**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”  
(УНИВЕРСИТЕТ ИТМО)**

**ЦЕНТР АВТОРИЗОВАННОГО ОБУЧЕНИЯ ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ**

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА**

*Разработка web сайта для планинга личных задач.*

Автор\_\_Одынец Иван Михайлович\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество) (Подпись)

**Центр авторизованного обучения информационным технологиям**

Наименование программы **«Python-разработчик»**

Руководитель \_\_Кузьмин К.М.\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

**К защите допустить**

|  |  |  |
| --- | --- | --- |
| *Заместитель директора ЦАО ИТ,*  *к.т.н.* |  | / *Т.В. Зудилова*/ |

Санкт-Петербург, 2023г.

Обучающийся\_\_\_\_\_\_\_\_\_\_\_Одынец И.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Группа 124/29

(Фамилия, И. О.)

Работа принята «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Работа выполнена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Секретарь ИАК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, И. О.) (подпись)

Листов хранения \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Демонстрационных материалов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОГЛАВЛЕНИЕ**

[**Введение 4**](#_Toc256011441)

[**Цель 6**](#_Toc256011442)

**1.** [**Описание системы 7**](#_Toc256011443)

[**1.1 Используемые фреймворки и библиотеки 7**](#_Toc256011444)

**1.2** [**Структура веб-приложения 10**](#_Toc256011445)

**1.3** [**Руководство пользователя 12**](#_Toc256011445)

**2.** [**Структура данных 17**](#_Toc256011443)

[**Заключение 1**](#_Toc256011446)**9**

[**Литература 20**](#_Toc256011449)

# Введение

В современном мире, когда время становится все более ценным ресурсом, необходимо уметь эффективно планировать свою жизнь и достигать поставленных целей. Одним из способов помочь себе в этом может стать использование специализированных инструментов для планирования задач.

Данный проект направлен на разработку web-сайта для планирования личных задач. Целью данной работы является создание удобного и функционального инструмента, который позволит пользователям более эффективно управлять своим временем и достигать своих целей.

В работе будет рассмотрена вся процедура создания сайта: начиная от выбора технологий и архитектуры, до разработки пользовательского интерфейса и функционала. Также будет уделено внимание анализу существующих решений на рынке и их сравнение с разрабатываемым проектом.

Ожидается, что данный web-сайт будет полезным инструментом для многих пользователей, помогая им эффективнее использовать свое время и достигать желаемых результатов.

В данной работе выбран язык программирования Python по нескольким причинам.

Во-первых, Python является одним из наиболее популярных языков программирования в мире, что означает, что существует множество библиотек и инструментов, которые могут быть использованы для разработки web-приложений. Например, для разработки web-приложений на Python широко используется фреймворк Django, который предоставляет множество готовых решений для работы с базами данных, обработки форм и маршрутизации запросов.

Во-вторых, Python обладает простым и понятным синтаксисом, что делает его очень привлекательным для начинающих программистов. Это также упрощает процесс разработки и поддержки кода, что важно для долгосрочного проекта.

В-третьих, Python является языком программирования, который имеет широкую сферу применения, и позволяет легко интегрироваться с другими технологиями и системами. Например, с помощью Python можно разрабатывать множество приложений, от научных вычислений до анализа данных и машинного обучения.

Таким образом, выбор Python для разработки web-сайта для планирования личных задач является обоснованным и позволит создать функциональное и эффективное приложение.

# Цель

Цель данной работы заключается в разработке web-сайта для планирования личных задач, который будет предоставлять пользователям удобный и функциональный инструмент для управления своим временем и достижения поставленных целей.

Для достижения данной цели необходимо решить следующие задачи:

Изучить существующие инструменты и решения для планирования задач, а также провести анализ их особенностей и преимуществ.

Выбрать архитектуру и технологии для разработки web-сайта, которые позволят реализовать все необходимые функциональные возможности и обеспечить надежность приложения.

Разработать пользовательский интерфейс web-сайта, который будет простым и удобным для использования, а также обеспечивать удобный доступ к основным функциональным возможностям.

Реализовать функциональность web-сайта, которая будет позволять пользователям создавать, редактировать задачи, а также получать уведомления о предстоящих событиях.

