



# Mendool

Présenté par Joanna Attali



A smiling man with a beard, wearing a dark suit jacket over a light-colored shirt, is positioned on the left side of the image. He is leaning forward slightly, with his right arm extended towards the right. The background is a solid dark green color.

# DEMANDE DU CLIENT

- SITE VITRINE
- ACCES ADMIN / GESTION DE SES PROJETS
- ACCES VISITEURS / CREATION DE COMMENTAIRES SOUS VIDEOS
- MAQUETTE FIGMA FOURNIE





[Accueil](#) [Mes Tournées](#) [Mes Sketches](#) [Mes Chroniques](#) [Se connecter](#)



## La vie d'artiste



"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum." Section 1.10.32 of "de Finibus Bonorum

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum." Section 1.10.32 of "de Finibus Bonorum

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum." Section 1.10.32 of "de Finibus Bonorum

Facebook Icon

Instagram Icon

Tiktok Icon

Twitter Icon



[Accueil](#) [Mes Tournées](#) [Mes Sketches](#) [Mes Chroniques](#) [Se connecter](#)



## Mes tournées

Endroit de la tournée

Date

Heure

reserver

Endroit de la tournée

Date

Heure

reserver

Endroit de la tournée

Date

Heure

reserver

Facebook Icon

Instagram Icon

Tiktok Icon





## Mes sketches



Facebook Icon

Instagram Icon

Tiktok Icon

Twitter Icon



Facebook Icon

Instagram Icon

Tiktok Icon



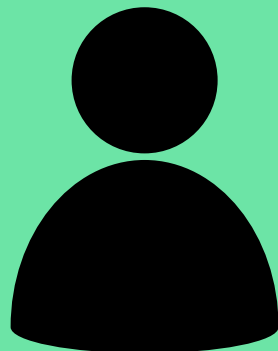
Accueil



Mes tournées



- Liste
- Ajout
- Supression
- Modification des tournées

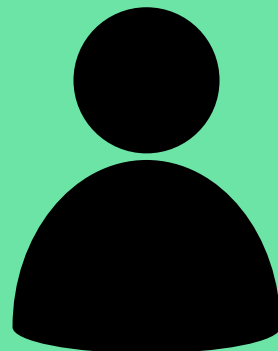


- Réserver une place sur le site de la Fnac

Mes sketches



- Liste
- Ajout
- Suppression
- Modification des sketches

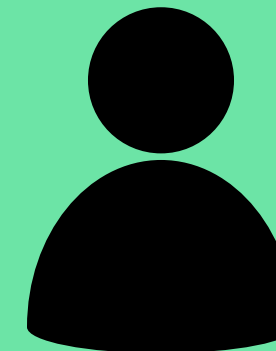


- Commenter un sketch (sous la vue d'un sketch)

Mes chroniques



- Liste
- Ajout
- Suppression
- Modification des chroniques



- Commenter une chronique (sous la vue d'une chronique)

