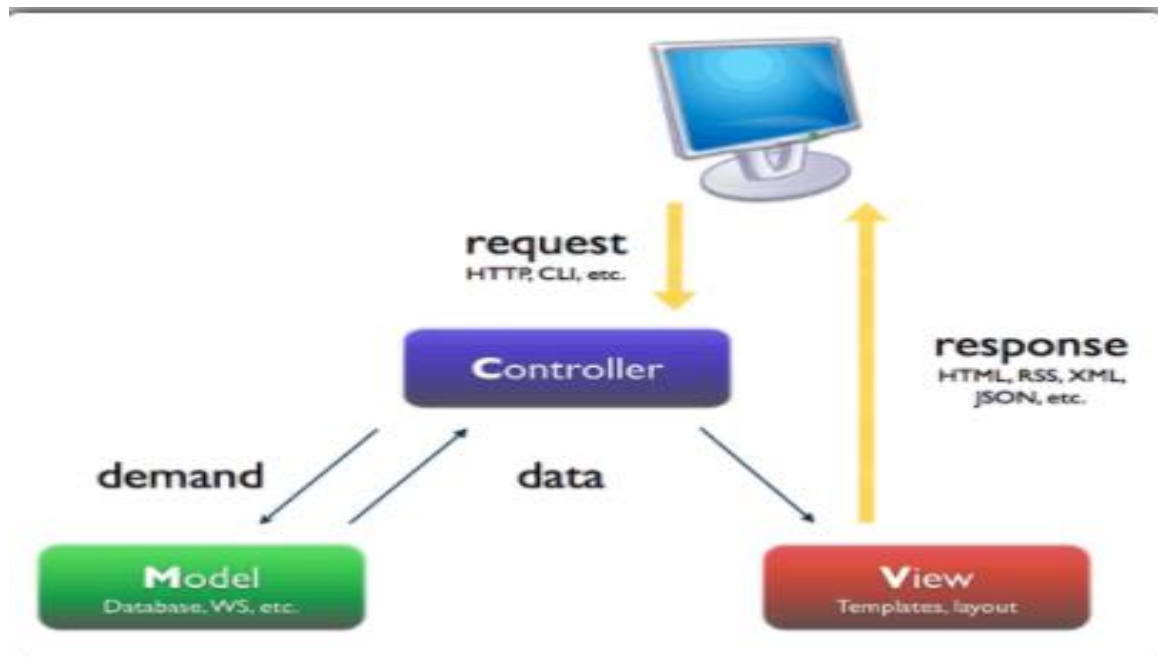


Rapport :

Atelier 1 : Mise en place d'un CRUD Etudiant



Nom : AOULEK

Prénom : Ibtissam

Présentation :

- l'objectif principal de cet atelier est la mise en place d'un CRUD contenant les opérations. création, mise à jour suppression et sélection à travers le framework développé from scratch.
- pour ce objectif on a travaillé avec un framework php MVC orienté Objet.

Le framework MVC orienté objet décrit une manière d'architecturer une application informatique en la décomposant en trois sous-parties :

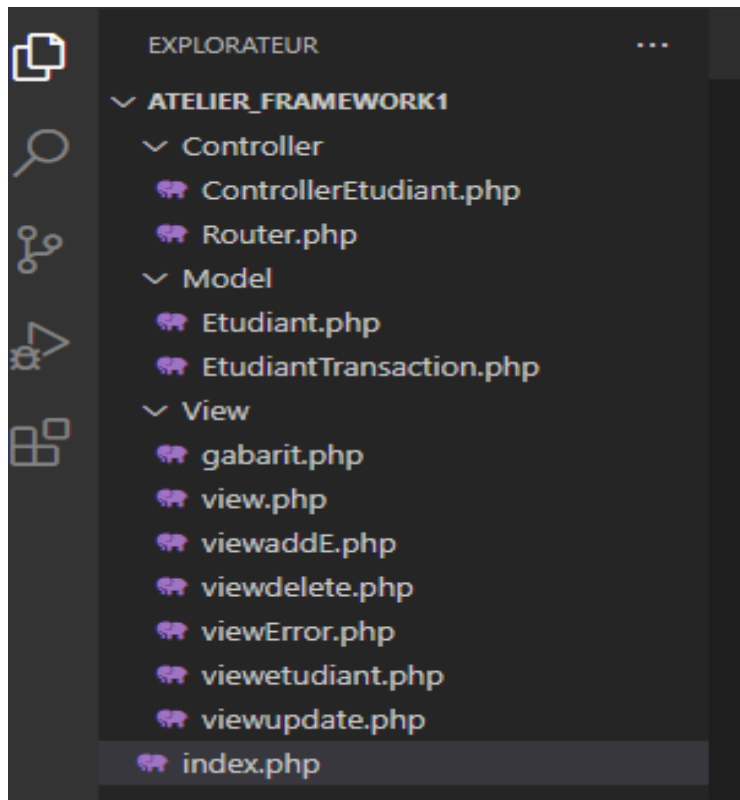
- la partie Modèle ;
- la partie Vue ;
- la partie Contrôleur.

