Федеральное государственное образовательное бюджетное учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Дипломный проект

Тема «Разработка информационной системы «Планировщик задач»

Студент Иванов Михаил Александрович

Учебная группа 4ПКС-120

Специальность 09.02.03 Программирование в компьютерных системах

Руководитель

|  |  |  |  |
| --- | --- | --- | --- |
| дипломного проекта | \_\_\_\_\_\_\_\_\_\_\_\_ |  | И.В. Сибирев |
| Консультант | (подпись) |  | (инициалы, фамилия) |
| дипломного проекта | \_\_\_\_\_\_\_\_\_\_\_\_ |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (при наличии)      Председатель предметной | (подпись) |  | (инициалы, фамилия) |
| (цикловой) комиссии | \_\_\_\_\_\_\_\_\_\_\_\_ |  | Т.Г.Аксенова\_ |
|  | (подпись)      Москва – 2024г. |  | (инициалы, фамилия) |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc168246954)

[Глава 1. Предпроектное исследование предметной области 5](#_Toc168246955)

[1.1. Описание предметной области 5](#_Toc168246956)

[1.2. Сравнительный анализ программ-аналогов 6](#_Toc168246957)

[1.3. Постановка задачи 8](#_Toc168246958)

[1.4. Характеристика инструментальных средств разработки 10](#_Toc168246959)

[Глава 2. Проектирование и реализация программы 12](#_Toc168246960)

[2.1. Анализ требований и определение спецификаций программного обеспечения 12](#_Toc168246961)

[2.2. Проектирование программного обеспечения 15](#_Toc168246962)

[2.3. Разработка программного обеспечения 18](#_Toc168246963)

[2.4. Тестирование программы 24](#_Toc168246964)

[2.5. Руководство по использованию веб-приложения 28](#_Toc168246965)

[ЗАКЛЮЧЕНИЕ 31](#_Toc168246966)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ИНТЕРНЕТ-РЕСУРСОВ 33](#_Toc168246967)

[ПРИЛОЖЕНИЕ 36](#_Toc168246968)

# **ВВЕДЕНИЕ**

В ходе выполнения практики разрабатывался индивидуальный проект по разработке информационной системы «Планировщик задач».

Современные возможности информационных технологий предоставляют уникальные перспективы для разработки высокоэффективных информационных систем. В рамках дипломного проекта предлагается создание информационной системы «Планировщик задач» с использованием фреймворка Django и языка программирования Python.

Цель проекта: повышение уровней организации рабочего процесса, эффективного управления временными ресурсами и повышении личной продуктивности.

Для выполнения целей проекта нужно выполнить следующие задачи:

- Разработать привлекательный и интуитивно понятный пользовательский интерфейс системы с применением технологии Django.

- Написать программный код, обеспечивающий эффективное управление логикой работы информационной системы, включая функции планирования и отслеживания задач.

- Создать механизм сохранения и загрузки данных о состоянии системы, включая сохранение планов, задач и другой важной информации для возможности последующего восстановления.

- Разработать механизм отображения результатов задач в отдельном окне для удобства пользователей.

Объектом исследования является веб-приложение.

Предметом исследования выступают технологии разработки информационной системы «Планировщик задач».

Основные методы исследования включают анализ литературных источников по теме проекта, изучение образовательных методик, а также сравнительный анализ существующих аналогичных информационных систем.

Актуальность проекта обосновывается не только растущей потребностью в эффективных инструментах управления задачами, но и важностью создания средств, способствующих повышению продуктивности и оптимизации рабочего процесса.

Основное функциональное предназначение разрабатываемого программного средства в рамках дипломного проекта заключается в предоставлении пользователям возможности эффективного планирования и управления задачами через инновационную информационную систему.

Для реализации данного проекта будут использованы инструментальные средства и языки программирования, такие как Python, Django, система управления базами данных (СУБД), diagrams.net. Дополнительно, для установки и визуализации будет задействован Actual Installer, а также инструменты Microsoft Word и Microsoft PowerPoint.

Таким образом, дипломный проект направлен на создание информационной системы «Планировщик задач», разработанной с использованием Django и Python. Цель проекта - предоставить пользователям увлекательный инструмент для эффективного управления своими задачами, способствуя осознанному развитию стремления к знаниям и достижению высоких результатов в учебе.

# **Глава 1. Предпроектное исследование предметной области**

## **1.1. Описание предметной области**

Важной составной частью в современной информационной технологии и сфере управления временем является разработка информационной системы «Планировщик задач». Эта система создается с целью обеспечения эффективного управления задачами, предоставляя пользователям инструмент для оптимизации и структурирования их рабочих процессов.

В современном образовании и бизнесе становится все важнее использование технологий, которые не только упрощают задачи, но и обеспечивают более эффективное использование времени. Информационная система, разрабатываемая в рамках данного дипломного проекта на базе Django и Python, объединяет в себе функциональность управления задачами и образовательные элементы, предоставляя пользователю возможность не только эффективно организовывать свои дела, но и улучшать навыки планирования.

Основной целью является предоставление инструмента для эффективного планирования и управления задачами. Система предлагает пользователю функции создания задач, установки приоритетов, определения сроков и отслеживания выполнения. Важным элементом является также интеграция образовательных компонентов, предоставляющих пользователю советы по эффективному управлению временем и задачами.

Дизайн и визуальное оформление информационной системы играют ключевую роль в создании удобной и привлекательной среды для пользователя. Интерфейс должен быть интуитивно понятным и легким в использовании.

Центральным элементом разработки системы является база данных, которая должна быть тщательно структурирована и постоянно обновляться, чтобы обеспечить актуальность и разнообразие функционала системы. Разработка логики системы включает в себя определение правил

планирования, механизм сохранения данных о задачах и определение уровней сложности для повышения эффективности использования системы.

## **1.2. Сравнительный анализ программ-аналогов**

Прежде чем приступать к реализации, проведем анализ и сравним существующие приложения «Планировщик задач». Анализ существующих информационных систем для планирования задач в таблице 1:

Таблица 1. Существующие планировщики задач

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Todoist** | **Trello** | **Microsoft To-Do** | **Any.do** |
| Функционал |  |  |  |  |
| Отправка задач | + | + | + | + |
| Прикрепление файлов | + | + | - | + |
| Установка сроков | + | + | + | + |
| Приоритет задач | + | + | + | + |
| Группировка задач | + | + | + | + |
| Интерфейс |  |  |  |  |
| Наличие тем | + | + | + | + |
| Смена языка | + | + | + | + |
| Стоимость |  |  |  |  |
| Бесплатная установка | + | + | + | + |
| Платная подписка | + | + | + | + |
| Кроссплатформенность |  |  |  |  |
| Мобильное приложение | + | + | + | + |
| Десктоп приложение | + | + | + | + |
| Веб-версия приложения | + | + | + | + |
| Хранилище задач | Сервер | Облако | Облако | Сервер |
| Синхронизация между устройствами | + | + | + | + |
| Распределение задач по проектам | + | + | + | + |
| Оповещения и напоминания | + | + | + | + |

При разработке информационной системы "Планировщик задач" будут учтены лучшие практики и функциональности из рассматриваемых веб-приложений для эффективного планирования и управления задачами.

Примеры функционала и интерфейса некоторых из представленных сайтов:

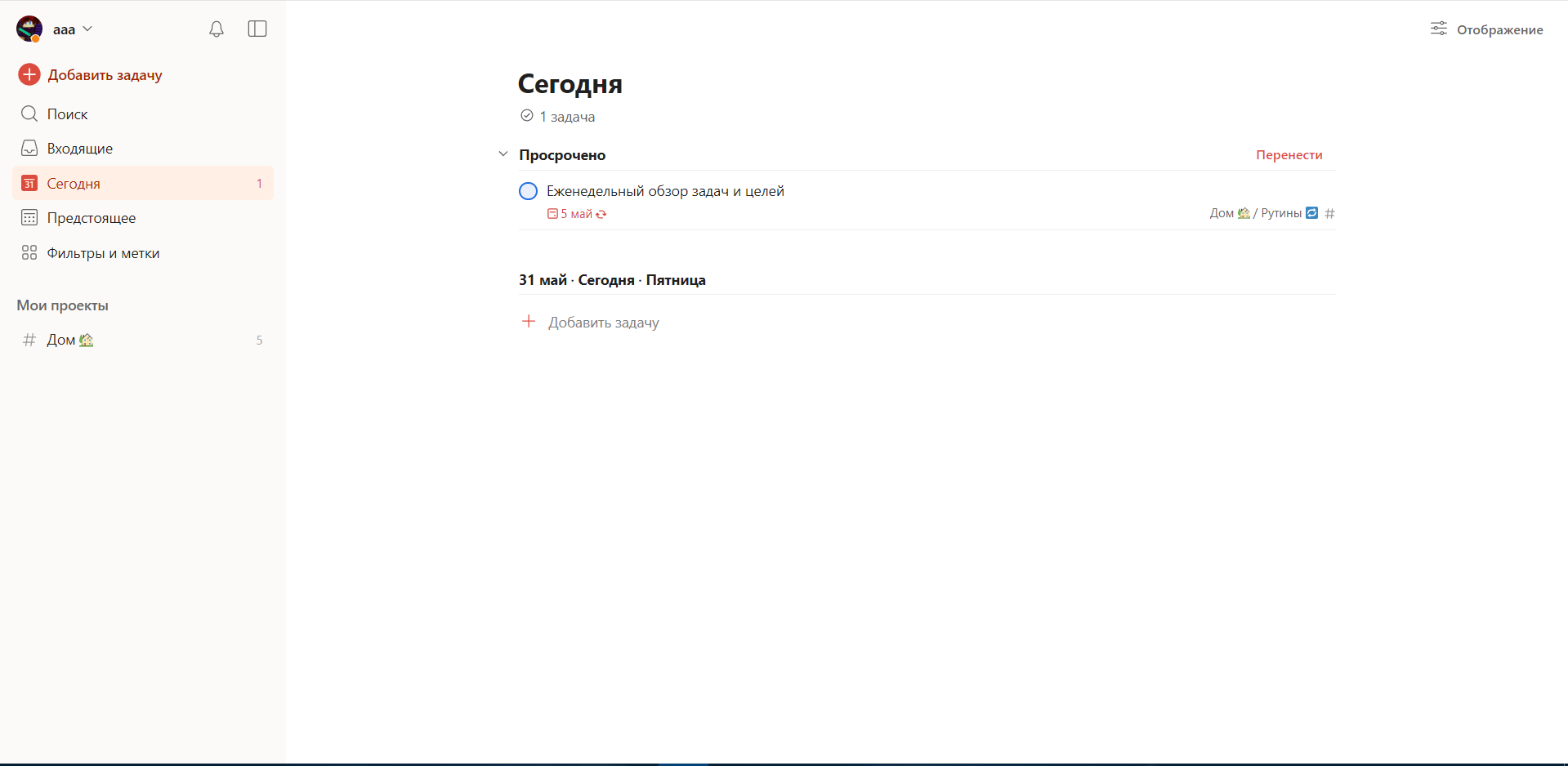
На рисунке 1 изображена главная страница сайта Todoist  