Таким образом, основной целью данной работы является разработка функционального и удобного web-сайта для планирования личных задач, который поможет пользователям более эффективно использовать свое время и достигать поставленных целей.

# 1. Описание системы

# 1.1 Используемые фреймворки и библиотеки

Django - это высокоуровневый фреймворк для разработки web-приложений на языке Python. Он был создан с целью упростить и ускорить процесс создания web-приложений, предоставляя разработчикам множество готовых инструментов и решений для работы с базами данных, формами, маршрутизации запросов и другими функциональными возможностями.

Основные преимущества Django:

Быстрая разработка приложений. Django предоставляет множество готовых инструментов и функциональных возможностей, что позволяет разработчикам быстро создавать и тестировать web-приложения.

Мощная архитектура. Django использует паттерн Model-View-Controller (MVC), который позволяет разделять приложение на логические компоненты и управлять ими независимо друг от друга.

Большое сообщество разработчиков. Django является одним из самых популярных фреймворков для web-разработки на Python, что означает, что существует множество сообществ, ресурсов и библиотек, которые могут помочь в разработке приложения.

Легкая интеграция с другими технологиями. Django легко интегрируется с другими технологиями и системами, что позволяет использовать его для разработки различных типов приложений, от веб-сайтов до веб-приложений.

Безопасность. Django предоставляет множество инструментов и механизмов для обеспечения безопасности приложения, включая защиту от атак, хранение паролей и контроль доступа.

Одним из функциональных требований было уведомления пользователю о предстоящих событиях. Для это необходим модуль способный создавать асинхронный задачи. Такими модулями являются celery и huey. Celery является популярным и широко используемым модулем в сообществе Python для управления асинхронными задачами и очередями. Однако huey имеет ряд преимуществ, поэтому

был выбран – huey.

Huey - это модуль для Python, который предоставляет простой и эффективный способ управления очередью задач в веб-приложениях. Он позволяет разработчикам легко создавать и управлять асинхронными задачами, такими как отправка электронных писем, обработка изображений или выполнение вычислительно тяжелых задач.

Основные преимущества Huey:

Простота разработки. Huey имеет интуитивно понятный API и простой синтаксис, что делает его более доступным для разработчиков, не имеющих большого опыта в области асинхронного программирования.

Легковесность. Huey имеет небольшой размер и использует меньше ресурсов, что позволяет снизить нагрузку на сервер и повысить производительность приложения.

Гибкость. Huey позволяет настроить множество параметров, таких как время жизни задачи, максимальное количество задач в очереди и другие, что позволяет адаптировать модуль под различные требования приложения.

Поддержка множества бэкэндов. Huey поддерживает множество бэкэндов для управления очередью задач, включая базы данных, Redis, sqlite и другие.

Расширяемость. Huey предоставляет механизмы для создания пользовательских задач и бэкэндов, что позволяет разработчикам создавать приложения с уникальной функциональностью.

В качестве источника уведомлений был выбран мессенджер Telegram. Для этого необходимо создать телеграм бота. Существует несколько python модулей для создания ботов, aiogram, python-telegram-bot, telebot(pyTelegramBotAPI) и другие. Был выбран pyTelegramBotAPI.

PyTelegramBotAPI - это модуль Python, который позволяет разработчикам создавать Telegram-ботов с помощью языка Python. Он предоставляет простой API для взаимодействия с Telegram-платформой и позволяет создавать ботов для различных целей, таких как обработка запросов пользователей, отправка уведомлений и многое другое.

Модуль pyTelegramBotAPI использует официальный Telegram Bot API, что обеспечивает надежность и стабильность взаимодействия с Telegram-платформой. Он позволяет легко создавать и настраивать ботов с помощью простого и интуитивно понятного API.

С помощью pyTelegramBotAPI можно создавать ботов с различными функциями, такими как отправка сообщений, получение информации о пользователе, работа с фото и видео, обработка команд и т.д. Благодаря простоте и удобству использования, этот модуль является популярным инструментом для создания Telegram-ботов в сообществе Python.

В качестве СУБД был выбран SQLite, так как он встроен в Django и структура таблиц в базе данных не является большой, помимо этого нет необходимости в дополнительном функционале как в PostgreSQL.

