

Rapport de Gestion des étudiants



ECOLE NATIONAL
DE SCIENCE
APPLIQUÉE AL
HOCEIMA

PRÉSENTÉ PAR:

CHAYMAE MERHRIQUI
SALMA KELLALI
SALMA FANNICH
CHADI MOUNTASSIR

ENCADRÉ PAR :

Mm A.Bengag



INTRODUCTION

REMERCIEMENT

LANGAGE UTILISÉE

**CONCEPTION DES PAGES
PROPOSÉS**

OUTILS UTILISÉS



INTRODUCTION :

La gestion des étudiants est un aspect essentiel de tout établissement d'enseignement, qu'il s'agisse d'une école, d'un collège ou d'une université. La gestion efficace des étudiants implique la collecte et la mise à jour des données des étudiants telles que les informations personnelles, les notes, les absences et la performance académique. Cela nécessite également une communication fluide avec les étudiants et leurs parents ou tuteurs.

Le projet de gestion des étudiants consiste à développer un système automatisé qui permettra de gérer toutes les activités liées aux étudiants de manière efficace. Il s'agit notamment de la gestion des admissions, de la gestion des inscriptions, de la gestion des notes, de la gestion des absences, de la communication avec les parents et de la génération de rapports.

Le système de gestion des étudiants sera capable de stocker les informations des étudiants et de les mettre à jour en temps réel. Il permettra également aux enseignants de saisir les notes et les absences des étudiants de manière efficace. Les parents pourront également accéder au système pour suivre la performance académique de leurs enfants.

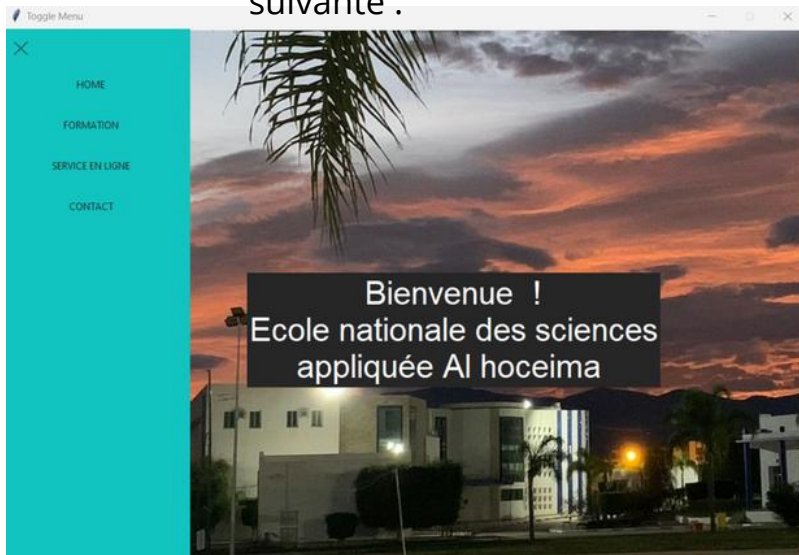
REMERCIEMENT :

Je tiens à exprimer ma sincère gratitude à notre professeur AMINA BENGAG pour nous avoir donné l'opportunité de travailler sur ce projet de gestion des étudiants. Cette expérience nous avons permis de mieux comprendre l'importance de la gestion efficace des étudiants dans un établissement d'enseignement.

Nous sommes reconnaissant envers notre professeur pour son soutien et ses conseils tout au long du projet. Sa passion pour l'enseignement et son expertise dans le domaine m'ont inspiré à m'impliquer davantage dans mes études et à développer mes compétences en matière de gestion de projet.

nous sommes fier du travail accompli et nous sommes convaincu que les compétences que nous avons acquises dans ce projet nous seront utiles dans notre future carrière. Encore une fois, nous remercions chaleureusement mon professeur pour cette opportunité et pour son soutien constant.

Notre première page est la suivante :



Ce code est un programme GUI (graphical user interface) écrit en Python avec l'utilisation de la bibliothèque Tkinter pour créer une fenêtre et un menu de navigation. Le programme comporte plusieurs fonctions qui définissent les différentes pages que l'utilisateur peut accéder en cliquant sur les boutons du menu. La page d'accueil est définie dans la fonction `home_page()`, la page de connexion est définie dans la fonction `login_page()`, la page de connexion en ligne est définie dans la fonction `seconnecter_page()` et la page de contact est définie dans la fonction `contact_page()`.

La fonction `toggle_win()` définit la mise en page du menu de navigation. Elle crée une nouvelle fenêtre avec un fond de couleur et définit les boutons avec des couleurs de fond et de texte différentes pour chaque bouton. Enfin, la fonction `dele()` est appelée pour supprimer le cadre de la fenêtre précédente après que l'utilisateur ait cliqué sur un bouton de navigation.

En somme, le code est destiné à créer une interface utilisateur graphique simple avec un menu de navigation pour permettre à l'utilisateur d'accéder facilement à différentes pages de l'application.

Le code utilise trois packages différents :

tkinter : C'est une bibliothèque Python qui permet de créer des interfaces graphiques utilisateur (GUI). Dans ce code, il est utilisé pour créer une fenêtre graphique, ajouter des widgets tels que des étiquettes et des boutons, et pour gérer les interactions avec l'utilisateur.