Рисунок 1. Главная страница сайта Todoist

На рисунке 2 изображена страница с добавленной задачей

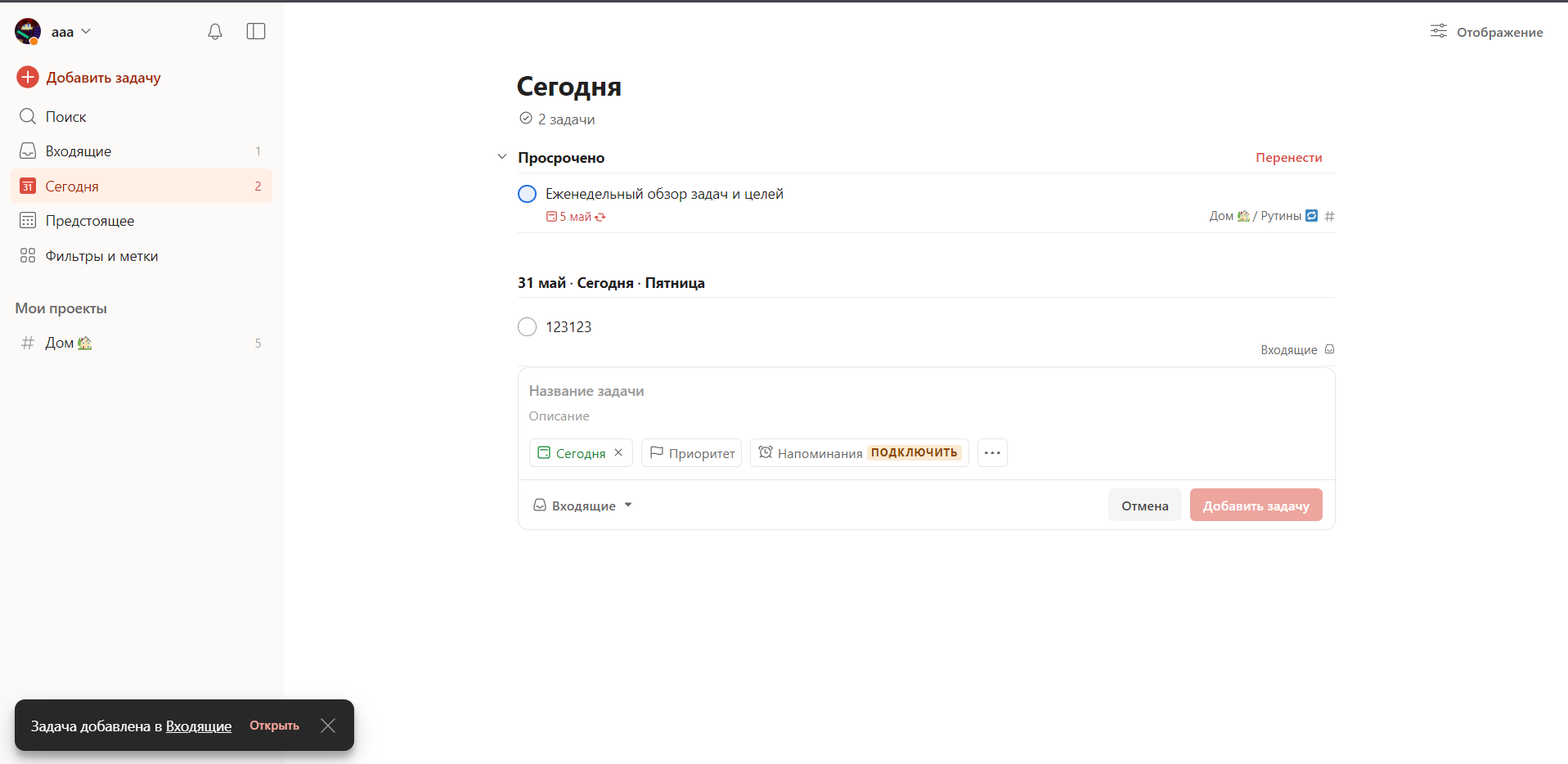


Рисунок 2. Страница с добавленной задачей

На рисунке 3 изображена страница планировщика задач Trello

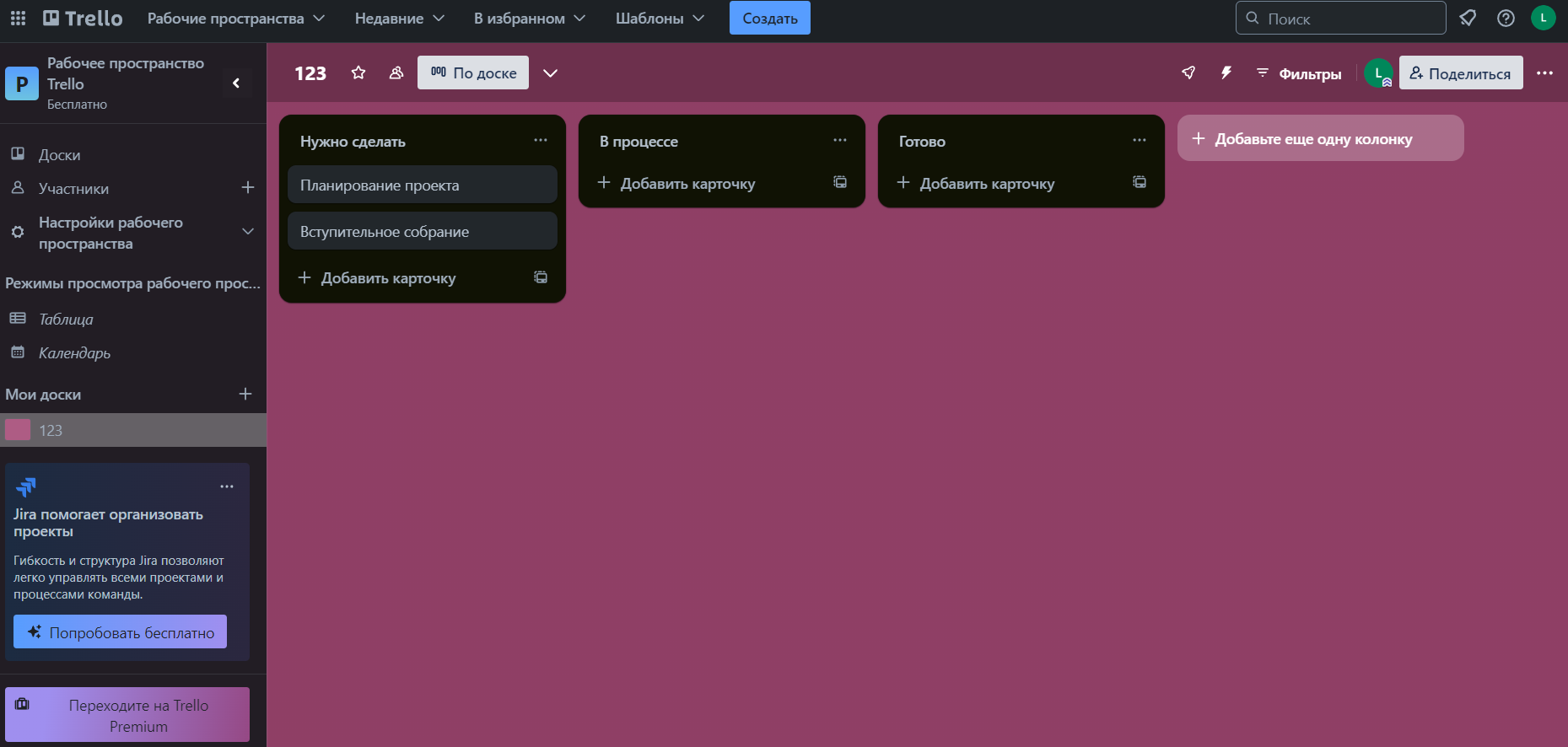


Рисунок 3. Страница планировщика задач Trello

На рисунке 4 изображена страница календаря Trello

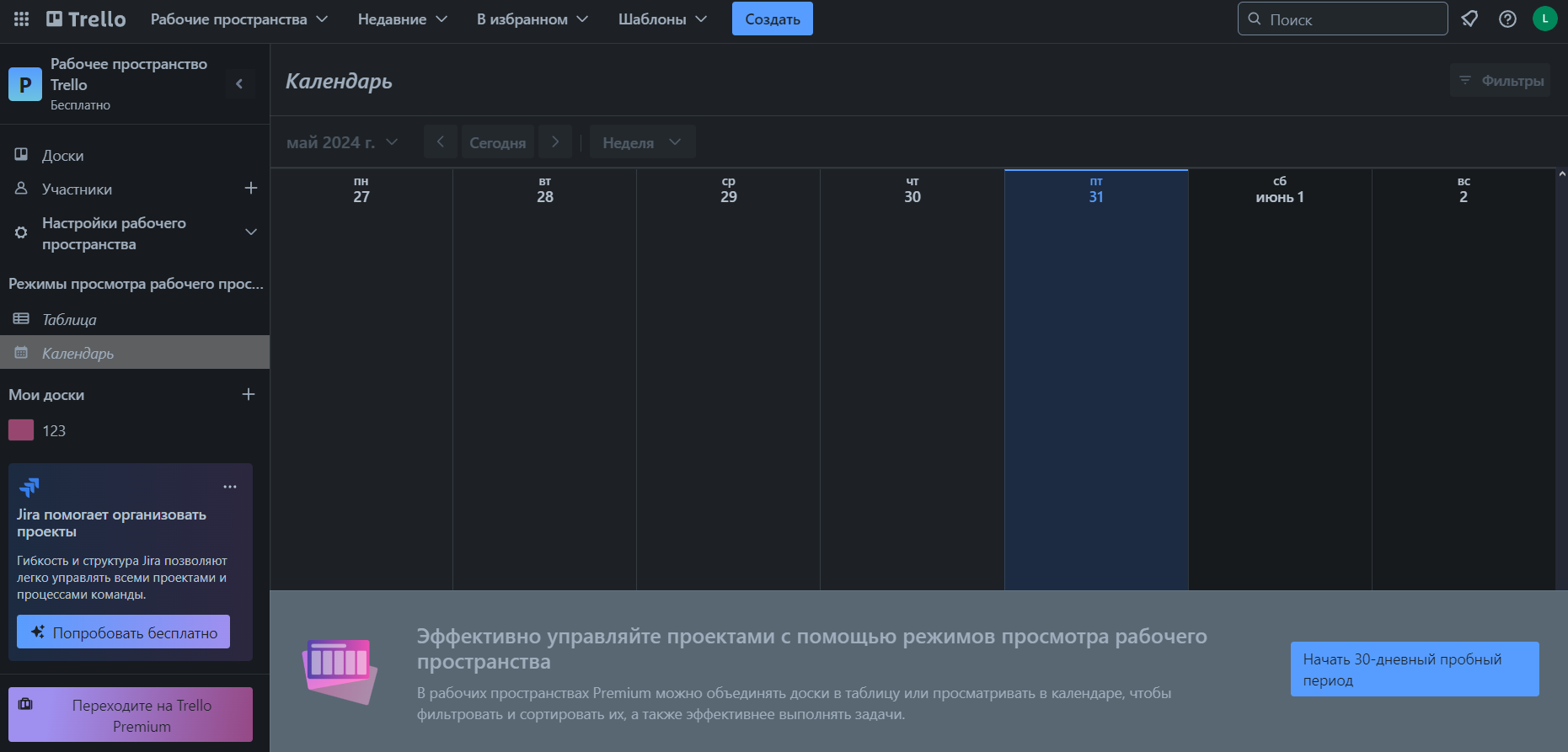


Рисунок 4. Страница календаря Trello

## **1.3. Постановка задачи**

Определение постановки задачи для разработки информационной системы «Планировщик задач» состоит в определении входной и выходной информации и других аспектов приложения.

Входной информацией для системы является: логин пользователя,

детали задачи (название, описание, срок выполнения, напоминание, повтор)

Выходной информацией для пользователя является: количество задач, статистика по выполненным задачам (количество выполненных, количество невыполненных), разные категории задач.

