**Année 2025-02-25**

RAPPORT DE PROJET

**NFE114 - FOAD : Systèmes d'information web (2024 - 2025 Semestre 1)**

LE PROJET

**Le système d’information d’un cabinet médical**

Projet réalisé par

**Sendhilvelan RANDON**

**Le projet a été déployé à l'adresse suivante :**

[**http://randon-sendhil-projet-php-2025.free.nf**](http://randon-sendhil-projet-php-2025.free.nf)

**GITHUB**

**https://github.com/RANDONsendhil/php\_project**

Projet encadré par

**M. Arnaud RENARD**

**SOMMAIRE**

1. **Introduction**

*1. Contexte du projet :*

*2. Objectif*

*3. Public cible*

*4. Technologies envisagées*

*5. Bénéfices attendus*

1. **Cahier des charges**

*1. Présentation générale*

*2. Fonctionnalités attendues*

*3. Acteurs du projet*

1. **Choix technologiques**
2. **Conception de l'application**

*1. Introduction*

*2. Justification du choix du MVC*

*3. Structure du Projet*

**V. Explication de la structure du projet**

*1. Racine du projet /app-cabinet-medical/*

*2. Explication des composants*

*3. Composants principaux de l'application*

**VI. Implémentation des Modules**

### ***1. Gestion de l’Authentification (loginComponent)***

### ***2. Gestion des Rendez-vous (appointmentComponent)***

**VII. Implémentation de Base de donné relationnel avec MySql.**

1. *Les Acteurs du Système*
2. *Explication de l’Organisation des Données*
3. *Scénario d’Utilisation*
4. *Modèle Conceptuel des Données (MCD)*
5. *Modèle Logique des Données (MLD)*
6. **Déploiement**
7. **Création de l’Hébergement sur InfinityFree**
8. **Configuration de la Base de Données avec phpMyAdmin**
9. **Transfert des Fichiers PHP via FTP**
10. **Configuration de la Connexion à la Base de Données**
11. **Tests et Mise en Production**

**XI . Conclusion**

***1. Expérience et Gains Acquis en Réalisant ce Projet***

***2. Conclusion***

1. **Introduction.**
2. **Context du projet :**

Le but de ce projet consiste à mettre en place un système d’information dans un cabinet médical, permettant de gérer les dossiers patients, les rendez-vous et les interactions entre les différents professionnels de santé du cabinet. Un cabinet médical de taille moyenne peut rencontrer des difficultés liées à la gestion des données, à la communication interne et à la coordination des soins. L'objectif de ce projet est donc de développer un système d’information performant et sécurisé afin d’optimiser le fonctionnement du cabinet.

1. **Objectif.**

Le projet vise à développer une application web permettant :

La gestion numérique des dossiers patients : stocker et accéder facilement aux informations médicales en toute sécurité.

La planification et gestion des rendez-vous : permettre aux patients de prendre des rendez-vous en ligne et aux professionnels d'optimiser leur emploi du temps.

L'automatisation des processus administratifs : gérer la facturation, les ordonnances et les courriers médicaux.

Une communication fluide entre les acteurs de la santé : messagerie interne et notifications pour le suivi des patients.

1. **Public cible**

Le système d’information s’adresse principalement aux :

* Médecins et spécialistes travaillant au sein du cabinet.
* Secrétaires et assistants médicaux.
* Patients souhaitant accéder à leurs dossiers et prendre rendez-vous en ligne.
* Autres professionnels de santé en interaction avec le cabinet.

### Bénéfices attendus

L'implémentation de ce système d'information permettra :

* Une amélioration significative de la gestion administrative et clinique.
* Une réduction des erreurs et des oublis grâce à l'automatisation.
* Une meilleure expérience patient avec des services numériques accessibles.
* Un gain de temps et une augmentation de la productivité des professionnels de santé.

Ce projet représente ainsi une avancée majeure pour optimiser le fonctionnement du cabinet médical et améliorer la qualité des soins dispensés.

### II. Cahier des charges

### 1. Présentation générale

L’objectif principal du système est d’améliorer la gestion et la coordination des activités du cabinet médical grâce à une plateforme numérique centralisée. Cette solution devra être intuitive, sécurisée et accessible aux différents utilisateurs.