PIL (Python Imaging Library) : C'est une bibliothèque Python utilisée pour manipuler des images. Dans ce code, elle est utilisée pour afficher une image en arrière-plan de la fenêtre.

tkinter as tk : Cette ligne permet de renommer la bibliothèque tkinter en tk, ce qui est un raccourci pour écrire moins de code en utilisant tk à la place de tkinter pour les widgets.

Notre première bouton **home** affichera :



Ce code est une interface graphique réalisée en utilisant la bibliothèque tkinter de Python. L'interface graphique comporte plusieurs éléments, notamment:

NewsTicker : une classe pour afficher un ticker de nouvelles en haut de la fenêtre. Les nouvelles sont stockées dans une liste de dictionnaires contenant le titre et le chemin d'accès à l'image de chaque nouvelle.

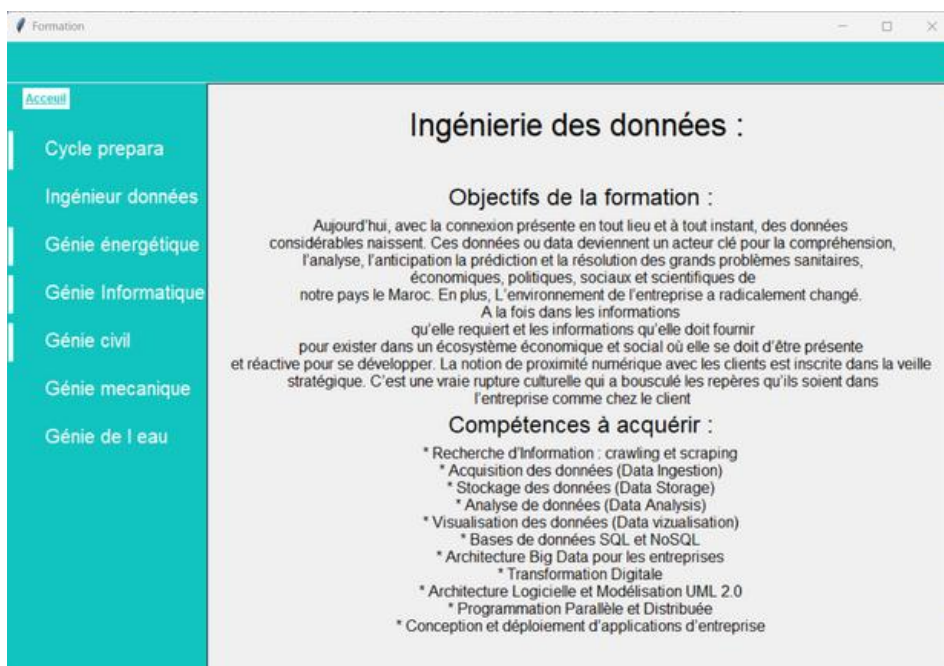
Des labels et des images sont utilisés pour afficher des informations sur l'école ENSA d'Al-Hoceima, telles que le nom de l'école, une brève description, une image de fond, etc.

Un bouton "Menu" pour rediriger l'utilisateur vers une autre page ("realacceuil" dans ce cas).

Une fonction login_page() qui sera appelée lorsque le bouton "Menu" est cliqué et qui détruit la fenêtre principale (root) et ouvre la nouvelle page "realacceuil".

Le code utilise également la bibliothèque Pillow pour manipuler et afficher des images. Les images sont stockées localement sur l'ordinateur de l'utilisateur.

Notre deuxième bouton **formation** affichera :



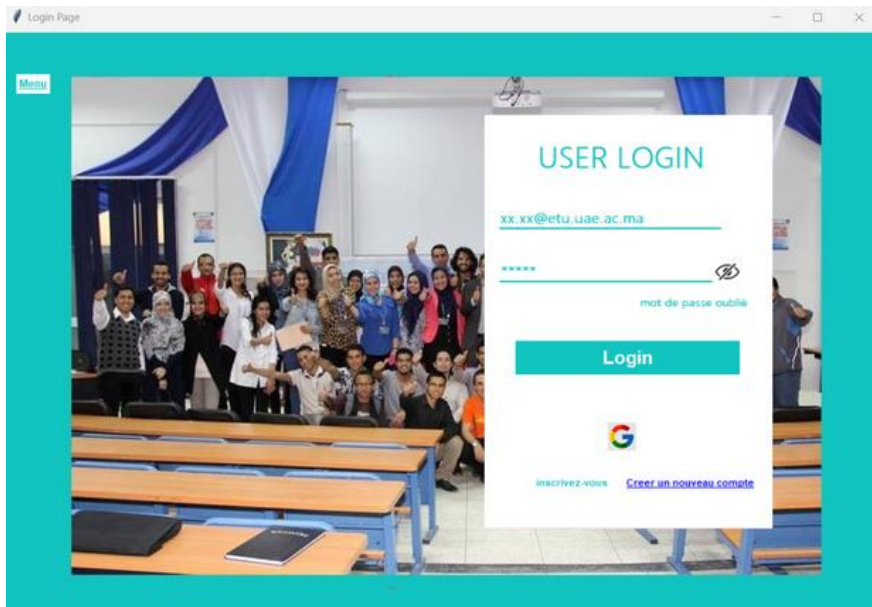
Ce code est écrit en Python et utilise le module tkinter pour créer une interface graphique. L'interface graphique contient une image de fond et plusieurs pages d'informations sur des formations spécifiques.