Логика работы системы:

Цель системы - обеспечить эффективное планирование и управление задачами пользователя. Пользователь создает, отслеживает и завершает задачи в соответствии с их приоритетом и сроками выполнения.

Правила взаимодействия:

Пользователь создает задачу, указывая ее детали.

Система предоставляет возможность отслеживания конец выполнения задачи.

Пользователь может изменять задачу (редактирование, удаление, продление)

Система предоставляет выполненные и невыполненные задачи.

Функциональные требования:

Обеспечить возможность создания и управления задачами.

Реализовать систему категорий для задач.

Иметь механизм отслеживания сроков выполнения задач.

Предоставить возможность просмотра и редактирования созданных задач.

Описание задач и их обработка:

Создание задачи: Пользователь должен иметь возможность создать новую задачу, указав ее детали и уровень приоритета.

Редактирование задачи: Пользователь может вносить изменения в уже созданные задачи (название, описание, сроки).

Отслеживание выполнения: Система должна отображать статус выполнения каждой задачи.

Уведомления: Система должна предоставлять пользователю уведомления о приближающихся сроках выполнения задач.

Оценка и статистика:

Разработать механизм оценки эффективности выполнения задач.

Позволить пользователю просматривать выполненные и невыполненные задачи.

Актуализация контента:

Регулярно обновлять систему новыми функциональности и улучшения.

Предоставить пользователям актуальные средства планирования.

Эксплуатационные требования:

Интерфейс пользователя: разработать интуитивно понятный интерфейс для комфортного взаимодействия с системой.

Поддержка платформ: обеспечить работу системы на операционной системе Windows 10.

Устойчивость и безопасность: гарантировать стабильную работу и защиту от сбоев и ошибок.

Подключение к сети интернет: обеспечить возможность подключения к сети для синхронизации данных.

Поддержка СУБД: интегрировать систему управления базами данных (СУБД) для эффективного хранения и доступа к данным.

## **1.4. Характеристика инструментальных средств разработки**

Современные инструменты разработки широко используются для создания и поддержки информационных систем для планирования задач. В этом проекте применяются следующие технологии:

Язык программирования Python выбран как основной язык для реализации логики работы с базой данных и обработки данных.

Фреймворк Django используется для создания веб-приложения, который предоставляет удобную архитектуру, встроенную административную панель и инструменты для разработки безопасных и масштабируемых приложений.

В процессе разработки используется Visual Studio Code, которая обеспечивает удобные возможности для написания, отладки и тестирования кода на Python.

Для создания пользовательского интерфейса используются HTML и CSS.

Система управления базами данных (СУБД) в данном случае используется SQLite, совместимая с Python и Django, для эффективного хранения, управления и извлечения данных.

Инструменты для создания диаграмм и схем для визуализации архитектуры системы, построения диаграмм баз данных и взаимодействий используются инструменты вроде draw.io или аналогичные.

В процессе разработки используются инструменты для создания документации и презентаций, такие как Microsoft Word и PowerPoint, а также для обработки графических изображений, например, draw.io.

JavaScript применяется для реализации логики работы кнопок и других элементов пользовательского интерфейса, обеспечивая динамическое взаимодействие пользователей с веб-приложением.

PHP используется для отправки писем и выполнения некоторых функций обработки данных, а также для интеграции различных функциональных модулей.

# **Глава 2. Проектирование и реализация программы**

## **2.1. Анализ требований и определение спецификаций программного обеспечения**

На рисунке 5 представлена диаграмма вариантов использования «Планировщика задач»



Рисунок 5. Диаграмма вариантов использования «Планировщика задач»

На рисунке 6 представлена функциональная диаграмма верхнего уровня

«Планировщика задач»

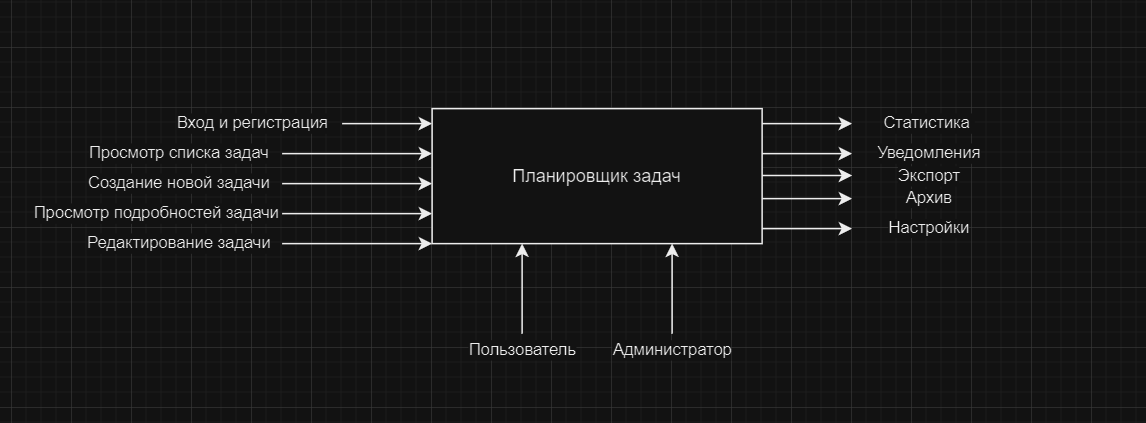
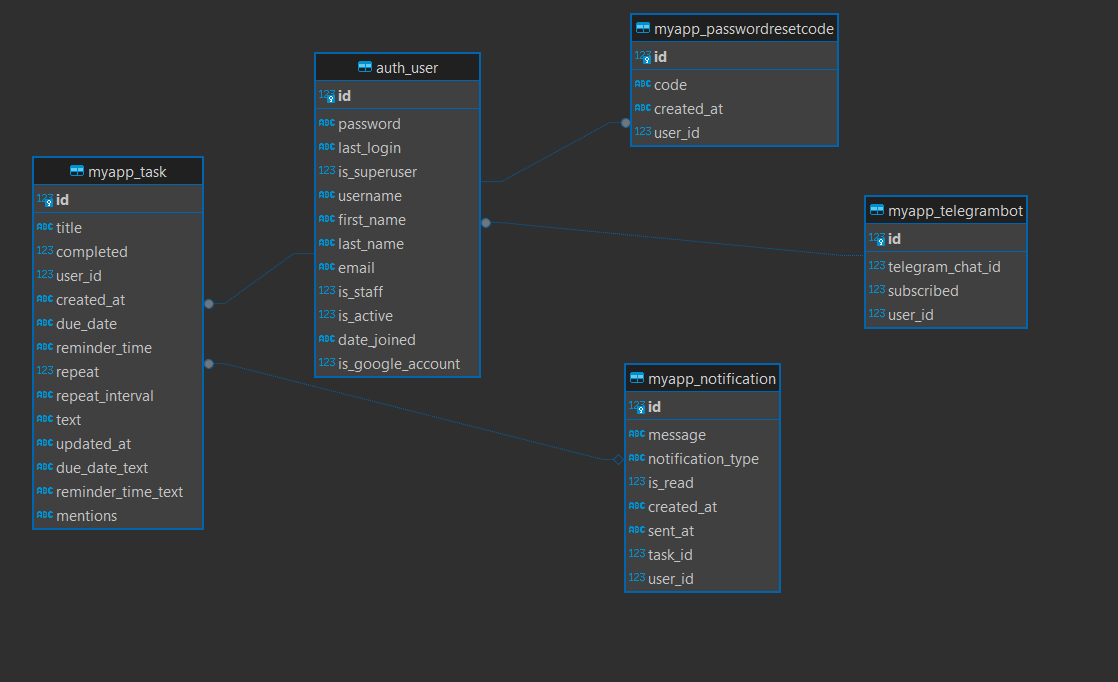


Рисунок 6. Функциональная диаграмма верхнего уровня

На рисунке 7 представлена ER-диаграмма

xРисунок 7. ER-диаграмма «Планировщика задач»

К ER-диаграмме идет словарь данных, который находится в Приложении №1. Ниже будут описаны таблицы базы данных:

Таблицы базы данных:

- auth\_user: в данной таблице хранится информация о пользователях, а именно их порядковый номер (id), пароль (password), дата последнего входа (last\_login), статус суперпользователя (is\_superuser), никнейм (username), имя (first\_name), фамилия (last\_name), электронная почта (email), статус сотрудника (is\_staff), статус активности (is\_active), дата регистрации (date\_joined) и статус использования аккаунта Google (is\_google\_account).

- myapp\_task: в этой таблице хранится информация о задачах. Она включает следующие поля: порядковый номер задачи (id), заголовок (title), статус завершенности задачи (completed), идентификатор пользователя (user\_id), дата создания (created\_at), дата выполнения (due\_date), время напоминания (reminder\_time), повтор задачи (repeat), интервал повтора (repeat\_interval), текст задачи (text), дата обновления (updated\_at), текстовое представление даты выполнения (due\_date\_text), текстовое представление времени напоминания (reminder\_time\_text) и упоминания (mentions).

- myapp\_passwordresetcode: в этой таблице хранятся коды для сброса пароля. Поля включают: порядковый номер (id), код (code), дата создания (created\_at) и идентификатор пользователя (user\_id).

- myapp\_telegrambot: в данной таблице хранится информация о подписке пользователей на уведомления в Telegram. Поля включают: порядковый номер (id), идентификатор чата в Telegram (telegram\_chat\_id), статус подписки (subscribed) и идентификатор пользователя (user\_id).

- myapp\_notification: в этой таблице хранятся уведомления для пользователей. Поля включают: порядковый номер (id), сообщение уведомления (message), тип уведомления (notification\_type), статус прочтения (is\_read), дата создания (created\_at), дата отправки (sent\_at), идентификатор задачи (task\_id) и идентификатор пользователя (user\_id).

Взаимосвязи между таблицами:

- auth\_user с myapp\_task: идентификатор пользователя (user\_id) из таблицы myapp\_task связывается с идентификатором пользователя (id) в таблице auth\_user.

- auth\_user с myapp\_passwordresetcode: идентификатор пользователя (user\_id) из таблицы myapp\_passwordresetcode связывается с идентификатором пользователя (id) в таблице auth\_user.

- auth\_user с myapp\_telegrambot: идентификатор пользователя (user\_id) из таблицы myapp\_telegrambot связывается с идентификатором пользователя (id) в таблице auth\_user.

- auth\_user с myapp\_notification: идентификатор пользователя (user\_id) из таблицы myapp\_notification связывается с идентификатором пользователя (id) в таблице auth\_user.

- myapp\_task с myapp\_notification: Идентификатор задачи (task\_id) из таблицы myapp\_notification связывается с идентификатором задачи (id) в таблице myapp\_task.

Таким образом, диаграмма показывает, как таблицы и их поля взаимодействуют друг с другом для обеспечения функциональности планировщика задач, включая управление пользователями, задачами, сбросом паролей и уведомлениями.

## **2.2. Проектирование программного обеспечения**

Далее представлены следующие схемы:

Структурная схема программы применяется для иллюстрации устройства программы и взаимодействия её компонентов. Это средство делает процесс чтения и осмысления кода более простым, демонстрируя как осуществляется последовательность операций и какие связи существуют между различными блоками кода.

Функциональная схема программы служит для наглядного представления взаимодействий между разными функциями и подпрограммами, а также показывает, как происходит передача данных внутри программы.