**2. Fonctionnalités attendues**

* **Gestion des patients** :
* Création, modification et suppression de dossiers patients.
* Historique des consultations et des traitements.
* **Gestion des rendez-vous** :
* Prise de rendez-vous en ligne et gestion des plages horaires.
* Rappels automatiques par email/SMS.
* **Gestion des professionnels de santé** :
* Profils des médecins et spécialités.
* Accès sécurisé aux dossiers patients selon les autorisations.
* **Automatisation administrative** :
* Gestion des factures et des paiements.
* Génération d’ordonnances et de courriers médicaux.
* **Sécurité et confidentialité** :
* Authentification par adresse mail unique et mot de passe

**3. Acteurs du projet**

* **Médecins** : Accès aux dossiers médicaux et gestion des consultations.
* **Secrétaires** : Gestion des rendez-vous et tâches administratives.
* **Patients** : Prise de rendez-vous et consultation des dossiers médicaux.
* **Administrateurs** : Gestion des accès et des paramètres du système.

**III. Choix technologique**

Le système sera développé en utilisant des technologies adaptées garantissant une interface intuitive et une haute sécurité :

* **Front-end** : Développement de l’interface utilisateur en **HTML, CSS, Bootstrap et jQuery** pour garantir une navigation fluide et réactive sur tous types d’écrans (ordinateurs). Bootstrap permettra une conception responsive et un design moderne.
* **Back-end** : Utilisation de **PHP** selon l’architecture **Modèle-Vue-Contrôleur (MVC)** afin de structurer le code de manière modulaire et évolutive, facilitant ainsi la maintenance et l’ajout de nouvelles fonctionnalités.
* **Base de données** : Mise en place d’un **système de gestion de bases de données relationnelles (MySQL)** assurant un stockage performant, sécurisé et structuré des informations médicales. L’optimisation des requêtes SQL garantira une rapidité d’accès aux données.
* **Hébergement et sécurité** : Hébergement du système sur **Infinity Free**, une plateforme offrant un hébergement stable et conforme aux normes de protection des données. Les mesures de sécurité comprendront le chiffrement des données sensibles, des certificats SSL, et des mécanismes de sauvegarde réguliers pour prévenir toute perte d’information.

**IV. Conception de l’application.**

## ****Choix de l'Architecture MVC pour le Projet****

### ****1. Introduction****

Dans le cadre du développement de notre application, nous avons choisi d’adopter l’architecture **Modèle-Vue-Contrôleur (MVC)**. Ce modèle de conception permet de structurer l’application en séparant la gestion des données, la logique de traitement et l'affichage, ce qui améliore la maintenabilité et la scalabilité du projet.

### ****2. Justification du choix du MVC****

L'architecture **MVC** offre plusieurs avantages essentiels pour notre projet :

* **Séparation des responsabilités** : chaque composant de l'application a un rôle bien défini.
* **Facilité de maintenance** : les modifications peuvent être effectuées sans impacter l’ensemble du système.
* **Réutilisation du code** : les modèles et vues peuvent être réutilisés dans différentes parties de l'application.
* **Évolutivité** : l’application peut facilement s’adapter à de nouveaux besoins sans refonte complète.

### ****3. Structure du Projet****

Notre projet est organisé selon la structure suivante :



Cette structure représente un **projet d’application web** pour un **cabinet médical**, conçu selon l’architecture **MVC (Modèle-Vue-Contrôleur)**. Chaque fonctionnalité principale est encapsulée dans un **composant dédié**, ce qui améliore l’organisation et la maintenabilité du projet.

**V. Explication de la structure du projet**

### ****Racine du projet**** /app-cabinet-medical/

La racine contient les fichiers globaux nécessaires au fonctionnement de l'application :