La fonction "login_page" permet de fermer l'interface actuelle et d'ouvrir une nouvelle interface appelée "realacceuil". Les fonctions suivantes, "cycle_preparatoire_page" et "genie_civil_page", "ingenierie de données", créent des cadres à l'intérieur de l'interface principale et ajoutent des étiquettes avec des informations sur les programmes de formation en cycle préparatoire et en génie civil, respectivement.

Le cadre créé par la fonction "cycle_preparatoire_page" contient des informations sur les objectifs de la formation en cycle préparatoire ainsi que les compétences à acquérir pendant la formation. Le cadre créé par la fonction "genie_civil_page" contient des informations sur les objectifs de la formation en génie civil, les débouchés et retombées de la formation, même chose pour les autres formations .

En fin de compte, l'interface graphique permet aux utilisateurs d'accéder aux informations sur différents programmes de formation.

Notre troisième bouton **Services en ligne** affichera :



Ce code utilise le module `tkinter` pour créer une interface graphique avec Python. Il permet de créer une page de connexion pour un site web en utilisant une base de données MySQL.

Le code commence par importer les modules nécessaires, notamment `tkinter`, `mysql.connector`, `PIL` et `messagebox`. Il contient plusieurs fonctions, chacune étant responsable d'une partie différente de la page de connexion.

La fonction `login_user()` est appelée lorsqu'un utilisateur clique sur le bouton de connexion. Elle vérifie que l'utilisateur a entré un nom d'utilisateur et un mot de passe,

puis elle essaie de se connecter à la base de données MySQL en utilisant les informations d'identification prédéfinies. Si la connexion réussit, la fonction vérifie si le nom d'utilisateur et le mot de passe correspondent à ceux stockés dans la base de données. Si les informations d'identification sont correctes, un message de bienvenue s'affiche. Sinon, un message d'erreur est affiché.

Les fonctions `login_page()`, `registr()`, `regirement_page()`, `signup_page()` sont appelées lorsque l'utilisateur clique sur les boutons correspondants et permettent de rediriger l'utilisateur vers la page de connexion, la page d'enregistrement, la page d'enregistrement ou la page d'inscription.

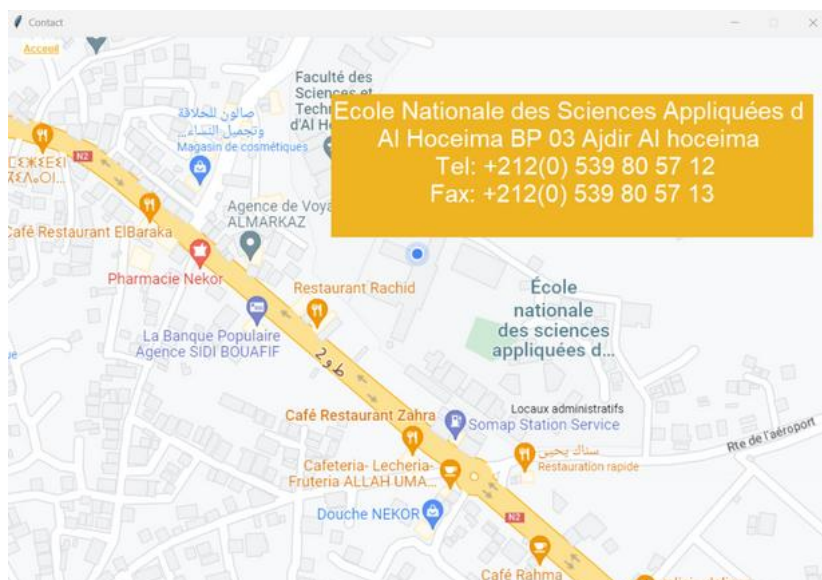
Les fonctions `on_entry()` et `pss_entry()` sont appelées lorsqu'un utilisateur clique sur les zones de texte correspondantes pour entrer son nom d'utilisateur et son mot de passe, respectivement. Ces fonctions effacent le texte "Nom d'utilisateur" ou "Mot de passe" des zones de texte pour faciliter l'entrée de l'utilisateur.

Les fonctions `hide()` et `show()` permettent de masquer ou d'afficher le mot de passe entré en cliquant sur un bouton avec une image d'un œil ouvert ou fermé.

La fonction `forget_pass()` est appelée lorsque l'utilisateur clique sur le bouton **"Mot de passe oublié ?"**. Elle affiche une nouvelle fenêtre dans laquelle l'utilisateur peut saisir son nom d'utilisateur et son nouveau mot de passe pour réinitialiser son mot de passe. La nouvelle information de mot de passe est stockée dans la base de données MySQL.

Un bouton **"Menu"** pour rediriger l'utilisateur vers une autre page ("realaccueil" dans ce cas).

Notre quatrième bouton **contact** affichera :



Le programme crée une fenêtre principale nommée "contact" avec une taille de 990x660 pixels et un arrière-plan de couleur #262626. Une image appelée "localisation.png" est chargée et affichée dans la fenêtre principale à l'aide d'un widget Label.

