



République du Sénégal
Un peuple-Un but-Une foi

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
DE LA RECHERCHE ET DE L'INNOVATION



UNIVERSITÉ IBA DER THIAM DE THIES
Master Sciences des Données et Applications

PROJET : PLATEFORME ÉTUDIANT

MEMBRES

- ◇ Ndeye Maïmouna Diop
Option : Statistique économétrie et modélisation
- ◇ Ndemba Ndiaye
Option : Statistique économétrie et modélisation
- ◇ El Hadji Malick Sarr
Option : Contrôle audit et data analytisc

Année: 2023/2024

Plan

INTRODUCTION

- I. MODELISATION DE L'APPLICATION
- II. CREATION DE LA BASE DE DONNEES
 AVEC MYSQL
- III. INTERFAÇAGE DE PYTHON ET DE MYSQL
- IV. CODAGE AVEC PYTHON

CONCLUSION

INTRODUCTION

Le Projet plateforme étudiant consiste à développer une application destinée aux étudiants grâce à l'utilisation combinée de PYTHON et MYSQL. L'objectif est de faciliter aux étudiants l'accès aux données les concernant. Ceci en leur fournissant une plateforme optimale avec une interface fluide et facile d'utilisation pour le renseignement et la modification d'informations.

La mise en œuvre de l'application s'inspire de la plateforme mise en place par l'université Iba Der Thiam de Thiès, qui répond aux difficultés rencontrées par les étudiants pour se renseigner, consulter et modifier des informations relatives à leur statut d'étudiant sans déplacement. Elle permet également aux étudiants de mieux suivre l'évolution de leur parcours universitaire.

I. MODELISATION DE L'APPLICATION

La Modélisation de l'application plateforme étudiant consiste à façonner un modèle de l'application avec Adobe Illustrator pour une vision globale des différents paramètres de l'application à concevoir, des éléments d'identité graphiques et de branding (logo, nom) à créer et pour pouvoir avoir un aperçu de l'application future.



On se retrouve avec ce modèle qu'on va s'efforcer de mettre en place.

II. CREATION DE LA BASE DE DONNEES AVEC MYSQL

La création de la base de données avec le logiciel MYSQL est fondamentale pour la mise en place de l'application en effet elle représente un bloc d'informations à fournir à notre application. Tels que : les noms

d'utilisateurs, identifiants etc.... De plus la base de données a une fonctionnalité de stockage.

Pour la création de l'application la base de données étudiant comportant les tables informations personnelles et parcours a été créer avec MYSQL.

La table information regroupe toutes les informations d'identifications propres à l'étudiant à savoir :

- Prénom & Nom de l'Etudiant ;
- Nationalité
- Adresse ;
- Id de l'étudiant ;
- Niveau d'étude ;
- Genre ;
- Situation Matrimoniale
- Email ;
- Mot de passe ;
- Date de naissance.

La clé primaire choisie pour cette table est l'id de l'étudiant pour son caractère unique et non nul.

La table parcours comporte des informations à citer :

- Niveau d'étude ;
- Spécialité ;
- UFR ;
- Année de validation ;
- Année scolaire ;
- Id de l'étudiant.

La clé primaire pour cette table correspond à la liaison des colonnes id et année de validation.

Dans notre cas il n'y a pas de clé étrangère car il n'y existe pas de colonne dans une table qui correspond a une clé primaire dans l'autre.

III. INTERFAÇAGE DE PYTHON ET DE MYSQL

L'interfaçage de python et de MySQL permet une utilisation combinée entre le logiciel de création de base de données : MySQL et le langage de programmation : python pour créer une application.

Le processus d'interfaçage permet de connecter les deux environnements grâce à l'utilisation de la bibliothèque MySQL Connector python contenant l'ensemble des outils pour réaliser la connexion et pouvoir directement avoir accès à la base de données via python.

IV. CODAGE AVEC PYTHON

Le coding constitue le paramétrage de notre application avec le langage de programmation python plus précisément avec l'utilisation de la bibliothèque tkinter pour ouvrir une fenêtre et paramétrer les options de l'application.

On va maintenant s'intéresser aux différentes parties du code et à leur fonctionnalité

<pre>import tkinter as tk from tkinter import messagebox import mysql.connector from mysql.connector import Error from tkinter import font</pre>	<p>Les bibliothèques `tkinter` et `mysql.connector` sont importées pour respectivement créer une interface graphique et se connecter à une base de données MySQL. Le module `messagebox` de `tkinter` permet d'afficher des boîtes de dialogue pour les erreurs et avertissements, tandis que `font` personnalise les polices. La classe `Error` de `mysql.connector` gère les erreurs liées à MySQL.</p>
--	---