# 1.2 Структура веб-приложения

Дерево проекта представляет из себя 5 каталогов easyplan, plannerapp, templates, tgbotapp, users и файлы db.sqlite3, huey\_db.db, manage.py. db.sqlite3 - это база данных для веб-приложения Django. huey\_db.db база данных для работы huey. manage.py это командный инструмент предназначенный для управления проектом, он предоставляет доступ к широкому набору команд для выполнения различных задач, таких как создание миграций базы данных, запуск сервера разработки, сборка статических файлов, создание суперпользователей и многое другое. Папка templates предназначена для хранения шаблонов HTML-страниц, которые используются для генерации динамических страниц веб-приложения. Шаблоны являются основным механизмом отображения данных в Django, и папка templates обеспечивает централизованное хранение всех шаблонов веб-приложения.

Папки plannerapp, tgbotapp, users являются приложениями Django. Приложение - это независимый модуль, который реализует какую-то конкретную функциональность, позволяет разбить веб-приложения на независимые компоненты, что делает код более структурированным и удобным для сопровождения. Кроме того, использование приложений позволяет повторно использовать код и легко добавлять новую функциональность в приложение. Подробнее о каждом приложении:

* plannerapp - в данном приложении реализован функционал работы с задачами. Здесь находится модель базы данных, функции отображения добавления и изменения задач. Также здесь реализованы формы Django и постановка задач в очередь huey.
* tgbotapp - в данном приложении реализовано взаимодействие с API телеграм. Модель базы данных пользователей телеграм бота. Функции сохранения id пользователя телеграм в бд. Функции отправки сообщений от имени бота.
* users - в данном приложении реализованы все функции для работы с пользователем. Функции аутентификации, регистрации, расширение стандартной базы данных пользователя и формы Django.

Каталог с названием проекта - это пакет Python, который содержит настройки, URL-адреса и другие файлы конфигурации для проекта:

* settings.py содержит различные настройки для проекта, такие как конфигурация базы данных, конфигурация статических файлов и установленные приложения.
* urls.py содержит конфигурацию маршрутизации URL для проекта.
* asgi.py и wsgi.py являются точками входа для серверов ASGI и WSGI соответственно.

Структура приложения Django имеет общий вид, который включает следующие элементы:

* admin.py содержит регистрацию моделей в интерфейсе администратора Django.
* apps.py содержит конфигурацию приложения.
* models.py определяет модели базы данных для приложения.
* views.py содержит функции, которые обрабатывают HTTP-запросы и возвращают HTTP-ответы.
* tests.py содержит модульные тесты для приложения.
* forms.py содержит формы для работы вводом и обработкой данных моделей.
* migrations каталог содержит миграции схемы базы данных для приложения.

# 1.3 Руководство пользователя

При открытии главной страницы пользователь видит страницу как изображено на рис. 1

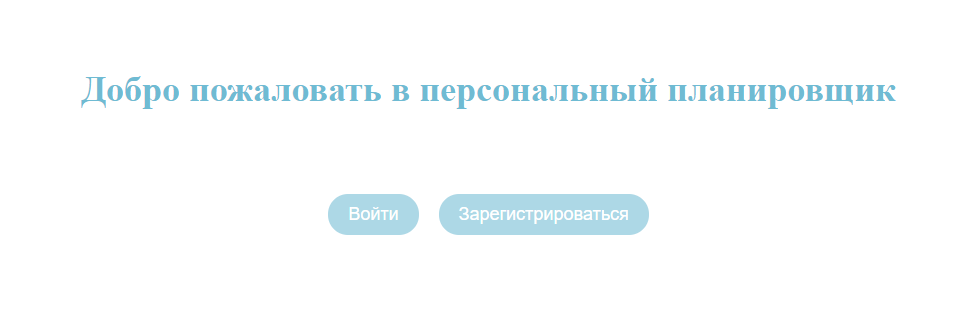


Рисунок - 1. Главная страница.

На главной странице нажав на кнопку Зарегистрироваться, открывается страница с регистрацией, где расположена форма для регистрации рис. 2.

