МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

**«Челябинский государственный университет»**

**(ФГБОУ ВО «ЧелГУ»)**

Математический факультет

Кафедра вычислительной механики и информационных технологий

**КУРСОВАЯ РАБОТА**

Автоматизация отчетности интернет магазина

Выполнил студент:

Сумкин Савелий Андреевич

Группы МТ-201

очной формы обучения

направления подготовки

02.03.02 - Фундаментальная информатика и информационные технологии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

Научный руководитель:

Должность:

Ученая степень:

Ученое звание:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

Челябинск

2024

# **Оглавление**

[**Введение** 1](#_Toc167908070)

[**1.1 Обоснование актуальности темы** 1](#_Toc167908071)

[**1.2 Цель, задачи и методы исследования** 2](#_Toc167908072)

[**1.3 Практическая значимость и структура работы** 2](#_Toc167908073)

[**2. Теоретическое и методологическое обоснование** 4](#_Toc167908074)

[**2.1 Автоматизация отчетности: понятие и основные подходы** 4](#_Toc167908075)

[**2.2 Технологии и инструменты для автоматизации отчетности в интернет-магазинах** 4](#_Toc167908076)

[**2.3 Преимущества и сложности автоматизации отчетности** 5](#_Toc167908077)

[**3. Практическая реализация автоматизации отчетности интернет-магазина** 7](#_Toc167908078)

[**3.1 Описание системы автоматизации отчетности** 7](#_Toc167908079)

[**3.1.1 Архитектура системы** 7](#_Toc167908080)

[**3.1.2 Используемые технологии и инструменты** 7](#_Toc167908081)

[**3.2 Разработка и внедрение телеграмм-бота для автоматизации отчетности** 8](#_Toc167908082)

[**3.2.1 Постановка задачи** 8](#_Toc167908083)

[**3.2.2 Описание функционала бота** 8](#_Toc167908084)

[**3.2.3 Используемый стек технологий** 8](#_Toc167908085)

[**3.3 Интеграция с Google sheets** 9](#_Toc167908086)

[**3.3.1 Описание Google Apps Script для обработки данных** 9](#_Toc167908087)

[**3.3.2 Создание и настройка Google Sheets** 9](#_Toc167908088)

[**4. Анализ результатов и оценка эффективности** 10](#_Toc167908089)

[**4.1 Результаты автоматизации отчетности** 10](#_Toc167908090)

[**4.2 Оценка преимуществ и недостатков внедрения автоматизации** 10](#_Toc167908091)

[**Заключение** 12](#_Toc167908092)

[**Список литературы** 13](#_Toc167908092)

[**Приложение №1** 14](#_Toc167908092)

[**Приложение №2** 17](#_Toc167908092)

# **Введение**

## **1.1 Обоснование актуальности темы**

Автоматизация отчетности является важной задачей для интернет-магазинов в условиях современного бизнеса. В последние годы наблюдается значительное увеличение объемов данных, которые необходимо обрабатывать для принятия обоснованных управленческих решений. Интернет-магазины ежедневно сталкиваются с необходимостью анализа больших объемов данных о продажах, запасах и других ключевых показателях. Вручную составление отчетов требует значительных временных и трудовых ресурсов, что повышает риск ошибок и задержек. Это, в свою очередь, может привести к потере конкурентоспособности и снижению эффективности бизнеса.

Современные технологии позволяют значительно упростить процессы сбора, обработки и анализа данных, что повышает качество управления бизнесом. Автоматизация отчетности помогает сократить время на составление отчетов, уменьшить вероятность ошибок и повысить оперативность принятия управленческих решений. Использование таких инструментов, как телеграм-боты и Google Sheets, позволяет интегрировать различные источники данных и обеспечить их быструю обработку.

Таким образом, актуальность данной темы обусловлена необходимостью повышения эффективности управления интернет-магазином через автоматизацию отчетности. Современные подходы и технологии автоматизации отчетности позволяют сократить затраты времени и ресурсов, улучшить качество принимаемых решений и повысить конкурентоспособность бизнеса. В условиях быстро меняющегося рынка и растущей конкуренции своевременное получение точной и полной информации становится критически важным.