На рисунке 8 представлена структурная схема «Планировщика задач»

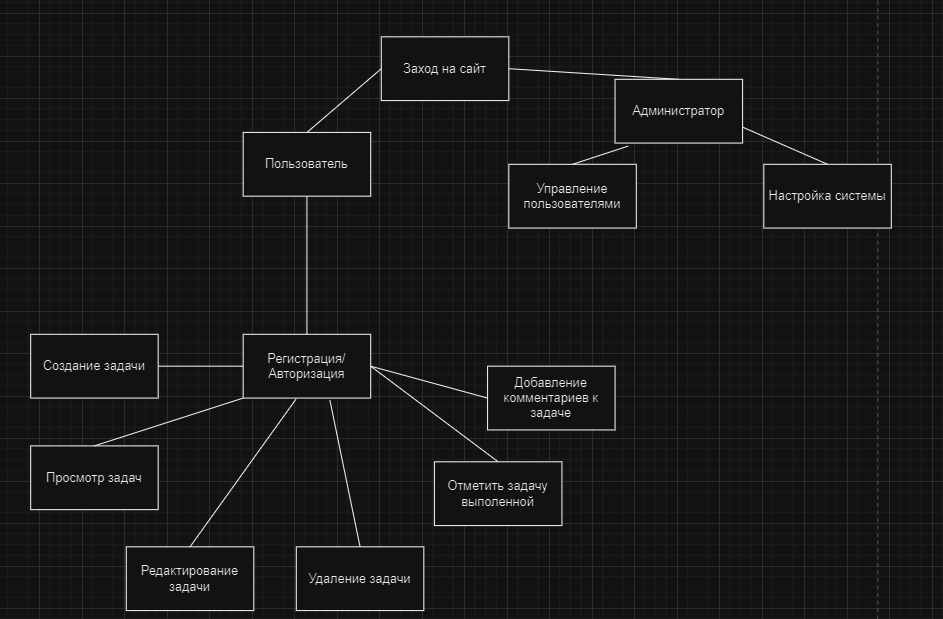


Рисунок 8. Структурная схема «Планировщика задач»

На рисунке 9 представлена функциональная схема «Планировщика задач»

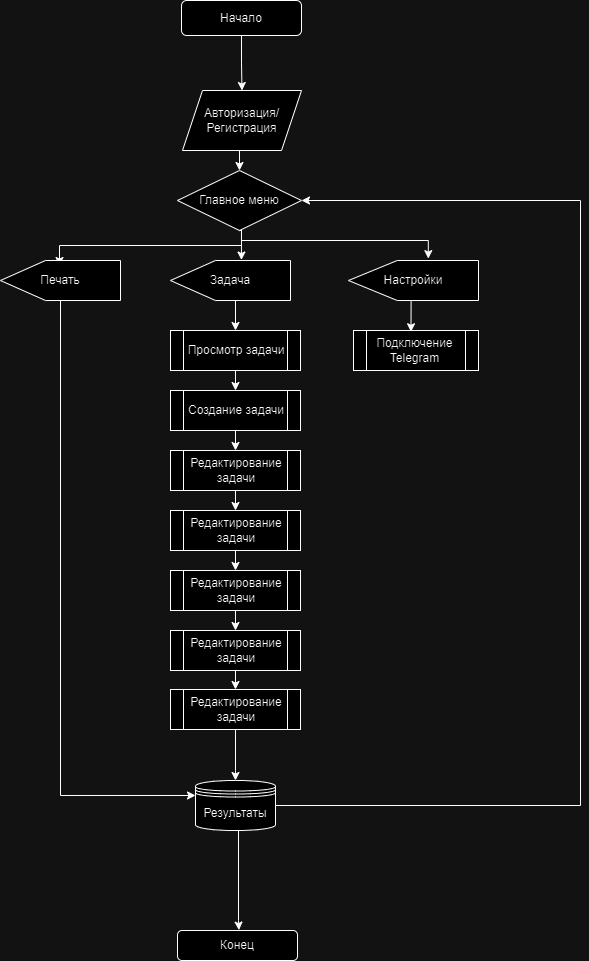


Рисунок 9. Функциональная схема «Планировщика задач»

## 2.3. Разработка программного обеспечения

Для создания приложения «Планировщика задач» будет использоваться язык программирования Python и фреймворк Django.

На рисунке 10 представлена панель управления сайта «Планировщика задач*»*.

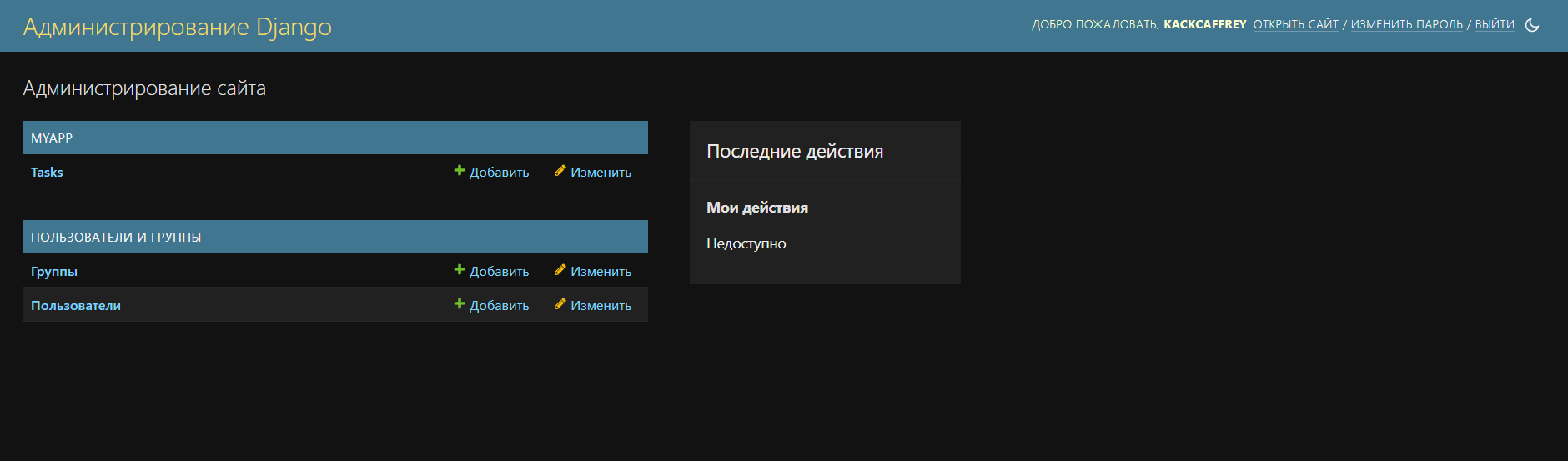


Рисунок 10. Панель управления «Планировщика задач»

В данной панелью управления сайта у нас имеется следующая функциональность доступна выдача прав администратора другим пользователям.

На рисунке 11 представлена страничка авторизации аккаунта

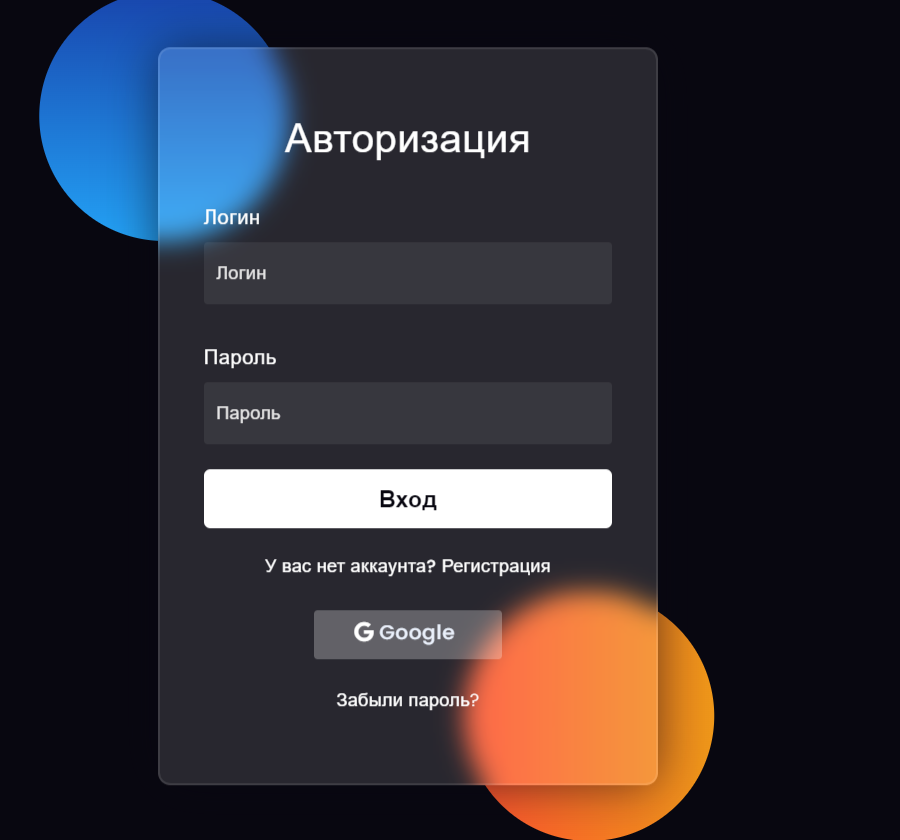


Рисунок 11. Страничка авторизации «Планировщика задач»

На данном рисунке изображена главная страничка авторизации планировщика задач. Отсюда можно перейти на страничку регистрации, авторизоваться по Google аккаунту, либо восстановить пароль от аккаунта.

На рисунке 12 изображена страничка регистрации пользователя.

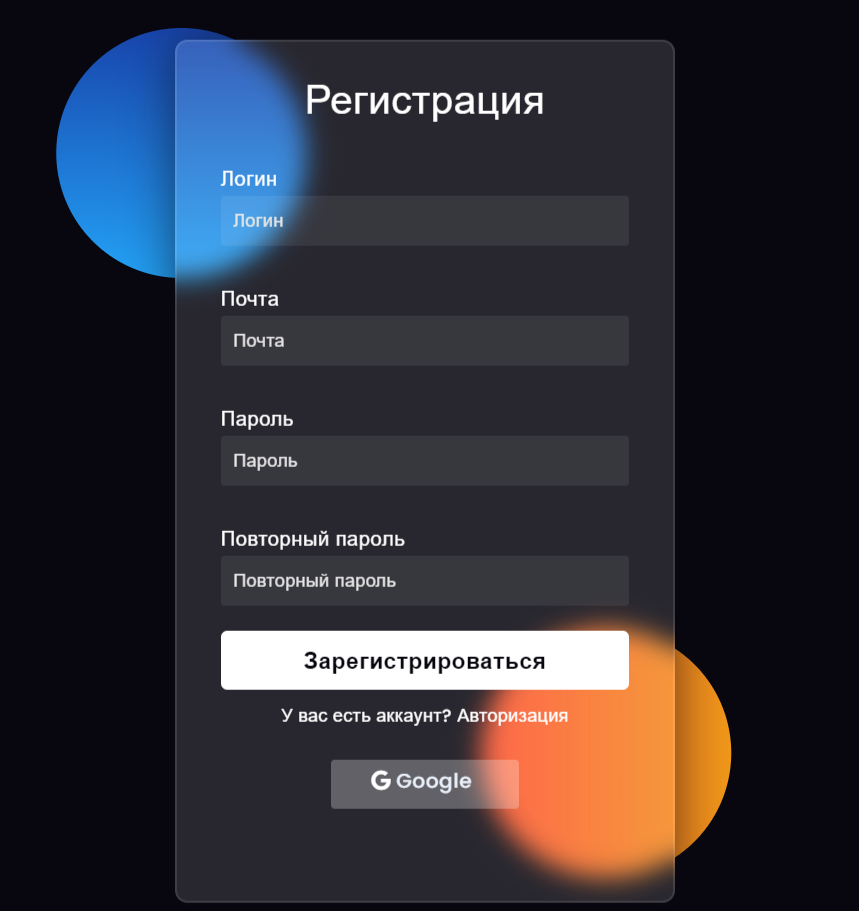


Рисунок 11. Страничка регистрации пользователя

На данном рисунке присутствует форма для регистрации нового пользователя в сервисе. Так же доступ регистрация по Google аккаунту.

