**Задание**

Курсовая работа состоит из двух частей. Первая часть, «Разработка сервиса», вам открылась в конце курса «Основы Django». Если вы уже выполнили эту часть, то сразу приступайте ко второй — «Расширение функционала».

Если вы еще не начинали работу над курсовым проектом, тогда выполняйте первую часть, а затем переходите ко второй.

Удачи!

В первой части проекта нужно создать структуру, реализовать интерфейс шаблонов, выполнить настройку моделей и создание CRUD-операций.

Во второй части необходимо расширить функциональность на основе уже написанного кода и собранного интерфейса, добавить новые возможности, такие как формы, аутентификация, права доступа и кеширование.

**Критерии оценки**

Наставник примет вашу работу, если в ней соблюдаются следующие условия. Проверьте работу по критериям перед тем, как отправить ее.

Посмотреть критерии

Чтобы удержать текущих клиентов бизнеса, часто используются вспомогательные, или «прогревающие», рассылки для информирования и привлечения.

В этом проекте вам нужно разработать **сервис управления рассылками, администрирования и получения статистики**.

**Техническое задание на разработку веб-приложения с рассылкой сообщений**

Создать веб-приложение на Django, которое позволяет пользователям управлять рассылками сообщений для клиентов. Приложение должно включать функциональность для создания, просмотра, редактирования и удаления рассылок, а также отправки сообщений по требованию.

Прежде чем приступить к написанию кода, ознакомьтесь со всеми условиями проекта (часть 1 и часть 2) и по возможности выполните декомпозицию.

**ЧАСТЬ 1. Разработка сервиса**

**1. Управление клиентами**

Реализовать возможность добавлять, просматривать, редактировать и удалять получателей рассылки (клиентов).

**Модель «Получатель рассылки»:**

* **Email** (строка, уникальное).
* **Ф. И. О.** (строка).
* **Комментарий** (текст).

**2. Управление сообщениями**

Реализовать возможность добавлять, просматривать, редактировать и удалять сообщения.

**Модель «Сообщение»:**

* **Тема письма** (строка).
* **Тело письма** (текст).

**3. Управление рассылками**

Реализовать возможность добавлять, просматривать, редактировать и удалять рассылки.

**Статусы рассылки:**

* **Создана —** рассылка была создана, но еще ни разу не была отправлена.
* **Запущена —** рассылка активна и была отправлена хотя бы один раз.
* **Завершена —** время окончания отправки рассылки прошло.

**Модель «Рассылка»:**

* **Дата и время первой отправки** (datetime).
* **Дата и время окончания отправки** (datetime).
* **Статус** (строка:

'Завершена'

,

'Создана'

,

'Запущена'

).

* **Сообщение** (внешний ключ на модель «Сообщение»).
* **Получатели** («многие ко многим», связь с моделью «Получатель»).

**4. Отправка сообщений по требованию**

Реализовать возможность отправки рассылки вручную через интерфейс пользователя и командную строку.

**5. Попытки рассылок**

Для каждой попытки отправки рассылки должны сохраняться следующие данные: дата и время попытки, статус (успешно / не успешно), ответ почтового сервера.

**Логика работы попытки рассылки**

**Попытка рассылки** — это запись о каждой попытке отправки сообщения по рассылке. Она содержит информацию о том, была ли попытка успешной, когда она произошла и какой ответ вернул почтовый сервер.

1. **Инициация отправки**.

Попытка рассылки создается каждый раз, когда запускается процесс отправки сообщений для конкретной рассылки.

1. **Определение получателей**.

Список клиентов, которым должно быть отправлено сообщение, определяется из выбранных клиентов для данной рассылки.

1. **Отправка сообщения**:
   * Для каждого клиента из списка выполняется отправка письма с помощью функции

send\_mail()

 из Django.

* + В случае успешной отправки письма создается запись в модели

Попытка\_рассылки

 со статусом «успешно».

* + В случае ошибки отправки, например из-за недоступности почтового сервера, создается запись со статусом «не успешно» и текстом ошибки в поле «Ответ почтового сервера».

1. **Сбор статистики**.

Каждая попытка отправки письма фиксируется, что позволяет отслеживать успешные и неуспешные попытки, а также общее количество попыток.

1. **Анализ и отчетность**.

Данные о попытках рассылок используются для формирования статистики и отчетности, предоставляя пользователю информацию о статусе и результатах его рассылок.

**Модель «Попытка рассылки»:**

* **Дата и время попытки** (datetime).
* **Статус** (строка:

'Успешно'

,

'Не успешно'

).

* **Ответ почтового сервера** (текст).
* **Рассылка** (внешний ключ на модель «Рассылка»).

**6. Главная страница**

На главной странице должно отображаться количество всех рассылок, количество активных рассылок (со статусом

'Запущена'

) и количество уникальных получателей.

**ЧАСТЬ 2. Расширение функциональности**

**7. Регистрация и аутентификация пользователей**

* Реализовать систему регистрации и аутентификации пользователей.
* Пользователи должны иметь возможность зарегистрироваться на сайте, подтвердив свой email.
* Должна быть реализована функция входа и выхода из системы.
* Необходимо предусмотреть возможность восстановления пароля.

**8. Статистика и отчеты**

Необходимо собирать и отображать информацию о количестве успешных/неуспешных попыток рассылок пользователя и количестве отправленных сообщений.

**9. Ограничение доступа**

* Пользователи могут управлять только своими рассылками и клиентами.
* Менеджеры могут просматривать все рассылки и клиентов, но не могут редактировать или удалять чужие данные.

**Описание ролей и прав доступа:**

1. **Пользователь**
   * Создание, просмотр, редактирование и удаление своих клиентов и рассылок.
   * Просмотр статистики по своим рассылкам.
2. **Менеджер**
   * Просмотр всех клиентов и рассылок.
   * Просмотр списка пользователей сервиса.
   * Блокировка пользователей сервиса.
   * Отключение рассылок.

**10. Кеширование**

Настроить серверное и клиентское кеширование для повышения производительности.

**\* Дополнительные задания**

**1. Автоматическая отправка сообщений по расписанию**

Реализовать автоматическую отправку рассылок по расписанию через библиотеку

django-apscheduler

 и написать скрипт для автоматической отправки сообщений по расписанию.

**2. Логирование**

Реализовать логирование основных событий и ошибок в работе приложения.

Дополнительные задания, помеченные звездочкой, желательно, но не обязательно выполнять.