## **1.2 Цель, задачи и методы исследования**

Целью данной работы является разработка и внедрение системы автоматизации отчетности интернет-магазина с использованием телеграм-бота и Google Sheets. Для достижения этой цели необходимо решить ряд задач, включающих теоретическое исследование автоматизации отчетности, анализ современных технологий и инструментов, разработку и тестирование системы автоматизации.

Задачи исследования:

1. Изучить теоретические основы автоматизации отчетности.
2. Рассмотреть современные технологии и инструменты для автоматизации отчетности.
3. Разработать телеграм-бота для автоматизации отчетности.
4. Интегрировать телеграм-бота с Google Sheets для обработки и хранения данных.
5. Провести тестирование разработанной системы и оценить её эффективность.

Методы исследования включают анализ научной литературы и практических источников, проектирование и разработку программного обеспечения, а также тестирование и оценку полученных результатов. В процессе исследования будут использоваться методы анализа данных, программирования и тестирования программного обеспечения. Теоретическая база работы основывается на трудах отечественных и зарубежных исследователей в области автоматизации бизнес-процессов и информационных технологий.

Таким образом, данная работа сочетает теоретическое исследование и практическую разработку системы автоматизации отчетности, что позволяет обеспечить комплексный подход к решению поставленной задачи.

## **1.3 Практическая значимость и структура работы**

Практическая значимость работы заключается в создании эффективного решения для автоматизации отчетности интернет-магазина, что позволяет сократить время на составление отчетов, уменьшить вероятность ошибок и повысить оперативность принятия управленческих решений. Внедрение системы автоматизации отчетности на базе телеграм-бота и Google Sheets обеспечивает интеграцию различных источников данных и их быструю обработку, что повышает качество управления бизнесом.

Структура работы включает введение, три главы основной части, заключение, список литературы и приложения. Введение обосновывает актуальность темы, формулирует цель, задачи и методы исследования, а также описывает практическую значимость работы. В первой главе рассматриваются теоретические и методологические основы автоматизации отчетности, включая понятие, значение и основные подходы к автоматизации, а также анализ современных технологий и инструментов. Вторая глава посвящена практической реализации системы автоматизации отчетности, включая разработку телеграм-бота и интеграцию с Google Sheets. В третьей главе проводится анализ результатов, оценка эффективности системы и примеры отчетов, сгенерированных системой. В заключении подводятся итоги работы, формулируются основные выводы и рекомендации по дальнейшему развитию темы.

Таким образом, данная работа представляет собой комплексное исследование и разработку системы автоматизации отчетности для интернет-магазина, что позволяет повысить эффективность управления бизнесом за счет использования современных технологий и инструментов.

# **2. Теоретическое и методологическое обоснование**

## **2.1 Автоматизация отчетности: понятие и основные подходы**

Автоматизация отчетности представляет собой процесс использования программных средств для создания, управления и анализа отчетов с минимальным участием человека. В контексте интернет-магазинов этот процесс играет ключевую роль в повышении эффективности управления бизнесом. Основное значение автоматизации отчетности заключается в сокращении времени, необходимого для сбора и обработки данных, снижении вероятности ошибок и повышении оперативности принятия решений.

Автоматизация отчетности позволяет исключить человеческий фактор, что значительно уменьшает риск возникновения ошибок и задержек, связанных с ручным вводом данных. В условиях высококонкурентного рынка и быстро меняющейся среды, наличие точных и своевременных данных становится критически важным для принятия обоснованных управленческих решений. Вручную составление отчетов требует значительных временных и трудовых затрат, что увеличивает риск ошибок и задержек.

Существуют различные подходы к автоматизации отчетности. Один из них – использование специализированных программных средств, которые могут автоматически собирать данные из различных источников, обрабатывать их и генерировать отчеты в заданном формате. Другой подход – разработка собственных решений с использованием доступных инструментов и технологий, таких как телеграм-боты и Google Sheets.

Программные средства для автоматизации отчетности могут быть интегрированы с различными информационными системами, что позволяет обеспечить бесшовное взаимодействие между различными компонентами системы. Такие инструменты могут автоматически обновлять данные в реальном времени, отправлять уведомления и создавать отчеты по заданным параметрам. Это позволяет руководителям интернет-магазинов оперативно получать необходимую информацию для принятия решений и улучшения управления бизнесом.

