

## Développement d'une application web avec Python et MySQL (Génie Mécatronique)

## Développement d'une application web avec Python et Oracle Apex (Génie Informatique)

### Objectif principale

Créer une application web en Python pouvant interagir avec une base de données MySQL (G.Méca) et Oracle Apex (G.Info) pour effectuer des requêtes basées sur les interactions des utilisateurs. Pour G. Indus, l'application web devrait être développée avec Vb Net et Access.

### Etape à suivre pour Génie Mécatronique :

- Démarrez MySQL via Docker ou XAMPP
- Importer la base de données
- Créer et démarrer le projet Python
- Vérifiez la connexion à la base de données via les informations d'identification.
- Astuce : utilisez `st.write()` pour examiner vos résultats.

### Description de la base de données

La base de données s'appelle GYM et concerne les activités dans une salle de sport. Elle est décrite par le schéma logique suivant (les clés primaires sont soulignées, les clés étrangères sont en italique) :

**ENTRAINEUR** (CodeE, Nom, Prénom, Date de naissance, Email, Numéro de téléphone)

**SEANCE** (Id-S, Nom, Type, Niveau)

**HORAIRE** (*CodeE*, Jour, Heure de début, Durée, *Id-S*, GymSalle)

Utilisez l'interface phpMyAdmin et Apex pour importer le script `createDB.sql` et vérifiez c'est réussi. Utilisez les interfaces pour explorer la base de données, les tables disponibles et les données enregistrées.

## Travail demandé :

Créez une application multipage pour afficher les principales informations contenues dans la base de données et permettre à l'utilisateur d'ajouter de nouvelles informations. En particulier :

1. Créez une page d'accueil personnalisée qui exploite la syntaxe Markdown de base ou les éléments Streamlit (Avec Streamlit, vous pouvez créer de belles applications web axées sur les données en seulement quelques lignes de code Python) pour présenter le projet, l'objectif et l'étudiant qui le réalise.

2. Créez une page pour afficher et filtrer les séances disponibles. La page doit être prise en charge par deux métriques pour afficher le nombre de séances et les types distincts disponibles. Les widgets d'entrée doivent être créés de manière à ce que les options suggérées soient basées sur les informations déjà présentes dans la base de données. L'utilisateur doit être en mesure d'afficher les informations sur les séances en les filtrant par plusieurs catégories (c'est-à-dire le type) et doit être en mesure de spécifier la plage de niveaux qui l'intéresse.

Dans un expander séparé, visualisez les horaires des séances sélectionnées. En cas de résultats vides, une erreur/avertissement correspondant doit être imprimé.

3. Créez une page pour afficher les entraîneurs disponibles. L'utilisateur doit avoir la possibilité de filtrer en tapant le nom de famille de l'entraîneur et en utilisant une plage de dates pour choisir par date de naissance (utilisez `datetime.date()` pour définir le `date_input` et transmettre la date sous forme de chaîne dans la requête). L'affichage ne doit pas être une table globale, mais divisé élément par élément (créez un dataframe et utilisez `df.iterrows` pour imprimer une ligne à la fois. Si les résultats sont vides, un message associé doit être affiché.

4. Créez une page pour afficher des graphiques sur les séances programmées. La page doit contenir deux graphiques : un graphique à barres qui montre le nombre de séances pour chaque plage horaire et un graphique à courbes qui montre le nombre de séances programmées en fonction du jour de la semaine.

5. Créez une page pour l'insertion de nouvelles séances via un formulaire approprié. Utilisez un formulaire de saisie qui demande toutes les données nécessaires à l'insertion d'une nouvelle séance dans la base de données (Id-S, Nom, Type, Niveau). Votre application doit vérifier que tous les champs ont une valeur et que la valeur de l'attribut Niveau est un entier compris entre 1 et 4 (utilisez `number_input`). En cas de données manquantes, de clé en double ou d'autres erreurs, l'application doit générer un message d'erreur. Si, par contre, les données saisies sont correctes et que l'opération d'insertion est réussie, un message d'insertion correcte doit être affiché.

6. Créez un formulaire pour insérer une nouvelle séance hebdomadaire dans la table Horaire. Le formulaire doit vous permettre de renseigner tous les champs nécessaires (CodeE, Jour, Heure de début, Durée, Id-S, GymSalle) liés à la programmation d'un nouveau créneau. La sélection de l'entraîneur doit s'effectuer via un menu déroulant contenant le CodeE des entraîneurs possibles générés par le contenu de la table de la base de données. De même, la sélection de la séance doit également s'effectuer via un menu déroulant alimenté par la base de données. Les autres champs sont des champs renseignés manuellement par l'utilisateur, à l'aide des widgets les plus adaptés (par exemple, curseur pour Heure de début, Durée) ou textuels.

