Муниципальное казенное общеобразовательное учреждение

средняя общеобразовательная школа №3 п. Михайловка

**Индивидуальный проект**

**«**Интерактивный веб-календарь событий с системой добавления пользовательских праздников**»**

Автор работы: Коротких Иван

ученик 7 класса «А»

Руководитель проекта:

Маркина Н.В., учитель математики, информатики

п. Михайловка, 2025 г.

**Содержание**

[Паспорт проекта 2](#_Toc72678431)

[Введение](#_Toc72678432)

Теоретическое обоснование проекта

Практическое описание проекта

[Перспективы](#_Toc72678456) развития проекта

Список литературы и интернет-источников

Приложения

**Паспорт проекта**

1. **Название проекта**

Интерактивный веб-календарь событий с системой добавления пользовательских праздников

1. **Проектная группа**

Коротких Иван

1. **Аннотация**

Проект представляет собой интерактивный веб-календарь событий, разработанный на языке программирования Python с использованием фреймворка **Streamlit**. Календарь позволяет пользователям просматривать события, добавлять новые мероприятия, а также создавать и управлять своими собственными праздниками.

Основные функции включают:

* Просмотр календаря с отображением текущих событий и праздников.
* Добавление новых мероприятий с указанием даты, времени и названия праздника.
* Создание пользовательских праздников, которые будут видны только автору.
* Управление созданными событиями: редактирование, удаление.

Проект предназначен для упрощения планирования личных и профессиональных дел, а также для удобства пользователей, желающих создать уникальные праздники для себя или своей группы.

Веб-приложение на Python, предоставляющее функционал интерактивного календаря с возможностью добавления и управления личными праздниками и событиями. Система включает личный кабинет пользователя, календарь событий и административную панель для управления общественными праздниками.

1. **Проблема**

Многие люди используют бумажные календари или электронные приложения для отслеживания важных дат и событий. Однако такие решения часто ограничены в функциональности и не позволяют легко добавлять собственные праздники или делиться ими с друзьями. В результате, пользователи сталкиваются с трудностями при планировании и организации своих мероприятий.

Наш проект предлагает пример пути решение этой проблемы путем создания интерактивного веб-календаря, который позволит пользователям не только просматривать стандартные праздники, но и добавлять свои собственные события, управлять ими, к сожалению, делиться с другими людьми календарём пока нельзя, требуются доработки.

1. **Актуальность**

В современном мире, когда информация становится всё более доступной и важной, умение эффективно планировать своё время и отслеживать важные события играет ключевую роль. Школьники, особенно в возрасте 12–13 лет (7 класс), начинают сталкиваться с большим количеством учебных заданий, внеклассных мероприятий и личных дел. Использование традиционных методов планирования, таких как бумажные календари или простые электронные приложения, уже не всегда удовлетворяет потребности растущего числа пользователей.

Наш проект "Интерактивный веб-календарь событий с системой добавления пользовательских праздников" предлагает один из способов решения этой проблемы, предоставляя более-менее удобный инструмент для управления временем и событиями. Он позволит школьникам лучше организовывать свою учебную и личную жизнь, а также даёт возможность создавать свои уникальные праздники. Это делает процесс планирования более гибким и интересным, что способствует развитию ответственности и самостоятельности у подростков. Календарь позволяет объединить различные типы календарей (личных, рабочих, образовательных) в едином интерфейсе.

1. **Цель проекта**

Цель нашего проекта — создание удобного и простого в использовании веб-календаря, который поможет школьникам и не только им эффективнее планировать своё время, добавлять личные события и праздники, а также управлять ими самостоятельно, вести учет личных и общественных праздников, планировать важные события.