## **2.2 Технологии и инструменты для автоматизации отчетности в интернет-магазинах**

Современные технологии предлагают множество инструментов для автоматизации отчетности, среди которых можно выделить телеграм-боты и интеграцию с Google Sheets. Телеграм-боты представляют собой программные приложения, работающие внутри мессенджера Telegram и взаимодействующие с пользователями через текстовые сообщения. Такие боты могут выполнять широкий спектр задач, включая сбор данных, генерацию отчетов и отправку уведомлений.

Google Sheets является облачным сервисом для создания и совместного редактирования электронных таблиц. Этот инструмент предоставляет мощные возможности для автоматизации обработки данных, включая интеграцию с различными API и возможность написания скриптов на языке Google Apps Script. Интеграция телеграм-бота с Google Sheets позволяет создать эффективную систему автоматизации отчетности, обеспечивающую сбор, обработку и хранение данных в режиме реального времени.

Кроме того, современные технологии включают использование API для интеграции различных систем и сервисов, что позволяет обеспечить бесшовное взаимодействие между различными инструментами и платформами.

Например, API Google Sheets позволяет программно взаимодействовать с таблицами, добавлять и обновлять данные, а также получать отчеты в различных форматах. Это делает процесс автоматизации отчетности более гибким и адаптируемым к конкретным требованиям бизнеса.Таким образом, современные технологии и инструменты для автоматизации отчетности предоставляют интернет-магазинам возможность значительно улучшить свои бизнес-процессы, повысить оперативность и точность отчетности, а также снизить затраты на ручной труд.

## **2.3 Преимущества и сложности автоматизации отчетности**

Преимущества автоматизации отчетности включают:

* Сокращение времени на составление отчетов: Автоматизация позволяет значительно сократить время, необходимое для сбора и обработки данных, что ускоряет процесс получения отчетов.
* Уменьшение вероятности ошибок: Исключение человеческого фактора снижает риск возникновения ошибок при составлении отчетов, что повышает точность и надежность данных.
* Повышение оперативности принятия решений: Своевременное получение точных данных позволяет быстро принимать обоснованные управленческие решения, что улучшает общую эффективность бизнеса.
* Улучшение качества данных: Автоматизация обеспечивает более точные и полные данные для анализа, что способствует более глубокому пониманию бизнес-процессов и выявлению проблемных областей.
* Повышение прозрачности и контрольности: Автоматизация отчетности позволяет отслеживать ключевые показатели эффективности (KPI), что повышает прозрачность и контроль за бизнес-процессами.

Вместе с тем, автоматизация отчетности сталкивается с рядом вызовов, включая:

* Технические сложности: Разработка и внедрение системы автоматизации требует значительных технических навыков и ресурсов. Необходимо обеспечить корректное функционирование всех компонентов системы и их интеграцию с существующими бизнес-процессами.
* Интеграция с существующими системами: Важно обеспечить бесшовное взаимодействие системы автоматизации с существующими информационными системами, чтобы избежать дублирования данных и обеспечить их консистентность.
* Обеспечение безопасности данных: Важно защитить данные от несанкционированного доступа и потерь. Необходимо обеспечить надежное шифрование данных и доступ только для авторизованных пользователей.

Таким образом, автоматизация отчетности представляет собой важное направление для повышения эффективности управления интернет-магазином. Современные технологии и инструменты, такие как телеграм-боты и Google Sheets, предлагают мощные возможности для автоматизации процессов сбора, обработки и анализа данных. Внедрение таких систем позволяет значительно сократить время и трудозатраты на составление отчетов, улучшить качество данных и повысить оперативность принятия управленческих решений.

# **3. Практическая реализация автоматизации отчетности интернет-магазина**

## **3.1 Описание системы автоматизации отчетности**

**3.1.1 Архитектура системы**

Архитектура системы автоматизации отчетности интернет-магазина включает несколько ключевых компонентов, которые обеспечивают сбор, обработку, хранение и представление данных. Центральным элементом системы является телеграм-бот, который взаимодействует с пользователями и выполняет запросы на получение отчетов. Бот подключен к Google Sheets, где хранятся все данные о заказах интернет-магазина. Для обработки и фильтрации данных используется Google Apps Script, который обеспечивает выполнение запросов к таблицам и возвращает результаты в формате, удобном для пользователя.

