{EPITECH}

KICK-OFF : MY CINEMA_

<Réalisation d’une API en PHP en POO/MVC et MySQL>

SOMMAIRE_

01 PRESENTATION DU PROJET

MY CINEMA_

< My Cinema est une **dashboard d'administration** permettant de gérer l'activité de son cinéma indépendant. Il doit permettre de gérer les films à l'affiche, les séances, et les salles du cinéma. >

< Contraintes techniques : >

- Backend (PHP, MySQL) et frontend (HTML, CSS, JS) indépendants
- Mise en place d'une architecture MVC, avec POO
- Développement du backend sous forme d'API
- Aucun framework PHP autorisé

02 POURQUOI UNE API ?_

CONCEVOIR UNE API_

< **API** *(Application Programming Interface)* : interface qui permet à des applications de **communiquer entre elles** de manière structurée, le plus souvent via des requêtes **HTTP** et des données **JSON**.
Permet de séparer le front-end du back-end, afin que chacun puisse évoluer indépendamment.
Objectif : centraliser la logique métier et les données, tout en rendant l'application plus maintenable, réutilisable et évolutive. />

< Différents types d'API : >

- **Open APIs** : API gouvernementales, OpenStreetMap (cartes, lieux), OpenWeather (météo),
- **Product APIS** : Stripe (paiements), Spotify (musique), YouSign (signatures électroniques), ...
- **Backend-for-frontend** (BFF): API interne conçue pour soi-même (exemple : **My Cinéma**)
- **Partner APIs** : API privée pour échanger entre 2 partenaires (exemple : réservation hôtelière depuis un site de réservation de vols).

03 L'ARCHITECTURE_

L'ARCHITECTURE

- **Architecture MVC** à mettre en oeuvre sur le projet :
  - Modèle : gestion des données
  - Vue : gestion de l'affichage *(sauf API)*
  - Contrôleur : Point "central" (lien modèle ⇔ vue)
- Utilisation de la **POO**
- **Objectif** : Réaliser un backend structuré et maintenable

Appuyez-vous sur le **bootstrap** du projet pour définir votre architecture.

04 ET ENSUITE ?_

ET ENSUITE ?

- **Aujourd'hui** : réalisation manuelle d'une architecture
- **Par la suite** : utilisation de frameworks :
  - Architecture (MVC) déjà en place
  - Utilisation de la POO
  - Intègrent une "boîte à outil" pour développeur *(interactions BDD, gestion d'utilisateurs, contrôle d'accès et de données, gestion d'URL, etc)*
  - Permettent la construction de page web, d'API, de scripts exécutables, ...
  - Très utilisés en entreprise

ET ENSUITE ?_

< La maîtrise de **l'architecture MVC** et de la **POO** est un prérequis indispensable à l'utilisation de frameworks PHP.

L'objectif pédagogique principal de ce projet est de vous **préparer** à l'utilisation de frameworks. />