1. **Задачи**

Для достижения цели проекта необходимо решить следующие задачи:

1. **Изучение основ языка Python**: Ознакомиться с базовыми конструкциями языка, такими как переменные, циклы, условия и функции.
2. **Разработка структуры календаря**: Определить основные элементы календаря (даты, месяцы, годы) и способы их представления в программе.
3. **Создание интерфейса**: Разработать простой графический интерфейс, используя библиотеки вроде stremlit, чтобы сделать календарь удобным для пользователя.
4. **Функционал добавления событий**: Написать код, который позволяет пользователю вводить данные о событии (название, дата) и сохранять их.
5. **Хранение данных**: Научиться использовать файлы или базы данных (например, SQLite) для сохранения введённых событий и праздников.
6. **Отображение событий**: Программировать вывод событий на экран в виде списка или календарной сетки.
7. **Редактирование и удаление событий**: Реализовать функции для изменения и удаления ранее добавленных событий.
8. **Тестирование программы**: Проверять работоспособность всех функций календаря и исправлять ошибки.
9. **Документирование проекта**: Составить инструкцию по использованию программы и описать основные этапы разработки.
10. **Презентация проекта**: Подготовить демонстрацию работы календаря перед одноклассниками.
11. **Ожидаемые результаты**

**Готовая программа**: Создание рабочей версии интерактивного веб-календаря, написанного на языке Python, с использованием подходящих библиотек и инструментов.

**Удобный интерфейс**: Программа должна иметь интуитивно понятный интерфейс, позволяющий пользователю легко ориентироваться и взаимодействовать с календарём.

**Функциональность добавления событий**: Пользователь сможет добавлять, редактировать и удалять события, указывая дату, время события.

**Система пользовательских праздников**: Реализована возможность создания и управления уникальными праздниками, видимыми только владельцу аккаунта.

**Локальное хранение данных**: Данные о событиях и праздниках сохраняются локально на устройстве пользователя, обеспечивая безопасность и конфиденциальность информации.

**Документация и инструкция**: Подготовка подробной документации и инструкции по установке и использованию программы, а также описание основных этапов разработки.

**Учебный эффект**: Ученик приобретёт практические навыки программирования на языке Python, улучшит понимание алгоритмов и структур данных, а также научится разрабатывать пользовательские интерфейсы.

**Демонстрационная версия**: Готовая программа будет представлена в виде демонстрационной версии, которую можно показать всем заинтересованным, демонстрируя функциональность и удобство использования.

**Одержать победу на конкурсе проектов**

1. **Технические характеристики**

**Используемые технологии:**

Python 3.x — основной язык программирования.

stremlit — библиотека для создания графического интерфейса.

**Функциональные возможности:**

1. Просмотр событий:

* Отображение текущего месяца с днями.
* Вывод списка событий на выбранную дату.
* Добавление событий:
* Возможность ввода названия события, даты, времени и описания.
* Сохранение событий в базу данных.

1. Редактирование и удаление событий:

* Изменение информации о существующем событии.
* Удаление ненужных событий.

1. Создание пользовательских праздников:

* Возможность добавить уникальный праздник с произвольной датой.
* Праздники видны только владельцу календаря.

1. Интерфейс:

* Простой и интуитивно понятный дизайн.
* Удобная навигация между месяцами и годами.

**Аппаратные требования:**

* Компьютер с операционной системой Windows, macOS или Linux.
* Минимум 512 МБ оперативной памяти.
* Процессор с частотой от 1 ГГц.

**Программные требования:**

* Установленный интерпретатор Python 3.x.
* Наличие библиотек stremlit (обычно предустановлены в стандартных дистрибутивах Python).

**Ограничения:**

* Программа работает только на одном компьютере, где она установлена.
* Нет возможности делиться событиями или праздниками с другими пользователями.

1. **Сроки реализации**

Данный проект был выполнен за одну неделю.

# Введение

В современном мире управление временем стало важным аспектом нашей жизни. Мы постоянно сталкиваемся с необходимостью планировать встречи, мероприятия, учёбу и другие дела. Бумажные календари постепенно уступают место электронным приложениям, которые предлагают больше возможностей для организации времени. Однако многие существующие решения либо сложны в использовании, либо не предоставляют достаточно свободы для настройки под индивидуальные нужды.

Именно поэтому я решил попробовать разработать интерактивный веб-календарь событий, который позволит пользователям удобно управлять своим временем, добавлять события и праздники, а также создавать собственные уникальные праздники. Этот проект выполнен на языке программирования Python, который является одним из самых популярных и простых для изучения языков среди новичков.

Моей целью было создать простое, но эффективное приложение, которое можно использовать для повседневного планирования. Я старался сделать интерфейс максимально понятным и доступным даже для тех, кто только начинает знакомиться с миром программирования. Проект включает в себя основные функции календаря, такие как просмотр событий, добавление новых дат и управление ими.