Основные компоненты архитектуры:

* Телеграм-бот: Интерфейс для взаимодействия с пользователями, реализующий команды для запроса отчетов за определенные даты, периоды, а также для экспорта данных в формате Excel.
* Google Sheets: Облачное хранилище данных, где хранятся все заказы интернет-магазина. Каждая строка таблицы представляет собой отдельный заказ, содержащий информацию о клиенте, товарах, цене и дате отправки.
* Google Apps Script: Скрипт, работающий на стороне сервера, который обрабатывает запросы от телеграм-бота, выполняет фильтрацию данных в Google Sheets и возвращает результаты.

Эта архитектура обеспечивает высокую гибкость и масштабируемость системы, позволяя легко добавлять новые функции и расширять функционал бота.

### **3.1.2 Используемые технологии и инструменты**

Для реализации системы автоматизации отчетности интернет-магазина были выбраны следующие технологии и инструменты:

* Telegram Bot API: Используется для создания и управления телеграм-ботом. API предоставляет методы для отправки и получения сообщений, обработки команд и управления ботом.
* Python: Основной язык программирования для разработки логики телеграм-бота. Используется библиотека python-telegram-bot, которая упрощает взаимодействие с Telegram Bot API.
* Google Sheets API: Обеспечивает программный доступ к данным в Google Sheets. Используется для чтения, записи и обновления данных в таблицах.
* Google Apps Script: Скриптовый язык, позволяющий автоматизировать задачи в Google Workspace. Используется для написания серверной логики, обрабатывающей запросы от телеграм-бота.
* Pandas: Библиотека для обработки и анализа данных в Python. Используется для работы с данными, полученными из Google Sheets, и для экспорта данных в формате Excel.

## **3.2 Разработка и внедрение телеграмм-бота для автоматизации отчетности**

### **3.2.1 Постановка задачи**

Основная задача разработки телеграм-бота для автоматизации отчетности заключается в создании удобного и эффективного инструмента для получения отчетов о заказах интернет-магазина. Бот должен предоставлять пользователям возможность запрашивать отчеты за конкретные даты и периоды, получать статистику заказов, а также экспортировать данные в формат Excel. Кроме того, бот должен быть прост в использовании и быстро реагировать на запросы пользователей.

**3.2.2 Описание функционала бота**

Функционал телеграм-бота включает следующие основные команды:

* /start: Приветственное сообщение и инструкция по использованию бота.
* /help: Список доступных команд и их описание.
* /report: Получение последнего отчета о заказах.
* /report\_day ГГГГ-ММ-ДД: Получение отчета за конкретный день.
* /report\_period ГГГГ-ММ-ДД ГГГГ-ММ-ДД: Получение отчета за определенный период.
* /stats ГГГГ-ММ-ДД: Получение статистики заказов за год, месяц и день.
* /export\_excel: Экспорт данных в формате Excel.

Бот должен уметь обрабатывать введенные пользователями команды, взаимодействовать с Google Sheets для получения необходимых данных и возвращать результаты в удобном для пользователя формате.

**3.2.3 Используемый стек технологий**

Для разработки телеграм-бота использовался следующий стек технологий:

* Python: Основной язык программирования для написания кода бота.
* python-telegram-bot: Библиотека для взаимодействия с Telegram Bot API.
* Requests: Библиотека для отправки HTTP-запросов к Google Apps Script.
* Pandas: Библиотека для обработки данных и экспорта отчетов в формате Excel.
* Google Apps Script: Серверный скрипт для обработки запросов к Google Sheets и выполнения фильтрации данных.

## **3.3 Интеграция с Google sheets**

**3.3.1 Описание Google Apps Script для обработки данных**

Google Apps Script используется для создания серверной логики, которая обрабатывает запросы от телеграм-бота и взаимодействует с Google Sheets. Основные функции скрипта включают:

* Обработка параметров запроса: Скрипт принимает параметры запроса, такие как дата или период, и выполняет соответствующую фильтрацию данных.
* Фильтрация и обработка данных: На основе переданных параметров скрипт выполняет фильтрацию данных в Google Sheets и формирует результаты.
* Возвращение результатов в формате JSON: Скрипт возвращает результаты запроса в формате JSON, который удобно использовать в телеграм-боте.

