

# Documentation Technique - Système de Gestion des Événements d'Entreprise

Lien vers mon dépôt GitHub: [https://github.com/Mohammad77Radwan/Entreprise\\_Events](https://github.com/Mohammad77Radwan/Entreprise_Events)

## Table de Matières

- [Documentation Technique - Système de Gestion des Événements d'Entreprise](#)
  - [Table des Matières](#)
  - [1. Introduction](#)
  - [2. Technologies Utilisées](#)
    - [Backend](#)
    - [Frontend](#)
    - [Infrastructure](#)
  - [3. Architecture du Projet](#)
    - [Structure des Fichiers](#)
    - [Schéma MVC](#)
  - [4. Modèle de Données](#)
    - [Diagramme MCD](#)
      - [Tables Principales](#)
  - [5. Installation et Déploiement](#)
    - [1. Cloner le dépôt](#)
    - [2. Démarrer les conteneurs](#)
    - [3. Accéder à l'application](#)
  - [6. Fonctionnalités](#)
    - [Fonctionnalités Principales](#)
      - [Exemple de Code :](#)
  - [7. Sécurité](#)
    - [Mesures Implémentées](#)
  - [8. Perspectives](#)
    - [Évolutions Possibles](#)
  - [9. Galerie Docker , Application Web & BD](#)

## 1. Introduction

Application web développée dans le cadre du BTS SIO SLAM pour la gestion des événements en entreprise. Permet :

- La création et gestion d'événements
- L'inscription des participants
- La visualisation des réservations
- La gestion des organisateurs

## 2. Technologies Utilisées

### Backend

- **PHP 8.2** (architecture MVC)
- **MySQL 8.0** (base de données)
- **PDO** (accès sécurisé à la base)

### Frontend

- **HTML5, CSS3, JavaScript**
- **Bootstrap 5** (interface responsive)
- **Font Awesome** (icônes)

### Infrastructure

- **Docker** (conteneurisation)
- **Apache** (serveur web)
- **Composer** (gestion des dépendances)

## [Structure des Fichiers](#) 3. Architecture du Projet

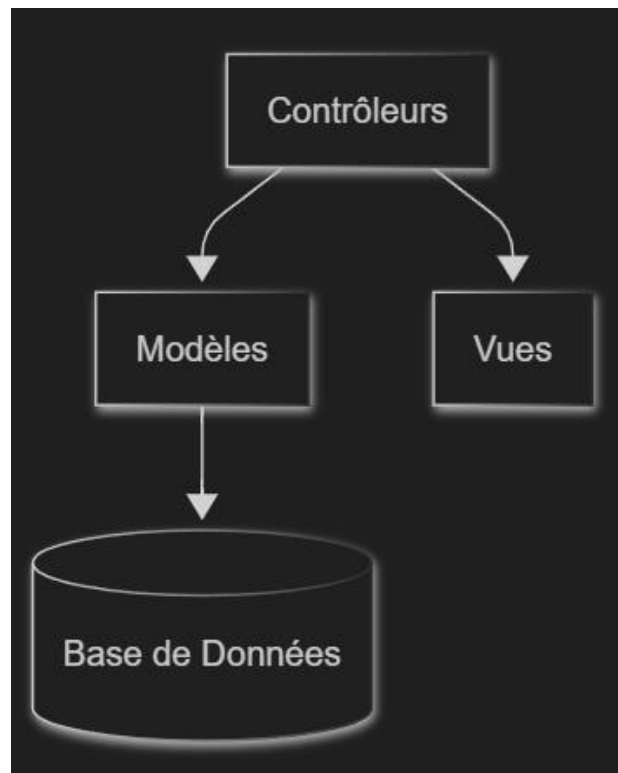
### Structure des Fichiers

```
entreprise_events/  
├── .gitignore  
├── docker-compose.yml  
├── README.md  
├── docs/  
│   ├── documentation-technique.md  
│   ├── MCD.png  
│   └── wireframes/  
├── src/  
│   ├── config/  
│   │   ├── database.php  
│   │   └── constants.php  
│   ├── controllers/  
│   │   ├── AuthController.php  
│   │   └── EventController.php
```

```
| | | └─ HomeController.php
| | | └─ OrganizerController.php
| | | └─ ParticipantController.php
| | | └─ ReservationController.php
| └─ models/
| | | └─ Database.php
| | | └─ Event.php
| | | └─ Organizer.php
| | | └─ Participant.php
| | | └─ Reservation.php
| | | └─ User.php
| └─ views/
| | | └─ auth/
| | | | └─ login.php
| | | | └─ register.php
| | | └─ events/
| | | | └─ create.php
| | | | └─ edit.php
| | | | └─ index.php
| | | | └─ show.php
| | | └─ organizers/
| | | | └─ index.php
| | | | └─ show.php
| | | └─ participants/
| | | | └─ index.php
| | | | └─ show.php
| | | └─ reservations/
| | | | └─ create.php
| | | | └─ index.php
| | | └─ partials/
| | | | └─ footer.php
| | | | └─ header.php
| | | | └─ navbar.php
| | | └─ layouts/
| | | | └─ main.php
| | | └─ home.php
| └─ utils/
| | | └─ Auth.php
| | | └─ Validator.php
```