- Inscription
- Connexion

## Langages utilisés

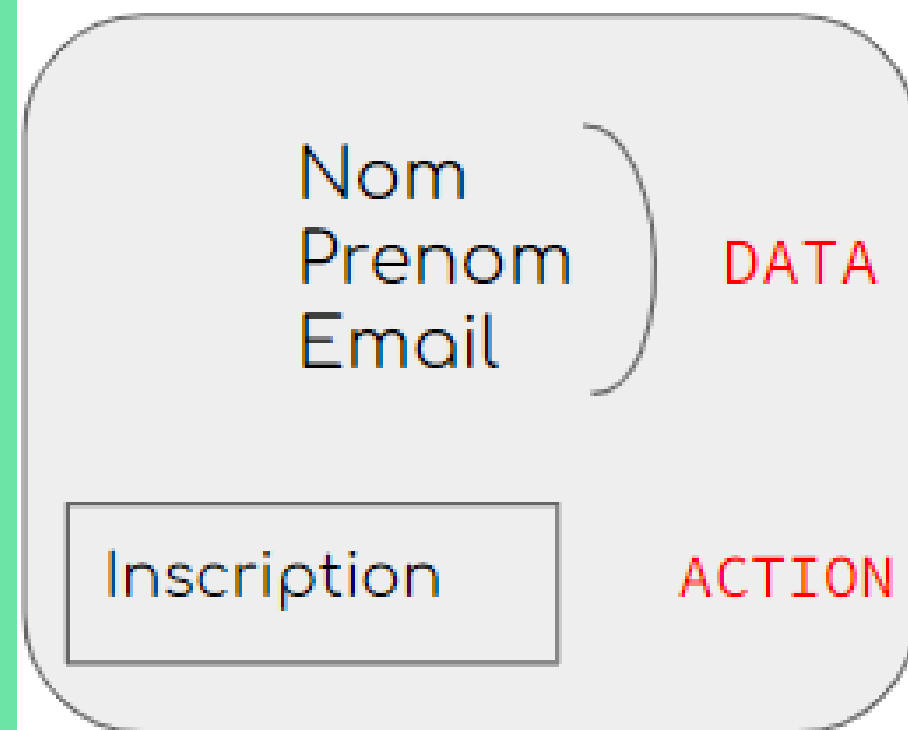


## Outils utilisés

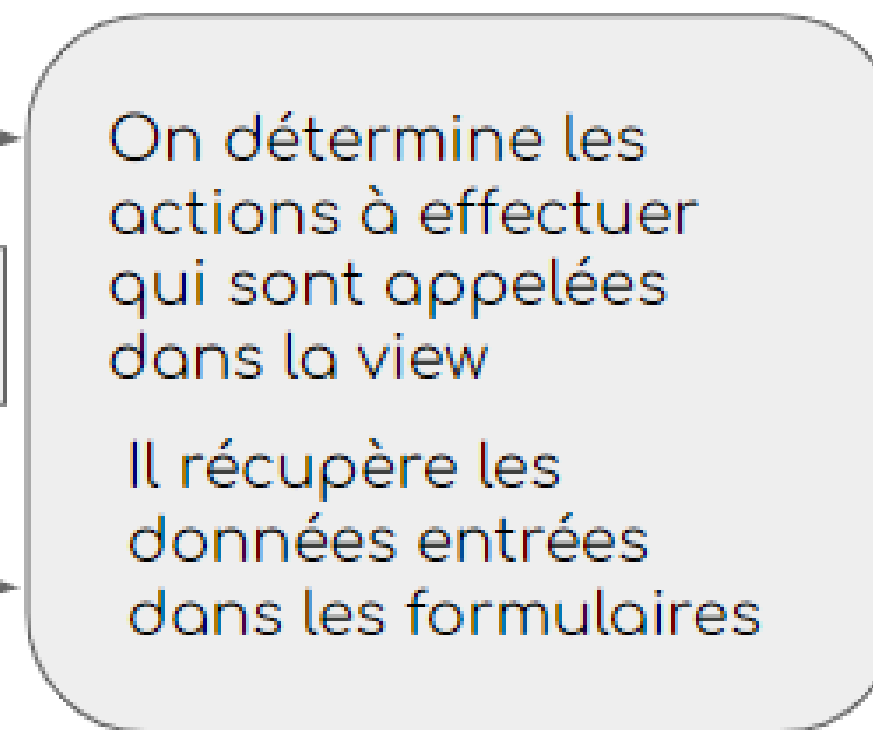
- JMerise
- PhpMyAdmin
- MVC
- POO
- CRUD

