### Руководство программиста

### Telegram-бот для шведского стола

### 1. Установка и запуск программы

### 1.1. Подготовка окружения

### Скачайте файлы программы из репозитория [TGbot\_SHVED\_STOL](https://github.com/MonsiQ/TGbot_SHVED_STOL.git" \t "_new).

### Убедитесь, что на вашем компьютере установлен Python версии 3.x и все необходимые библиотеки:

### python-dotenv==1.0.1

### aiogram==3.14.0

### psycopg2-binary==2.9.10

### requests==2.32.3

### pillow==11.0.0

### pytz==2024.2

### timezonefinder==6.5.4

### Для установки зависимостей, используйте команду:

### pip install -r requirements.txt

### 1.2. Подключение базы данных

### База данных в проекте хранится в локальных файлах, подключение к PostgreSQL не требуется.

### 1.3. Конфигурация

### Получите токен для вашего Telegram-бота у [BotFather](https://t.me/BotFather" \t "_new).

### Вставьте токен в env файл:

### BOT\_TOKEN=Ваш токен

### 1.4. Запуск

### Запустите бота с помощью команды:

### python main.py

### Для взаимодействия с ботом в Telegram используйте команду /start.

### 2. Проверка работоспособности

### Убедитесь, что база данных успешно подключена и программа взаимодействует с ней.

### Проверьте корректность работы бота, отправку заявок на авторизацию и заказ блюд через админ-панель.

### 3. Поддержка программы

### 3.1. Обновления

### Регулярно обновляйте Python, а также библиотеки telebot, psycopg2, xlwt, datetime до актуальных версий.

### 3.2. Отслеживание ошибок

### Проводите мониторинг ошибок от пользователей и своевременно исправляйте их.

### 3.3. Уведомление пользователей

### При добавлении новых функций или исправлении ошибок, уведомляйте пользователей через обновления в репозитории GitHub.

### 3.4. Документация

### Обновляйте документацию программы, включая инструкции по устранению проблем, FAQ и храните её в репозитории.

### 3.5. Резервное копирование

### Регулярно создавайте резервные копии базы данных и данных пользователей.

### 4. Модернизация программы

### 4.1. Сбор обратной связи

### Анализируйте отзывы пользователей для улучшения программы.

### 4.2. Создание новой ветки для разработки

### При разработке новых функций создавайте новую ветку с помощью команды:

### git checkout -b feature-branch

### 4.3. Разработка функций

### Используйте Python и библиотеки для реализации новых функций. Обеспечьте тестирование кода.

### 4.4. Тестирование

### Протестируйте новые функции на наличие ошибок, прежде чем выпустить их.

### 4.5. Обновление документации

### После добавления новых функций обновите документацию, добавив описание изменений.

### 4.6. Отправка изменений в репозиторий

### После тестирования и подготовки изменений отправьте их в репозиторий:

### git push origin feature-branch