Projet SA LAC

1. Introduction 1
2. Architecture et Organisation du Code 2
3. Analyse Détaillée des Composants 4

1. Introduction

- **1.1. Objectif du Projet**
  Le projet SALAC est une application web conçue pour simuler les opérations d'une agence de location immobilière. Son objectif principal est de fournir une plateforme en ligne où les utilisateurs peuvent consulter les biens disponibles à la location (appartements et chambres), obtenir des informations sur l'agence, et potentiellement soumettre des demandes de location. Ce projet a été réalisé dans le cadre de mon BTS SIO, ce qui implique une dimension pédagogique et une démonstration de compétences en développement web.

- **1.2. Fonctionnalités Principales**
  Les fonctionnalités clés du projet incluent :
  - Affichage du catalogue de biens : Présentation des résidences, des appartements et des chambres avec leurs caractéristiques (type, loyer, photos) [cite: accueil.php, residence.php].
  - Gestion des informations de contact : Fourniture des coordonnées de l'agence et d'un formulaire de contact pour les demandes d'informations [cite: Coordonnees.php].
  - Téléchargement de documents : Mise à disposition du dossier de candidature au format PDF [cite: accueil.php, DossierInscription.pdf].
  - Navigation et structure du site : Organisation des différentes sections du site web pour une expérience utilisateur claire [cite: header.php, index.php].

2. Architecture et Organisation du Code

- **2.1. Structure des Fichiers**
  Le projet est organisé en plusieurs fichiers PHP, chacun ayant une responsabilité spécifique :
  - index.php : Ce fichier agit comme le contrôleur frontal. Il reçoit toutes les requêtes du navigateur et détermine quel contenu afficher en fonction de l'URL. Il inclut également les fichiers d'en-tête (header.php) et de pied de page (footer.php) pour assurer une structure cohérente sur toutes les pages [cite: index.php].

  - accueil.php : Affiche la page d'accueil du site, qui contient une présentation de l'agence, une description des services offerts et des instructions pour les utilisateurs [cite: accueil.php].

  - connect.php : Ce fichier est dédié à la gestion de la connexion à la base de données MySQL. Il établit la connexion en utilisant PDO (PHP Data Objects) et stocke les informations d'identification (serveur, nom d'utilisateur, mot de passe) [cite: connect.php].

  - Coordonnees.php : Affiche les coordonnées de l'agence (adresse, téléphone, email, réseaux sociaux) et inclut un formulaire permettant aux utilisateurs de poser des questions ou de demander de la documentation [cite: Coordonnees.php].

  - header.php : Contient le code HTML de l'en-tête du site, incluant la barre de navigation. Il effectue également une requête à la base de données pour récupérer la liste des résidences, qui est ensuite utilisée pour générer le menu de navigation [cite: header.php].

  - footer.php : Contient le code HTML du pied de page, affichant généralement des informations de copyright et des liens vers les réseaux sociaux [cite: footer.php].

  - residence.php : Affiche les détails d'une résidence spécifique, notamment

sa description, sa localisation et la liste des appartements ou chambres disponibles [cite: residence.php].

- DossierInscription.pdf : Un document PDF contenant le formulaire que les candidats locataires doivent remplir pour soumettre leur demande [cite: DossierInscription.pdf].

- **2.2. Flux d'Exécution**
  Le flux d'exécution typique d'une requête utilisateur est le suivant :

1. L'utilisateur saisit une URL dans son navigateur (par exemple, www.salac.fr/index.php?page=residence&codeRes=1).
2. Le serveur web reçoit la requête et exécute le fichier index.php.
3. index.php inclut connect.php pour établir une connexion à la base de données.
4. index.php inclut header.php, qui récupère les données des résidences depuis la base de données et génère la barre de navigation.
5. index.php détermine quelle page afficher en se basant sur le paramètre page dans l'URL ($_GET["page"]).
6. index.php inclut le fichier PHP correspondant à la page demandée (par exemple, residence.php). Si c'est residence.php, ce fichier récupère les détails de la résidence et des appartements depuis la base de données.
7. index.php inclut footer.php pour afficher le pied de page.
8. Le serveur envoie la page HTML complète au navigateur de l'utilisateur.