# ARCHITECTURE MVC

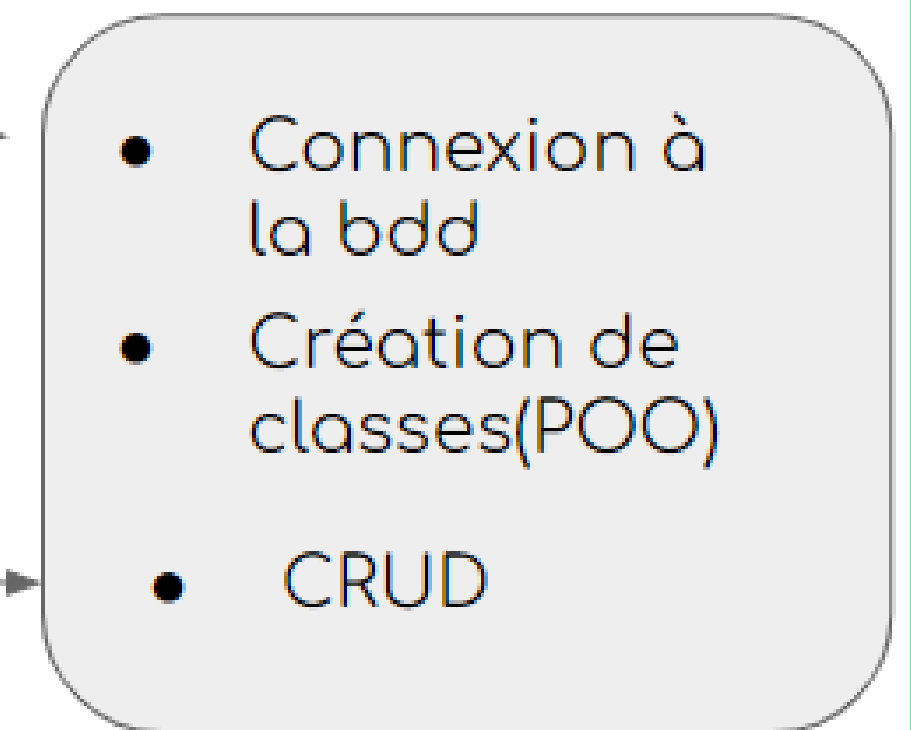
## VIEW



## CONTROLLER



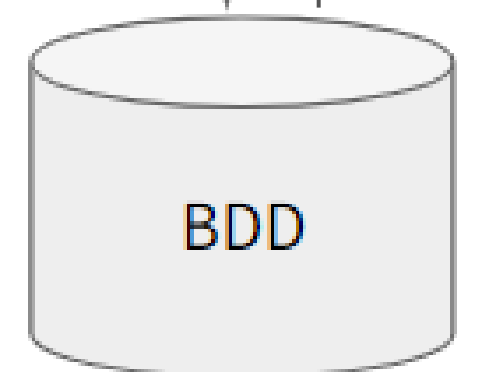
## MODEL



Affichage de la page avec les infos récupérées

- Liste des sketches
- Ajout de sketches
- Modification d'un sketch
- Suppression d'un sketch

Grâce aux requêtes SQL

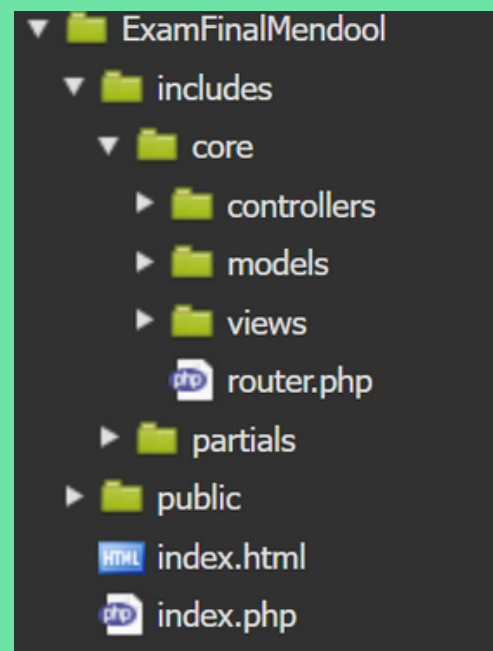




# Architecture

## Index.php

- Point d'entrée de l'application
- Fichier exécuté en premier lorsque l'utilisateur accède à l'application via l'URL
- Appel au routeur