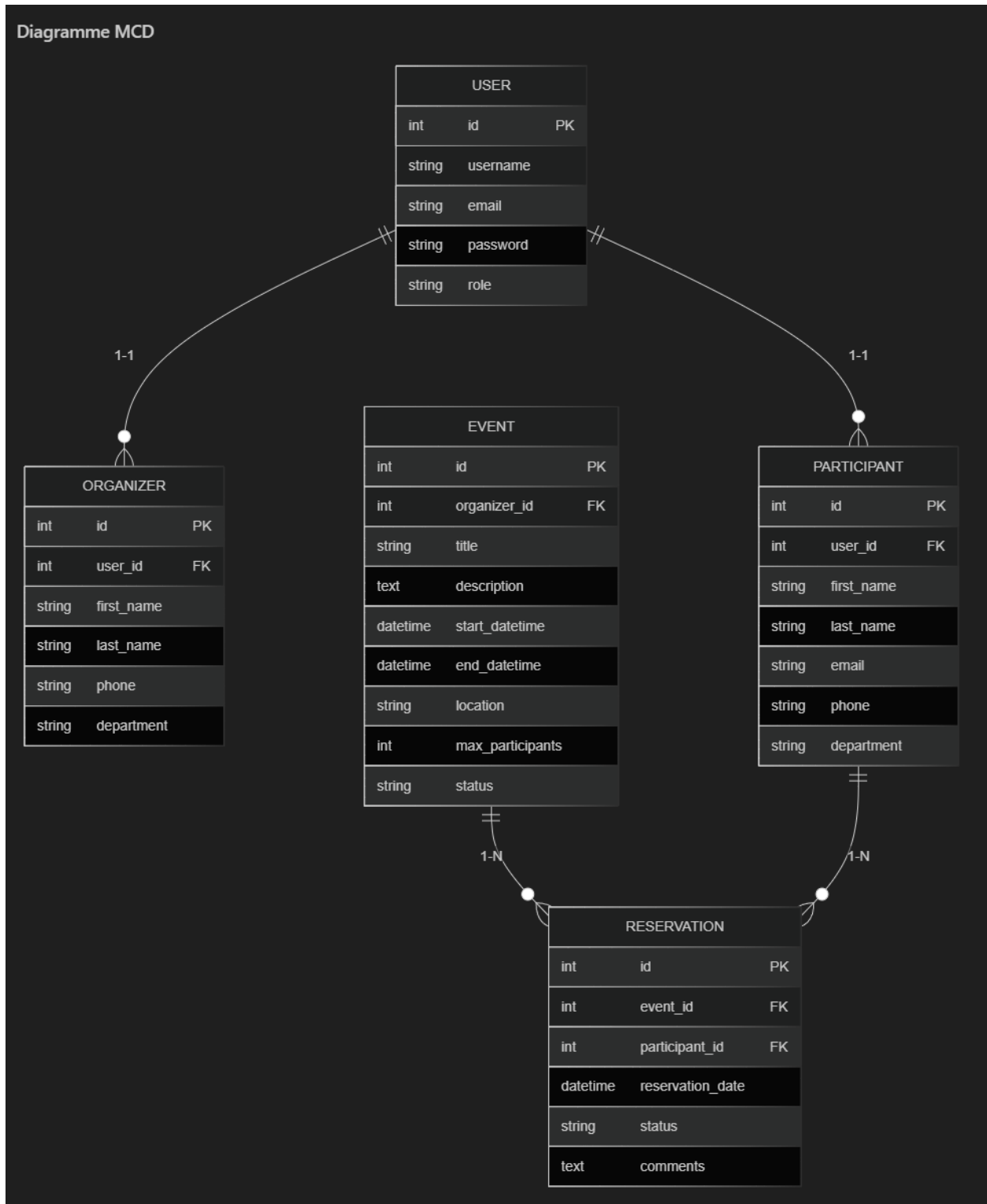
```
| | | └─ helpers.php
| | └─ assets/
| | | └─ css/
| | | | └─ style.css
| | | └─ js/
| | | | └─ main.js
| | | └─ images/
| └─ index.php
└─ docker/
    └─ mysql/
        └─ init.sql
    └─ php/
        └─ Dockerfile
```

