

## Workshop : Développement d'une Mini-Application Web

### Objectif Général

Chaque squad doit concevoir et développer une application web complète en **2 jours**, en intégrant **POO PHP, UML, Laravel (API) et JavaScript (DOM, Ajax)**. L'objectif est de renforcer les compétences à travers **un projet structuré** appliquant les bonnes pratiques de développement.

---

## Plan du Workshop

### **1** Phase de Conception (Matin - Jour 1)

- ◆ **Définition du Besoin & Rédaction d'un Mini Cahier des Charges**
    - Identification du problème à résoudre.
    - Définition des utilisateurs et des fonctionnalités principales.
  - ◆ **Modélisation en UML**
    - **Diagramme de Cas d'Utilisation** (définir les interactions utilisateur).
    - **Diagramme de Classes** (modéliser la structure de l'application).
- 

### **2** Phase Backend : Développement de l'API avec Laravel (Après-midi - Jour 1 & Matin - Jour 2)

- ◆ **Implémentation des Modèles & POO PHP**
  - Création des **modèles Laravel (Eloquent)** en respectant les principes de **POO avancée** (héritage, encapsulation, polymorphisme...).
  - Gestion des **relations entre les entités** via les **migrations** et les **clés étrangères**.
- ◆ **Développement des Routes et Contrôleurs**
  - Mise en place des **routes API** en respectant les conventions REST.
  - Création des **Controllers** avec la logique métier appliquant **les principes SOLID**.
- ◆ **Interactions avec la Base de Données**
  - Manipulation des données avec **Eloquent ORM**.

- Utilisation de **fonctions SQL avancées** (jointures, agrégats).
- 

### ③ Phase Frontend : Interface Dynamique avec JavaScript (Après-midi - Jour 2)

#### ♦ Création de l'Interface avec JavaScript (DOM, Ajax)

- Intégration d'une **interface dynamique** qui communique avec l'API Laravel.
- Utilisation d'**Ajax pour l'échange de données** avec l'API.
- Gestion des erreurs et retour utilisateur.




#### ♦ Finalisation et Présentation des Projets

- Chaque squad prépare une **démonstration** de son projet.
  - Présentation des choix techniques et des défis rencontrés.
- 

## Exigences Techniques

- ✓ **POO avancée en PHP** (héritage, abstraction, polymorphisme, encapsulation).
  - ✓ **Modélisation UML** (diagramme de cas d'utilisation, diagramme de classes).
  - ✓ **Base de données relationnelle** avec SQL (relations, clés étrangères, requêtes optimisées).
  - ✓ **Développement d'une API avec Laravel** en suivant les bonnes pratiques REST.
  - ✓ **Interface utilisateur dynamique** avec **JavaScript (DOM, Ajax)**.
  - ✓ **Respect des bonnes pratiques** (Clean Code, SOLID, MVC).
- 

## Contraintes et Organisation

-  **Chaque squad travaille sur un projet unique.**
  -  **Les projets doivent être finalisés et présentés dans le délai imparti (2 jours).**
  -  **L'architecture MVC doit être respectée dans Laravel.**
-

## Exemples de Projets

- ♦ Plateforme de mise en relation entre associations et bénévoles.
- ♦ Application de gestion de cours et d'évaluations pour une école.
- ♦ Système de gestion d'événements et d'inscriptions.
- ♦ Application de signalement des zones polluées.
- ♦ Toute autre idée, validée par le formateur.