

Documentation Technique - Projet Congrès

1. Introduction

Le projet Congrès est une application web de gestion d'événements permettant l'administration d'un congrès, y compris les inscriptions, la gestion des congressistes, des sessions, des activités, des hébergements et la facturation. Il s'adresse principalement aux administrateurs et aux participants aux congrès.

1.1 Objectifs du projet

- Faciliter la gestion des inscriptions aux congrès
- Gérer les participants et leurs informations
- Administrer les sessions et activités disponibles
- Gérer l'hébergement des participants
- Générer et suivre les factures
- Fournir une interface conviviale pour les administrateurs et utilisateurs

2. Architecture technique

L'application est construite sur une architecture MVC (Modèle-Vue-Contrôleur) en PHP, sans utilisation de framework spécifique.

2.1 Technologies utilisées

- **Backend:** PHP 7.4+
- **Frontend:** HTML5, CSS3, JavaScript
- **Base de données:** MySQL/MariaDB
- **Serveur web:** Apache
- **Environnement de développement:** WAMP (Windows, Apache, MySQL, PHP)

3. Structure du projet

```
congres/
└── class/                  # Classes du modèle
    ├── Congressiste.php
    ├── Facture.php
    └── ...
```

```
|── config/                      # Configuration
|   └── config.php                # Configuration BDD
├── controller/                  # Contrôleurs
|   └── ...
└── css/                         # Feuilles de style
    └── style.css                 # Style principal
├── index.php                     # Point d'entrée
└── js/                           # Scripts JavaScript
    └── ...
└── vue/                          # Vues
    ├── entete.php                # En-tête commun
    ├── pied.php                  # Pied de page commun
    ├── vueConnexion.php          # Vue de connexion
    ├── vueFacture.php            # Vue des factures
    ├── vueDetailFacture.php     # Détail d'une facture
    └── ...
```

