

Présentation prestataire

2.1- Rôle du prestataire dans le projet Orientation+ :

Le prestataire est l'entité ou l'équipe de développement chargée de la conception, de l'implémentation et de la maintenance technique de la plateforme *Orientation+*. Il s'agit ici de garantir que la solution numérique livrée répond aux besoins définis dans le cahier des charges et respecte les contraintes techniques et fonctionnelles.

Responsabilités principales :

- Analyse des besoins clients et rédaction du cahier des charges.
- Conception technique (choix de l'architecture et des technologies).
- Développement des fonctionnalités principales et des interfaces utilisateur.
- Tests de performance, sécurité et compatibilité multiplateforme.
- Maintenance et mises à jour régulières après la livraison.

2.2 Équipe prestataire :

L'équipe prestataire est composée de plusieurs profils spécialisés pour assurer une exécution optimale du projet :

- Chef de projet :**
 - Coordonne l'ensemble des étapes et communique avec le client.
 - Valide les livrables à chaque étape (analyse, développement, tests).
- Développeurs (Front-end et Back-end) :**
 - Back-end** : Gestion des bases de données, développement des fonctionnalités PHP et sécurisation des données.
 - Front-end** : Développement des interfaces utilisateurs (HTML, CSS, JavaScript).
 - Implémentation des frameworks comme Bootstrap pour le design responsive.
- Spécialiste UX/UI :**
 - Conçoit une interface intuitive et ergonomique.
 - Assure l'accessibilité et l'adaptabilité sur différents supports (ordinateurs, tablettes, smartphones).
- Technicien QA (Quality Assurance) :**
 - Teste les fonctionnalités pour identifier et corriger les bugs avant la mise en production.
 - Vérifie la conformité avec les exigences définies dans le cahier des charges.

2.3 Technologies utilisées par le prestataire :

- Back-end :**
 - PHP avec le modèle MVC (Modèle-Vue-Contrôleur) :**
Permet une structuration claire du code et une séparation entre logique métier, interface utilisateur et base de données.
 - Exemple d'implémentation :**

```
class UserModel {
    public function getUserByEmail($email) {
        global $conn;
        $stmt = $conn->prepare("SELECT * FROM users WHERE email = ?");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        return $stmt->get_result()->fetch_assoc();
    }
}
```

- Base de données :**
 - MySQL** : Gestion des données utilisateurs, fiches métiers et fichiers uploadés.
 - Création de tables pour structurer les informations.

Exemple de table users :

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    role ENUM('eleve', 'professeur') NOT NULL
);
```

- Front-end :**
 - HTML, CSS, JavaScript (Bootstrap)** pour une interface responsive et moderne.
 - Utilisation d'icônes pour rendre les fonctionnalités intuitives (ex. : Font Awesome).**Exemple d'interface utilisateur :**

```
<div class="login-form">
    <h2><i class="fas fa-user"></i> Connexion</h2>
    <form method="post" action="login.php">
        <input type="email" name="email" placeholder="Email" required>
        <input type="password" name="password" placeholder="Mot de passe" required>
        <button type="submit">Se connecter</button>
    </form>
</div>
```

- Hébergement :**
 - Serveur **Apache ou NGINX** pour l'hébergement des fichiers et la gestion des requêtes.
 - Support de protocoles HTTPS pour sécuriser les échanges de données.
- Sécurité :**
 - Hashage des mots de passe avec **bcrypt**.
 - Contrôle d'accès basé sur les rôles (élève ou professeur).
 - Validation des fichiers uploadés pour prévenir les vulnérabilités.

2.4 Étapes réalisées par le prestataire :

- Analyse et conception :**