**3.3.2 Создание и настройка Google Sheets**

Для интеграции с телеграм-ботом необходимо создать и настроить Google Sheets:

1. Создание таблицы: Создайте новую таблицу в Google Sheets и добавьте необходимые столбцы, такие как Имя, Email, Телефон, ID заказа, Продукты, Цена, Валюта и Дата отправки.
2. Заполнение данных: Внесите данные о заказах интернет-магазина в соответствующие столбцы таблицы.
3. Настройка доступа: Настройте доступ к таблице, чтобы скрипт Google Apps Script мог читать и записывать данные. Убедитесь, что таблица доступна для чтения и записи с помощью API.

Таким образом, интеграция с Google Sheets обеспечивает эффективное хранение и обработку данных о заказах интернет-магазина, а использование Google Apps Script позволяет автоматизировать процессы генерации и фильтрации отчетов.

# **4. Анализ результатов и оценка эффективности**

## **4.1 Результаты автоматизации отчетности**

Автоматизация отчетности в интернет-магазине привела к значительному повышению эффективности и точности процесса генерации отчетов. Основные результаты включают:

1. Снижение временных затрат: Ранее сотрудники тратили значительное время на ручное создание отчетов, теперь эта задача выполняется автоматически за считанные секунды. Это позволяет сотрудникам сосредоточиться на более важных задачах, улучшая общую продуктивность.
2. Увеличение точности данных: Ручное создание отчетов подвержено человеческим ошибкам. Автоматизация исключает этот фактор, обеспечивая высокую точность данных. Все отчеты формируются на основе актуальных данных, хранящихся в Google Sheets, что минимизирует вероятность ошибок.
3. Повышение прозрачности и доступности данных: Система автоматизации позволяет легко и быстро получить доступ к необходимым отчетам. Пользователи могут запросить отчеты за конкретные даты или периоды, а также получить статистику заказов, что способствует лучшему пониманию текущего состояния дел в интернет-магазине.

В целом, внедрение системы автоматизации отчетности принесло значительные улучшения в операционной деятельности интернет-магазина, обеспечив более эффективное использование ресурсов и повышение качества управления данными.

## **4.2 Оценка преимуществ и недостатков внедрения автоматизации**

Преимущества:

1. **Скорость и эффективность**: Автоматизация значительно ускоряет процесс генерации отчетов, экономя время сотрудников и повышая их продуктивность.
2. **Точность и надежность**: Исключение человеческого фактора уменьшает количество ошибок, обеспечивая высокую точность данных.
3. **Удобство и доступность**: Пользователи могут легко запросить необходимые отчеты в любое время, что повышает прозрачность и доступность данных.
4. **Гибкость и масштабируемость**: Система легко адаптируется к изменяющимся потребностям бизнеса, позволяя добавлять новые функции и улучшать существующие процессы.

Недостатки:

1. **Ограничения функционала**: В некоторых случаях автоматизированные системы могут не покрывать все специфические потребности бизнеса, требуя доработок и кастомизации.

**Заключение**

В заключение следует отметить, что автоматизация отчетности интернет-магазина представляет собой значительный шаг вперед в управлении бизнесом. Внедрение системы автоматизации позволило существенно сократить время на генерацию отчетов, повысить точность и надежность данных, а также улучшить доступность и прозрачность информации. Это, в свою очередь, привело к повышению общей эффективности работы интернет-магазина и улучшению качества принятия решений на основе точных и актуальных данных.

Процесс автоматизации, подробно описанный в данной работе, охватывает все ключевые аспекты – от постановки задачи до реализации и оценки результатов. Использование телеграм-бота и интеграция с Google Sheets показали высокую эффективность в автоматизации отчетности. Бот обеспечивает удобный и быстрый доступ к необходимым данным, а Google Sheets служит надежным источником информации и платформой для хранения и обработки данных.

Результаты внедрения системы автоматизации отчетности подтверждают ее высокую полезность и значимость. Система позволяет оперативно получать данные о заказах, продажах и других ключевых показателях, что способствует улучшению управленческих решений и оперативному реагированию на изменения в бизнес-среде. Важно отметить, что автоматизация отчетности также способствует повышению уровня удовлетворенности клиентов, так как позволяет быстрее и точнее обрабатывать их запросы и заказы.