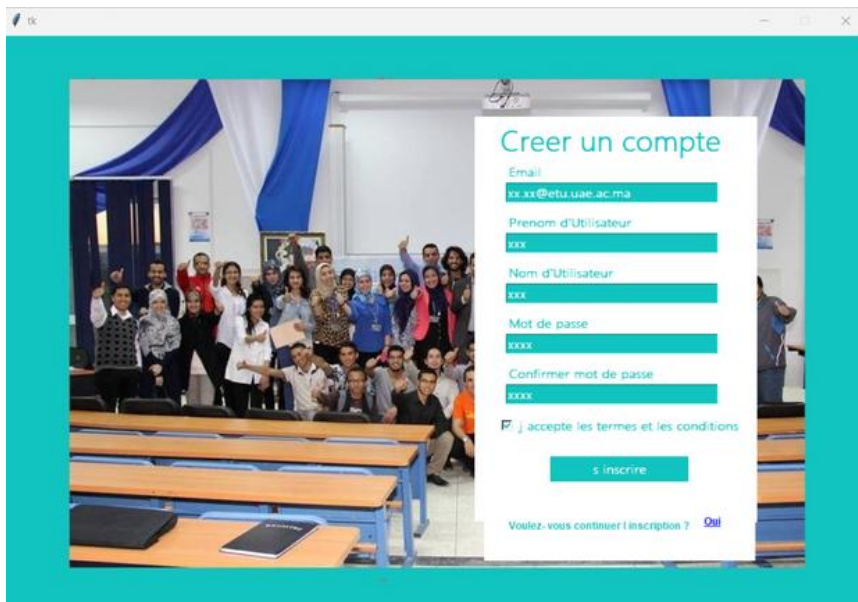
Un widget Label est également utilisé pour afficher du texte dans la fenêtre principale, notamment une adresse et des informations de contact pour l'école nationale des sciences appliquées d'Al Hoceima.

Un bouton **"Accueil"** est créé et placé dans la fenêtre principale à l'aide d'un autre widget Button. Ce bouton est associé à une fonction appelée "login_page", qui sera exécutée lorsque l'utilisateur cliquera sur le bouton.

La fonction "login_page" ferme la fenêtre "contact" et importe un autre fichier Python appelé "realaccueil". Enfin, la méthode "mainloop" est appelée sur l'objet "contact" pour afficher la fenêtre principale et attendre que l'utilisateur interagisse avec l'interface.

En revenant a notre troisième bouton **Services en ligne si le cas de l'étudiant n'est pas s'inscrit**

En cliquant le bouton **Créer un nouveau compte(pour s'inscrire):**



Ce code est une interface graphique réalisée à l'aide du module tkinter en Python. Il permet à l'utilisateur de s'inscrire sur un site web en entrant son adresse e-mail, son prénom, son nom et son mot de passe.

Le programme commence par importer les modules nécessaires, tels que tkinter pour l'interface graphique, PIL pour la gestion d'image et mysql.connector pour la communication avec une base de données MySQL.

La fonction `clear()` efface tous les champs de saisie de l'utilisateur, elle est appelée lorsque l'utilisateur appuie sur le bouton "Effacer".

La fonction `login_page()` permet à l'utilisateur de revenir à la page de connexion en détruisant la fenêtre d'inscription actuelle et en appelant le module "seconnecter".

La fonction `registrement_page()` permet à l'utilisateur de revenir à la page d'inscription en détruisant la fenêtre de connexion actuelle et en appelant le module "registration".

La fonction `connect_database()` est appelée lorsque l'utilisateur appuie sur le bouton "S'inscrire". Elle vérifie que tous les champs requis sont remplis, que les mots de passe entrés correspondent et que l'utilisateur a accepté les termes et conditions.

Ensuite, elle essaie de se connecter à la base de données MySQL en utilisant les informations de connexion prédéfinies. Si elle échoue, elle affiche un message d'erreur.

Elle crée ensuite une base de données appelée "userdata" si elle n'existe pas déjà et crée une table appelée "data" si elle n'existe pas déjà. Cette table contient les champs "id", "email", "first_name", "last_name" et "password".

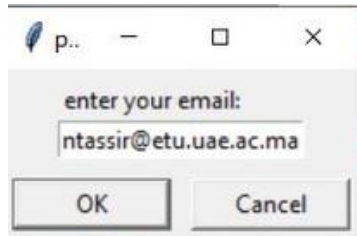
Ensuite, elle vérifie si l'utilisateur existe déjà dans la table en exécutant une requête SELECT sur la table avec l'adresse e-mail entrée par l'utilisateur. Si un utilisateur avec la même adresse e-mail est trouvé, un message d'erreur est affiché.

Sinon, elle insère les informations de l'utilisateur dans la table en utilisant une requête INSERT. Elle confirme ensuite l'inscription de l'utilisateur en affichant un message de succès et en effaçant tous les champs de saisie par une bouton appele **OUI**.