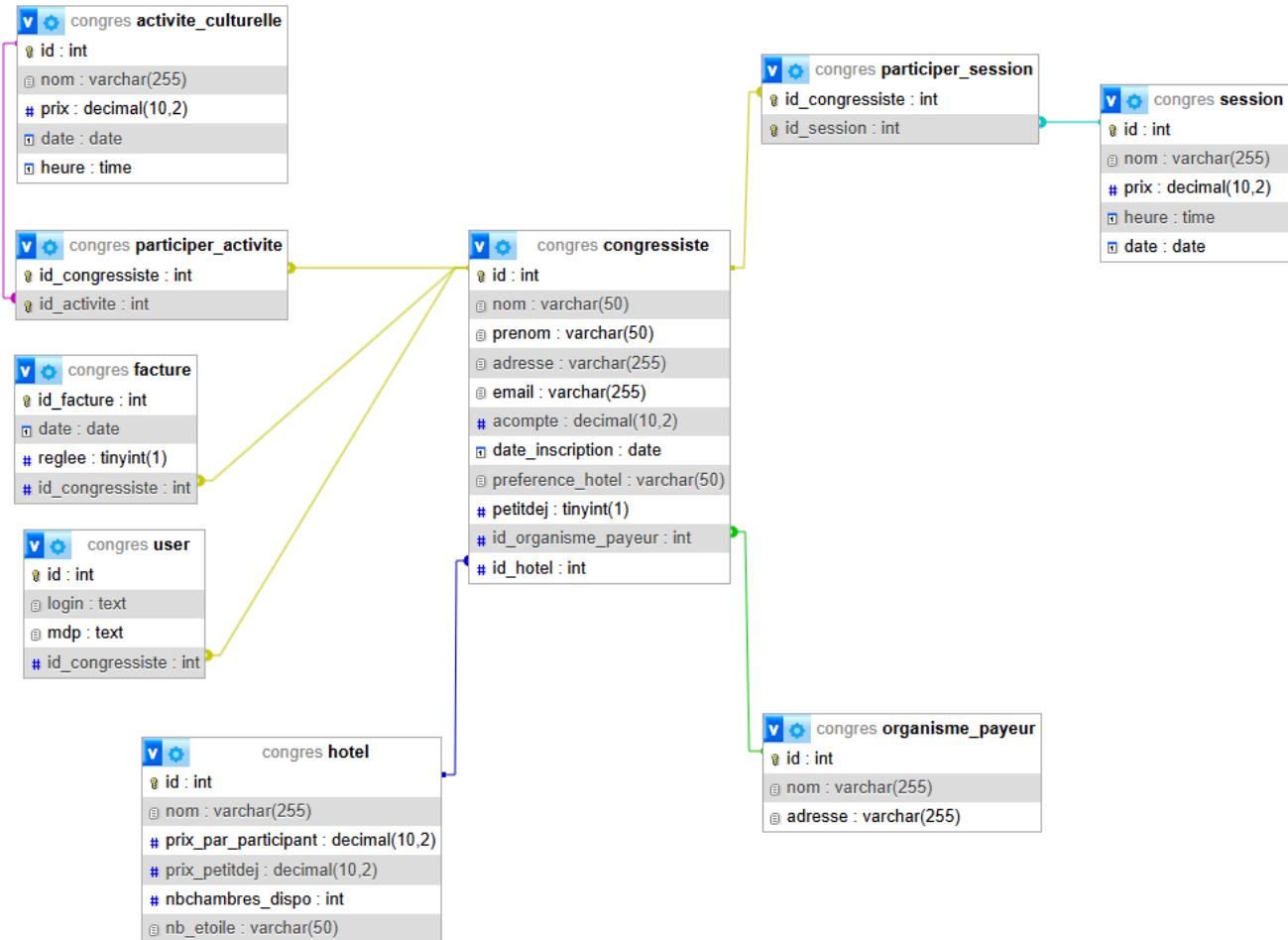
4. Base de données

4.1 Schéma de la base de données

La base de données comprend les tables principales suivantes:

- `congressiste` : Informations sur les participants
- `hotel` : Informations sur les hôtels disponibles
- `session` : Sessions disponibles pendant le congrès
- `activite` : Activités culturelles disponibles
- `facture` : Factures générées
- `session_congressiste` : Relation entre sessions et congressistes
- `activite_congressiste` : Relation entre activités et congressistes
- `utilisateur` : Informations d'authentification

4.2 BDD



4.3 Requêtes principales

Les principales requêtes SQL utilisées dans l'application se trouvent dans les classes du modèle, notamment pour:

- Récupération des congressistes avec leurs informations
- Calcul des coûts pour les factures
- Gestion des inscriptions aux sessions et activités

5. Fonctionnalités principales

5.1 Gestion des congressistes

- Inscription de nouveaux participants
- Modification des informations personnelles
- Association avec un organisme payeur
- Consultation de la liste des participants

5.2 Gestion des sessions

- Création et modification des sessions
- Inscription des congressistes aux sessions
- Suivi des inscriptions

5.3 Gestion des activités culturelles

- Création et modification des activités
- Inscription des congressistes aux activités
- Suivi des participations

5.4 Gestion de l'hébergement

- Association des congressistes avec des hôtels
- Options pour les petits déjeuners
- Calcul des coûts d'hébergement

5.5 Facturation

- Génération automatique des factures
- Suivi des paiements
- Édition des factures en PDF
- Gestion des statuts (réglé/non réglé)

6. Système d'authentification

6.1 Fonctionnement

L'authentification est gérée via des sessions PHP. Les utilisateurs peuvent être:

- Administrateurs: accès complet au système
- Utilisateurs standard: accès limité à leurs propres informations

```
// Extrait simplifié du mécanisme d'authentification
if(isset($_POST['validerConnexion'])) {
    $login = $_POST['login'];
    $mdp = $_POST['mdp'];

    // Vérification dans la base de données
    // ...

    // Crédit de la session
    $_SESSION['id_utilisateur'] = $utilisateur['id'];
```

```
$_SESSION['is_admin'] = $utilisateur['is_admin'];
// ...
}
```

6.2 Sécurité

- Mots de passe hashés avec les méthodes sécurisées de PHP
- Protection contre les attaques par injection SQL via l'utilisation de requêtes préparées
- Validation des données utilisateur

7. Gestion des factures

7.1 Crédation de facture

Les factures sont générées automatiquement en fonction des inscriptions des congressistes:

- Sessions sélectionnées
- Activités choisies
- Options d'hébergement
- Acomptes versés

```
// Extrait simplifié de la création de facture
function createFacture($id_congressiste, $date, $total) {
    $stmt = $this->connexion->prepare(
        "INSERT INTO facture (id_congressiste, date, montant, reglee)
         VALUES (?, ?, ?, 0)
    ");
    $stmt->execute([$id_congressiste, $date, $total]);
    // ...
}
```

7.2 Calcul des montants

Les montants des factures sont calculés en additionnant:

- Le coût de l'hébergement incluant les options
- Le coût des sessions
- Le coût des activités culturelles
- Moins les acomptes déjà versés

7.3 Génération PDF

Les factures peuvent être exportées au format PDF via une bibliothèque PHP.