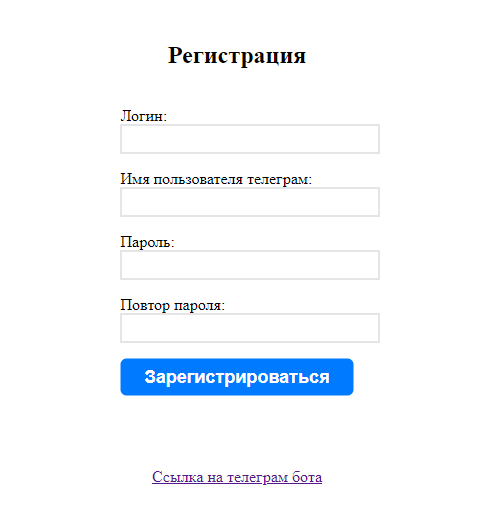


Рисунок - 2. Страница регистрации.

После заполнения формы и нажатии кнопки Зарегистрироваться, если пользователь не активировал бота в своем телеграмм аккаунте, появится сообщение с просьбой перейти по ссылке в телеграм. В телеграме необходимо установить имя пользователя, если оно не установлено и в боте нажать кнопку Запустить рис. 3.

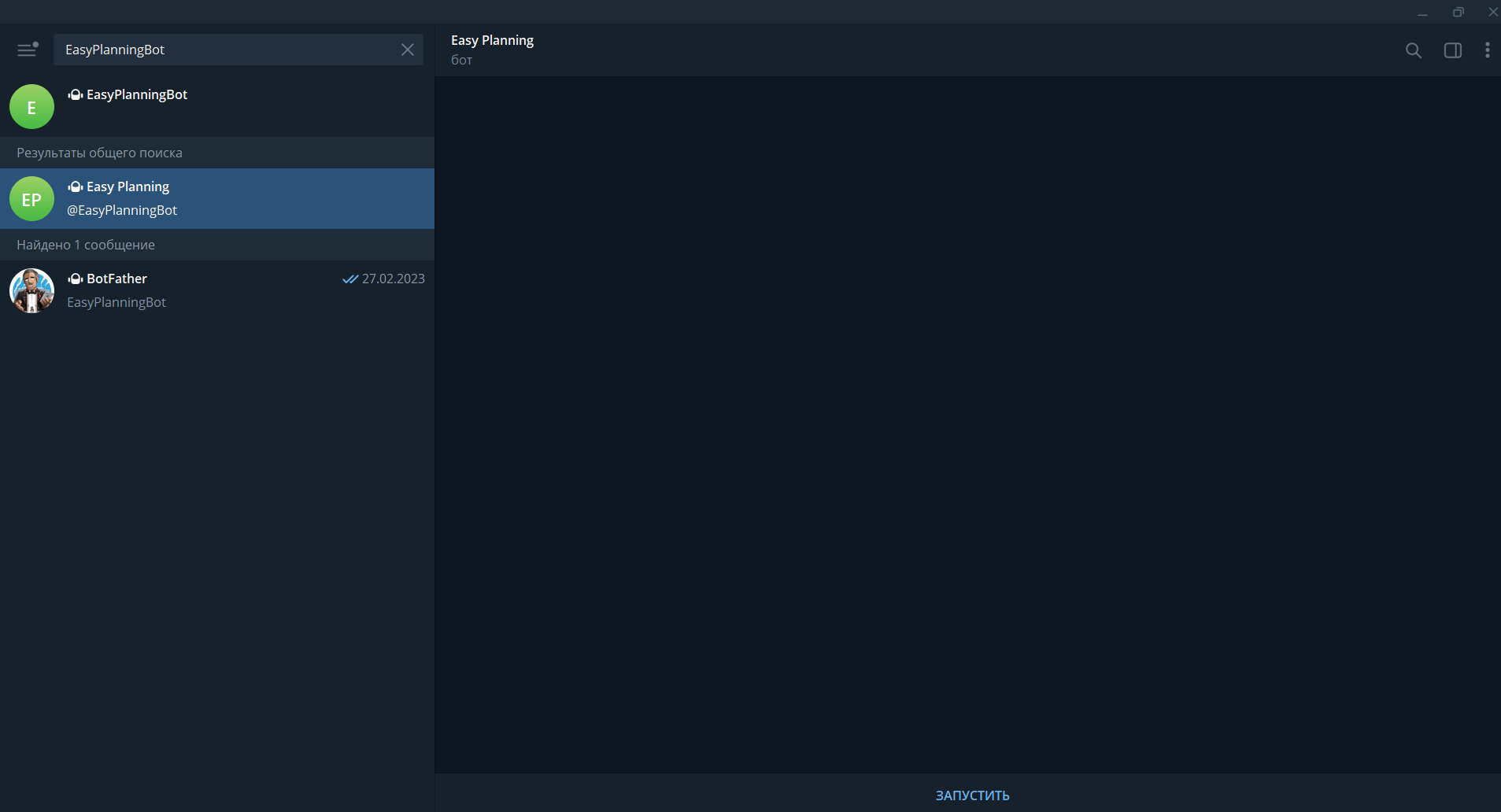


Рисунок - 3. Телеграм бот.

Далее необходимо перейти на страницу регистрации, до заполнить форму регистрации и нажать кнопку Зарегистрироваться. После этого откроется страница входа в планировщик рис. 4.

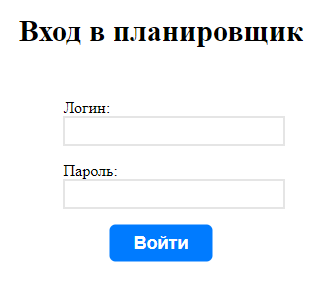


Рисунок - 4. Страница входа.

На странице входа необходимо ввести свои логин, пароль и нажать кнопку Войти. После этого откроется страница с вашими задачами рис. 5. Если у вас еще нет задач будет написано “у вас нет ни одной задачи”.

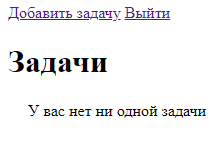


Рисунок - 5. Страница без задач.