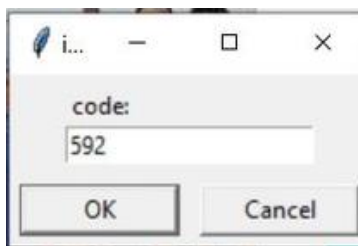
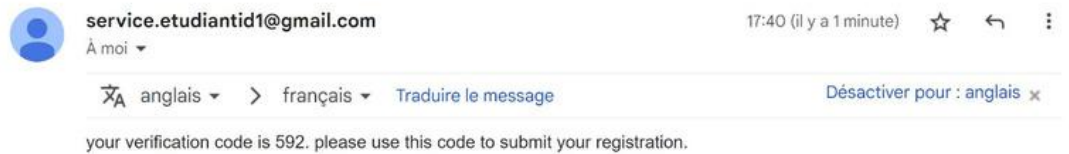
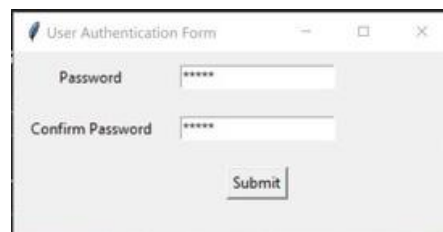
La fenêtre d'inscription est ensuite créée, elle contient une image de fond et un cadre blanc pour les champs de saisie et les étiquettes correspondantes. Chaque champ de saisie est précédé d'une étiquette pour indiquer le champ de saisie attendu.

Enfin, la fenêtre est affichée à l'utilisateur et attend ses interactions jusqu'à ce que l'utilisateur ferme la fenêtre.

Dans la page de connexion ,si l'utilisateur oublie son mot de passe il va cliquer sur 'mot de passe oublié' et il va s'afficher :

A small dialog box titled 'p..' with a close button. It contains the text 'enter your email:' and a text input field with the email 'ntassir@etu.uae.ac.ma'. Below the input field are two buttons: 'OK' and 'Cancel'.

un code de vérification va nous envoyer vers notre email :

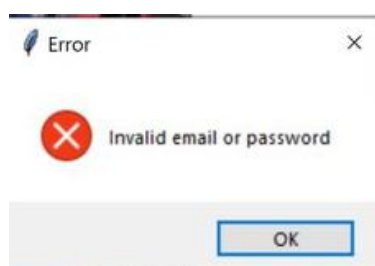
A small dialog box titled 'i...' with a close button. It contains the text 'code:' and a text input field with the code '592'. Below the input field are two buttons: 'OK' and 'Cancel'.A 'User Authentication Form' window. It has two input fields: 'Password' and 'Confirm Password', both masked with asterisks. Below these fields is a 'Submit' button.

En saisie le code

En remplir le nouveau mot de passe Et Bon connexion

A 'Welcome' dialog box with a close button. It features an information icon and the text 'Login is succesful'. At the bottom is an 'OK' button.

Et maintenant si l'utilisateur a saisie une email ou mot de passe incorrect . ce message d'erreur va s'afficher

An 'Error' dialog box with a close button. It displays a red 'X' icon and the text 'Invalid email or password'. At the bottom is an 'OK' button.

Notre code lié avec base de données:

Lorsqu'on saisit notre informations dans **SignUp** il va lier automatiquement dans notre base donnée

```
MariaDB [userdata]> select * from data;
```

id	email	first_name	last_name	password
1	salma.fannich@etu.uae.ac.ma	salma	fannich	12345
2	chadi.mountassir@etu.uae.ac.ma	chadi	mountassir	3456
3	chaymae.mehrioui@etu.uae.ac.ma	chaymae	mehrioui	4567
4	salma.kellali@etu.uae.ac.ma	salma	kellali	5678

```
4 rows in set (0.000 sec)

MariaDB [userdata]>
```

Le code est écrit en Python et utilise les modules Tkinter, io, datetime, filedialog, **simplifiedialog, messagebox, PIL, os, Combobox, openpyxl et mysql.connector.** Il définit une interface graphique pour l'enregistrement d'étudiants.

Le programme commence par importer les modules nécessaires et créer une fenêtre avec le titre "**Registrement d'Etudiant**", une taille de 1250x700 pixels et une position initiale de (210, 100) sur l'écran. La fenêtre utilise un arrière-plan bleu foncé (#06283D).

Ensuite, différentes fonctions sont définies pour gérer les actions de l'utilisateur dans l'interface graphique. La fonction Exit() ferme la fenêtre. La fonction registrement_page() ferme la fenêtre actuelle et ouvre une nouvelle fenêtre pour l'enregistrement d'un nouvel étudiant. La fonction registr() ferme la fenêtre actuelle et ouvre une nouvelle fenêtre pour se connecter. La fonction get_id_tab() ouvre une nouvelle fenêtre pour permettre à l'utilisateur de récupérer son ID en cas d'oubli. La fonction showimage() permet à l'utilisateur de sélectionner une image pour l'afficher dans l'interface graphique. La fonction registration_no() génère un numéro d'enregistrement unique pour chaque étudiant. La fonction clear() réinitialise tous les champs de la fenêtre. La fonction verif(num) permet de vérifier que l'utilisateur entre le bon code avant de pouvoir sauvegarder ses modifications. La fonction save() enregistre les données saisies par l'utilisateur dans un fichier Excel et dans une base de données MySQL.

La fenêtre contient différents widgets, tels que des libellés (Labels), des entrées (Entry), des boutons (Button), des images (Image) et des listes déroulantes (Combobox), pour permettre à l'utilisateur d'entrer des informations sur l'étudiant à enregistrer. Le programme utilise également une grille pour organiser les différents widgets dans la fenêtre.