* **index.php** : Point d’entrée principal qui initialise l’application et charge le **routeur**.
* **router.php** : Gère la navigation entre les différentes pages et redirige les requêtes vers les contrôleurs appropriés.
* $routes = [
* '/' =>  BASE\_PATH . '/userCreationComponent/public/index.php',
* '/userCreation' =>  BASE\_PATH . '/userCreationComponent/public/index.php',
* '/login' =>  BASE\_PATH . '/loginComponent/view/login.php',
* '/login/submit-login'  =>   BASE\_PATH . '/loginComponent/public/index.php',
* '/home' =>  BASE\_PATH . '/user/homeComponent/public/index.php',
* '/profil' =>   BASE\_PATH . '/user/profilComponent/public/index.php',
* '/updateProfil' =>   BASE\_PATH . '/user/profilComponent/public/index.php',
* '/name' =>   BASE\_PATH . '/utilisateur/public/index.php',
* '/home/selectDoctor/appointment'  =>   BASE\_PATH . '/appointmentComponent/public/index.php',
* '/home/selectDoctor'  =>   BASE\_PATH . '/appointmentComponent/public/index.php',
* '/home/bookAppoinment'  =>   BASE\_PATH . '/appointmentComponent/public/index.php',
* '/home/displayAppointments'  =>   BASE\_PATH . '/appointmentComponent/public/index.php',
* '/contact-assistant'  =>   BASE\_PATH . '/user/profilComponent/public/index.php',
* '/find-doctors'  =>   BASE\_PATH . '/user/profilComponent/public/index.php',
* '/emergency-numbers'  =>   BASE\_PATH . '/user/profilComponent/public/index.php',
* ];
* **Dossiers /config/, /images/, /css/, /js/** :
  + **/config/** : Contient les fichiers de configuration comme la connexion à la base de données.
  + **/images/** : Stocke les images utilisées dans l’application.
  + **/css/** et **/js/** : Contiennent les fichiers CSS et JavaScript pour la mise en page et l’interactivité.
  1. **Explication des composants**

Chaque fonctionnalité est organisée en **composants** distincts qui suivent la structure **MVC** :

* **controllers/** → Contient les fichiers contrôleurs qui gèrent la logique de traitement.
* **model/** → Contient les modèles qui interagissent avec la base de données.
* **view/** → Contient les fichiers d'affichage (HTML/PHP).
* **public/index.php** → Point d'entrée du composant, qui appelle le contrôleur correspondant.

### ****Composants principaux de l'application****

#### adminComponent/ ****→ Gestion des administrateurs****

Ce module gère les actions des administrateurs, comme l'ajout ou la gestion des utilisateurs et des rendez-vous.

* **controllers/controllerAdmin.php** : Gère les requêtes liées à l’administration.
* **model/admin.php** : Contient les fonctions d'accès aux données des administrateurs.
* **view/admin.php** : Affiche les interfaces administratives.
* **public/index.php** : Charge le contrôleur et affiche la page d'administration.

#### appointmentComponent/ ****→ Gestion des rendez-vous****

Permet aux patients de prendre, modifier ou annuler des rendez-vous.

* **controllers/controllerAppointment.php** : Gère la logique métier des rendez-vous.
* **model/appointment.php** : Contient les requêtes SQL pour manipuler les rendez-vous.
* **view/appointment.php** : Page de prise et gestion des rendez-vous.
* **public/index.php** : Point d’accès pour ce module.

#### loginComponent/ ****→ Gestion de l’authentification****

📌 Permet aux utilisateurs (patients et administrateurs) de se connecter à leur compte.

* **controllers/controllerLogin.php** : Vérifie les identifiants et gère la connexion/déconnexion.
* **model/login.php** : Accède à la base de données pour vérifier les identifiants.
* **view/login.php** : Affiche le formulaire de connexion.
* **public/index.php** : Charge le contrôleur et affiche la page de connexion.

#### user/ ****→ Composants liés aux utilisateurs****

Ce dossier regroupe les fonctionnalités dédiées aux patients/utilisateurs.

##### **📍** homeComponent/ **→ Page d’accueil utilisateur**

* **controllers/controllerHome.php** : Gère le contenu dynamique de la page d’accueil.
* **model/home.php** : Récupère des informations liées à l’utilisateur (ex. : ses prochains rendez-vous).
* **view/home.php** : Affiche la page d’accueil.

##### **📍** profilComponent/ **→ Gestion du profil utilisateur**

* **controllers/controllerProfil.php** : Gère l’édition et l’affichage du profil.
* **model/profil.php** : Récupère et met à jour les informations de l’utilisateur.
* **view/profil.php** : Affiche la page du profil avec formulaire de mise à jour.