#### [4. Modèle de Données](#)Schéma MVC



## 4. Modèle de Données

### Diagramme MCD



### *Tables Principales*

1. users : Gestion des comptes
2. organizers : Informations des organisateurs
3. participants : Informations des participants
4. events : Détails des événements
5. reservations : Liaison participants-événements

## 5. Installation et Déploiement

### Prérequis

- Docker et Docker Compose
- Git

### Instructions :

#### *1. Cloner le dépôt*

git clone [https://github.com/votre-repo/entreprise\\_events.git](https://github.com/votre-repo/entreprise_events.git) cd entreprise\_events

#### *2. Démarrer les conteneurs*

docker-compose up -d --build

#### *3. Accéder à l'application*

<http://localhost:8000>

Rôle	Identifiants
Administrateur	admin / password
Organisateur	org1 / password
Participant	part1 / password

## 6. Fonctionnalités

### Fonctionnalités Principales

Fonctionnalité	Route	Fichiers Concernés
Authentification	/auth/login	AuthController.php, User.php
Création d'événements	/events/create	EventController.php
Gestion des participants	/participants	ParticipantController.php
Système de réservation	/reservations/create	ReservationController.php

*Exemple de Code :*

```
// Création d'un événement (EventController.php)
public function create() {
    Auth::redirectIfNotOrganizer();

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $eventId = $this->eventModel->create(
            $_SESSION['user_id'],
            $_POST['title'],
            $_POST['description'],
            $_POST['start_datetime'],
            $_POST['end_datetime'],
            $_POST['location'],
            (int) $_POST['max_participants']
        );
        // Redirection après création
    }
}
```

## 7. Sécurité

### Mesures Implémentées

#### Protection des données :

- Préparation des requêtes SQL (PDO)
- Échappement HTML (`htmlspecialchars`)

#### Authentification :

- Hachage des mots de passe (`password_hash`)
- Gestion des sessions sécurisées

#### En-têtes HTTP :

- CSP (Content Security Policy)
  - XSS Protection
  - Referrer Policy
- 

## 8. Perspectives

### Évolutions Possibles

#### Fonctionnalités :

- Export PDF des participants
- Calendrier interactif (`FullCalendar.js`)
- Notifications par email

