD & Déploiement :

- Pipeline GitHub Actions pour déployer automatiquement la branche main.
 - **npm run build** pour le frontend et **Docker** pour le backend.
-

Suivi et Avancement du Projet

Timeline :

- **Jour 1-2** : Mise en place du backend (Express.js, MySQL).
- **Jour 3** : Création de la base de données SQL.

- **Jour 4** : Création du frontend (React + Vite).
- **Jour 5** : Intégration API et frontend.
- **Jour 6** : Tests, finalisation et déploiement.

Suivi Agile :

- Daily Scrum pour l'avancement.
- Sprint Review et Retrospective en fin de développement.

Livrables fournis

- 📁 **Dépôt GitHub** public avec historique Git propre.
- 📄 **Fichiers SQL** complets (schéma + transactions).
- 🎨 **Charte graphique** (polices, couleurs, wireframes).
- 📄 **Manuel d'utilisation** PDF.
- 📄 **Documentation technique** (Architecture, MCD, CI/CD, tests).