

Менеджер паролей

# ИДЕЯ

Сделать менеджер паролей, который позволит безопасно хранить их и также легко получать к ним доступ, по типу KeePass.



# СТЕК

Для интерфейса был выбран распространённый инструмент для создания приложений для Windows – PyQt6.

Для хранения данных используется база данных SQLite.

Для многопользовательского доступа и аутентификации был использован API Telegram для отправки одноразового кода.

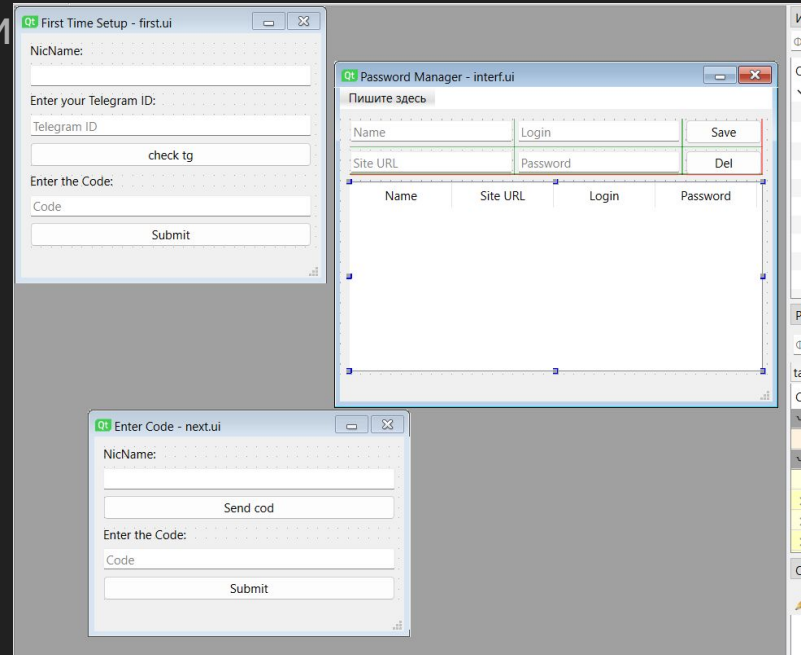


6



# вид приложения

В дизайне предусмотрено три окна: два из них являются авторизационными, а третье — основным. Все окна были выполнены в Qt6 Designer с использованием QPushButton, QLineEdit и



# регистрация и 2FA

Для регистрации в приложении необходимо знать свой chat\_id в Telegram (его можно получить через бота @ShowJsonBot).

Отправка кода реализована отдельной функцией с использованием модуля requests. Это позволяет использовать приложение на разных устройствах и избегать ошибок подключения к серверу, которые могли бы возникнуть при использовании Telebot.

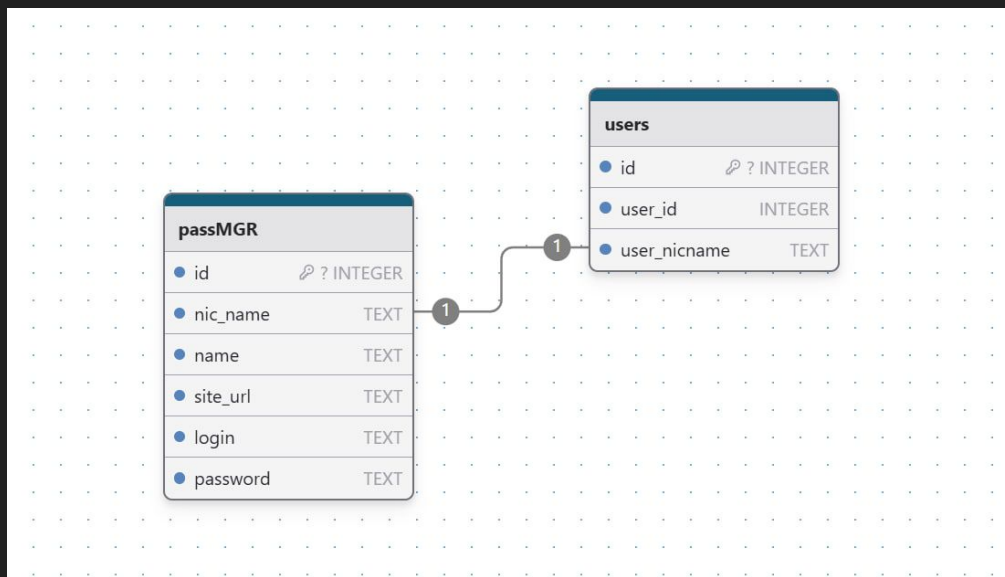
```
def gev_auth_cod():
    cod = ''.join(str(random.randint(a: 1, b: 9)) for _ in range(5))
    return cod

3 usages  ▲ dItemnd
def sand_code(user_id: int, login_code: str):
    response = requests.post(
        url=f"https://api.telegram.org/bot{os.getenv('TOKEN')}/sendMessage",
        params={
            "chat_id": user_id,
            "text": f"Ваш пароль для входа в приложение: ||{login_code} ||",
            "parse_mode": "MarkdownV2"
        }
    )

    print(response.status_code)
    return response.status_code
```

# структура базы данных

База данных состоит из двух таблиц: users и passMGR. В таблице users хранятся все зарегистрированные пользователи, а в таблице passMGR — пароли пользователей. Эти таблицы связаны по полю user\_nickname.



# проверка правильности заполнения полей

Проверка реализована для полей site\_url и password. Для поля site\_url используется строгая проверка с помощью регулярного выражения, и если оно не совпадёт, пользователь не будет добавлен в таблицу. Проверка для

```
pattern_password = r'^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&#])[A-Za-z\d@$!%*?&#]{8,}$'  
pattern_url = r'^(https?:\/\/)?([a-zA-Z0-9-]+\.)+[a-zA-Z]{2,}(:\d+)?(\/[^\s]*)?$'
```

