

Projet de session – Plan de travail

Voici une suggestion du plan de travail à suivre lors du développement du projet :

Étape	Indications
1. Identifier les fonctionnalités offertes par l'application et le public cible.	À partir de la description du projet
2. Identifier les différents acteurs et de quelle(s) façon(s) ils utiliseront le système (c'est-à-dire, les fonctionnalités que le système offre à chacun des acteurs).	À partir de la description du projet
3. Effectuer la modélisation des données et réaliser le diagramme entités-associations puis en déduire le schéma de la base de données.	<ul style="list-style-type: none"> • Modélisation de données, • Diagramme entités-associations, • Normalisation
4. Implémenter la base de données (sous MySQL).	<ul style="list-style-type: none"> • LDD de SQL, • MySQL, • PhpMyAdmin • Kit WAMP (UwAmp, XAMPP,...)
5. Faire une esquisse de la page d'accueil et des autres pages, si elles ont un gabarit différent.	Outil de prototypage d'interface (exemples : <i>Balsamiq</i> https://balsamiq.com/ , <i>Pencil</i>)
6. Identifier les différents écrans et les éléments composant les interfaces.	Outil de prototypage d'interface (exemples : <i>Balsamiq</i> , <i>Pencil</i>)
7. Commencer le codage et l'implémentation de l'interface (HTML 5 et CSS 3 ou, de préférence, <i>Bootstrap</i> ou équivalent).	<ul style="list-style-type: none"> • HTML 5, • CSS 3, • Bootstrap ou un autre framework CSS (conseillé)
8. Créer les classes entités et les DAO.	<ul style="list-style-type: none"> • Programmation objet en PHP, • Pattern DAO, • LMD de SQL (SELECT, INSERT, DELETE, UPDATE), • Pattern Singleton, • Requêtes paramétrées
Evaluation d'état d'avancement : à confirmer/infirmier Date : à préciser, le cas échéant Contenu : <ul style="list-style-type: none"> - Modèle de données, - Écrans des interfaces 	
9. Mettre en place l'architecture MVC Model 2 au niveau de la couche présentation (contrôleur frontal, action-builder, ...)	<ul style="list-style-type: none"> • Programmation objet en PHP, • Pattern MVC2 (Model 2, Front controller), • Pattern Builder, • Inclusion (include, require, ...)
10. Implémenter les fonctionnalités de l'application de manière itérative et incrémentale	<ul style="list-style-type: none"> • Programmation objet en PHP, • Couche métier (services) • Pages PHP (vues),

- Redirection,
- Injection de dépendances,
- Problème du double-submit,
- ...

Evaluation finale :

Date : à venir

Contenu :

- Tout