

## **Projet :**

### ***Application Web JEE pour la Gestion des Locations d'Immeubles***

#### **I. Introduction**

*Le projet consiste à développer une application web en Java Enterprise Edition (JEE) pour gérer les locations d'immeubles. Cette application permettra aux propriétaires d'immeubles de gérer leurs propriétés, aux locataires de consulter les offres de location et aux administrateurs de gérer l'ensemble des utilisateurs et des opérations de location.*

#### **II. Objectifs**

##### **1. Gestion des Immeubles :**

- Ajouter, modifier et supprimer des immeubles.
- Ajouter des détails pour chaque immeuble, tels que l'adresse, le nombre d'unités, les équipements disponibles, etc.

##### **2. Gestion des Unités de Location :**

- Ajouter, modifier et supprimer des unités de location (appartements) dans les immeubles.
- Gérer les détails de chaque unité, tels que le nombre de pièces, la superficie, le loyer mensuel, etc.

##### **3. Gestion des Locataires :**

- Inscription et gestion des profils des locataires.
- Consultation des offres de location disponibles.
- Envoi de demandes de location et suivi de l'état de ces demandes.

##### **4. Gestion des Locations :**

- Enregistrement des contrats de location.
- Suivi des paiements des loyers.
- Génération de reçus et gestion des relances pour les paiements en retard.

##### **5. Administration :**

- Gestion des utilisateurs (propriétaires, locataires, administrateurs).
- Suivi des activités et génération de rapports.
- Configuration des paramètres de l'application.

#### **III. Technologies Utilisées**

##### **• Langages :**

- Java EE (Jakarta EE)
- Serveur d'application : Apache Tomcat
- Framework : Java EE (Servlets, JSP, EJB)

##### **• Base de données :**

- MySQL ou PostgreSQL

##### **• ORM :**

- JPA/Hibernate

#### **IV. Fonctionnalités Détails**

##### **1. Gestion des Immeubles :**

- **Ajouter un immeuble** : Formulaire pour ajouter un nouveau bien immobilier avec ses détails (nom, adresse, description, équipements, etc.).
  - **Modifier un immeuble** : Formulaire pour modifier les informations d'un immeuble existant.
  - **Supprimer un immeuble** : Option pour supprimer un immeuble et toutes ses unités associées.
2. **Gestion des Unités de Location** :
- **Ajouter une unité** : Formulaire pour ajouter une nouvelle unité de location avec ses détails (numéro d'unité, nombre de pièces, superficie, loyer, etc.).
  - **Modifier une unité** : Formulaire pour modifier les informations d'une unité de location existante.
  - **Supprimer une unité** : Option pour supprimer une unité de location.
3. **Gestion des Locataires** :
- **Inscription** : Formulaire d'inscription pour les nouveaux locataires.
  - **Connexion** : Système d'authentification pour les locataires inscrits.
  - **Profil locataire** : Page permettant aux locataires de voir et de modifier leurs informations personnelles.
  - **Consulter les offres** : Liste des unités de location disponibles avec options de filtrage (par prix, localisation, nombre de pièces, etc.).
  - **Demande de location** : Formulaire pour envoyer une demande de location pour une unité spécifique.
4. **Gestion des Locations** :
- **Contrat de location** : Formulaire pour enregistrer un nouveau contrat de location.
  - **Suivi des paiements** : Tableau de bord pour suivre les paiements des loyers (date, montant, statut).
  - **Reçus et relances** : Génération de reçus de paiement et envoi de relances pour les loyers impayés.
5. **Administration** :
- **Gestion des utilisateurs** : Interface pour ajouter, modifier et supprimer des utilisateurs (propriétaires, locataires, administrateurs).
  - **Suivi des activités** : Tableau de bord pour suivre les activités récentes et les statistiques d'utilisation de l'application.
  - **Rapports** : Génération de rapports sur les locations, les paiements, les utilisateurs, etc.
  - **Configuration** : Paramètres de l'application (notifications, paramètres système, etc.).

## V. Structure du Projet

### 1. Modèle de données :

- **Entités** : Immeuble, Unité de location, Locataire, Contrat de location, Paiement, Utilisateur
- **Relations** : Un immeuble peut avoir plusieurs unités de location. Un locataire peut avoir plusieurs contrats de location. Un contrat de location est associé à un immeuble et un locataire.

### 2. Architecture de l'application :

- **MVC (Model-View-Controller)** : Pour séparer les préoccupations et organiser le code.
- **DAO (Data Access Object)** : Pour accéder aux données de la base de données.

- *Services : Pour la logique métier.*
- *Contrôleurs : Pour gérer les requêtes HTTP.*
- *Pages : Pour les pages JSP*

## **VI. Développement**

### **1. Développement :**

- *Utilisation d'un IDE comme IntelliJ IDEA ou Eclipse.*
- *Utilisation de Maven pour la gestion des dépendances.*

### **2. Tests : (BONUS)**

- *Écriture de tests unitaires et d'intégration avec JUnit.*
- *Utilisation de MockMVC pour tester les contrôleurs Spring.*
- *Tests fonctionnels pour les différentes fonctionnalités de l'application.*