#### ****3.5 -**** userCreationComponent/ ****→ Création d’un compte utilisateur****

Permet aux nouveaux utilisateurs de s’inscrire sur la plateforme.

* **controllers/controllerUserCreation.php** : Gère la validation et l’ajout d’un nouvel utilisateur.
* **model/userCreation.php** : Ajoute un nouvel utilisateur dans la base de données.
* **view/userCreation.php** : Affiche le formulaire d’inscription.

## ****Configuration et fichiers globaux****

**/config/** → Contient les fichiers de configuration :

* **database.php** : Fichier de connexion à la base de données.
* **utils.php** : Contient des fonctions utilitaires pour toute l’application.

VI . **Implémentation des Modules**

### ****1. Gestion de l’Authentification (loginComponent)****

**Objectif** : Permettre aux utilisateurs (patients et administrateurs) de se connecter de manière sécurisée.

#### ****Modèle :**** model/login.php

Ce fichier contient la connexion à la base de données et la vérification des identifiants.

#### ****Contrôleur :**** controllers/controllerLogin.php

Gère la validation des identifiants et la création des sessions utilisateur.

#### ****Vue :**** view/login.php

Affiche le formulaire de connexion.

### ****2. Gestion des Rendez-vous (appointmentComponent)****

**Objectif** : Permet aux utilisateurs de prendre et gérer leurs rendez-vous médicaux.

#### ****Modèle :**** model/appointment.php

#### Contrôleur : controllers/controllerAppointment.php

#### Vue : view/appointment.php

L’implémentation de notre application suit une approche **modulaire et organisée** en respectant l’architecture **MVC**. Chaque module est indépendant et interagit via un **routeur central** qui gère les redirections. Grâce à cette conception, l’application est **facilement extensible** et **maintenable**.

**VII. Implémentation de Base de donné relationnel avec MySql.**

Ce projet met en place la **Base de donné relationnel avec MySql** permettant de gérer les utilisateurs (patients et administrateurs), les médecins et les rendez-vous.

## ****Les Acteurs du Système****

1. **Patient (Utilisateur) :**
   * Peut s’inscrire et se connecter à la plateforme.
   * Peut consulter la liste des médecins disponibles et leurs spécialités.
   * Peut réserver un rendez-vous en choisissant une date et un médecin.
   * Peut gérer ses rendez-vous ( annulé ou modifé).
2. **Administrateur :**
   * Gère les utilisateurs (patients).
   * Valide, annule ou reporte les rendez-vous.
   * Gère la liste des médecins et leurs disponibilités.
   * Peut consulter et organiser les rendez-vous à venir.

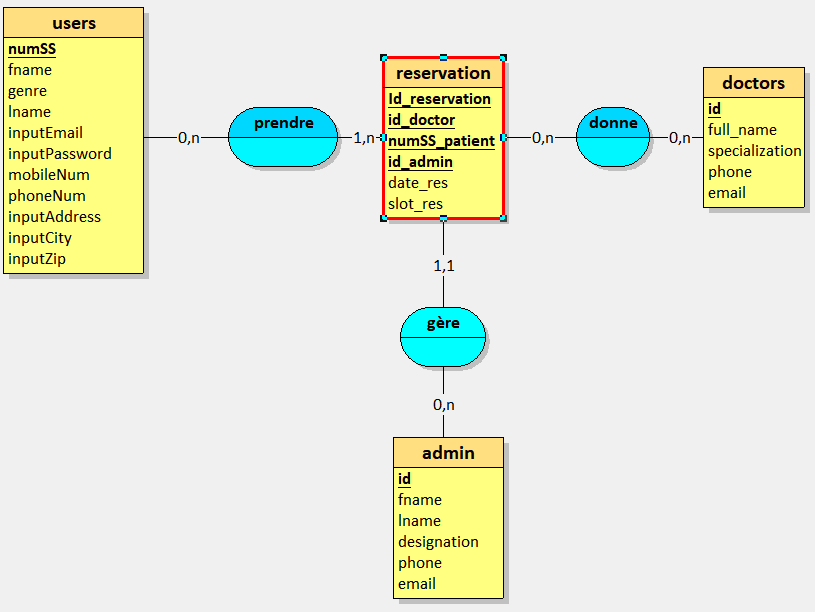
## ****Explication de l’Organisation des Données****