Ce modèle de conception (« *design pattern* ») a été imaginé à la fin des années 1970 pour le langage Smalltalk afin de bien séparer le code de l'interface graphique de la logique applicative. Il est utilisé dans de très nombreux langages : bibliothèques Swing et Model 2 (JSP) de Java, *frameworks* PHP, ASP.NET MVC, etc.

- La partie **Modèle** d'une architecture MVC encapsule la logique métier (« *business logic* ») ainsi que l'accès aux données. Il peut s'agir d'un ensemble de fonctions (Modèle procédural) ou de classes (Modèle orienté objet).
- La partie **Vue** s'occupe des interactions avec l'utilisateur : présentation, saisie et validation des données.
- La partie **Contrôleur** gère la dynamique de l'application. Elle fait le lien entre l'utilisateur et le reste de l'application.

Notre framework MVC nous a donné plusieurs services essentiels : configuration, interactions avec la base de données, routage des requêtes, URL génériques, sécurisation des données

les résultats des fichiers obtenus dans notre projet :



Atelier Framework1 :

Router.php :

- ce fichier permet de rentre le routage automatique et générique.
- La méthode principale routerRequete() de cette classe instancie un objet ContollerEtudiant en fusionnant les données des variables superglobales \$_GET et \$_POST, afin de pouvoir analyser toute requête issue soit d'une commande HTTP GET, soit d'une commande HTTP POST.

Ensuite, cette méthode fait appel à deux méthodes internes afin d'instancier le contrôleur approprié et d'exécuter l'action correspondant à la requête reçue.

ControllerEtudiant.php :

- ce fichier contient des méthodes public qui seront appelée après chaque clic de l'utilisateur on peut dire c'est lui qui gère l'application selon les besoin de l'ulisateur ; après le fichier ContollerEtudiant dirige l'utilisateur vers la bonne view soit pour modifier, supprimer ou ajouter.

Le fichier etudiant.php :

- contient une classe qui définit un étudiant et ces caractéristiques.

le fichier EtudiantTranslation :

- contient tous les opérations CRUD sur la base de données

un répertoire View/ contient toutes les vues :

- chaque vue est définie dans un sous-répertoire de View/ contient les interfaces qui existent dans notre **CRUD**.
- les fichiers gabarit.php (gabarit commun à toutes les vues) et viewError.php (affichage du message d'erreur) sont stockés dans le répertoire **View/**.

le fichier index.php qui contient l'action par défaut.

le résultat de ce framework MVC est :

Mise en place d'un CRUD Etudiant

Ajouter

| Nom | Prenom | Age | CNE | Operations |
|--------|----------|-----|---------|--|
| AOULEK | IBTISSAM | 20 | jhik52 | <div>Modifier</div> <div>Supprimer</div> |
| aoulek | khadija | 19 | kjhh589 | <div>Modifier</div> <div>Supprimer</div> |

Mise en place d'un CRUD Etudiant

Nom

AOULEK

Prenom

IBTISSAM

Age

20

CNE

jhik52

Modifier

Mise en place d'un CRUD Etudiant

Nom:

Prenom:

Age:

CNE:

Ajouter