Таким образом, этот проект не только полезен сам по себе, я надеюсь, но и служит отличным примером того, как можно начать изучать программирование и создавать полезные приложения с нуля.

**Актуальность** работы заключается в том, что заинтересованные люди получают простой и удобный веб-календарь на Python, позволяющий легко добавлять и управлять событиями, а также создавать свои праздники.

**Целью проекта** является создание простого и удобного веб-календарь на Python, который поможет пользователям планировать события и добавлять свои праздники, улучшая организацию времени.

**Задачи проекта**: изучить основы Python, разработать интерфейс календаря, реализовать функции добавления, редактирования и удаления событий, а также создать систему пользовательских праздников.

**Продукт проекта** — это простой веб-календарь, написанный на Python, с функциями добавления, редактирования и удаления событий, а также возможностью создания пользовательских праздников.

**Объект исследования** — процессы планирования и организации времени с помощью электронных календарей, а также разработка программного обеспечения для этих целей.

**Предмет исследования** — методы и технологии создания веб-календаря на языке Python, обеспечивающие удобное добавление и управление событиями, включая пользовательские праздники.

**В работе использованы следующие методы:** изучение информации в интернете, методы объектно-ориентированного программирования на Python, а также подходы к разработке графического интерфейса с помощью библиотеки stremlit

**Теоретическое обоснование проекта**

При создании интерактивного веб-календаря событий важно учитывать несколько ключевых аспектов, связанных с управлением временем и организацией информации. В основе любого календаря лежит структура данных, позволяющая хранить и обрабатывать даты, события и повторяющиеся праздники. Для реализации такой структуры можно использовать различные подходы, начиная от простых списков и заканчивая реляционными базами данных.

**Сбор информации**

На этапе сбора информации важно определить основные функциональные требования к проекту. Это включает:

1. **Анализ существующих решений**. Изучение популярных приложений-календарей, таких как Google Calendar, Microsoft Outlook и др., чтобы понять их сильные и слабые стороны.
2. **Определение потребностей целевой аудитории**. Поскольку проект выполняется учеником 7-го класса, важно учесть особенности возраста и уровня подготовки. Например, интерфейс должен быть простым и интуитивно понятным, а функционал — достаточным для базовых нужд, таких как добавление школьных мероприятий, дней рождения друзей и семейных праздников.
3. **Выбор технологий**. Python был выбран благодаря своей простоте и доступности для начинающих программистов.

**Анализ информации**

После сбора информации необходимо провести её анализ, чтобы выделить ключевые компоненты будущего проекта:

1. **Структура данных**. Определение формата хранения событий и праздников. Важно предусмотреть поля для даты, времени, названия события.
2. **Интерфейс пользователя**. Разработка макета окна приложения, расположение кнопок и полей ввода. Интерфейс должен быть минималистичным, чтобы не перегружать пользователя лишней информацией.
3. **Алгоритмы обработки данных**. Нужно продумать, как будут обрабатываться запросы на добавление, изменение и удаление событий. Например, проверка корректности вводимой даты или уведомление о совпадении событий.
4. **Безопасность и приватность**. Хотя проект выполняется для личного использования, важно предусмотреть базовые меры защиты данных, такие как шифрование паролей или ограничение доступа к файлам базы данных.

**Практическое описание проекта**

Чтобы запустить этот код на своем компьютере, вам нужно сначала установить Python, а затем необходимые библиотеки для работы с приложением на Streamlit. Вот шаги, которые вам нужно выполнить:

### Шаг 1: Установка Python

1. **Скачайте Python**:
   * Перейдите на [официальный сайт Python](https://www.python.org/downloads/" \t "_new).
   * Скачайте последнюю стабильную версию Python (рекомендуется версия 3.8 или выше).
2. **Установите Python**:
   * Запустите установочный файл и следуйте инструкциям. Убедитесь, что галочка "Add Python to PATH" выбрана при установке.