На рисунке 12-13 изображен процесс авторизации/регистрации по Google аккаунту.

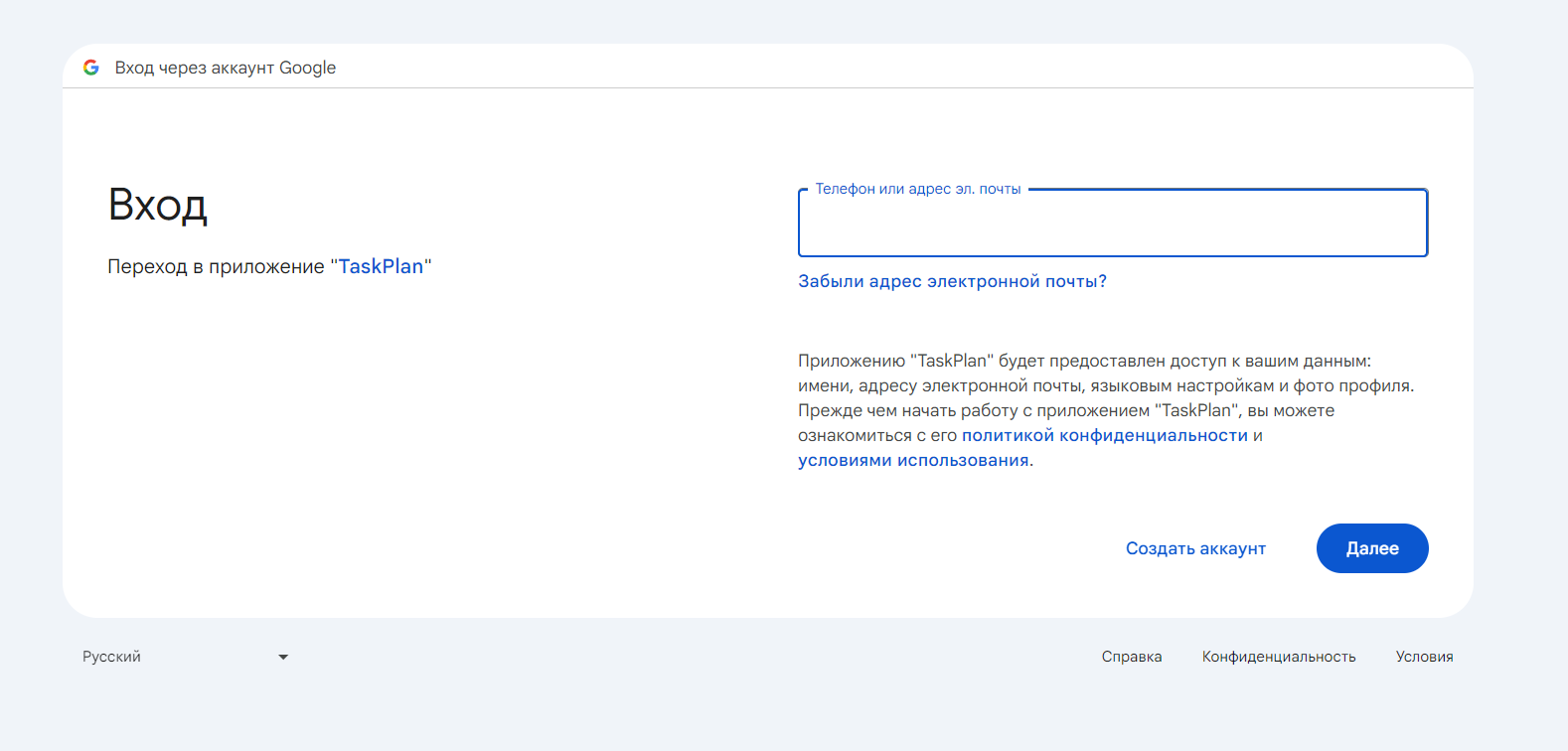


Рисунок 12. Авторизация/Регистрация по Google аккаунту

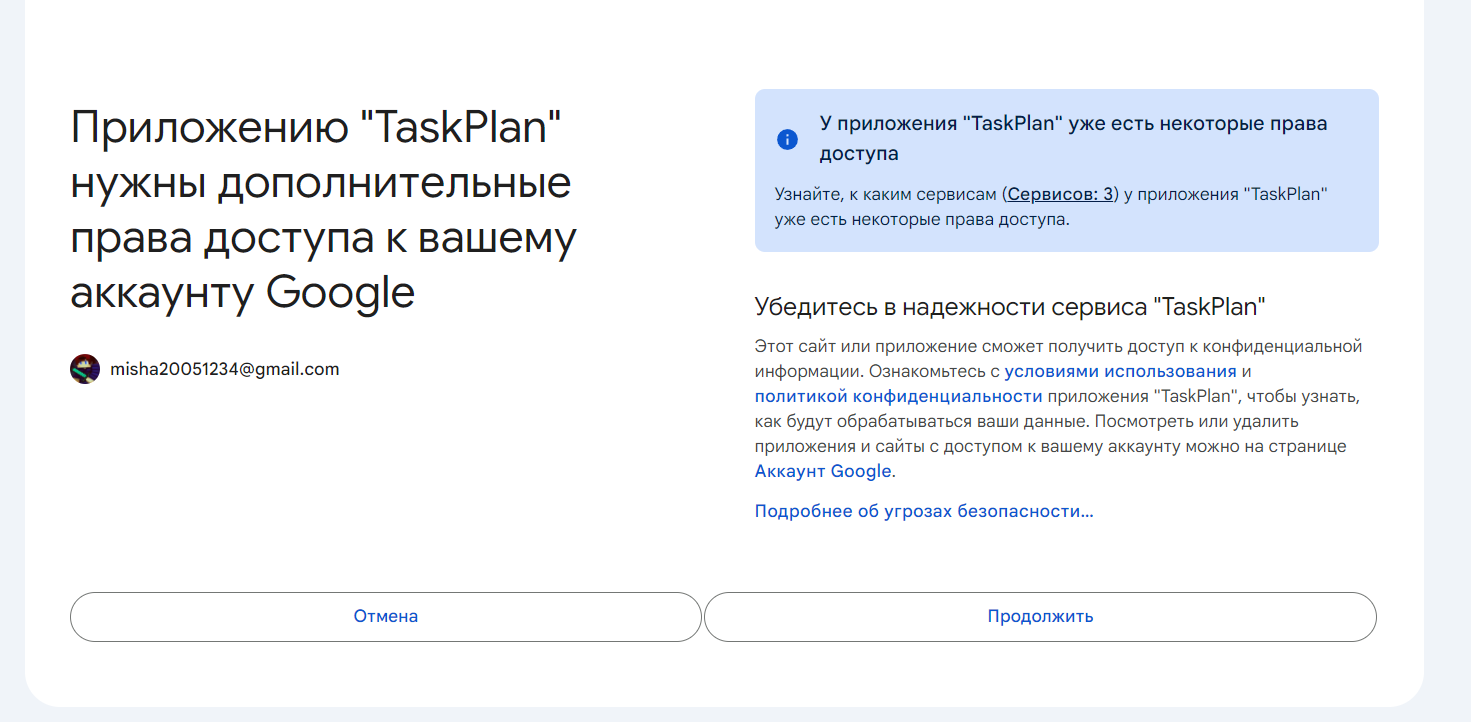


Рисунок 13. Процесс авторизации/регистрации по Google

На страничке авторизации (рис. 10) изображена так же кнопка «Забыли пароль?», при нажатии открывается смена пароля, где сначала запрашивается почта, а после уже идет смена пароля.

На рисунке 14-15 изображена система восстановление пароля

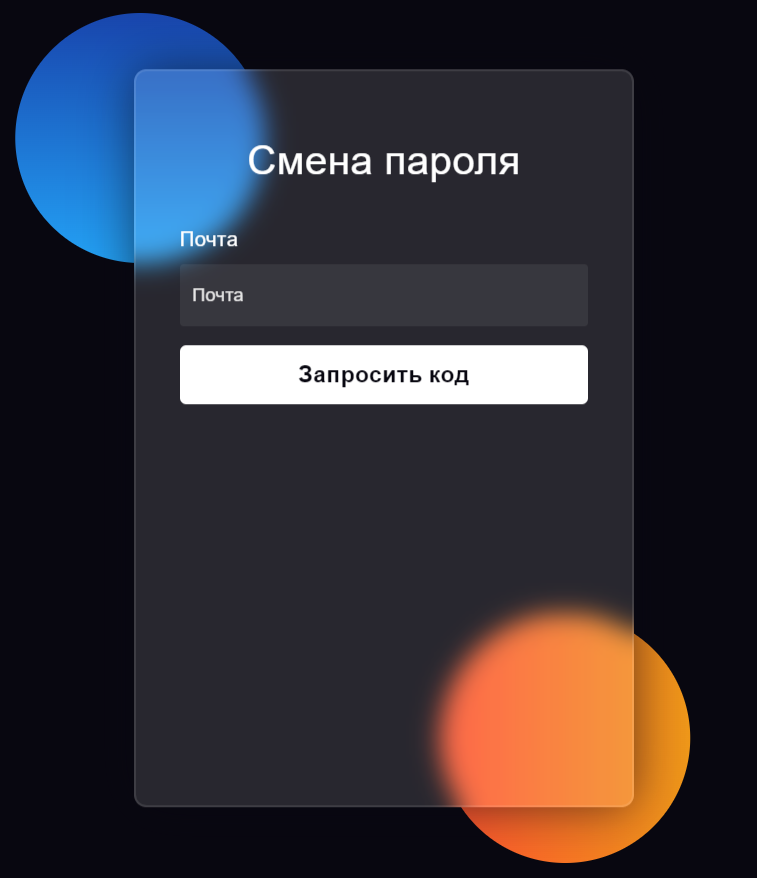


Рисунок 14. Ввод почты и запрос кода с почты

Код реализации отправки письма на почту:

|  |
| --- |
| <?php  if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {      $to = $\_POST['to'];      $subject = $\_POST['subject'];      $message = $\_POST['message'];      $headers = "From: your\_email@example.com";(изменю)      if (mail($to, $subject, $message, $headers)) {          echo json\_encode(['success' => true]);      } else {          echo json\_encode(['success' => false]);      }  } else {      echo json\_encode(['success' => false, 'message' => 'Invalid request method']);  }  ?> |

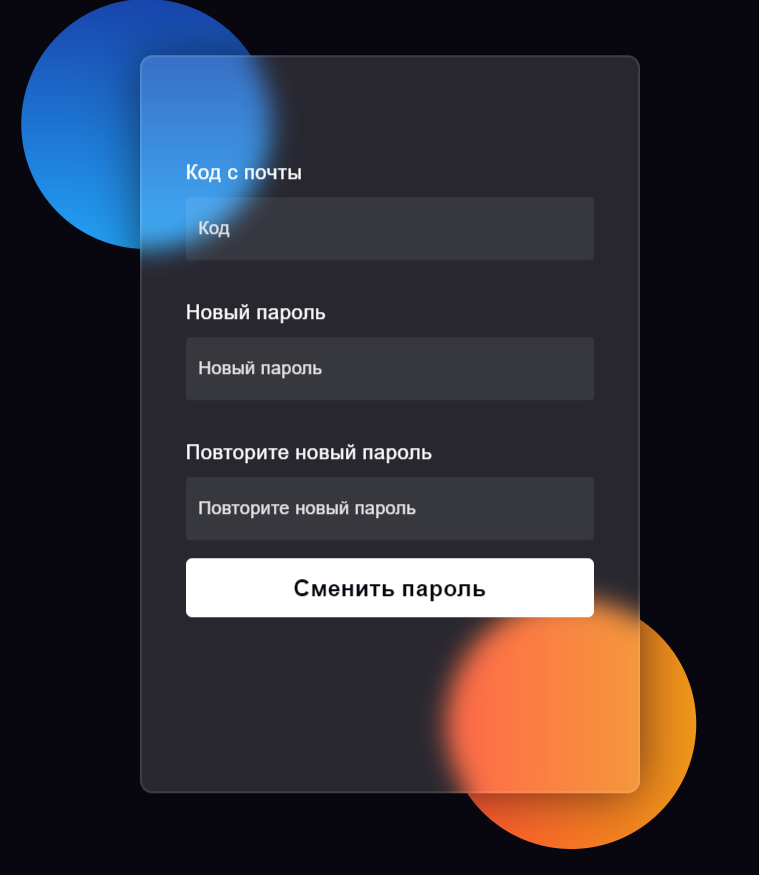


Рисунок 15. Смена пароля после подтверждения почты

После авторизации идет переход на основную страничку сайта, откуда будут уже все изменения отображаться.