La fonction update() met à jour les informations de l'étudiant dans la base de données MySQL. Les informations sont récupérées depuis les champs d'entrée de l'interface graphique et stockées dans des variables. Les données sont ensuite insérées dans la base de données à l'aide de la bibliothèque mysql-connector-python.

La fonction selection() est appelée pour récupérer le genre de l'étudiant à partir d'un bouton radio sélectionné dans l'interface graphique.

La fenêtre principale de l'interface graphique contient un en-tête, une zone de recherche et un bouton de mise à jour.

La zone de recherche permet de rechercher un étudiant en entrant son numéro d'inscription.

Les champs d'entrée de l'interface graphique permettent de saisir les détails de l'étudiant tels que son nom, sa date de naissance, son genre, sa classe, son baccalauréat, sa nationalité, le nom de son père, le travail de son père, le nom de sa mère et le travail de sa mère.

La fonction registration_no() est appelée pour générer un numéro d'inscription unique pour chaque nouvel étudiant inscrit.

Si l'utilisateur veut modifier ses propres informations , il faut d'abord saisir son code pour pouvoir accéder a ses données

Notre code lié avec base de données:

Lorsqu'on saisit notre informations dans **Registration** il va lier automatiquement dans notre base donnée

id	email	first_name	last_name	password	date_naissance	genre	class	nationalite	bacaloret	nom_pere	travaille_pere	nom_mere	travaille_mere	image
1	chadi.mountassir@etu.uae.ac.ma	Chadi	Mountassir	chadi	2002-07-26	Male	i.d	Marocain	Bac Marocain	Hamid	enseignant	Chaibia	femme au foyer	[BLOB - 147.2 KiB]
4	salma.kellali@etu.uae.ac.ma	salma	Kellali	vgHBX1f@42	2002-09-04	Female	i.d	Nerlandais	Bac Marocain	Abdel Rahim	gangster	Zahra	architecte	[BLOB - 147.2 KiB]

Lorsqu' on termine notre registration on retourne à la page de connexion pour accéder à **Myeservice**

Et voici notre

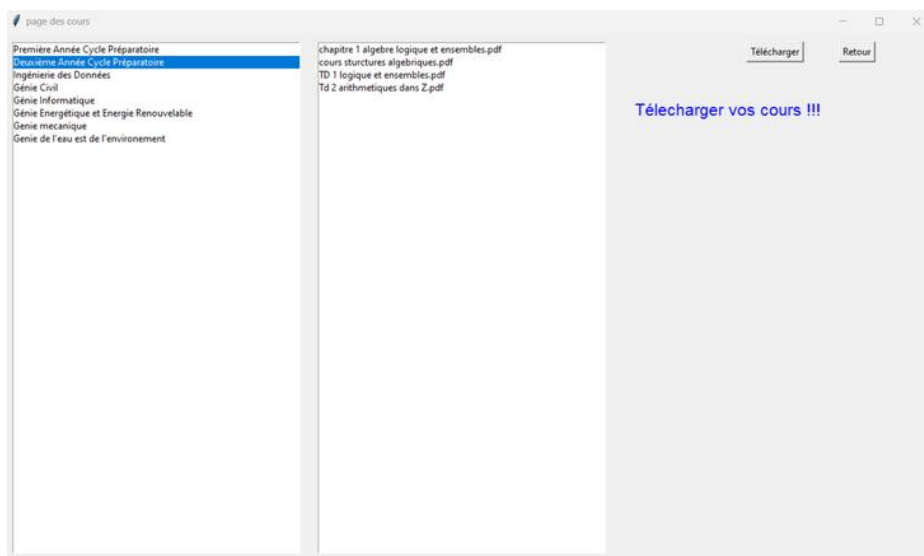
Eservice :



L'interface graphique contient un fond d'écran et des boutons pour naviguer vers différentes pages. Les boutons sont stylisés avec une couleur de fond et de texte différente ainsi que des effets de survol de la souris. Il y a également un bouton de menu qui ouvre un menu latéral avec des boutons supplémentaires pour naviguer vers les différentes pages.

Le programme est conçu pour un système de **gestion des étudiants**, avec des pages pour les cours, les cartes étudiantes, les résultats, les clubs, etc. Le code utilise également la **bibliothèque PIL (Pillow)** pour manipuler des images, telles que les icônes et les images de fond d'écran.

Notre première bouton **cours** affichera :



Le code commence par définir un dictionnaire nommé **FILIERES_DOSSIERS** qui contient les noms des filières comme clés et les chemins des dossiers contenant les cours pour chaque filière comme valeurs.

Ensuite, une fonction nommée **"animate_text"** est définie pour animer le texte dans un canevas. La fonction **"download_course"** est également définie pour télécharger un cours.

Ensuite, une instance de la classe Tk est créée en appelant le constructeur `tk.Tk()`. Le titre de la fenêtre est défini avec la méthode **"title()"**.