## Routeur

- Récupère le paramètre page dans l'URL
- En fonction du paramètre récupéré il fait appel au controller dans lequel sera établit les actions à mener

```
session_start();
$page = $_GET['page'] ?? 'index';
$action = $_GET['action'] ?? 'view';

switch ($page){
    case 'index':{
        require_once "includes/core/controllers/controller_index.php";
        break;
    }
    case 'sketch':{
        require_once "includes/core/controllers/controller_sketch.php";
        break;
    }
}
```

# Controller

- Il détermine les différentes actions à établir.
- Il fait le lien entre les models et les views en récupérant et en envoyant des données.
- On fait appelle à des fonctions que l'on aura établies dans notre DAO.

## Ex du controller sketch

```
case 'add':{
    require_once "includes/core/models/daoSketch.php";
    if (empty($_POST)){//si$_POST est vide c'est qu'il n'a pas ete rempli
        $unSketch = new Sketch();//donc on a doit afficher un nouvel objet pour a
    }else{
        // Je viens de valider le formulaire : j'ai cliqué sur Submit
        $unSketch = new Sketch();//pour creer cet objet je recupere les données ent
            $_POST['title'],//la [valeur]=name du champs input sur le form view t
            $_POST['url']);

        if (insertSketch($unSketch)){//je teste si qd j envoie un sketch ds la bd
            //l insertion s'est bien passée dc je retourne sur la liste des sket
            header('Location: ?page=s sketch&action=list');
        }else{//l insertion s'est mal passée j affiche un message(qui est stoqué
            $message = "Erreur d'enregistrement !";
        }
    }
}
```

# Model

- Dossier : informations de connexion à notre bdd ,nos classes et nos fonctions avec nos requêtes SQL (fonctions rappelées dans le controller).

## Exemple sur le cas d'ajout d'un sketch

```
function insertSketch(Sketch $newSketch): bool { //pour renvoyer vrai ou faux pour dire
    // INSERT DANS LA BDD
    $conn = getConnection();

    $SQLQuery = "INSERT INTO videos(titre,lien_video,id_type)
    VALUES (:titre, :lien_video,:id_type)";

    $SQLStmt = $conn->prepare($SQLQuery);
    //le bindvalue affecte la valeur qu'on a mit ds le controller pour renseigner la va
    $SQLStmt->bindValue(':titre', $newSketch->getTitre(), PDO::PARAM_STR);
    $SQLStmt->bindValue(':lien_video', $newSketch->getLienVideo(), PDO::PARAM_STR);
    $SQLStmt->bindValue(':id_type', $newSketch->getType()->getId(), PDO::PARAM_INT);
    ;

    if (!$SQLStmt->execute()){ //si l eexecution s est mal passée je renvoie false sinn t
        return false;
    }else{
        return true;
    }
}
```

# View

- Permet d'avoir un visuel et d'appeler nos actions

```
<?php
    $admin = $_SESSION['admin'] ?? false;
    if($admin == true){
        print('<div class="button_ajout">');
        print ('<a href="index.php?page=sketch&action=add" title="ajout sketch">Ajoute
    print('</div>');
    }
?>
```