На рисунке 16 отображена главная страничка сайта.

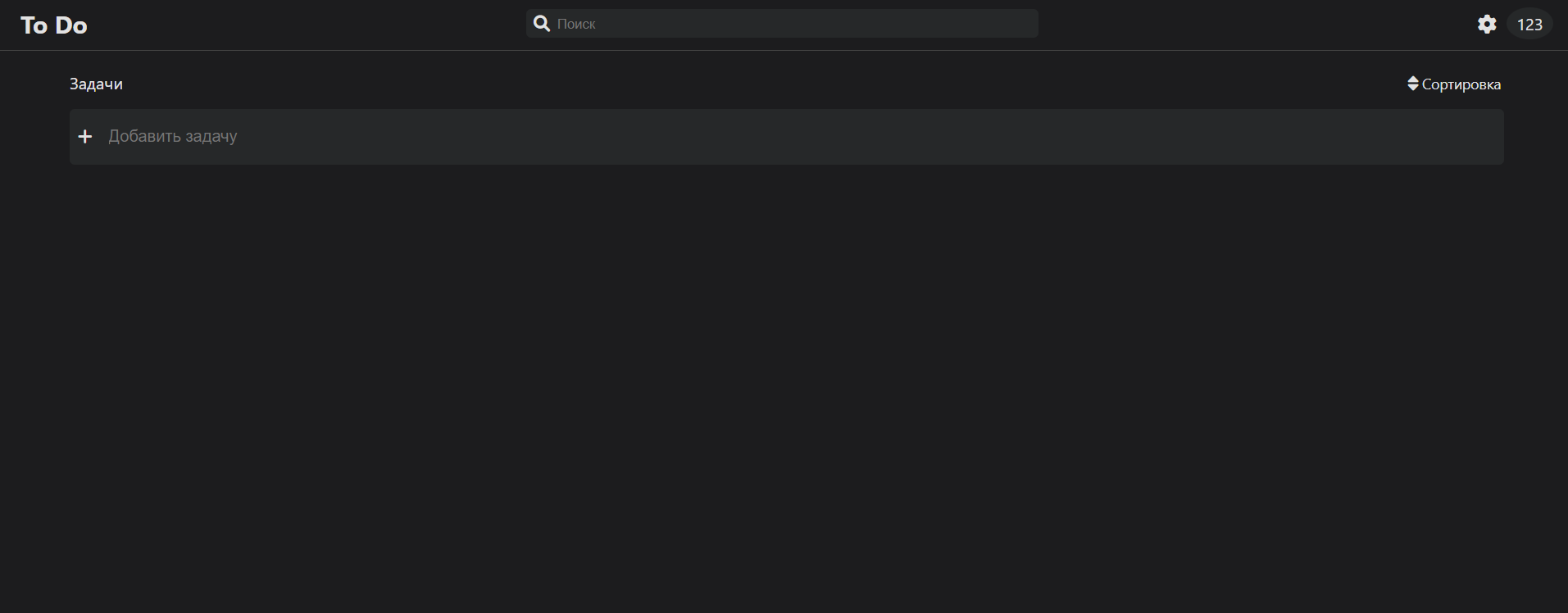


Рисунок 16. Главная страничка сайта

При нажатии на «+», либо при нажатии на строчку «Добавить задачу» происходит раскрытия меню, где можно выбрать дополнительные настройки задач.