L'application vérifie que l'utilisateur n'essaie pas d'inclure dans la programmation des séances d'une durée supérieure à 60 minutes et que le jour indiqué est un jour compris entre le lundi et le vendredi. L'insertion d'une nouvelle séance dans l'horaire doit être autorisée et exécutée si et seulement si aucune autre séance n'est programmée pour la même séance et le même jour de la semaine (indice : utilisez les valeurs d'entrée pour interroger et vérifier qu'il n'y a aucun enregistrement). Si la demande d'insertion respecte les contraintes indiquées et l'insertion se termine avec succès, un message d'insertion correcte doit être affiché, sinon un message d'erreur doit être notifié (le message d'erreur doit indiquer le type de problème qui a provoqué l'erreur).

Toutes les pages doivent être personnalisées avec des éléments de texte (à l'aide de démarques ou des widgets prédéfinis de python) afin que les titres, sous-titres et paragraphes mettent en évidence ce qui est représenté. En plus de générer les bonnes requêtes, pour rendre la visibilité et l'interface plus intuitives et organisées, les principaux éléments de mise en page doivent être utilisés : extensions, colonnes, onglets.

# Développement d'une application web avec Visual Basic Net et Access database (Génie Industriel)

## Objectif :

La conception d'une base de données de gestion était nécessaire pour faciliter les opérations quotidiennes des salariés, notamment pour l'insertion de dizaines de nouvelles offres chaque semaine. L'ancienne méthode utilisée nécessitait une mise à jour manuelle des feuilles Excel et il était souvent nécessaire de comparer les anciennes offres ; Toutefois, la consultation du Fichier Unique s'est révélée fastidieuse en raison de l'énorme quantité de lignes. La nouvelle base de données peut être exécutée dans son intégralité à travers des formulaires qui peuvent être facilement utilisés même par des utilisateurs sans la moindre connaissance de Microsoft Access et de la programmation en général. Le remplissage automatique des cellules est alors exploité de manière adéquate, facilitant la modification continue des données et la saisie cyclique des informations.

Les formulaires de comparaison permettent de visualiser simultanément le produit analysé proposé par différents fabricants, en fournissant un rapport fiable pour choisir un prix qui convient au client en le comparant avec ceux déjà sur le marché.

## Description de la base de données :

**CLIENTS** (Id\_client, Nom\_ctl, TaxeCtl, Adresse\_Clt, Tel\_Clt, Magasin\_Clt, TVA, Id\_Pays)

**FOURNISSEURS** (Id\_Fournisseur, Nom\_Fr, Taxe\_Fr, Adresse\_Fr, Tel\_Fr, TVA, Id\_Pays)

**PRODUITS** (Id\_Produit, Id\_Marque, Code\_Fabriqueur, Nom\_Prod, Certification, Poids, lot\_de\_production, unité\_de\_production, longueur\_Prod, Largeur\_Prod, Hauteur\_Prod, catégorie, Sous\_categorie, Prix par boîte)

**FABRIQUANT- MARQUE** ( Code\_Fabriqueur , Id\_Marque, Id\_Fournisseur)

**PAYS** (Id\_Pays, pays)

## Travail demandé :

Les formulaires présents dans la base de données donnent la possibilité aux utilisateurs d'interagir avec toutes les tables existantes sans y accéder directement. L'ouverture du programme active automatiquement une interface dans laquelle on indique l'utilisateur au formulaire nécessaire pour la fonction qu'il souhaite utiliser : insérer, modifier ou visualiser/Recherche

L'insertion : contient les 3 formulaires d'insertion (fournisseur, client, produit)

La modification : contient les 3 formulaires de modification ( client, fournisseur avec la marque qu'il commercialise, produit)

La recherche : : c'est la section la plus conséquente car elle contient les formulaires d'affichage des données (fournisseur, client, produit, marque), les formulaires d'affichage des offres (un pour le référencement et un ciblage pour filtrer les offres par rapport au client ou au fournisseur) et les formulaires pour comparaison entre offres (par produit, par client ou par fournisseur).

L'application doit donner la possibilité de créer une nouvelle offre aux clients intéressés par un produit :

Le formulaire de saisie le plus important est « Insérer une nouvelle offre ». L'offre consiste à proposer un produit choisi à un client intéressé. S'il est accepté, le produit est "répertorié" et inséré par la chaîne de distribution dans ses points de vente.

Compte tenu de la grande quantité de produits à proposer chaque semaine, le formulaire a pour but d'automatiser et de rendre le plus efficace possible le travail de l'employé chargé de gérer les offres.

Le formulaire vous demande dans un premier temps de choisir le client (et la TVA appliquée pour le type de produit dans son pays), le fournisseur et le produit à proposer. L'insertion de ce dernier remplit automatiquement les champs du formulaire avec les données du produit (dimensions, poids, coûts, certifications et données logistiques).