### Шаг 2: Установка необходимых библиотек

1. **Открыть командную строку**:
   * На Windows: Нажмите Win + R, введите cmd и нажмите Enter.
   * На macOS: Откройте приложение "Terminal".
   * На Linux: Откройте терминал.
2. **Обновите пакетный менеджер pip**:

Введите в командную строку

python -m pip install --upgrade pip

1. **Установите Streamlit и другие необходимые библиотеки**: Введите следующую команду:

pip install streamlit pillow

### Шаг 3: Создание файла с кодом

1. **Создайте новый файл**:
   * Создайте новый текстовый файл с расширением .py, например calendar\_app.py.
2. **Скопируйте код**:
   * Скопируйте код, который вы предоставили, и вставьте его в файл calendar\_app.py.

### Шаг 4: Запуск приложения

1. **Перейдите в директорию, где находится ваш файл**: используйте команду cd для перехода в директорию. Например:

cd путь\к\вашему\каталогу

1. **Запустите приложение Streamlit**: Введите следующую команду:

streamlit run calendar\_app.py

1. **Откройте приложение в браузере**: После выполнения команды вы увидите сообщение о запуске сервера, обычно с адресом http://localhost:8501. Откройте этот адрес в вашем веб-браузере.

### Шаг 5: Использование приложения

Теперь вы можете использовать приложение, добавлять праздники, создавать расписание и многое другое.

### Объяснение кода

* + Импорт библиотек: Импортируются библиотеки для работы с Streamlit, датами и изображениями.
  + Словарь праздников: Содержит названия и даты праздников.
  + Функции:
  + display\_calendar(): Отображает ближайшие праздники.
  + add\_holiday(): Позволяет пользователю добавить свой праздник.
  + schedule\_events(): Позволяет создать расписание мероприятий с уведомлениями.
  + Основная часть приложения: Выводит заголовок и вызывает функции для отображения календаря и добавления мероприятий.

### Обратите внимание

Если вы хотите расширить функциональность, например, добавить уведомления на основе времени, вам потребуется дополнительная логика и, возможно, использование сторонних библиотек.

Давайте разберем каждый элемент представленного кода, написанного на Python с использованием библиотеки Streamlit для создания веб-приложения, которое функционирует как календарь праздников и планировщик мероприятий.

import streamlit as st from datetime import datetime, timedelta from PIL import Image import os

1. **Импортируем библиотеки:**
   * streamlit — библиотека для создания веб-приложений на Python.
   * datetime и timedelta — модули для работы с датами и временем.
   * PIL (Python Imaging Library) — библиотека для обработки изображений (она здесь не используется, но импортирована).
   * os — модуль для взаимодействия с операционной системой (также импортирован, но не используется).

# Создаем список праздников holidays = { "Новогодние праздники": datetime(2023, 12, 31), "Рождество": datetime(2024, 1, 7), "Международный женский день": datetime(2024, 3, 8), }

1. **Определяем словарь holidays:**
   * Словарь, в котором ключами являются названия праздников, а значениями — соответствующие даты, заданные с помощью datetime.

# Функция для отображения календаря с праздниками def display\_calendar(): st.header("Календарь праздников") current\_date = datetime.now()

1. **Объявляем функцию display\_calendar():**
   * Заголовок для раздела с календарем праздников устанавливается с помощью st.header().
   * current\_date получает текущее время и дату.

for holiday, date in holidays.items(): if date.date() >= current\_date.date(): st.markdown(f"\*\*{holiday} :\*\* {date.strftime('%d-%m-%Y')}")

1. **Итерируем по праздникам:**
   * Перебираем элементы в словаре holidays.
   * Проверяем, что дата праздника не меньше текущей даты.
   * Если дата удовлетворяет условию, отображаем название праздника и его дату в формате "дд-мм-гггг".

if st.button('☀️', key=str(date)): st.success(f"Вы нажали на иконку праздника: {holiday}")

1. **Добавляем кнопку для праздника:**
   * Создаем кнопку с иконкой солнца (☀️).
   * Если пользователь нажимает на кнопку, показываем сообщение об успехе.