1. **Gestion des utilisateurs :**
   * Chaque **utilisateur** (patient ou administrateur) possède un identifiant unique, un nom, un email et un mot de passe.
   * L’utilisateur a un **rôle** qui peut être **patient** ou **administrateur**.
2. **Gestion des médecins :**
   * Chaque **médecin** possède un identifiant unique, un nom et une spécialité.
   * Les patients peuvent voir les spécialités disponibles et choisir un médecin correspondant.
3. **Gestion des rendez-vous :**
   * Un **rendez-vous** est associé à un **patient** et à un **médecin**.
   * Il comporte une **date et une heure**.
   * Son **statut** peut être **en attente**, **confirmé** ou **annulé**.

## ****Scénario d’Utilisation****

1. **Prise de rendez-vous par un patient**
   * Le patient se connecte à son compte.
   * Il consulte la liste des médecins et leurs spécialités.
   * Il choisit un médecin et une date disponible.
   * La demande de rendez-vous est enregistrée avec le statut **"en attente"**.
2. **Consultation des rendez-vous**
   * Le patient peut consulter ses rendez-vous et leur statut.
   * L’administrateur peut voir tous les rendez-vous organisés.

**Modèle Conceptuel des Données (MCD) du Système de Réservation de Rendez-vous :**

****

**Modèle Logique des Données (MLD) du Système de Réservation de Rendez-vous :**

**user**(genre,numSS,lname,fname,inputEmail,inputPassword,mobileNum,phoneNum,inputAddress,inputCity,inputZip)

**doctors**(id,full\_name,specialization, phone, email)

**reservation**(id\_reservation,id\_doctor,numssPatient,date\_res,slot\_res)

**admin**(id,fname,lname,designation,phone,email)

**VIII. Déploiement**

Le déploiement consiste à **mettre en ligne l’application PHP/MYSQL** pour qu’elle soit accessible aux utilisateurs. Voici les étapes à suivre :

### ****Déploiement du Système de Réservation de Rendez-vous Médicaux avec InfinityFree et phpMyAdmin****

L’hébergement du projet est réalisé sur **InfinityFree**, un service d’hébergement gratuit, avec **phpMyAdmin** pour la gestion de la base de données. Le processus de déploiement suit plusieurs étapes clés.

## ****Création de l’Hébergement sur InfinityFree****

1. Inscription sur [**InfinityFree**](https://www.infinityfree.net/) et accès au **cPanel**.
2. Création d’un **nouvel hébergement** avec un sous-domaine gratuit ou un domaine personnalisé.
3. Accès aux informations d’hébergement, y compris l’**URL du site**, le **nom d’utilisateur FTP**, et l’**hôte MySQL**.

## ****Configuration de la Base de Données avec phpMyAdmin****

1. Accès à la section **"MySQL Databases"** dans le cPanel.
2. Création d’une **nouvelle base de données**, récupération des informations nécessaires :
   * **Nom de la base de données** (ex: epiz\_12345678\_cabinet).
   * **Hôte MySQL** (ex: sql123.infinityfree.com).
   * **Nom d’utilisateur et mot de passe MySQL**.
3. Ouverture de **phpMyAdmin** et **importation du fichier SQL** contenant la structure et les données de la base.

## ****Transfert des Fichiers PHP via FTP****

1. Utilisation d’un client FTP comme **FileZilla**.
2. Connexion avec les **identifiants FTP** fournis par InfinityFree :
   * **Hôte** : ftp.epizy.com
   * **Nom d’utilisateur** : fourni par InfinityFree
   * **Mot de passe** : défini lors de la création de l’hébergement
3. Navigation vers le dossier /htdocs/ et **téléversement des fichiers du projet**.

## ****Configuration de la Connexion à la Base de Données****

1. Modification du fichier **config/database.php** avec les informations correctes :

private $host = "sql305.infinityfree.com:3306";

private $username = "if0\_38266839";

private $password = "AT8u8qJztPkKz";

private $database = "if0\_38266839\_cnam\_php";

1. Sauvegarde et mise à jour du fichier sur le serveur via FTP.

## ****Tests et Mise en Production****

1. Accès au site via l’**URL fournie par InfinityFree**.

## randon-sendhil-projet-php-2025.free.nf

1. Vérification du bon fonctionnement :
   * Connexion et inscription des utilisateurs.
   * Prise de rendez-vous.
   * Gestion des réservations par l’administrateur.
2. Correction des éventuels problèmes et validation finale.

