

Documentazione del Progresso

Preambolo:

Intraprendendo un progetto simile in Python dopo aver già lavorato su una soluzione analoga in Java, ho beneficiato di una base di conoscenze pregresse. Tuttavia, mi sono reso conto che, a differenza di JavaFX, in Python è necessario costruire l'applicazione da zero.

Primo Passo:

Il primo obiettivo era individuare una libreria Python adatta alla creazione di applicazioni. Dopo una valutazione, ho scelto Tkinter. Si tratta di una libreria standard che offre un set completo di strumenti per la creazione di interfacce grafiche utente (GUI), inclusi finestre, bottoni, menu, caselle di testo, etichette e altri elementi interattivi.

Secondo Passo:

Per comprendere il funzionamento di Tkinter, è fondamentale capire che essa crea un loop che si esegue ciclicamente a intervalli regolari per garantire una reattività all'interfaccia. Pertanto, il primo passo consiste nella creazione della finestra principale e nella configurazione delle sue dimensioni e altre proprietà. Gli elementi della GUI interagiscono attraverso i cosiddetti Trace, ossia delle funzioni che monitorano e gestiscono i cambiamenti eseguiti dall'utente nell'interfaccia.

Terzo Passo:

Per permettere all'applicazione di interagire con un database, è necessario utilizzare una libreria adatta. Nel

mio caso, ho scelto mysql.connector. Una volta ottenuti gli elementi necessari per stabilire la connessione con il database, è opportuno creare un file di configurazione (.env) contenente le informazioni sensibili, come credenziali di accesso al database.

Quarto Passo:

Al fine di semplificare l'utilizzo dell'applicazione da parte degli utenti, ho implementato una funzione per gestire il processo di accesso tramite un prompt di login. Questo permette di comunicare eventuali errori o problemi di autenticazione all'utente in modo chiaro e immediato.

Quinto Passo:

Durante l'interazione dell'utente con l'interfaccia grafica utente (GUI), vengono eseguite delle query che sono gestite da una funzione apposita. I risultati di queste query devono essere salvati in un documento specifico. Per gestire questa operazione, ho integrato la libreria Docx nel progetto.

Nei punti in cui è necessario sostituire del testo dinamicamente all'interno del documento, ho utilizzato dei placeholder. Tramite l'utilizzo della funzione "Trova e Sostituisci", è possibile sostituire automaticamente questi placeholder con i dati appropriati.

Spiegazione Librerie:

import os

Questa libreria fornisce funzioni per interagire con il sistema operativo. È utilizzata per eseguire comandi nel prompt dei comandi (o nella shell) e per gestire file, directory e variabili di ambiente.

import sys

Questa libreria fornisce accesso a alcune variabili e funzioni specifiche del sistema Python. È utilizzata per interagire con le proprietà dei file e i percorsi di sistema. Tuttavia, l'uso specifico menzionato, ovvero comunicare la grandezza del file, sarà rimosso dopo la fase di produzione.

from dotenv import load_dotenv

Questa libreria è utilizzata per leggere le variabili d'ambiente da file di configurazione .env. È comunemente usata per gestire dati sensibili come le informazioni di accesso al database, permettendo di mantenere tali informazioni al sicuro e separate dal codice sorgente.

import tkinter as tk

Questa è una libreria per la creazione di interfacce grafiche (GUI) in Python. È utilizzata per creare finestre, pulsanti, caselle di testo e altri elementi di un'interfaccia utente.

from tkinter import ttk

Questa libreria fornisce widget aggiuntivi e stili di widget per l'interfaccia utente di Tkinter. È utile per

sostituire le funzioni obsolete di Tkinter e migliorare l'aspetto delle GUI.

from docx import Document

Questa libreria permette di creare, modificare e leggere file di documento di Microsoft Word (.docx). È utile per automatizzare la generazione di documenti, ad esempio report o lettere.

import shutil

Questa libreria fornisce funzioni ad alto livello per operazioni di copia e manipolazione di file e directory. È utile per copiare, spostare o eliminare file e directory da Python.

import mysql.connector

Questa libreria fornisce un'interfaccia Python per comunicare con database MySQL. È utilizzata per eseguire query, inserire dati, aggiornare dati e altro ancora nel database MySQL.

from datetime import date

Questa libreria fornisce funzionalità per lavorare con date e orari. È utilizzata per ottenere la data corrente dal sistema e può essere utile per registrare o manipolare date nei documenti o nel database.