8. Styles et interface utilisateur

8.1 Système de design

L'application utilise un système de design cohérent avec des variables CSS pour:

- Les couleurs primaires et secondaires
- Les espacements (spacing)
- Les rayons des bordures (border-radius)
- Les ombres
- Les typographies

```
:root {  
    /* Couleurs */  
    --primary: 59, 130, 246;      /* Bleu */  
    --secondary: 139, 92, 246;    /* Violet */  
    --error: 220, 38, 38;        /* Rouge */  
    --success: 16, 185, 129;     /* Vert */  
  
    /* Espacement */  
    --spacing-1: 0.25rem;  
    --spacing-2: 0.5rem;  
    --spacing-4: 1rem;  
    --spacing-6: 1.5rem;  
    --spacing-8: 2rem;  
  
    /* Typographie */  
    --txt-sm: 0.875rem;  
    --txt-base: 1rem;  
    --txt-lg: 1.125rem;  
    --txt-xl: 1.25rem;  
    /* ... */  
}
```

8.2 Composants UI

Les principaux composants UI incluent:

- Formulaires stylisés
- Cartes (cards) pour l'affichage des données
- Boutons avec différentes variantes

- Tableaux responsifs
- Messages d'alerte
- En-têtes et pieds de page cohérents

8.3 Responsive design

L'application est conçue pour s'adapter à différentes tailles d'écran grâce à:

- Des media queries
- Des layouts flexibles
- Des grilles CSS
- Des composants adaptatifs

```
@media (max-width: 768px) {
    .cards-container {
        grid-template-columns: 1fr;
    }

    .button-container {
        flex-direction: column;
    }
}
```

9. Sécurité

9.1 Protection contre les injections SQL

Toutes les requêtes utilisent des requêtes préparées PDO pour empêcher les injections SQL:

```
$stmt = $this->connexion->prepare("SELECT * FROM users WHERE login = ?");
$stmt->execute([$login]);
```

9.2 Validation des données

Les données utilisateur sont validées avant traitement:

- Filtrage des entrées
- Vérification des types
- Échappement à l'affichage avec `htmlspecialchars()`

9.3 Protection XSS

Protection contre les attaques cross-site scripting:

- Échappement systématique des données affichées
- Validation des entrées

9.4 Gestion des sessions

- Utilisation sécurisée des sessions PHP
- Régénération des identifiants de session
- Paramètres de cookies sécurisés

10. Installation et configuration

10.1 Prérequis

- Serveur web Apache
- PHP 7.4 ou supérieur
- MySQL 5.7 ou MariaDB 10.2+
- Extensions PHP: mysqli, pdo_mysql, gd, mbstring

10.2 Étapes d'installation

1. Cloner le dépôt dans le répertoire web
2. Importer le fichier SQL pour créer la base de données
3. Configurer les accès à la base de données dans config/config.php
4. Configurer les droits d'accès aux répertoires
5. Accéder à l'application via l'URL du serveur

10.3 Configuration

Le fichier config/config.php contient les paramètres principaux:

```
// Connexion à la base de données
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'username');
define('DB_PASSWORD', 'password');
define('DB_NAME', 'congres');

// Autres configurations
define('APP_NAME', 'Gestion de Congrès');
define('APP_URL', 'http://localhost/congres');
```

11. Dépendances

11.1 Bibliothèques PHP

- PHP Data Objects (PDO) pour l'accès à la base de données
- Bibliothèque FPDF/TCPDF pour la génération des PDF

11.2 Bibliothèques Frontend

- Aucun framework frontend n'est utilisé, uniquement du CSS et JavaScript natifs
-

Cette documentation technique fournit une vue d'ensemble du projet Congrès. Pour plus de détails sur des fonctionnalités spécifiques, consultez les commentaires dans le code source ou les documents de spécification associés.