**Декларация кода**

Этот код представляет собой приложение на Python, использующее библиотеку customtkinter для создания графического пользовательского интерфейса (GUI) и mysql.connector для взаимодействия с базой данных MySQL. Приложение предназначено для управления пользователями, товарами, заказами и адресами в контексте некой системы управления.

**Основные компоненты кода:**

1. **Настройки подключения к базе данных:**
   * db\_config: Словарь, содержащий параметры подключения к базе данных, такие как имя пользователя, пароль, хост и имя базы данных.
2. **Данные для входа:**
   * user\_data: Словарь, содержащий данные пользователей (логин, пароль и роль) для аутентификации.
3. **Функция login\_window():**
   * Создает окно входа в систему, где пользователи могут ввести свои логин и пароль.
   * Проверяет введенные данные и, в зависимости от роли пользователя, открывает соответствующее окно (управление пользователями, просмотр для обычного пользователя).
4. **Функция users\_window(role):**
   * Создает окно управления пользователями.
   * Содержит функции для получения списка пользователей из базы данных, добавления, редактирования и удаления пользователей.
   * Использует ttk.Treeview для отображения информации о пользователях в виде таблицы.
5. **Функция products\_window():**
   * Создает окно управления товарами.
   * Содержит функции для получения, добавления, редактирования и удаления товаров.
6. **Функция orders\_window():**
   * Создает окно управления заказами.
   * Содержит функции для получения, добавления и удаления заказов.
7. **Функция addresses\_window():**
   * Создает окно управления адресами.
   * Содержит функции для получения, добавления, редактирования и удаления адресов.
8. **Запуск приложения:**
   * В конце кода вызывается функция login\_window(), которая запускает приложение.

**Примечания:**

* Код использует try-except блоки для обработки ошибок, связанных с базой данных, что позволяет пользователю получать уведомления об ошибках.
* Интерфейс приложения выполнен в темной теме с использованием библиотеки customtkinter, что делает его более современным и удобным для восприятия.
* Каждое окно имеет возможность навигации к другим функциям приложения, что делает его многофункциональным.

**Возможные улучшения:**

* Добавление более сложной логики аутентификации и безопасности (например, хеширование паролей).
* Реализация системы ролей и прав доступа для более гибкого управления пользователями.
* Улучшение пользовательского интерфейса (UI) с использованием дополнительных элементов управления и стилей.
* Оптимизация запросов к базе данных для повышения производительности.