3. Analyse Détaillée des Composants

- **3.1. index.php (Contrôleur Frontal)**
  Ce fichier est crucial car il orchestre l'affichage de toutes les pages du site.

```
<?php
    $pagesAutorisee=["accueil", "residence","Coordonnees"];
    include './connect.php';
    include './header.php';

    if(isset($_GET["page"])){
        $page = $_GET["page"];
        }
    else{
        $page = 'accueil';
    }

    $page = isset($_GET["page"]) &&!empty($_GET["page"]) &&
in_array($_GET['page'],$pagesAutorisee) ? $_GET['page'] :
"accueil";
    include $page.'.php';
    include './footer.php';
?>
```

  - $pagesAutorisee : Ce tableau définit les pages qui peuvent être affichées par le site. C'est une mesure de sécurité simple pour empêcher l'inclusion de fichiers non autorisés.

  - include './connect.php'; et include './header.php'; : Ces lignes incluent les fichiers de connexion à la base de données et d'en-tête, respectivement.

  - Gestion de $_GET["page"] : Ce code détermine quelle page afficher. Si le paramètre page est présent dans l'URL, il est utilisé. Sinon, la page

d'accueil (accueil) est affichée par défaut. La présence du paramètre est vérifiée, s'il n'est pas vide, et s'il est dans le tableau $pagesAutorisee.

- include $page.'.php'; : Cette ligne inclut le fichier PHP correspondant à la page à afficher.

- include './footer.php'; : Inclut le pied de page.

- **3.2. connect.php (Connexion à la Base de Données)**
  Ce fichier gère la connexion à la base de données.

```php
<?php
    // Fichier de connexion à la base de données
    $server="localhost"; // Localhost car il est déjà sur le
conteneur
    $nomBD="SALAC"; // Nom de la BDD
    $login=base64_decode("YWRtaW5lcg==");
    $psw=base64_decode("QXplcnR5MTIz");

    try { // Test de fonctionalité
        $bdd= new
PDO("mysql:host=$server;dbname=$nomBD",$login,$psw);
    }
    catch(Exeception $e) { // Renvoie le message d'erreur
        echo 'Erreur: '.$e->getMessage."\n";
    }

?>
```

  - $server, $nomBD, $login, $psw : Ces variables stockent les informations d'identification de la base de données.

    Important : Le nom d'utilisateur et le mot de passe sont stockés encodés en base64. C'est une pratique *très* peu sûre. Dans une application réelle, ces informations devraient être stockées de manière beaucoup plus sécurisée (variables d'environnement, fichiers de configuration protégés).
  - try...catch : Ce bloc de code tente d'établir une connexion à la base de données en utilisant PDO. Si une erreur se produit, elle est capturée et un message d'erreur est affiché.

- **3.3. header.php (En-tête du Site)**
  Ce fichier génère l'en-tête HTML et la barre de navigation. Il récupère également la liste des résidences depuis la base de données.

```
<?php
$sql = "SELECT codeRes, nomRes
FROM Residences
ORDER BY nomRes;";
$reponse = $bdd -> query($sql);
$residences=$reponse->fetchall(PDO::FETCH_ASSOC);
?>
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet" href="./bootstrap/css/bootstrap.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.7.2/font/boot
strap-icons.css">
    <script src="bootstrap/js/bootstrap.js"></script>
    <title>location appartement</title>
</head>
<body class="d-flex flex-column min-vh-100">
    <header>
    <nav class="navbar navbar-expand-lg bg-body-tertiary
navbar-dark bg-dark bg-gradient" data-bs-theme="dark">
        <div class="container-fluid">
            <span class="navbar-brand">
                <a href="./index.php"
class="text-decoration-none">
                    <img src="./photos/logoSALAC.png" alt="logo"
height="75" class="d-inline-block align-text-middle">
                </a>
                SA LAC
            </span>
            <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
```

```
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link active"
href="./index.php?page=accueil">Accueil</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="./index.php?page=Coordonnees">Coordonnées</a>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle "
data-bs-toggle="dropdown" href="#" role="button"
                           aria-expanded="false">Résidences</a>
                        <ul class="dropdown-menu">
                            <?php foreach ($residences as
$residence){?>
                                <li>
                                    <a class="dropdown-item"
href="./index.php?page=residence&codeRes=<?php echo
$residence["codeRes"]?>">
                                        <?php echo
htmlentities($residence["nomRes"], ENT_QUOTES) ?>
                                    </a>
                                </li>
                            <?php } ?>
                        </ul>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link"
href="./administration/index.php?page=login">Administration</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
    </header>
```

- Requête SQL : La requête SELECT codeRes, nomRes FROM Residences ORDER BY nomRes; récupère les codes et noms de toutes les résidences de la base de données, triés par nom.