На рисунках 17-19 представлены дополнительные опции настройки задач

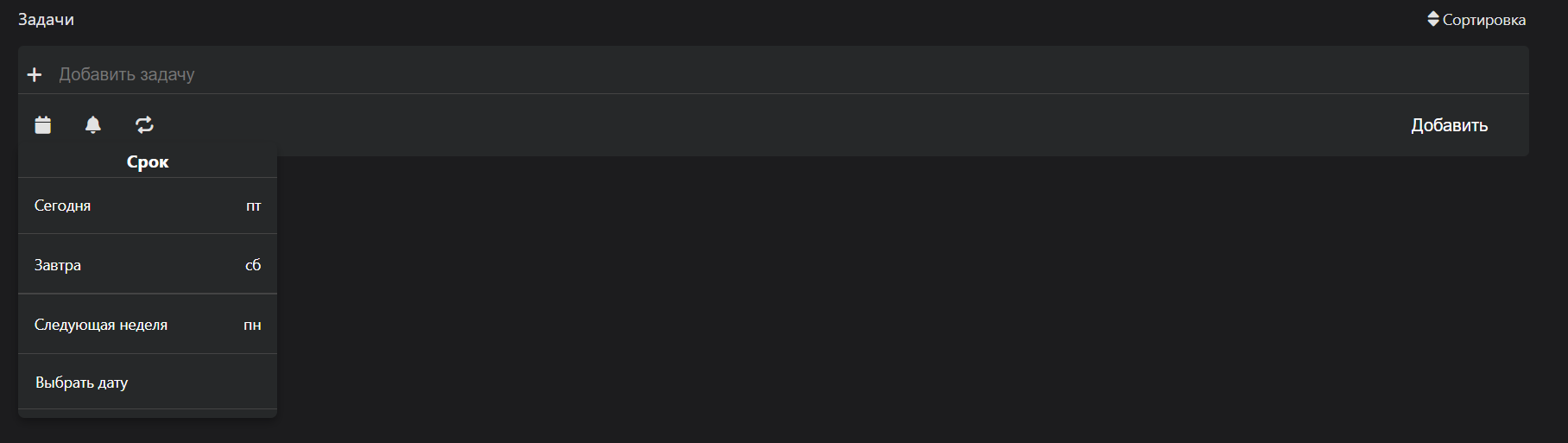


Рисунок 17. Выбор срока задачи

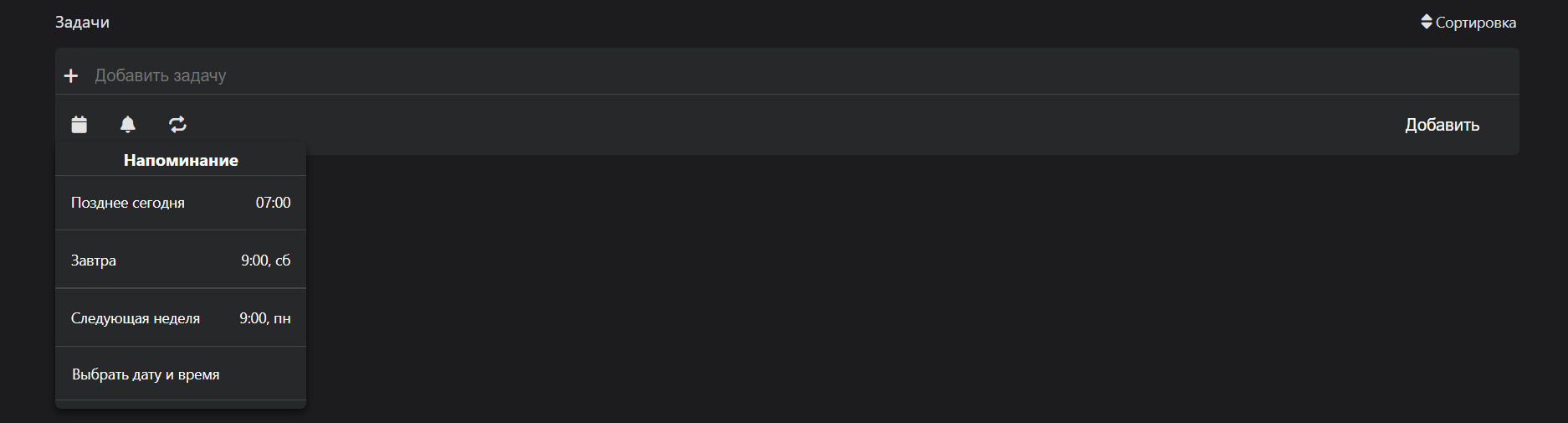


Рисунок 18. Выбор напоминания задачи

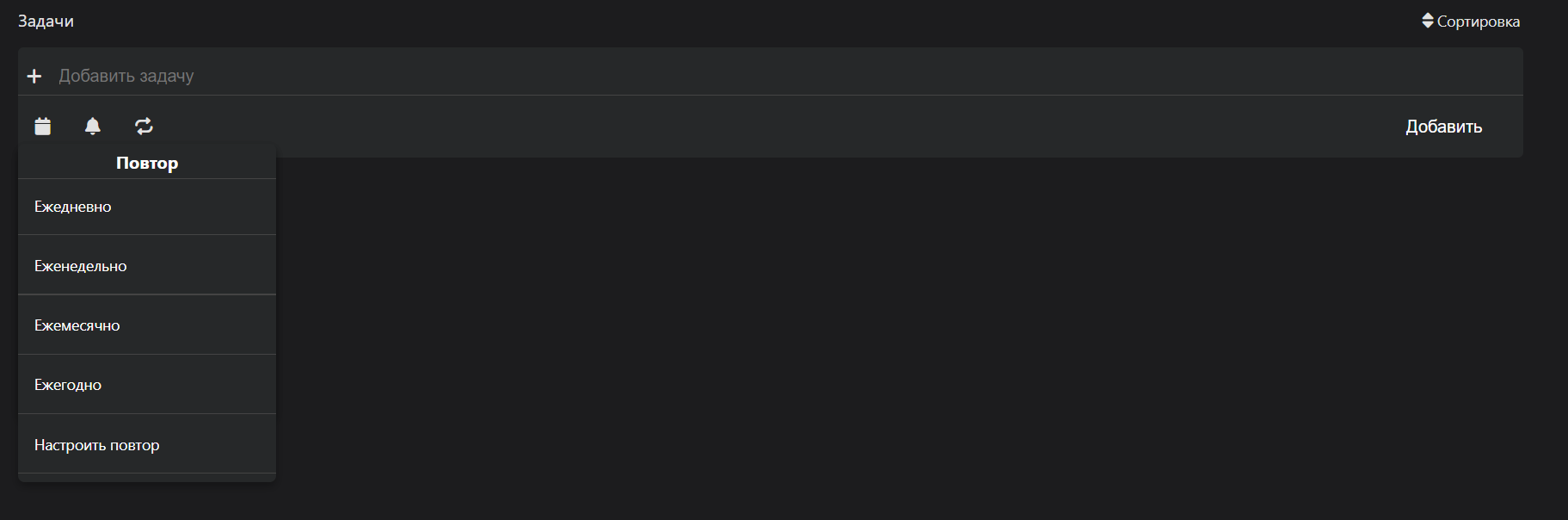


Рисунок 19. Выбор повтора задачи

После выбора любых из действий появляется возможность добавить задачу

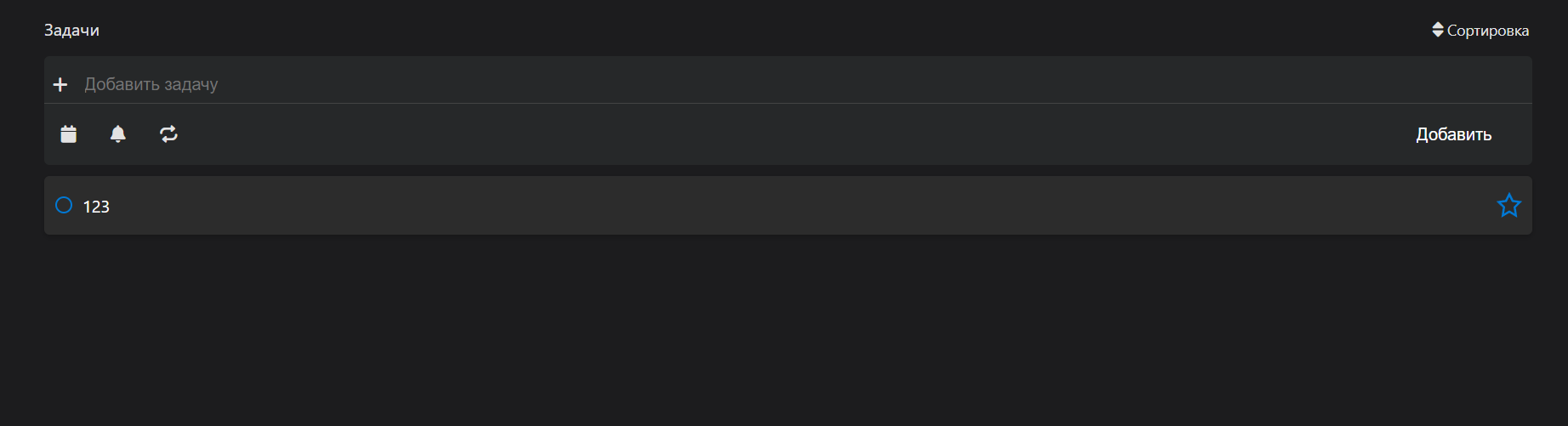


Рисунок 20. Отображение задач

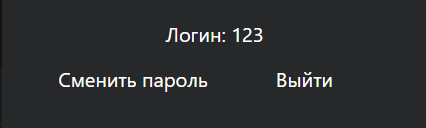


Рисунок 21. Меню профиля

В меню профиля можно перейти в смену пароля, так же виден логин аккаунта и есть кнопка выхода с профиля, с возращением на страничку авторизации.

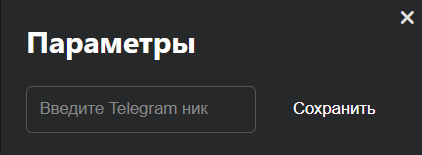


Рисунок 22. Меню настройки для сохранения Telegram аккаунта.

## 2.4. Тестирование программы

На рисунках 20-21 представлено тестирование вкладки «Авторизация»

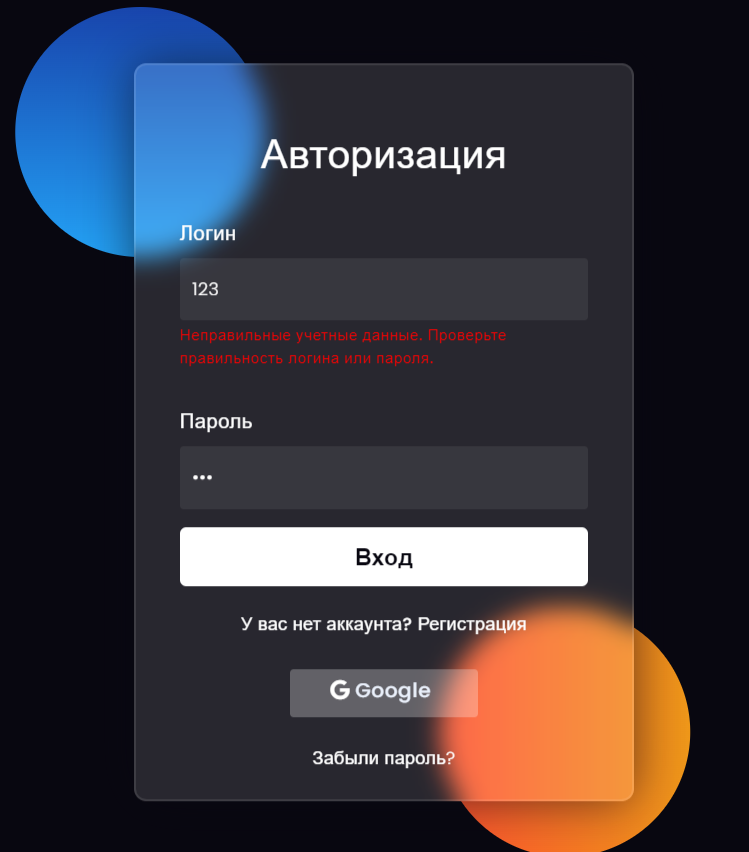


Рисунок 20. Ошибка при неверных данных

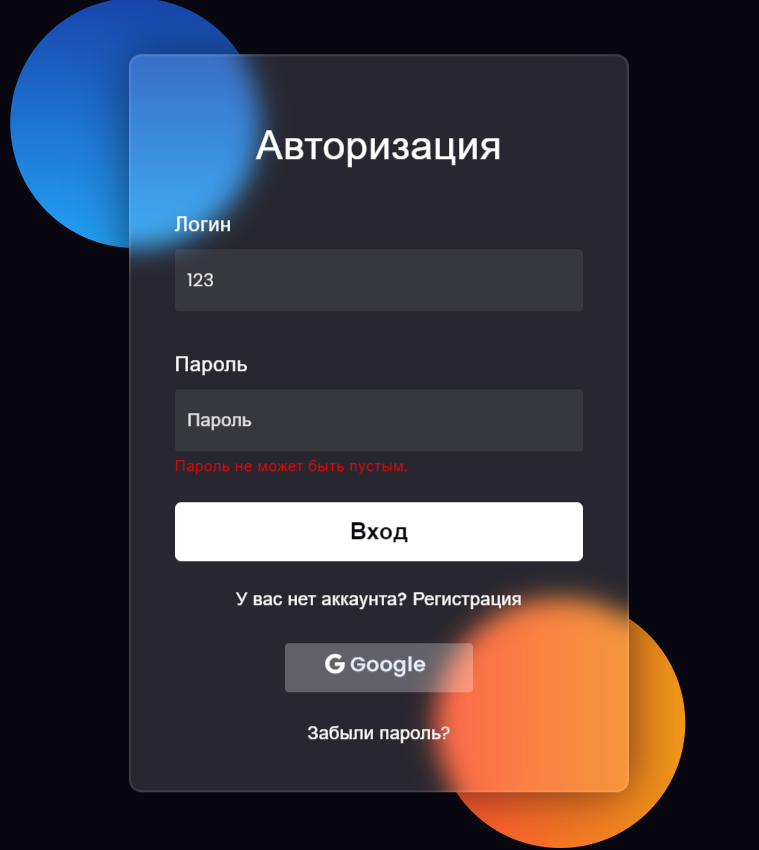


Рисунок 21. Ошибка при пустом пароле

На рисунках 22-26 представлены скриншоты тестирования окна «Регистрация»

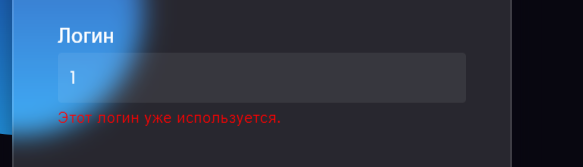


Рисунок 22. Ошибка, что логин уже используется

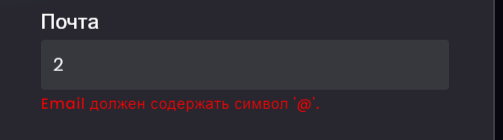


Рисунок 23. Ошибка, что должен соблюдаться форма почты

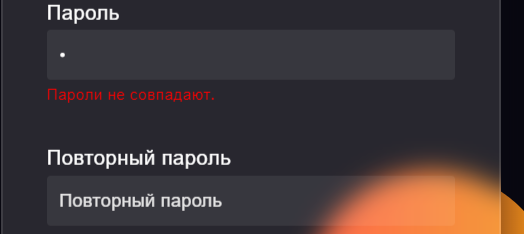


Рисунок 24. Ошибка, что пароли не совпадают

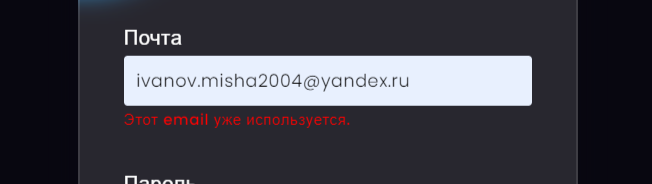


Рисунок 25. Ошибка, что email уже занят

Код реализации проверки, что email уже занят:

|  |
| --- |
| emailInput.addEventListener("blur", function() {          if (emailInput.value.includes('@')) {              fetch(`/check\_email/?email=${emailInput.value}`)                  .then(response => response.json())                  .then(data => {                      const message = data.is\_taken ? "Этот email уже используется." : "";                      displayError(emailError, message);                  });          }      }); |

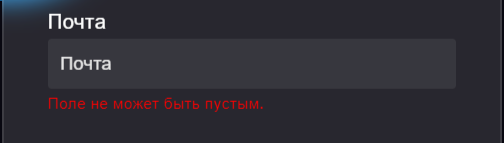


Рисунок 26. Проверка на пустые поля

На рисунках 27-31 представлено тестирование окна «Восстановление пароля»

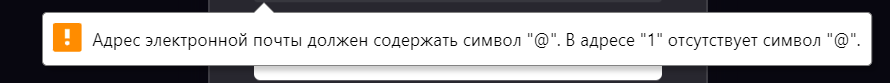


Рисунок 27. Ошибка на отсутствие символа @

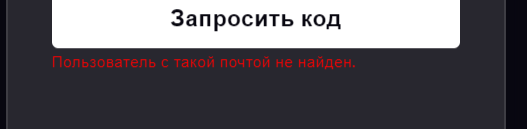


Рисунок 28. Ошибка, что пользователь не найден



Рисунок 29. Ошибка, что введен неполный адрес почты

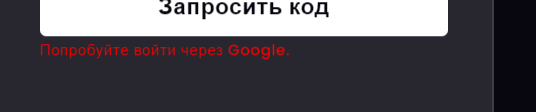


Рисунок 30. Ошибка, если ввести почты, которая совпадает с Google

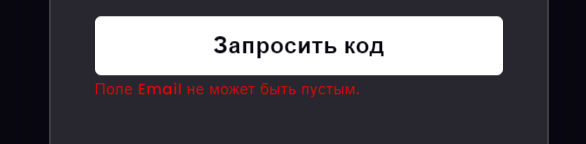


Рисунок 31. Проверка на заполнение Email

На рисунках 32-34 представлено тестирование окна «Смена пароля»

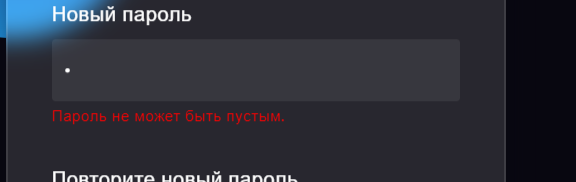


Рисунок 32. Ошибка на пустое поле

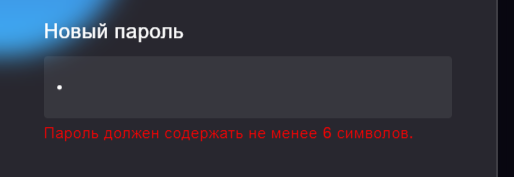


Рисунок 33. Ошибка, что минимальная длина пароля 6

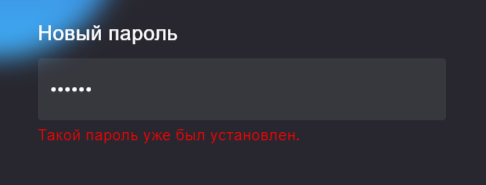


Рисунок 34. Ошибка, что такой пароль уже стоит

## **2.5. Руководство по использованию веб-приложения**

1) Общие Сведения о Программе

Веб-приложение предназначено для планирования задач и управления ими. Пользователи могут создавать, редактировать и удалять задачи, а также устанавливать напоминания и сроки выполнения. Веб-приложение разработано с использованием языка программирования Python и фреймворка Django.