# Sécurité côté Back End

- htmlspecialchars,
  - conditions,
  - vérification email et hachage du mot de passe
- Exemple sur case Inscription

```
case 'inscription':{
    //j'arrive sur le form d'inscription
    if (empty($_POST)){
        //s'il est vide je crée une nouvelle instance ( donc un nouveau user)
        $unePersonne = new User();
    }else{
        // si le form est pas vide je recupere les infos
        $nom = htmlspecialchars($_POST['nom'], ENT_QUOTES, 'UTF-8');//cette valeur c'est le name du champs input sur le form view form
        $prenom = htmlspecialchars($_POST['prenom'], ENT_QUOTES, 'UTF-8');
        $email = htmlspecialchars($_POST['email'], ENT_QUOTES, 'UTF-8');
        $login = htmlspecialchars($_POST['login'], ENT_QUOTES, 'UTF-8');
        $password = trim($_POST['password']);//ici je sors les espaces eventuels insérés ds le password

        if (!empty($nom) && !empty($prenom) && !empty($login) && strlen($nom) <= 20 && strlen($prenom) <= 20 && strlen($login) <= 20
            && preg_match("/^[a-zA-Z]+$/", $nom) && preg_match("/^[a-zA-Z]+$/", $prenom) && preg_match("/^[a-zA-Z0-9]+$/", $login)){

            if(!empty($email) && filter_var($email, FILTER_VALIDATE_EMAIL) && strcmp($_POST['email'], $_POST['emailConfirm']) == 0//validation email
                && !empty($password) && strlen($password) >= 8){
                $unePersonne = new User(
                    $login,
                    password_hash($password, PASSWORD_ARGON2ID),//on fait appel a une fonction php qui hache le password
                    true,
                    $nom,
                    $prenom,
                    $email,
                );
            }
        }
    }
}
```

- Bindvalue (contre les injections SQL)

## Exemple sur la fonction UserExists

```
function userExists(string $login):bool{  
    $conn = getConnexion();  
  
    $SQLQuery = "SELECT COUNT(id) as existe  
FROM userauth  
WHERE login = :login";  
  
    $SQLStmt = $conn->prepare($SQLQuery);  
    $SQLStmt->bindValue(':login',$login, PDO::PARAM_STR);  
    $SQLStmt->execute();  
  
    $SQLRow = $SQLStmt->fetch(PDO::FETCH_ASSOC);  
    $loginTrouve = $SQLRow['existe'];  
  
    $SQLStmt->closeCursor();  
  
    return ($loginTrouve>0);  
}
```

# Sécurité côté FRONT

## HTML

- maxlength
- required

Exemple HTML sur le nom :

```
<div>
  <label for="nom"></label>
  <input type="text" id="nom" name="nom" placeholder="Nom" required maxlength= 20>
</div>
```

# JS

- Vérification de soumission d'adresses mails valident (utilisation de Regex)
- Comparaison des adresses mail
- Contrainte de minimum 8 caractères pour le mot de passe

Exemple de JS : Création d'un message d'erreur pour l'utilisateur si le mot de passe est inférieur à 8 caractères.

```
const inscriptionForm = document.getElementById('inscription_form');
//ici je crée une fonction et je mets ds une var le bouton de connexion et je crée
function checkPasswordLength(passwordField) {
  const btnConnexion = document.querySelector(".connexion-btn");
  const elementP = document.createElement('p');

  //ici je cree le contenu de mon <p>
  elementP.textContent = "Le mot de passe doit contenir au moins 8 caractères";
  elementP.style.color = "red";
  elementP.style.textAlign = "center";
  elementP.style.display = "none";
  //Ici j'ajoute mon élément <p> juste après le champ de saisie de mot de passe a
  //passwordField.nextSibling sert à insérer le <p> juste après le champ de saisie
  passwordField.parentNode.insertBefore(elementP, passwordField.nextSibling);
  //Ici j'ecoute l'evenement à la pression sur les touche sur le champs de saisi
  passwordField.addEventListener('keypress', function(event) {
    //si la valeur entrée ds le champs et <8 caracteres je desactive le bouton
    if(event.target.value.length<8){
      btnConnexion.disabled = true;
      elementP.style.display = "block";
    }
    //sinon je laisse mon bouton actif et je desactive la visibiilté du <p>
  }else{
    btnConnexion.disabled = false;
    elementP.style.display = "none";
  }
}
```







# Merci pour votre attention