#### Techniques :

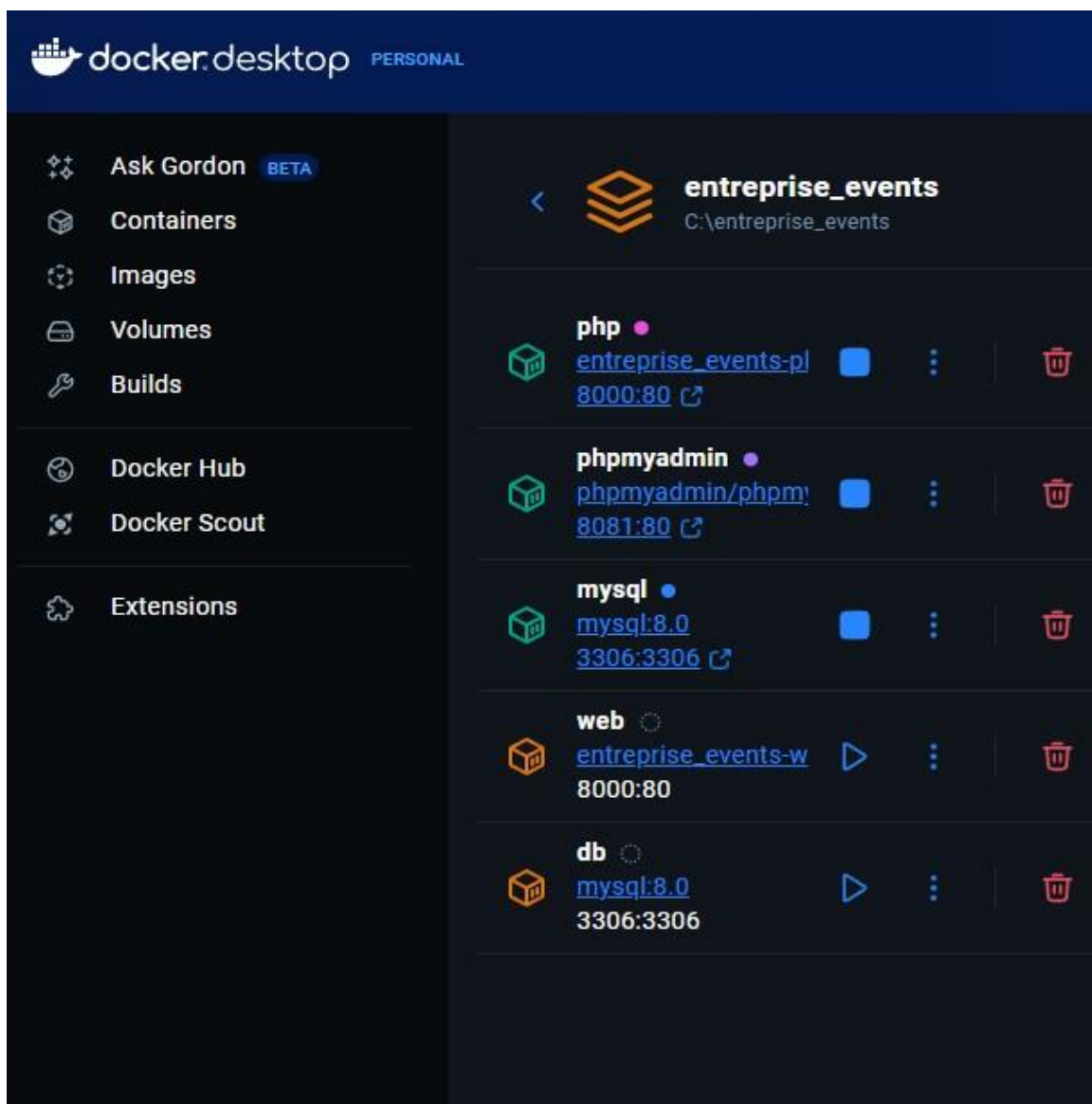
- Migration vers Symfony/Laravel
- API REST pour mobile
- Intégration SSO



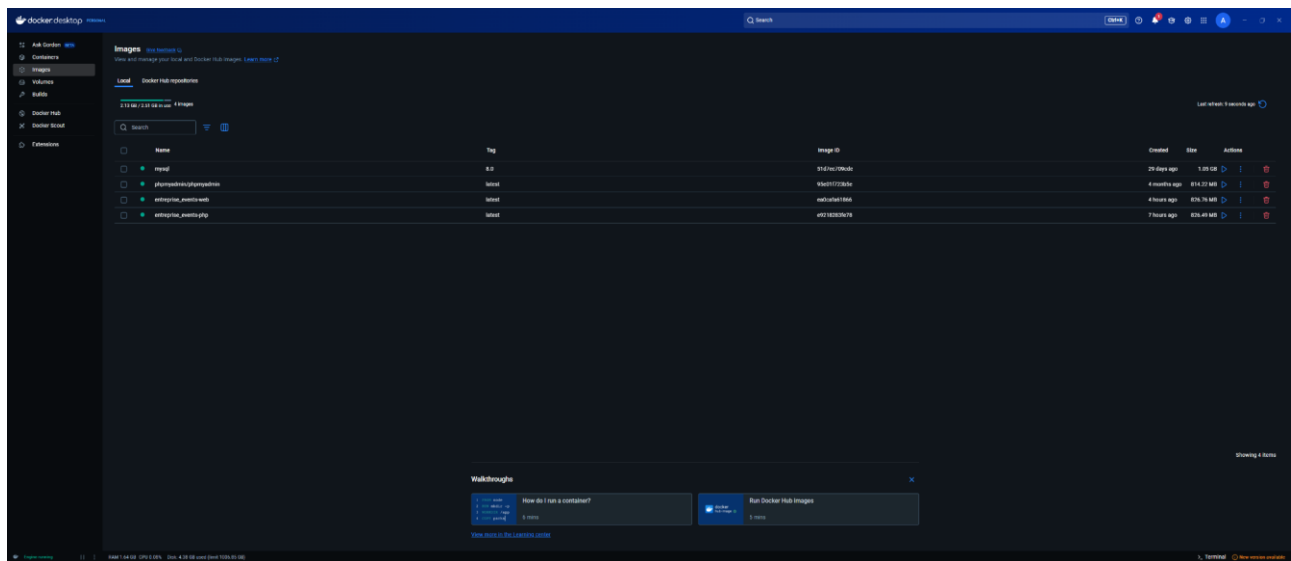
## 9. Galerie Docker , Application Web & BD