Нажав на кнопку Добавить задачу откроется страница с формой добавление задачи рис. 6.

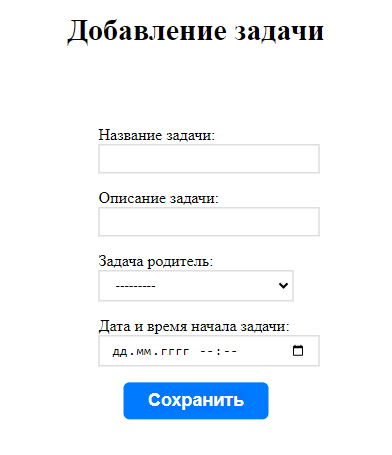


Рисунок - 6. Страница добавления задачи.

На форме добавления задачи необходимо ввести название задачи. Поле описание задачи необязательное для заполнения. Далее следует выбрать в выпадающем списке задачу родителя, если оставить это поле пустым, то добавленная задача на самом верхнем уровне. Дату и время можно ввести вручную или нажать на символ календаря рис.7 и выбрать дату и время.

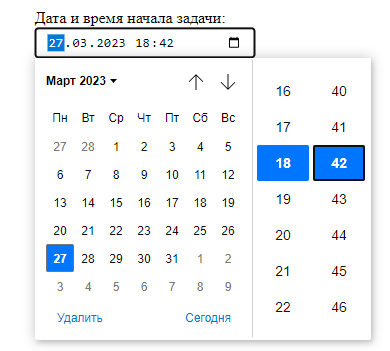


Рисунок - 7. Всплывающее окно ввода даты и времени.

Далее необходимо нажать на кнопку Сохранить и откроется страница с уже добавленной задачей. На рисунке 8 представлен пример с несколькими задачами в виде древовидной структуры.

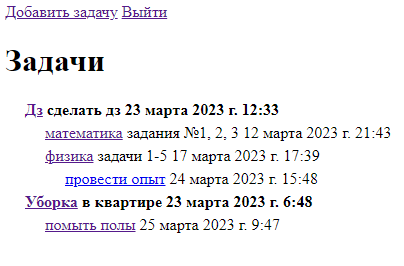


Рисунок -8. Страница с задачами.

На каждую из этих задач можно нажать и откроется страница с редактированием задачи рис. 9.