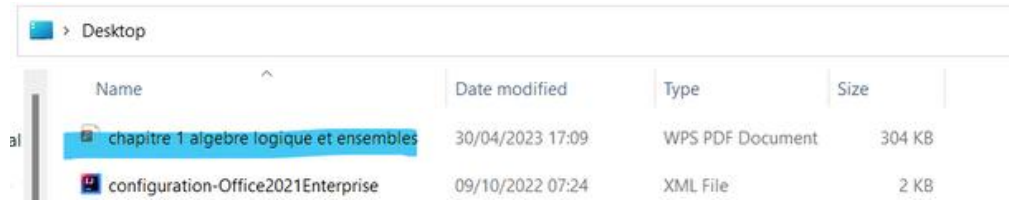
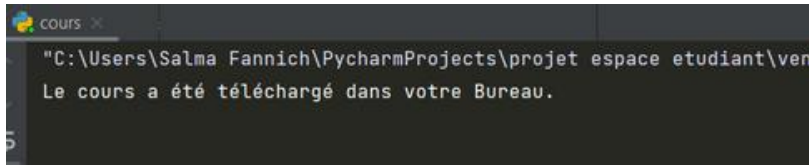
Une **listebox** est créée pour afficher les filières disponibles et une autre listebox est créée pour afficher les cours disponibles pour la filière sélectionnée. Les listes sont créées à l'aide de la méthode **"Listbox()"**. Les filières sont ajoutées à la listebox en utilisant la méthode **"insert()"** de la listebox.

Lorsqu'une filière est sélectionnée dans la listebox des filières, la listebox des cours est mise à jour pour afficher les cours disponibles pour cette filière. Cette mise à jour est réalisée grâce à la fonction **"update_courses"**, qui est appelée lorsque l'événement **"ListboxSelect"** est déclenché.

Un bouton nommé **"Télécharger"** est créé pour télécharger le cours sélectionné. Lorsque ce bouton est cliqué, la fonction **"download_course"** est appelée.

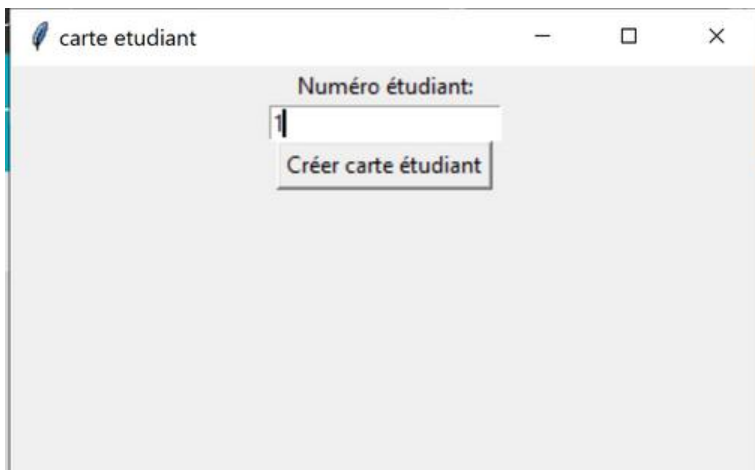
Enfin, un canevas est créé pour afficher le texte animé, et la fonction **"animate_text"** est appelée pour animer le texte dans le canevas.

Lorsqu'on clique sur télécharger , le cour va telecharger sur le bureau de notre utilisateur.



Notre deuxième bouton **carteEtudiant** affichera :

lorsque l'utilisateur saisie le "**Numero d'etudiant**" , une carte etudiant va creer automatiquement en liant sa code avec base de donnée



La fonction **creer_carte_etudiant()** crée une nouvelle fenêtre Tkinter en utilisant la méthode **Toplevel()**. Elle crée également un canevas en utilisant la classe **Canvas()** de Tkinter pour dessiner la carte étudiant. La méthode **create_rectangle()** de la classe **Canvas()** est utilisée pour dessiner un rectangle blanc pour la carte. L'image de l'étudiant est ajoutée à la carte étudiant en utilisant la méthode **create_image()** de la classe **Canvas()**. Les labels avec le nom, prénom et date de naissance de l'étudiant sont ajoutés à la carte en utilisant la classe **Label()** de Tkinter.

Enfin, l'interface graphique est créée en utilisant la classe **Tk()** de Tkinter. Un champ de saisie pour le numéro d'étudiant et un bouton pour créer la carte étudiant sont ajoutés à l'interface. Lorsque l'utilisateur clique sur le bouton, la fonction **creer_carte_etudiant()** est appelée pour afficher la carte étudiant.

Remarque : Le chemin d'accès à l'image de l'étudiant est spécifié en utilisant la ligne de code `etudiant_image = tk.PhotoImage(file='t1.png')`. Si l'image se trouve dans un répertoire différent, le chemin d'accès doit être modifié en conséquence.

Notre troisième bouton **marksheet** affichera :

The screenshot shows a Tkinter window titled 'Marksheet Generator'. The main window has a purple header with the 'Marksheet' logo and the text 'Create your own Marksheet'. Below the header, there are two sections: 'Student's Information' and 'Subjects Information'. The 'Student's Information' section contains four input fields: 'Name of the Student', 'Date of Birth', 'Programme/Class', and 'Registration Number'. The 'Subjects Information' section contains five input fields for 'Subject I' through 'Subject V', each with a corresponding 'Marks Obtained (Out of 20)' field. At the bottom, there is a summary row showing 'Total: 0', 'Percentage (%): 0', 'Grade: XXXX', and 'Result: XXXX'. Below this, there are four buttons: 'Display Result' (green), 'Generate Marksheet' (blue), 'Reset Entries' (yellow), and 'Exit' (red).