### Galerie Docker

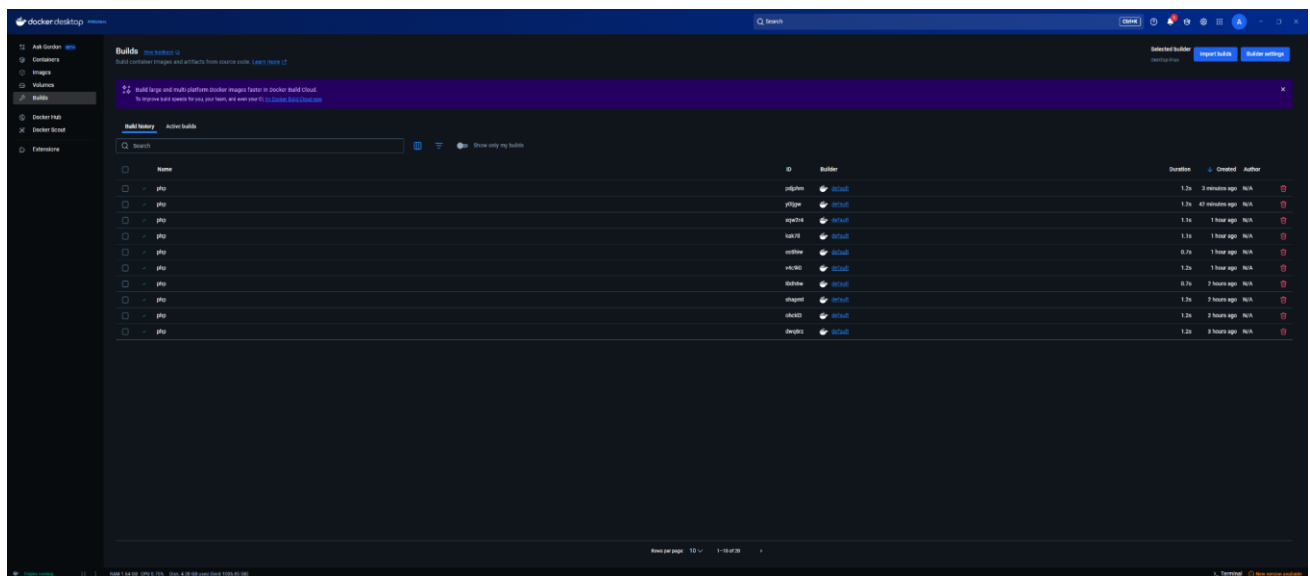
Ces captures présentes l'environnement Docker utilisé pour l'exécution de l'application. On y voit les conteneurs en fonctionnement, orchestrés via `docker-compose`, ce qui facilite le déploiement et l'isolation des services comme Apache, MySQL et PHP.