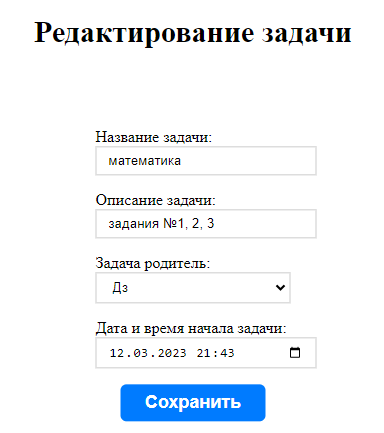


Рисунок - 9. Страница редактирования задачи.

Форма для редактирования задачи выглядит также, как и для добавления задачи, однако она уже заполнена данными. При нажатии кнопки Выйти на рисунке 8 пользователь выйдет из системы и перейдет на главную страницу.

# 2. Структура данных

Модель базы данных, которая используются в данном проекте изображена на рисунке 10.

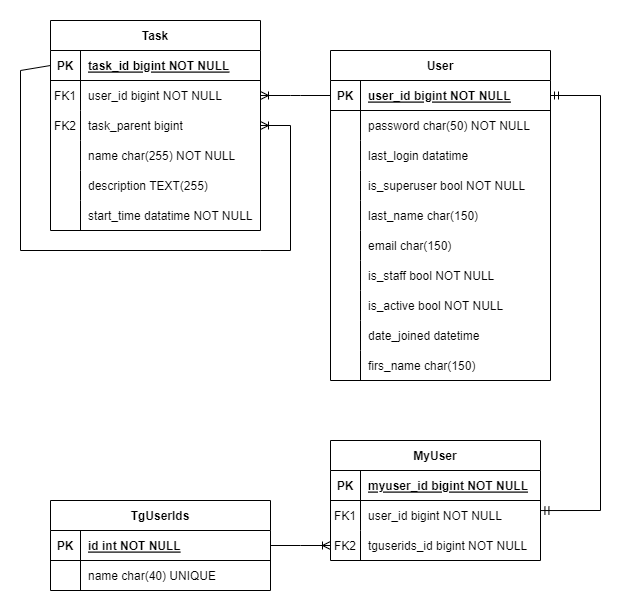


Рисунок - 10. Схема связей данных в базе данных.

В таблице Task – задача, находятся такие данные о задачах, как какого пользователя это задача, какая задача родитель, название, описание, дата и время на которое запланирована задача. Таблица User - стандартная модель Django, которая содержит информацию о пользователе. Таблица MyUser – расширение стандартной модели User. TgUserIds – таблица, в которой хранятся id и имя пользователя бота телеграм.

# Заключение

В результате выполнения данной работы был разработан веб-сайт для планирования личных задач, используя фреймворк Django и Python-модули Huey и pyTelegramBotAPI.

Были достигнуты следующие цели:

* изучение фреймворка Django и его основных компонентов;
* изучение Python-модуля Huey и его использование для управления фоновыми задачами в веб-приложении;
* изучение Python-модуля pyTelegramBotAPI и его использование для создания Telegram-бота, связанного с веб-приложением;
* разработка функционального веб-сайта для планирования личных задач, способного обрабатывать задачи фоновыми процессами и взаимодействовать с пользователями через Telegram-бот.

Разработка данного веб-сайта показала, что фреймворк Django обладает широким набором инструментов и функций для разработки высокоуровневых веб-приложений. Python-модуль Huey предоставляет мощный механизм для управления фоновыми задачами в веб-приложении, а Python-модуль pyTelegramBotAPI делает возможным быстрое создание Telegram-ботов, связанных с веб-приложением.

В итоге, данная работа показала, что с помощью инструментов Python и Django можно создавать функциональные веб-приложения с использованием современных технологий и инструментов.

# Список используемых источников

1. Документация Django [Электронный ресурс] –URL:<https://docs.djangoproject.com/en/4.1/> (дата обращения 10.03.2023).
2. Документация Huey [Электронный ресурс] –URL: <https://huey.readthedocs.io/en/latest/> (дата обращения 10.03.2023).
3. Документация PyTorch [Электронный ресурс] –URL: <https://pytorch.org/docs/stable/index.html> (дата обращения 10.03.2023).
4. Документация Python [Электронный ресурс] –URL: <https://docs.python.org/3.10/> (дата обращения 10.03.2023).