<pre># Configuration de la base de données configuration = { 'user': 'root', 'password': 'donotforgetme3', 'host': '127.0.0.1', 'database': 'Etudiants', 'raise_on_warnings': True }</pre>	<p>Une bibliothèque est créée pour stocker les différentes informations de connexion a la base de données MySQL.</p>
<pre># Création de la connexion et gestion des erreurs try: connexion = mysql.connector.connect(**configuration) print("Connexion réussie") except mysql.connector.Error as err: print(f"Erreur lors de la connexion : {err}") exit()</pre>	<p>On tente de se connecter à la base de données. En cas de succès, « Connexion réussie » s'affiche, sinon un message d'erreur apparaît.</p>
<pre># Fonction pour changer la couleur des boutons en fonction de l'endroit où on se trouve def update_button_colors(active_button): buttons = [info_button, parcours_button, autre_button] for button in buttons: button.config(bg='#F5F5DC', fg='black') active_button.config(bg='#0B3D91', fg='white')</pre>	<p>Cette fonction permet de changer l'apparence des boutons dans l'interface graphique en fonction de notre emplacement dans l'application.</p>
<pre># Fonction pour masquer les champs de connexion def hide_login_fields(): email_label.pack_forget() email_entry.pack_forget() password_label.pack_forget() password_entry.pack_forget() login_button.pack_forget()</pre>	<p>La fonction retire de l'interface les éléments de la page de connexion sans les détruire, permettant un accès ultérieur si nécessaire.</p>
<pre># Fonction pour récupérer les données d'un seul étudiant avec email et mot de passe def get_student_data(): email = email_entry.get() password = password_entry.get() if not email or not password: messagebox.showwarning("Avertissement", "Veuillez entrer l'email et le mot de passe.") return try: ... except mysql.connector.Error as err: messagebox.showerror("Erreur", f"Erreur lors de la récupération des données : {err}")</pre>	<p>Cette fonction récupère l'email et le mot de passe, puis cherche les informations correspondantes dans la table 'InformationsPersonnelles'. Si elles sont trouvées, les données sont affichées ; sinon, un message d'erreur est affiché.</p>

<pre>def get_student_parours(): email = email_entry.get() try: cursor = connexion.cursor() query = """ ... cursor.execute(query, (email,)) rows = cursor.fetchall() if rows: ... else: messagebox.showinfo("Information", "Aucun parcours trouvé pour cet étudiant.") cursor.close() except mysql.connector.Error as err: messagebox.showerror("Erreur", f"Erreur lors de la récupération des données : {err}")</pre>	<p>Cette fonction récupère le parcours de l'étudiant en effectuant une jointure entre les tables 'InformationsPersonnelles' et 'Parcours'.</p>
<pre># Création de la fenêtre principale appli = tk.Tk() appli.title('Plateforme étudiant') appli.config(bg='#FFFFFF') appli.geometry('600x800') # Définir les polices title_font = font.Font(family="Helvetica", size=16, weight="bold") text_font = font.Font(family="Verdana", size=12)</pre>	<p>Cette partie du code crée une fenêtre avec des propriétés telles que le titre ou encore la couleur de fond. De plus elle définit les différentes polices à utiliser dans l'interface.</p>
<pre># Champs de saisie pour l'email et le mot de passe email_label = tk.Label(appli, text="Entrez l'email :", bg='#FFFFFF', fg='black', font=title_font) email_label.pack(pady=5) email_entry = tk.Entry(appli, width=30, font=text_font) email_entry.pack(pady=5) password_label = tk.Label(appli, text="Entrez le mot de passe :", bg='#FFFFFF', fg='black', font=title_font) password_label.pack(pady=5) password_entry = tk.Entry(appli, show='*', width=30, font=text_font) password_entry.pack(pady=10) # Boutons de connexion login_button = tk.Button(appli, text="Se connecter", command=get_student_data, font=title_font) login_button.pack(pady=10)</pre>	<p>On met en place ici l'interface de la page de connexion avec notamment des champs pour saisir l'email et le mot de passe et un bouton connexion.</p>
<pre># Cadre pour le texte d'affichage text_frame = tk.Frame(appli) text_area = tk.Text(text_frame, wrap=tk.WORD, height=15, width=70, font=text_font) text_area.pack() # Cadre de navigation nav_frame = tk.Frame(appli) info_button = tk.Button(nav_frame, text="Information", command=get_student_data, font=title_font) info_button.pack(side=tk.LEFT, padx=5, pady=5) parcours_button = tk.Button(nav_frame, text="Parcours", command=get_student_parours, font=title_font) parcours_button.pack(side=tk.LEFT, padx=5, pady=5) autre_button = tk.Button(nav_frame, text="Autre", command=None, font=title_font) # espace réservé pour une implementation autre_button.pack(side=tk.LEFT, padx=5, pady=5)</pre>	<p>Ce bout de code crée l'interface de l'application après une connexion réussie, comprenant un cadre de navigation avec les boutons "Information", "Parcours" et "Autre", ainsi qu'une zone de texte pour afficher les données de l'étudiant.</p>

<pre>appli.mainloop()</pre>	Lance la boucle principale de l'application, gère les actions de l'utilisateur et maintient la fenêtre de l'application ouverte.
<pre># Fermer la connexion à la base de données if connexion.is_connected(): connexion.close() print("Connexion à la base de données fermée.")</pre>	Ferme la connexion a la base de données.

Enfin on se retrouve avec cette interface graphique pour notre application en utilisant les informations de l'utilisateur Eden Diop.

CONCLUSION

La **Plateforme Étudiant**, conçue en utilisant **Python** et **MySQL**, simplifie l'exécution des tâches pédagogiques en offrant une gestion fluide des informations personnelles et en permettant aux étudiants de suivre l'évolution de leur parcours académique, un droit fondamental dans l'enseignement. Grâce à son interface intuitive, elle améliore l'accessibilité et la transparence des données étudiantes.

Cependant, plusieurs améliorations peuvent être envisagées pour enrichir et compléter l'application :

- **Ajout de nouvelles fonctionnalités** : Intégrer des sections supplémentaires, telles que les relevés de notes, les documents

administratifs ou la planification des examens, pour offrir une vue d'ensemble plus complète aux étudiants.

- **Amélioration de l'interface utilisateur** : Ajouter des fonctionnalités permettant aux utilisateurs de s'inscrire directement via l'application ou de modifier leurs informations personnelles, tout en rendant l'interface plus interactive et agréable.
- **Renforcement de la sécurité** : Mettre en place des mesures de protection supplémentaires.

Ces améliorations contribueraient à rendre la plateforme non seulement plus fonctionnelle, mais également plus sécurisée et conviviale pour les utilisateurs.