**VIII. Conclusion.**

Le déploiement du système d’information du cabinet médical sur **InfinityFree** avec **phpMyAdmin** a permis de mettre en ligne une solution accessible aux patients et aux administrateurs. L'utilisation d'InfinityFree comme hébergeur a offert une alternative gratuite et efficace pour l’hébergement du projet, tandis que phpMyAdmin a facilité la gestion de la base de données.

. L’intégration d’un modèle **MVC (Modèle-Vue-Contrôleur)** a structuré l’application pour une meilleure maintenabilité et évolutivité.

Ce projet peut être amélioré par l’optimisation des performances, l’ajout d’un certificat **SSL** pour sécuriser les échanges, ainsi que l’intégration de nouvelles fonctionnalités comme les rappels de rendez-vous par e-mail ou SMS. En conclusion, ce déploiement marque une étape clé dans la mise en production d’un service fonctionnel et accessible en ligne. 🚀

### ****1. Expérience et Gains Acquis en Réalisant ce Projet****

La conception et le déploiement du système d’infomation du cabinet médical m’ont permis d’acquérir une expérience précieuse dans plusieurs domaines du développement web.

#### ****Maîtrise de l’Architecture MVC****

L’application a été développée en suivant le modèle **MVC (Modèle-Vue-Contrôleur)**, ce qui a permis de structurer le code de manière modulaire et maintenable. Cette approche a renforcé la compréhension de la séparation des responsabilités dans une application web.

#### ****Gestion d’une Base de Données****

La conception du **MCD** et du **MLD**, ainsi que leur implémentation avec **MySQL**, ont permis de mieux appréhender la modélisation des données et les relations entre les différentes entités. L’importation et la gestion des données via **phpMyAdmin** ont renforcé la pratique de la gestion des bases de données.

## *****Développement Local avec WAMP Server*****

Avant le déploiement en ligne, l’application a d’abord été développée et testée en local avec **WAMP Server**. Cela a permis de :  
 **Configurer un serveur Apache et MySQL localement** pour exécuter l’application sans connexion Internet.  
 **Tester le bon fonctionnement du code PHP et des requêtes SQL** avant de les mettre en production.  
 **Utiliser phpMyAdmin en local** pour créer, gérer et tester la base de données.  
 **Détecter et corriger les erreurs** avant de passer à la mise en ligne, ce qui a facilité le débogage.

## *****Hébergement et Déploiement sur InfinityFree*****

Après la validation du fonctionnement en local, le projet a été **déployé sur InfinityFree**, ce qui a permis de :  
 **Apprendre à configurer un hébergement web** et à transférer les fichiers du projet via **FTP**.  
 **Mettre en place une base de données distante** en exportant le schéma depuis **WAMP Server** et en l’important sur **phpMyAdmin** d’InfinityFree.  
**Adapter le fichier de configuration** (config/database.php) pour utiliser les identifiants et l’hôte de la base de données en ligne.  
 **Gérer les différences entre environnement local et distant**, notamment les permissions des fichiers et les restrictions de l’hébergement gratuit.

#### ****Sécurisation et Optimisation****

La mise en place de la gestion des utilisateurs avec un système de connexion sécurisé (hachage des mots de passe) a permis d’approfondir les notions de sécurité web. L’optimisation des requêtes SQL et l’amélioration des performances du site ont également été des points clés du projet.

#### ****Expérience Globale en Développement Web****

Ce projet a permis d’acquérir une expérience concrète en développement **PHP/MySQL**, en intégration avec **HTML, CSS et JavaScript**, et en gestion de projet. Il a aussi renforcé les compétences en **résolution de problèmes**, notamment face aux erreurs rencontrées lors du déploiement.

### ****Conclusion.****

La réalisation de ce projet a été une expérience enrichissante, combinant **développement backend, gestion de base de données et déploiement web**. Elle a permis d’acquérir des compétences pratiques essentielles et d’améliorer la capacité à concevoir et mettre en ligne une application fonctionnelle et sécurisée.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*