**Список литературы**

1)Документация по Python. [Электронный ресурс]. URL: https://docs.python.org/3/

2)Документация по API Telegram Bot. [Электронный ресурс]. URL: https://core.telegram.org/bots/api

3)Как парсить json в python. [Электронный ресурс]. URL: https://ru-brightdata.com/blog/how-tos-ru/parse-json-data-with-python

4) Google apps script API. [Электронный ресурс]. URL: https://developers.google.com/apps-script/api/concepts?hl=en#:~:text=The%20Google%20Apps%20Script%20API,use%20the%20Apps%20Script%20editor

5)Johnson A. "Integration of Google Sheets with Business Processes". Journal of Business Information Systems, 2020, Страницы 78-91.

**Приложение №1  
Код телеграмм бота**from telegram import Update, InputFile  
from telegram.ext import Application, CommandHandler, ContextTypes  
import requests  
import pandas as pd  
from io import BytesIO  
import asyncio  
import nest\_asyncio  
  
# Замените YOUR\_TELEGRAM\_BOT\_TOKEN на токен вашего Телеграм-бота  
TOKEN = '7044724125:AAGRUbChusByOgnkh2q\_sW4nCkr5tlEP-dU'  
# Замените URL на URL вашего веб-приложения Google Apps Script  
GOOGLE\_SCRIPT\_URL = 'https://script.google.com/macros/s/AKfycbwsgfpXgO8hleKGpdxD3qCNGOOSVO4JM89v6SgJr1KlGkaZuQ9ae3mnlHtfG3BWGFZR/exec'  
  
async def start(update: Update, context: ContextTypes.DEFAULT\_TYPE) -> None:  
 await update.message.reply\_text('Привет! Используйте /report для получения отчета.\nИспользуйте /help для получения списка доступных команд.')  
  
async def help\_command(update: Update, context: ContextTypes.DEFAULT\_TYPE) -> None:  
 help\_text = (  
 "/start - Запустить бота\n"  
 "/report - Получить последний отчет\n"  
 "/report\_day ГГГГ-ММ-ДД - Получить отчет за конкретный день\n"  
 "/report\_period ГГГГ-ММ-ДД ГГГГ-ММ-ДД - Получить отчет за конкретный период\n"  
 "/stats ГГГГ-ММ-ДД - Получить статистику заказов (год, месяц, день)\n"  
 "/export\_excel - Экспортировать отчет в формате Excel"  
 )  
 await update.message.reply\_text(help\_text)  
  
async def fetch\_report(url: str) -> str:  
 try:  
 response = requests.get(url)  
 response.raise\_for\_status()  
 data = response.json()  
 report\_text = ''  
 for entry in data:  
 report\_text += (  
 f"Имя: {entry['Name']}\n"  
 f"Email: {entry['Email']}\n"  
 f"Телефон: {entry['Phone']}\n"  
 f"ID заказа: {entry['orderid']}\n"  
 f"Продукты: {entry['products']}\n"  
 f"Цена: {entry['price']} {entry['Currency']}\n"  
 f"Отправлено: {entry['sent']}\n\n"  
 )  
 return report\_text  
 except requests.exceptions.RequestException as e:  
 return f"Ошибка HTTP-запроса: {e}"  
 except ValueError:  
 return "Не удалось декодировать ответ JSON."  
  
async def report(update: Update, context: ContextTypes.DEFAULT\_TYPE) -> None:  
 report\_text = await fetch\_report(GOOGLE\_SCRIPT\_URL)  
 await update.message.reply\_text(report\_text)  
  
async def report\_day(update: Update, context: ContextTypes.DEFAULT\_TYPE) -> None:  
 if context.args:  
 date = context.args[0]  
 report\_text = await fetch\_report(f"{GOOGLE\_SCRIPT\_URL}?date={date}")  
 await update.message.reply\_text(report\_text)  
 else:  
 await update.message.reply\_text('Пожалуйста, укажите дату в формате ГГГГ-ММ-ДД.')  
  
async def report\_period(update: Update, context: ContextTypes.DEFAULT\_TYPE) -> None:  
 if len(context.args) == 2:  
 start\_date, end\_date = context.args  
 report\_text = await fetch\_report(f"{GOOGLE\_SCRIPT\_URL}?start\_date={start\_date}&end\_date={end\_date}")  
 await update.message.reply\_text(report\_text)  
 else:  
 await update.message.reply\_text('Пожалуйста, укажите начальную и конечную даты в формате ГГГГ-ММ-ДД.')  
  