Ce code est un programme de calcul de notes et de résultats scolaires en utilisant la bibliothèque Tkinter de Python. Le programme contient des fonctions pour calculer la somme, le pourcentage, la note et le résultat en fonction des notes des matières entrées par l'utilisateur. Le programme comporte également des fonctions pour vérifier les erreurs et afficher le résultat dans une fenêtre Tkinter.

Le programme commence par importer les modules et les widgets nécessaires. Ensuite, il définit une fonction pour passer à la page d'accueil et des fonctions pour calculer la somme, le pourcentage, la note et le résultat. Il y a également une fonction pour vérifier les erreurs et une fonction pour afficher le résultat.

Lorsque l'utilisateur entre les noms des matières et leurs notes, le programme utilise la fonction de vérification des erreurs pour s'assurer que toutes les

entrées sont valides. Si elles le sont, le programme utilise les fonctions de calcul pour déterminer la somme, le pourcentage, la note et le résultat et les affiche dans une fenêtre Tkinter.

Si les notes ne sont pas valides, une boîte de dialogue d'erreur est affichée et les entrées sont réinitialisées. Si les champs de matière sont vides, une boîte de dialogue d'erreur est affichée et les champs sont réinitialisés. Si le résultat est un échec, la note est définie sur "F". Le bouton "Générer le bulletin de notes" est également désactivé après l'affichage du résultat pour empêcher les modifications.

Lorsque l'utilisateur entre ses propres notes , le programme va afficher :

The screenshot shows the results of a student's performance. The student's information is displayed as follows:

Institut: ensah
Registration No: 1
Roll No : chaymae merhrioui
Student Name: 06/07/2002
Date of Birth: id1

Subject List	Marks
python	19
datamining	20
java	12
sql	14
statistiques	13

Grand Total : 78.0
POURCENTAGE : 15.6
Grade : Mention Bien
Result : PASS

Notre code lié avec base de données:

Lorsqu'on saisit notre notes dans **Marksheet** il va lier automatiquement dans notre base de données avec les notes et id

note1	note2	note3	note4	note5	data_id
12	2	3	18	15	1

Notre quatrième bouton **Clubs** affichera :

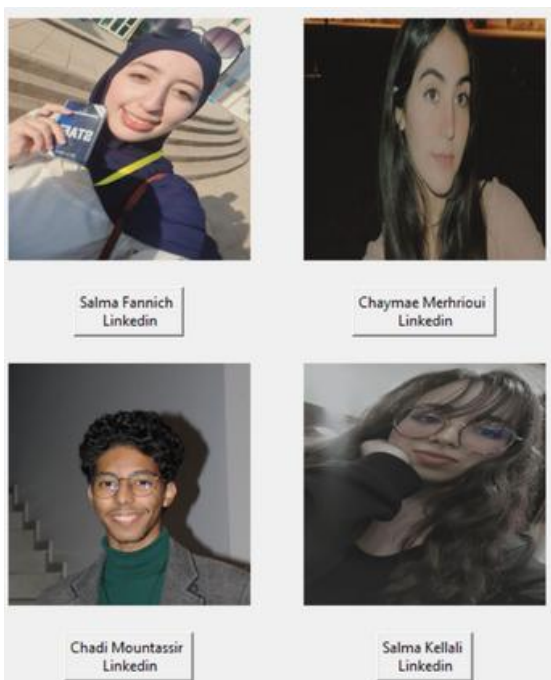


Ce code est un script Python qui utilise la bibliothèque Tkinter pour créer une interface utilisateur graphique (GUI) pour un programme d'activités parascolaires. Le programme compte quatre clubs : le Club Data, le Club Génie Civil et le Club CCT, Club 01. Chaque club a sa propre page avec ses objectifs et ses événements.

Nous avons créé des fonctions pour la page de chaque club et nous avons utilisé la méthode **pack()** pour ajouter des étiquettes à chaque cadre. Et nous avons également défini une fonction **login_page()** qui détruit la fenêtre courante et importe un autre module nommé **eservicehome**.

Dans l'ensemble, notre code semble être le début d'un joli programme d'interface graphique pour un programme d'activités parascolaires. Cependant, nous pouvons envisager de créer un menu principal ou une page d'accueil qui renvoie à la page de chaque club au lieu d'afficher chaque page dès que le programme démarre. De plus, nous souhaitons d'inclure plus de fonctionnalités, telles que permettre aux utilisateurs de s'inscrire à un club ou de répondre à des événements.

Notre cinquième bouton **Contacter nous** affichera :



Ce code affiche une fenêtre principale, des labels et des boutons pour afficher notre images et notre liens vers nos profils LinkedIn d'pour nous contacter au cas des problèmes .

Le code utilise également le module **webbrowser** pour ouvrir les liens LinkedIn dans un navigateur Web lorsque les boutons sont cliqués.

Pour chaque bouton, le code utilise la fonction **lambda** pour définir une fonction anonyme qui ouvre le lien dans un nouvel onglet du navigateur. Les liens sont passés comme arguments à la fonction **open_new_tab()** du module **webbrowser**.

Fin

```
}  
render() {  
  return (  
    <React.Fragment>  
      <div className="py-4">  
        <div className="py-2">  
          <Title name="Product List" />  
          <div class="row">  
            <ProductCard />  
            {products.map((product) => <ProductCard key={product.id} />)}  
          </div>  
        </div>  
      </div>  
    </React.Fragment>  
  )  
}
```