2) Структура Программы

Программа состоит из следующих компонентов:

- главная страница: отображает текущие задачи и взаимодействует с пользователем.

- логика приложения: управляет созданием, редактированием и удалением задач, а также обработкой напоминаний.

- графический интерфейс пользователя обеспечивает удобную навигацию и визуализацию задач.

3) Настройка веб-приложения

Веб-приложение не требует дополнительной настройки, однако вы можете настраивать задачи под свои запросы, а также подключить Telegram.

4) Проверка Программы

Методы проверки:

- выполнить проверки по разным сценариям использования веб-приложения

5) Дополнительные возможности

Дополнительные функциональные возможности отсутствуют в текущей версии веб-приложения.

6) Сообщения пользователю

Сообщения во время использования:

- уведомления по завершению задач

Руководство программиста для веб-приложения планировщик задач

1) Назначение и условия использования

Веб-приложение предназначено для управления задачами и напоминаниями. Оно работает на платформе Windows требует соответствия минимальным системным требованиям Django.

2) Характеристики программы

Интуитивно понятный интерфейс пользователя.

Возможность установки напоминаний и сроков выполнения, а так же количество повторов.

Отдельное оповещение о выполнении задачи.

3) Обращение к программе

Программа запускается через веб-браузер. Пользователь выбирает опции в главном меню, используя мышь или клавиатуру.

4) Входные и выходные данные

Входной информацией для системы является: логин пользователя,

детали задачи (название, описание, срок выполнения, напоминание, повтор)

Выходной информацией для пользователя является: количество задач, статистика по выполненным задачам (количество выполненных, количество невыполненных), разные категории задач.

Руководство оператора для веб-приложения планировщика задач

1) Назначение программы

Веб-приложение предназначено для организации и управления задачами, предоставляя пользователям возможность создавать, редактировать и удалять задачи, а также устанавливать напоминания.

2) Условия выполнения программы

Операционная система: Windows 10

Процессор: 2.0 GHz и выше

Оперативная память: 1 ГБ и выше

Веб-браузер: Совместимый с современными стандартами

3) Выполнение программы

Загрузка и запуск:

- Запуск браузера и переход на URL веб-приложения

- Вход в учетную запись или регистрация.

Начало работы:

- Создание и управление задачами через интерфейс веб-приложения

- Управление задачами

- Использование интерфейса для создания, редактирования, удаления задач и установки напоминаний.

Завершение работы:

- Выход из учетной записи через иконку профиля.

4) Сообщения оператору

Программа выводит сообщения о выполнении или провале задач и другие системные сообщения об ошибках, если они возникают.

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения индивидуального проекта по разработке информационной системы «Планировщик задач»на основе фреймворка Django и языка программирования Python достигнуто успешное выполнение поставленных задач и целей. Проект направлен на создание эффективного инструмента для организации рабочего процесса, управления временными ресурсами и повышения личной продуктивности.

В ходе работы над проектом решены следующие задачи:

Разработка пользовательского интерфейса с использованием технологии Django. Проектирование и реализация привлекательного и интуитивно понятного интерфейса системы, обеспечивающего удобство использования для пользователей.

Написание эффективного программного кода. Создание логики работы информационной системы, включая функции планирования и отслеживания задач, с использованием языка программирования Python.

Механизм сохранения и загрузки данных. Разработка механизма сохранения и загрузки данных о состоянии системы, включая сохранение планов, задач и другой важной информации для последующего восстановления.

Отображение результатов задач. Реализация механизма отображения результатов задач в отдельном окне, обеспечивающего удобство взаимодействия и анализа для пользователей.

Объектом исследования стала информационная система "Планировщик задач", а предметом исследования — средства разработки, включая Django и Python. Исследование проведено с использованием методов анализа литературных источников, изучения образовательных методик и сравнительного анализа аналогичных информационных систем.

Актуальность проекта обоснована не только растущей потребностью в эффективных инструментах управления задачами, но и важностью создания средств, способствующих повышению продуктивности и оптимизации рабочего процесса.

Таким образом, дипломный проект по созданию информационной системы на Django и Python успешно реализован, предоставляя пользователям инновационный инструмент для эффективного управления задачами и содействуя развитию стремления к знаниям и достижению высоких результатов в учебе и работе.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ИНТЕРНЕТ-РЕСУРСОВ**

*Законодательные и нормативные акты:*

1. ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления – 92 с.
2. ГОСТ 7.32-2017 Отчет о научно-исследовательской работе. Структура и правила оформления – 47 с.
3. ГОСТ 7.82-2001 Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления – 39 с.
4. ГОСТ Р 7.0.100-2018 Библиографическая запись. Библиографическое описание. Общие требования и правила составления – 122 с.
5. ГОСТ Р 7.0.12-2011 Библиографическая запись. Сокращение слов и словосочетаний на русском языке. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2004. – 169 с.
6. ГОСТ Р 7.0.5-2008 Библиографическая ссылка. Общие требования и правила составления – 32 с.
7. Единая система программной документации. – М.: Стандартинформ, 2005. – 128 с.

*Учебная и научная литература:*

1. Агальцов, В.П. Математические методы в программировании: учебник / В.П. Агальцов. – 2-е изд., перераб. и доп. – Москва: ИД «ФОРУМ»: ИНФРА-М, 2023. – 240 с.
2. Мюллер Д. П. Си Шарп для чайников – Москва, Санкт-Петербург 2019 ООО "Диалектика" – Текст: электронный. – Режим доступа: <https://vk.com/doc161644846_634870821?hash=y8mbUO5UhKRlCR4tJkTnQsTg7BV4cQQhbNC6FkByWTP&dl=f4zdcFgzbAIpG5Qn0BCz1QINJJ49cQORMlAhF4bX8js> (дата обращения: 24.03.2024).

*Интернет документы:*

1. Аксенова Т.Г. Онлайн-курс по технологии разработки программного обеспечения [Электронный ресурс]. – Режим доступа:<https://classroom.google.com/u/2/w/NTc2MzExNTI0MTY0/t/all> (дата обращения: 24.03.2024).
2. Полное руководство по языку программирования Python и фреймворку Django. – [Электронный ресурс]. – Режим доступа: http://example.com/python-django-tutorial/ (дата обращения: 23.03.2024).
3. Ресурс для сравнительного анализа задачных менеджеров Any.do. [Электронный ресурс]. – Режим доступа: https://www.any.do/ (дата обращения: 23.03.2024).
4. Ресурс для сравнительного анализа задачных менеджеров Microsoft To-Do. [Электронный ресурс]. – Режим доступа: https://to-do.microsoft.com/ (дата обращения: 22.03.2024).
5. Ресурс для сравнительного анализа задачных менеджеров Todoist. [Электронный ресурс]. – Режим доступа: https://todoist.com/ (дата обращения: 21.03.2024).
6. Ресурс для сравнительного анализа задачных менеджеров Trello. [Электронный ресурс]. – Режим доступа: https://trello.com/ (дата обращения: 21.03.2024).
7. Руководство по работе в среде Visual Studio. – [Электронный ресурс]. – Режим доступа: https://docs.microsoft.com/ru-ru/visualstudio/ (дата обращения: 20.03.2024).
8. Форум программистов и сисадминов - [Электронный ресурс]. – Режим доступа: https://www.cyberforum.ru/unity/thread1230990.html (дата обращения: 21.03.2024).
9. Django Python Professional Modern Chat App UI Tutorial. [Электронный ресурс]. – Режим доступа: https://youtu.be/V9DkvcT27WI?si=IZqGPf56ZE-hPwxy (дата обращения: 21.03.2024).
10. Python Django UI | How to Design Chat Messaging App in Django. [Электронный ресурс]. – Режим доступа: https://youtu.be/tUSCm\_t6Ypw?si=6kq\_yLs2WYn6MAt (дата обращения: 21.03.2024).
11. Python Django UI Tutorials: 21 - Custom Dialog System Message Box Popup. [Электронный ресурс]. – Режим доступа: https://youtu.be/jrgT-fbV2tM?si=rpP3KNq5OeZyygz4 (дата обращения: 22.03.2024).

### ПРИЛОЖЕНИЕ

**Приложение №1**

**Словарь данных базы данных**

Ниже представлена таблица 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| Словарь данных | | | |
| auth\_user | | | |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| Первичный | id | Нет | Идентификатор пользователя |
|  | password | Да | Пароль |
|  | last\_login | Нет | Последний вход |
|  | is\_superuser | Нет | Проверка на администратора |
|  | username | Да | Никнейм пользователя |
|  | first\_name | Да | Имя пользователя |
|  | last\_name | Нет | Фамилия (при наличии) |
|  | email | Да | Email пользователя |
|  | is\_google\_account | Да | Регистрация через Google или нет |
| myapp\_notification | | | |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| Первичный | id | Нет | Идентификатор |
|  | message | Да | Сообщение задачи |
|  | is\_read | Да | Прочитано или нет |
|  | is\_created | Да | Когда было создано |
|  | sent\_at | Да | Когда было отправлено |
|  | task\_id | Да | Идентификатор задачи |
|  | user\_id | Да | Идентификатор пользователя |
| myapp\_passwordresetcode | | | |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| Первичный | id | Нет | Идентификатор |
|  | code | Да | Код для восстановления |
|  | created\_at | Да | Дата создания кода |
|  | user\_id | Да | Идентификатор пользователя |
| myapp\_task | | | |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| Первичный | id | Нет | Идентификатор |
|  | title | Нет | Название |
|  | competed | Нет | Выполнена ли задача |
|  | user\_id | Да | Идентификатор пользователя |
|  | created\_at | Да | Дата создания |
|  | due\_date | Нет | Дата окончания |
|  | reminder\_time | Нет | Напоминание |
|  | repeat | Нет | Повторение задачи |
|  | repeat\_interval | Нет | Сколько раз повторение |
|  | text | Да | Содержание задачи |

Продолжение таблицы 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
|  | updated\_at | Да | Дата изменения |
|  | due\_date\_text | Нет | Текст с иконок |
|  | reminder\_time\_text | Нет | Текст с иконок |
|  | mentions | Нет | Напоминание |
| myapp\_telegrambot | | | |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| Первичный | id | Нет | Идентификатор |
|  | telegram\_chat\_id | Да | Идентификатор чата |
|  | subscribed | Да | Включена ли опция |
|  | user-id | Да | Идентификатор пользователя |