- Boucle foreach : Cette boucle parcourt les résultats de la requête et génère les liens HTML pour chaque résidence dans le menu déroulant.

- htmlentities() : Cette fonction est utilisée pour échapper les caractères spéciaux dans le nom de la résidence, ce qui est important pour la sécurité (prévention des XSS).

- **3.4. residence.php (Détails d'une Résidence)**
  Ce fichier affiche les informations détaillées d'une résidence spécifique, y compris la liste des appartements disponibles.

```
<?php

    if(isset($_GET["codeRes"])&&!empty($_GET["codeRes"])){
        $codeRes = $_GET["codeRes"];
        $sql = "SELECT *
            FROM Residences
            WHERE codeRes = '$codeRes';";
        $reponse = $bdd -> query($sql);
        $residence=$reponse->fetch();
        $sql = "SELECT codeRes, resContientTypes.codeType,
libelleType, loyerType, photoType
        From typesAppartStudio
        INNER JOIN resContientTypes ON
resContientTypes.codeType=typesAppartStudio.codeType
        WHERE codeRes='$codeRes';";
        $reponse = $bdd -> query($sql);
        $appartements=$reponse->fetchall(PDO::FETCH_ASSOC);
        $pIcone = $residence["parkingRes"]==1 ? "check-circle-fill
text-success" : "x-circle-fill text-danger";
        $pAscenseur = $residence["assenceurRes"]==1 ?
"check-circle-fill text-success" : "x-circle-fill text-danger";
    }

?>
<main class="container pt-3 ">
    <h1 class="text-center text-dark">
        La résidences :
        <span class="text-secondary"><?php echo
$residence["nomRes"]?></span>
    </h1>
    <hr>

    <div class="row">
        <div class="col">
            <div class="card mb-3">
                <div class="row g-0">
                    <div class="col-md-2">
```

```
                        <img src="./photos/<?php echo
$residence["photoRes"]?>" class="img-fluid rounded-start m-1"
alt="<?php echo $residence["nomRes"]?>">
                    </div>
                    <div class="col-md-4 align-self-center">
                        <div class="card-body p-0 ps-3">
                            <h4 class="card-title">
                                Résidence <?php echo
$residence["nomRes"]?>
                            </h4>
                            <p class="card-text"><?php echo
$residence["adresseRes"]?><br><?php echo $residence["cpRes"]?>
<?php echo $residence["villeRes"]?></p>
                        </div>
                    </div>
                    <div class="col-md-6 align-self-center">
                        <div class="card-body">
                            <div class="row border-start pb-4">
                                <div class="col">
                                    <h6 class="card-subtitle
text-muted text-center">
                                        Classification
                                        <?php for
($i=1;$i<=$residence["classificationRes"];$i++) {?>
                                            <i class="bi
bi-star-fill text-warning"></i>
                                        <?php } ?>
                                    </h6>

                                </div>
                            </div>
                            <div class="row border-start">
                                <div class="col">
                                    <h6 class="card-subtitle
text-muted text-center">
                                        Ascenseur <i class="bi
bi-<?php echo $pAscenseur?>"></i>
                                    </h6>
                                </div>
                                <div class="col">
                                    <h6 class="card-subtitle
text-muted text-center">
```

```
                                        Parking
                                        <i class="bi bi-<?php echo
$pIcone?>"></i>
                                    </h6>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>

        <div class="card-group">
            <?php foreach($appartements as $appartement){ ?>
                <div class="card">
                <div class="card-body pt-2">
                    <h5 class="card-title"><?php echo
$appartement["libelleType"]?></h5>
                    <p class="card-text">Loyer : <?php echo
$appartement["loyerType"]?>€</p>
                </div>
                <img src="./photos/<?php echo
$appartement["photoType"]?>" class="card-img-top img-fluid"
alt="T1">
            </div>
            <?php } ?>
        </div>
</main>
```

- Récupération de codeRes : Le code commence par vérifier si le paramètre codeRes est présent dans l'URL ($_GET["codeRes"]). Si c'est le cas, il récupère sa valeur.
- Requêtes SQL : Deux requêtes SQL sont exécutées :
  - La première récupère les informations de la résidence correspondant au codeRes.
  - La seconde récupère les informations sur les types d'appartements disponibles dans cette résidence.
- Affichage des données : Le reste du code HTML affiche les informations de la résidence et une liste des appartements disponibles à l'aide d'une

boucle foreach.