### Docker Overview



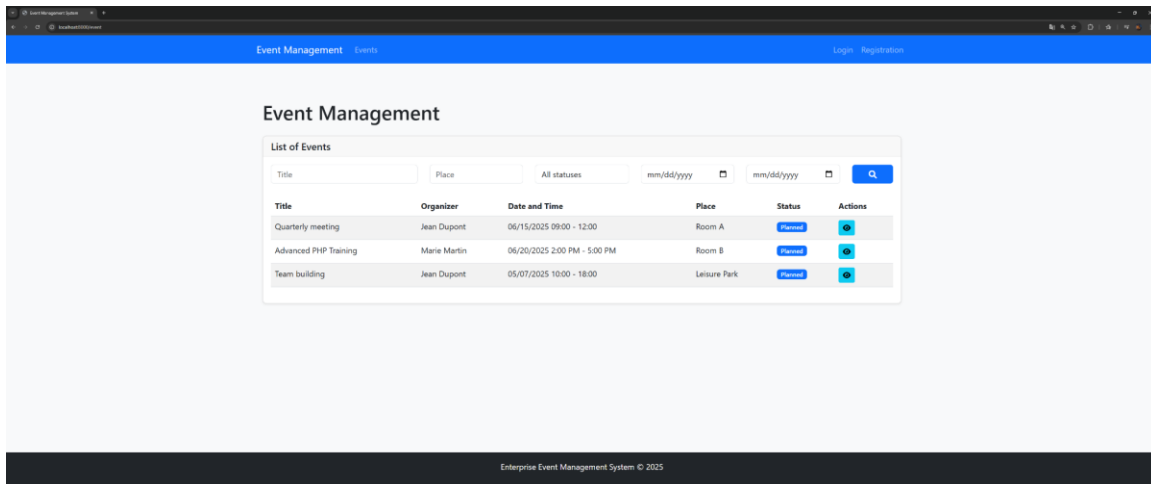
## Docker Containers



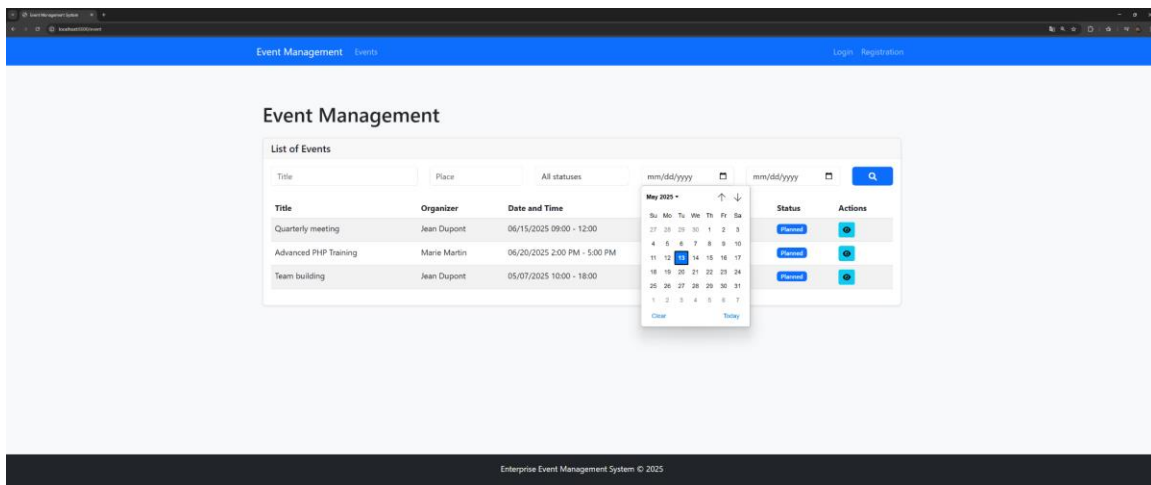
## Docker Build History

## Application Web

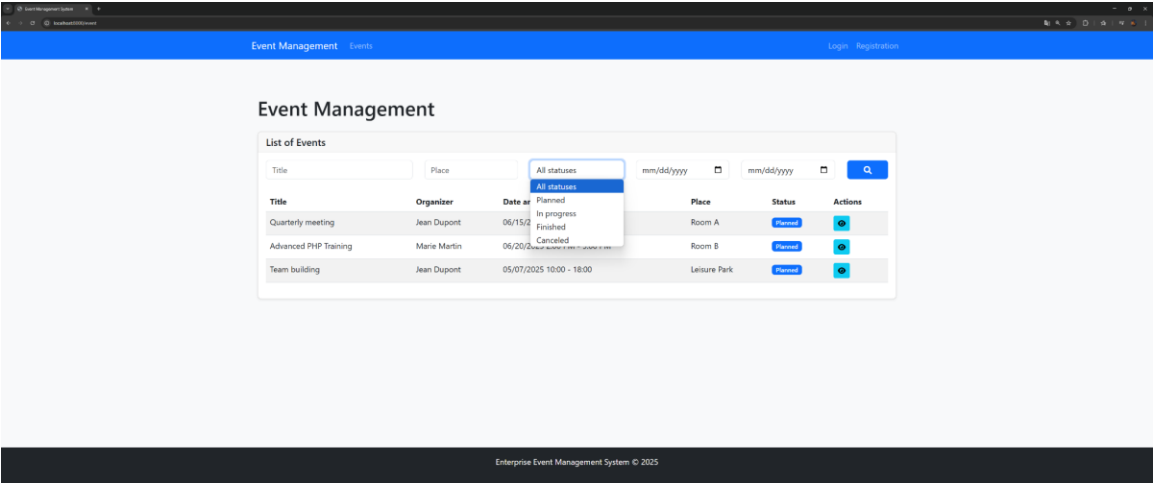
*Interface utilisateur de l'application de gestion d'événements. On y retrouve les principales fonctionnalités : tableau de bord, création d'événements, liste des participants et système de réservation. L'interface est conçue avec Bootstrap 5 pour garantir une expérience fluide et responsive.*