async def stats(update: Update, context: ContextTypes.DEFAULT\_TYPE) -> None:  
 if context.args:  
 date = context.args[0]  
 response = requests.get(GOOGLE\_SCRIPT\_URL)  
 if response.status\_code == 200:  
 data = response.json()  
 year\_count = sum(1 for entry in data if entry['sent'].startswith(date[:4]))  
 month\_count = sum(1 for entry in data if entry['sent'].startswith(date[:7]))  
 day\_count = sum(1 for entry in data if entry['sent'].startswith(date))  
 stats\_text = (  
 f"Заказы за {date[:4]}: {year\_count}\n"  
 f"Заказы за {date[:7]}: {month\_count}\n"  
 f"Заказы за {date}: {day\_count}\n"  
 )  
 await update.message.reply\_text(stats\_text)  
 else:  
 await update.message.reply\_text('Не удалось получить данные из Google Sheets.')  
 else:  
 await update.message.reply\_text('Пожалуйста, укажите дату в формате ГГГГ-ММ-ДД.')  
  
async def export\_excel(update: Update, context: ContextTypes.DEFAULT\_TYPE) -> None:  
 response = requests.get(GOOGLE\_SCRIPT\_URL)  
 if response.status\_code == 200:  
 data = response.json()  
 df = pd.DataFrame(data)  
 output = BytesIO()  
 with pd.ExcelWriter(output, engine='xlsxwriter') as writer:  
 df.to\_excel(writer, index=False, sheet\_name='Заказы')  
 output.seek(0)  
 await update.message.reply\_document(InputFile(output, filename='orders.xlsx'))  
 else:  
 await update.message.reply\_text('Не удалось получить данные из Google Sheets.')  
  
async def main() -> None:  
 application = Application.builder().token(TOKEN).build()  
  
 application.add\_handler(CommandHandler("start", start))  
 application.add\_handler(CommandHandler("help", help\_command))  
 application.add\_handler(CommandHandler("report", report))  
 application.add\_handler(CommandHandler("report\_day", report\_day))  
 application.add\_handler(CommandHandler("report\_period", report\_period))  
 application.add\_handler(CommandHandler("stats", stats))  
 application.add\_handler(CommandHandler("export\_excel", export\_excel))  
  
 await application.run\_polling()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 nest\_asyncio.apply()  
 asyncio.run(main())

**Приложение №2**

function doGet(e) {

  try {

    const spreadsheetId = '17XXQrE8tSyit6FoTMHkrKtmPTnhHb17G4V-7qLiwOwk';

    const sheetName = 'Лист1';

    const sheet = SpreadsheetApp.openById(spreadsheetId).getSheetByName(sheetName);

    if (!sheet) {

      throw new Error('Sheet not found');

    }

    const data = sheet.getDataRange().getValues();

    const headers = data[0];

    const rows = data.slice(1);

    const result = rows.map(row => {

      let obj = {};

      headers.forEach((header, index) => {

        obj[header] = row[index];

      });

      return obj;

    });

    const date = e.parameter.date;

    const startDate = e.parameter.start\_date;

    const endDate = e.parameter.end\_date;

    const realtime = e.parameter.realtime;

    if (date) {

      const filtered = result.filter(entry => entry['sent'].split(' ')[0] === date);

      return ContentService.createTextOutput(JSON.stringify(filtered)).setMimeType(ContentService.MimeType.JSON);

    }

    if (startDate && endDate) {

      const filtered = result.filter(entry => {

        const sentDate = new Date(entry['sent']);

        return sentDate >= new Date(startDate) && sentDate <= new Date(endDate);

      });

      return ContentService.createTextOutput(JSON.stringify(filtered)).setMimeType(ContentService.MimeType.JSON);

    }

    if (realtime) {

      return ContentService.createTextOutput(JSON.stringify(result)).setMimeType(ContentService.MimeType.JSON);

    }

    return ContentService.createTextOutput(JSON.stringify(result)).setMimeType(ContentService.MimeType.JSON);

  } catch (error) {

    return ContentService.createTextOutput('Error: ' + error.message);

  }

}