

Sujet de Projet :

Conception et Développement d'une Plateforme de Gestion Associative avec Base de Données Relationnelle

Objectifs du projet :

Créer une plateforme web permettant à une association étudiante de gérer ses activités.

Les fonctionnalités incluent :

- Gestion des membres
- Gestion des événements
- Gestion des projets collaboratifs
- Forum interne pour la communication

Étapes de développement

1. Analyse des besoins

Les principales fonctionnalités à implémenter sont :

- **Authentification** : Inscription, connexion et gestion des rôles (administrateur, membre).
- **Gestion des membres** : Ajouter, modifier et supprimer des membres.
- **Gestion des événements** : Créer, modifier et afficher un calendrier d'événements.
- **Gestion des projets** : Créer des projets, attribuer des membres et suivre leur progression.
- **Forum interne** : Espace pour publier des messages et échanger entre membres.

2. Technologies suggérées

- **Back-end** : PHP avec un framework au choix ou pur PHP.
- **Base de données** : MySQL, PostgreSQL, SQLite.
- **Front-end** : HTML5, CSS3, Bootstrap pour le design.
- **Serveur local** : XAMPP ou WAMP pour le développement.
- **Versioning** : Git/GitHub.

3. Structure de la base de données

Vous devez présenter une modélisation claire et cohérente des données nécessaires au fonctionnement de la plateforme. Cette modélisation doit répondre aux besoins fonctionnels identifiés dans le projet, en garantissant une gestion efficace des informations et des relations entre les différentes entités.

Modélisation Conceptuelle

- Fournir une analyse des entités principales et des relations entre elles.
- Identifier les attributs pertinents pour chaque entité en fonction des fonctionnalités de la plateforme.
- Justifier les choix de conception en lien avec les besoins exprimés (exemple : gestion des membres, événements, projets, etc.).

Contraintes et Intégrité des Données

- Définir les contraintes nécessaires pour garantir l'intégrité des données (unicité, relations entre entités, etc.).
- Expliquer comment ces contraintes assurent la cohérence des informations dans le système.

Optimisation et Normalisation

- Décrire les démarches entreprises pour éviter les redondances et assurer la normalisation des données.
- Justifier les choix effectués pour optimiser les performances (exemple : utilisation d'index ou autres mécanismes).

4. Structure des fichiers

Voici une organisation simple pour les fichiers du projet :

```
/public
  index.php          --> Point d'entrée principal
  /css               --> Fichiers CSS (incluant Bootstrap)
  /js                --> Scripts JavaScript
/config
  database.php       --> Configuration de la base de données
/views
  header.php         --> En-tête commun
  footer.php         --> Pied de page commun
  /users             --> Pages liées aux utilisateurs
    login.php        --> Connexion
    register.php     --> Inscription
  /events            --> Pages liées aux événements
/controllers         --> Logique métier (PHP)
models               --> Classes pour interagir avec la base de données
```

Livrables Attendus

1. Application Web Fonctionnelle :

- Interface utilisateur responsive avec Bootstrap.
- Back-end en PHP (ou autre langage au choix).

2. Dépôt Git Organisé :

- Code source commenté.
- Fichier README.md expliquant l'installation et les dépendances.
- Historique des commits clair.

3. Mini-Rapport (5-8 pages maximum) :

- Introduction et objectifs.
- Analyse des besoins et schéma de la base de données (MCD, MLD).
- Choix techniques (PHP/MySQL vs alternatives).
- Liste de l'ensemble des requêtes SQL illustrant comment interagir avec la base de données pour répondre aux besoins fonctionnels du projet.
- Difficultés rencontrées et solutions apportées.
- Perspectives d'amélioration.

4. Présentation de 20 Minutes :

- Contexte et objectifs (2 min).
- Architecture technique (6 min).
- Démonstration de l'application (6 min).
- Questions/Réponses (6 min).

5. Bonus (1 au choix si Temps)

- **Renouvellement automatique des adhésions** : Envoyer des rappels automatiques pour renouveler les adhésions et permettre un paiement en ligne.
- **Intégration des réseaux sociaux** : Permettre le partage d'événements ou d'activités sur les réseaux sociaux directement depuis la plateforme.
- **Évaluation post-événement** : Intégrer un système de feedback pour recueillir les avis des participants après un événement.

Critères d'Évaluation

Critère	Détails
Cohérence de la base	Schéma relationnel normalisé, clés étrangères, gestion des contraintes.
Fonctionnalités de base	Respect des spécifications (authentification, gestion membres/événements).
Fonctionnalités avancées	Implémentation et originalité des 2 options choisies.
Qualité du code	Structuration MVC, sécurité, commentaires, tests unitaires.
Documentation	Clarté du rapport, schémas, explication des choix techniques.
Présentation orale	Démonstration fluide, réponses aux questions, justification des solutions.

Barème indicatif :

- Base de données (40%)
- Application (30%)
- Documentation (15%)
- Présentation (15%)