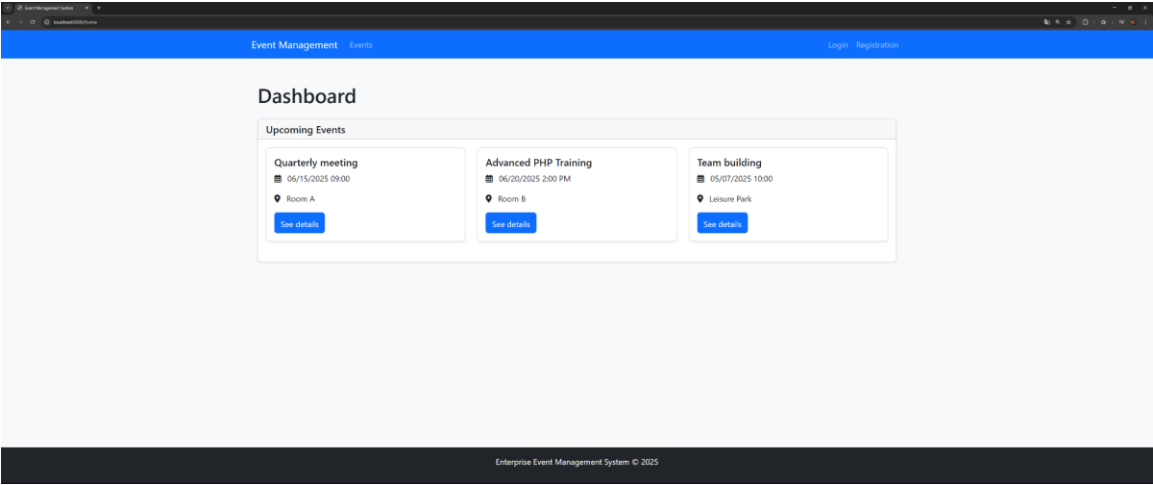
## Web Home



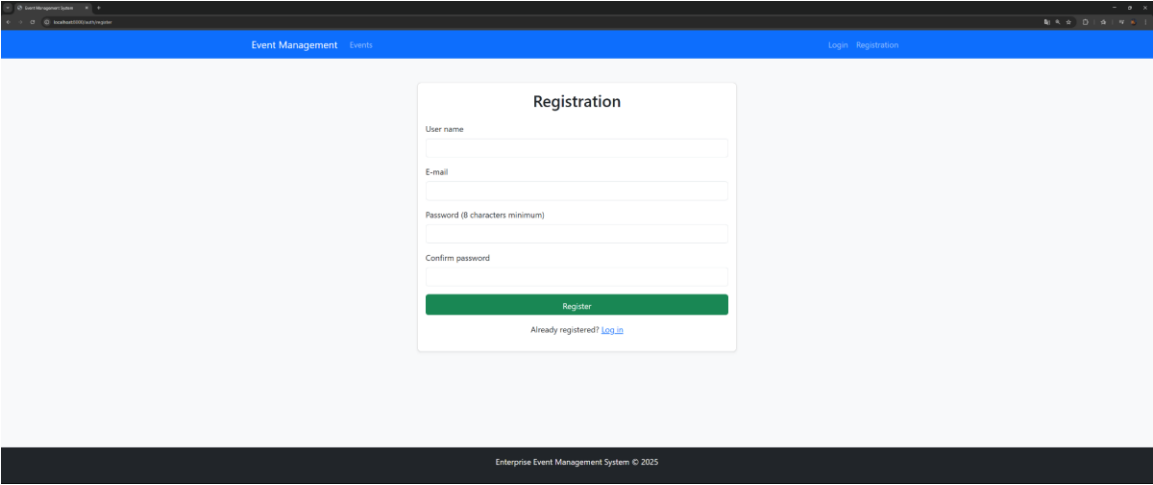
## Web Calendar View



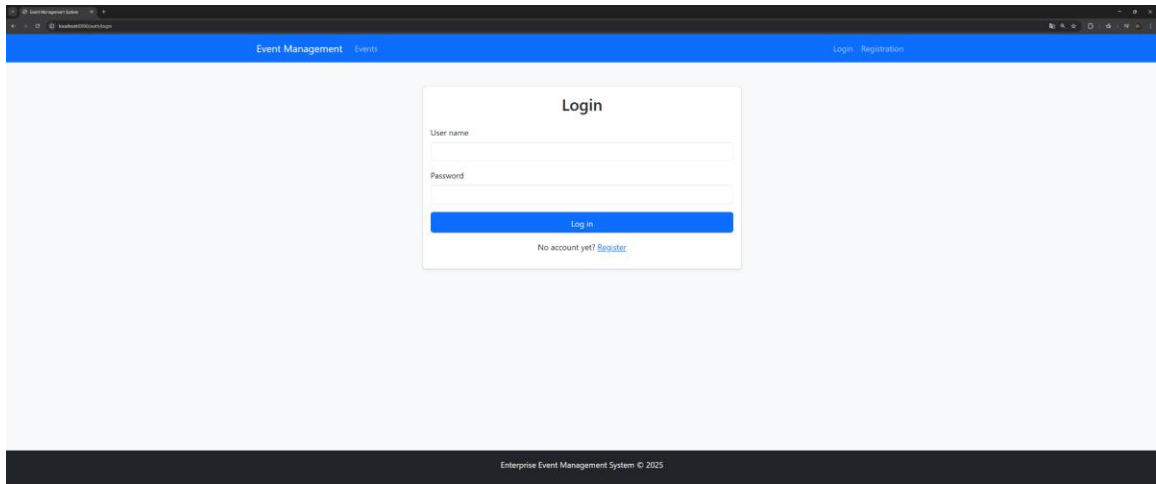
## Web Status Filter



## Web Dashboard



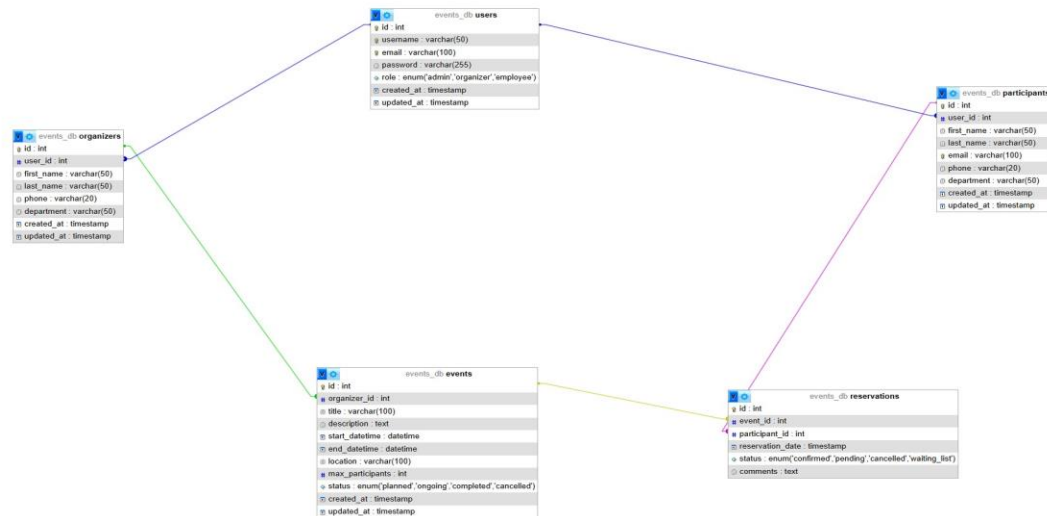
## Web Registration



## Web Login

## BD (Base de Données)

Diagramme MCD de la base de données relationnelle. Il met en évidence les principales entités : utilisateurs, organisateurs, participants, événements et réservations. Ce modèle assure la cohérence des données et facilite les opérations de gestion via des relations bien définies.



## Model Relationnelle