# **Техническое задание**

# **«Разработка CRM-системы»**

Разработайте CRM-систему в виде веб-приложения для управления клиентами.

Основные функции приложения:

* авторизация пользователя;
* создание, редактирование и просмотр предоставляемых услуг;
* создание, редактирование и просмотр рекламной кампании;
* создание, редактирование и просмотр потенциальных клиентов;
* создание, редактирование и просмотр контракта для клиента;
* перевод потенциального клиента в активного;
* подсчёт и отображение статистики по рекламным кампаниям: сколько привлечено потенциальных клиентов, сколько из них перешло в активных.

## Страницы для работы с услугами

* Страница отображения списка активных клиентов: у каждой записи должен быть уникальный идентификатор, например название. Записи — ссылки на переход к детальной странице. Рядом с ними нужно добавить кнопки для удаления, а на странице — общую кнопку для создания новой записи.
* На детальной странице должна быть неизменяемая форма с данными о записи, а также кнопка для редактирования и удаления записи.
* Страница редактирования должна быть предзаполненной и доступной для редактирования.
* На странице создания записи должна быть пустая форма для заполнения.

## Страницы для работы с рекламными кампаниями

* Аналогично страницам для работы с услугами.

## Страницы для работы с потенциальными клиентами

* Аналогично страницам для работы с услугами.
* Рядом с каждой записью на странице списка должна быть ссылка для перевода потенциального клиента в активного. Так пользователь переходит на страницу создания активного клиента.

## Страницы для работы с контрактами

* Аналогично страницам для работы с услугами.

## Страницы для работы с активными клиентами

* Страница отображения списка активных клиентов: у каждой записи должен быть уникальный идентификатор, например название. Записи — ссылки на переход к детальной странице. Рядом с ними нужно добавить кнопки для удаления.
* На детальной странице должна быть неизменяемая форма с данными о записи, а также кнопка для редактирования и удаления записи.
* Страница редактирования должна быть предзаполненной и доступной для редактирования.
* На странице создания записи поле о потенциальном клиенте должно быть предзаполненным, а остальные — пустыми.

## Роли пользователей

В системе должны быть преднастроенные роли пользователей.

* Администратор может создавать, просматривать и редактировать пользователей, назначать им роли и разрешения. Такой функционал реализует административная панель Django.
* Оператор может создавать, просматривать и редактировать потенциальных клиентов.
* Маркетолог может создавать, просматривать и редактировать предоставляемые услуги и рекламные кампании.
* Менеджер может создавать, просматривать и редактировать контракты, смотреть потенциальных клиентов и переводить их в активных.
* Все роли могут смотреть статистику рекламных кампаний.

## Создание услуги

Приложение принимает информацию об услуге:

* название;
* описание;
* стоимость.

## Создание рекламной кампании

Приложение принимает:

* название рекламной кампании;
* рекламируемую услугу;
* канал продвижения;
* бюджет на рекламу.

## Создание потенциального клиента

Приложение принимает данные о клиенте:

* Ф. И. О.;
* телефон;
* email;
* рекламную кампанию, из которой он узнал об услуге.

## Создание контракта

Приложение принимает сведения о контракте:

* название;
* предоставляемую услугу;
* файл с документом;
* дату заключения;
* период действия;
* сумму.

## Создание активного клиента

Активного клиента можно создать только из списка потенциальных.

Приложение принимает:

* данные о потенциальном клиенте;
* данные о контракте.

## Подсчёт статистики об успешности рекламных кампаний

Статистику считают по нескольким критериям:

* число клиентов, привлечённых рекламной кампанией;
* число клиентов, перешедших из потенциальных в активных;
* соотношение дохода от контрактов и расходов на рекламу.

## Зависимость объектов



Изображение: Skillbox

## Требования к коду

* Пользователей и роли нужно создавать с помощью стандартного административного интерфейса Django.
* Все остальные функции CRM-системы должны быть доступны вне административного интерфейса Django.
* Нужно использовать PostgreSQL как систему управления базами данных (СУБД). О том, как установить её, вы можете узнать в инструкциях: [установка на Windows](https://postgrespro.ru/windows), [установка на Linux](https://postgrespro.ru/docs/postgrespro/10/binary-installation-on-linux) и [настройка Django](https://django.fun/docs/django/4.2/ref/databases/#postgresql-notes).
* Необходимо подключить Pylint, чтобы автоматически проверять код на соответствие стилевому оформлению. Чтобы разобраться в этом подробнее, прочитайте [инструкцию по использованию для Pylint](https://docs.google.com/document/d/17Ix5Snheubk645CVF2J737wH7kC2ZPNe1Rit3UXHXsg/edit#heading=h.qw8atkej5zqt).

## Дополнительные требования (по желанию)

Вы можете существенно улучшить код:

* использовать аннотации типов и настроить автоматизированную проверку с помощью mypy;
* добавить unit-тесты с применением библиотеки Pytest.