Логика работы:

1. Создать базу данных **domclick**(либо через консоль postgresql, либо через pgadmin)
2. В консоли python выполнить **from register import db**, затем **db.create\_all()**

Будет создана таблица Register(id, name, username, otd, password\_hash)

1. Запустить приложение **register.py –** создает пользователь admin **,** пароль 111 (поля )
2. В консоли python выполнить **from app import db**, затем **db.create\_all()**

Буду созданы все рабочие таблицы приложения:

аplicat (полная информация о заявки)

department (отделы)

register (регистрация пользователей)

statuses (статус заявки)

tips (тип заявки)

1. Далее следует заполнить разделы

Регистрация пользователя системы (можно )\*

Добавление статуса заявки (В работе, Закрыта, Открыта)\*

Добавление типа заявки (Ремонт, Обслуживание, Консультация)\*

\*-можно провести запись в данные таблицы при их создании, чтобы не было ошибок в фильтрах, но в академических целях сделан ручной ввод.

Добавление новых отделов (Отдел ремонта, отдел обслуживания, консультативный отдел)

Все заявки: В данном раздели работники могут выбрать заявку, и/или изменить ее статус. При изменении статуса автоматически записывается текущий пользователь, который внес изменения статуса заявки.

В данном разделе можно просмотреть **реализацию фильтров** по статусам заявок (В работе, Закрыта, Открыта) \*

\*- фильтр жестко привязан к регистру и написанию при заведении статуса в таблице statuses

Реализация дополнительных фильтров возможна через app.py, как это сделано для статусов заявок с созданием шаблона html, или с выводом на любую удобную форму

1. Для примера реализована возможность редактирования (удаление/добавление) нескольких таблиц (statuses, tips и т.д.)
2. Все формы, за исключением создания заявки закрыт для незарегистрированных пользователей