# Функция для добавления собственных праздников def add\_holiday(): st.header("Добавить свой праздник") holiday\_name = st.text\_input("Введите название праздника") holiday\_date = st.date\_input("Выберите дату праздника", value=datetime.now())

1. **Функция для добавления праздников:**
   * Заголовок для раздела добавления праздников.
   * holiday\_name получает ввод названия праздника.
   * holiday\_date — ввод даты с предустановленным значением текущей даты.

if st.button("Добавить"): if holiday\_name and holiday\_date: holidays[holiday\_name] = holiday\_date st.success(f"{holiday\_name} добавлен на {holiday\_date.strftime('%d-%m-%Y')}") else: st.error("Пожалуйста, заполните все поля")

1. **Кнопка для добавления:**
   * Если кнопка "Добавить" нажата, проверяем, заполнены ли поля.
   * Если да, добавляем новый праздник в словарь и показываем сообщение об успешном добавлении.
   * Если нет, выводим сообщение об ошибке.

# Функция для создания расписания с уведомлениями def schedule\_events(): st.header("Создать расписание мероприятий") event\_name = st.text\_input("Введите название мероприятия") event\_date = st.date\_input("Выберите дату мероприятия", value=datetime.now()) notify\_time = st.time\_input("Выберите время уведомления", value=datetime.now().time())

1. **Функция для создания расписания:**
   * Заголовок для раздела с расписанием мероприятий.
   * Получаем ввод названия мероприятия, даты и времени уведомления.

if st.button("Добавить мероприятие"): if event\_name: st.success(f"{event\_name} добавлено на {event\_date} в {notify\_time}. Уведомление будет отправлено.") # Здесь можно добавить логику для отправки уведомлений (например, с помощью библиотеки

**Перспективы развития проекта**

1. **Расширение функционала**: Добавление возможности уведомления пользователей о предстоящих событиях через email или SMS, интеграция с популярными календарями (Google Calendar, Яндекс.Календарь и др.).
2. **Улучшение интерфейса**: Внедрение более современного дизайна, поддержка различных цветовых схем и тем оформления.
3. **Поддержка многопользовательской среды**: Разработка механизма совместного использования календаря несколькими пользователями, возможность обмена событиями и создания общих праздников.
4. **Мобильная версия**: Адаптация приложения для мобильных устройств, создание нативных приложений для iOS и Android.
5. **Интеграция с социальными сетями**: Возможность публикации событий в социальных сетях и приглашение друзей на мероприятия.
6. **Аналитические инструменты**: Введение статистики использования календаря, анализ частоты и типов добавляемых событий.

Эти перспективы позволят превратить начальный проект в полноценное многофункциональное приложение, востребованное широким кругом пользователей.

# Заключение

В ходе выполнения проекта "Интерактивный веб-календарь событий с системой добавления пользовательских праздников" была создана простая, но функциональная программа на языке Python. Этот проект позволил мне углубить знания в области программирования, освоить работу с графическим интерфейсом и базами данных, а также развить навыки самостоятельного решения задач.

Разработанное приложение обладает основными функциями, необходимыми для эффективного планирования времени: добавлением событий, редактированием и удалением записей, а также созданием индивидуальных праздников. Несмотря на простоту интерфейса, программа удобна в использовании и подходит для повседневной организации личных и учебных дел.

Кроме того, работа над проектом дала мне возможность познакомиться с процессом разработки программного обеспечения от идеи до готового продукта. Я изучил теоретические аспекты создания приложений, проанализировал существующие решения и применил полученные знания на практике.

Данный проект имеет хорошие перспективы для дальнейшего развития. В будущем можно расширить функционал, добавив уведомления, интеграцию с мобильными устройствами и поддержку многопользовательского режима. Это сделает программу ещё более полезной и востребованной.

Таким образом, выполнение данного проекта стало важным этапом в моем обучении программированию и позволило получить ценные навыки, которые пригодятся в дальнейшем развитии как программиста.

# 

# Литература и Интернет-ресурсы:

1. Лутц М. Изучаем Python. — СПб.: Символ-Плюс, 2011.
2. Дейтел Х.М., Дейтел П.Дж. Как программировать на Python. — М.: Бином, 2018.
3. Дронов В.А. Программирование на Python для начинающих. — М.: Эксмо, 2019.
4. Официальная документация Python: https://docs.python.org/3/
5. Документация по библиотеке Tkinter: https://docs.python.org/3/library/tkinter.html
6. Учебник по SQLite: https://www.sqlitetutorial.net/
7. Статья "Основы работы с Tkinter" на сайте Habrahabr: https://habr.com/ru/post/254773/
8. Руководство по созданию простого календаря на Python: https://python-scripts.com/create-a-calendar-with-python