ФЕДЕРАЛЬНOЕ ГOСУДАРСТВЕННOЕ БЮДЖЕТНOЕ

OБРАЗOВАТЕЛЬНOЕ УЧРЕЖДЕНИЕ

ВЫСШЕГO OБРАЗOВАНИЯ

«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Институт информатики, математики и робототехники

Кафедра информационных технологий и компьютерной математики

Направление подготовки:

09.03.03 Прикладная информатика

Направленность (профиль) образовательной программы:  
Информационные и вычислительные технологии

Проектный практикум

Проектная работа

Создание приложения “ Mail Sender”

|  |  |
| --- | --- |
| Проверил:  доц., к.ф.-м.н.  Колонских Д.М. | Выполнили: студенты 3 курса  очной формы обучения группы ПИ-3ИВТ221Б  Санников М.А., Ерастова П.С., Душенко А.Р. |

УФА – 2025

**Оглавление**

[Введение 3](#_Toc198907433)

[Техническое задание 5](#_Toc198907434)

[Теоретическая часть 8](#_Toc198907435)

[1.1 Преимущества и недостатки приложения. 9](#_Toc198907436)

[1.2 Алгоритм работы приложения. 10](#_Toc198907437)

[1.3 Реализация работы приложения. 10](#_Toc198907438)

[1.4 Отклик приложения при нажатии на кнопки. 14](#_Toc198907439)

[Пример использования приложения “ Mail Sender ”. 15](#_Toc198907440)

[Заключение 16](#_Toc198907441)

[Литература 17](#_Toc198907442)

[Приложение. Код программы 18](#_Toc198907443)

# Введение

Идея проекта заключается в создании удобного консольного приложения, предоставляющего пользователям возможность для отправки электронных писем через SMTP-сервер. Основная цель – упростить процесс массовой рассылки писем с возможностью кастомизации тем и текста.

Ключевые особенности:

* Простота использования – не требует графического интерфейса, работает через командную строку.
* Гибкость – поддерживает отправку писем нескольким получателям (из файла или ручного ввода).
* Кастомизация – позволяет задавать тему и текст письма перед отправкой.
* Автономность – не зависит от веб-сервисов (SendGrid, Mailchimp), работает напрямую с SMTP.

Целевая аудитория:

* Разработчики, которым нужен простой инструмент для тестирования email-рассылок.
* Администраторы, отправляющие уведомления на группы email-адресов.
* Студенты, изучающие работу с SMTP в Python.

**Обзор**

Существует множество решений для отправки писем:

* SendGrid, Mailchimp — облачные сервисы с API.
* PHPMailer, Nodemailer — библиотеки для отправки писем из кода.
* Почтовые клиенты (Thunderbird, Outlook) — требуют ручного ввода.

Отличие Mail Sender в простоте и автономности: это консольное приложение, не требующее сложной настройки. Оно заполняет нишу локальных SM-клиентов с минимальными зависимостями, в отличие от облачных сервисов и тяжелых почтовых программ.

**Технические требования**

Постановка задачи:

Разработать консольное Python-приложение, которое:

• Подключается к SMTP-серверу (Gmail, Yandex, Mail.ru и др.).

• Принимает список email-адресов получателей (из файла или вручную).

• Позволяет задавать тему и текст письма.

• Логирует процесс отправки (успех/ошибка).

Отличия от аналогов:

• Не требует API-ключей – работает напрямую с SMTP.

• Минимальные зависимости – только Python SMTP lib

# Техническое задание

**Область применения**

Проект представляет собой приложение для отправки электронных писем через SMTP-сервер. Программа предназначен для:

* Автоматизации уведомлений

**Цель работы**: требуется создать приложение на языке программирова­ния Python с возможностью отправки писем через SMTP.  
**Задачи:**

1. Реализовать подключение к SMTP-серверу.
2. Добавить поддержку XLSX-файлов с email-адресами.
3. Обеспечить кастомизацию текста.

**Требования к системе**

**Функциональные требования:**

* **Подключение к SMTP:** Поддержка TLS/SSL, аутентификация логин/пароль.
* **Загрузка получателей:** Чтение email-адресов из файла.
* **Настройка письма:** Возможность задать тему и текст
* **Логирование:** Запись результатов отправки в консоль или файл (success/fail).

**Нефункциональные требования:**

* **Производительность:** Отправка 100+ писем за раз без зависаний.
* **Безопасность:** Оповещение о неверных SMTP-настройках (без хранения пароля в открытом виде).
* **Удобство:** Четкие подсказки в консоли при вводе данных.

**Архитектура системы**

1. **Взаимодействие:**
   * Пользователь - UI (Python) - SMTP (Python).
2. **Компоненты:**
   * **SMTP-клиент:** Взаимодействие с сервером через библиотеку SMTP.
   * **Парсинг получателей:** Загрузка email-адресов из файла.
   * **Логирование:** Запись статусов отправки.

**Технологический стек**

* **Язык:** Python 3.12+.
* **Библиотеки:**
  + smtplib – для работы с SMTP.
* **Сборка:**
* **Среда:**

**Пользовательский интерфейс**

* **Кнопки:** войти, отправить.
* **Формат ответа:**

Окно с информацией об успешном входе.

Окно с информацией об успешной отправке писем.

**API и интеграции**

* **smtplib**

**Операционные условия эксплуатации**

Система представляет собой Python-приложение, работающее на сервере или локальной машине с минимальными требованиями к ресурсам.

**Минимальная конфигурация сервера:**

* CPU: 1 ядро (x86-64).
* Оперативная память: 512 МБ.
* Дисковое пространство:100 МБ (для JRE, библиотек и временных файлов).
* ОС:Linux (Ubuntu/Debian), Windows Server, macOS.

**Сетевые требования:**

* Интернет-подключение со скоростью не менее 1 Мбит/с (для работы с SMTP-серверами).
* Открытые порты: SMTP: 25, 587 (TLS) или 465 (SSL) – в зависимости от настроек почтового сервиса.

**Системные требования для пользователей**

Для доступа к приложению с ПК, требуется:

* Операционная система: Windows 7 или выше, MacOS 10.12+, Linux.
* Оперативная память: 1 ГБ RAM.
* Место на жестком диске: 200 Мбайт.

**График выполнения работ**

Разработка приложения была разбита на несколько этапов:

1. **Анализ SMTP-протокола (1 день)**
2. **Разработка ядра (3 дня)**
3. **Доработка интерфейса (2 дня)**
4. **Тестирование и отладка (1 день)**

# Теоретическая часть

Mail Sender — это приложение для автоматизированной отправки электронных писем через SMTP-сервер. Программа разработана на Python с использованием стандартных библиотек для работы с электронной почтой. Основное назначение - упрощение массовых рассылок для администраторов, разработчиков и других пользователей, которым необходимо отправлять уведомления или письма группе получателей.

## Преимущества и недостатки приложения.

Преимущества:

1. Простота использования: интуитивно понятный интерфейс, минимальные требования к настройке.
2. Кроссплатформенность: приложение работает на Windows, Linux и macOS.
3. Быстрая настройка: достаточно указать параметры SMTP-сервера
4. Автономность: не требует подключения к облачным сервисам
5. Гибкость: поддержка различных SMTP-серверов, возможность загрузки списка получателей из файла

Недостатки:

1. Ограничения платформы: нельзя реализовать сложный пользовательский интерфейс, как в мобильных приложениях.
2. Зависимость от сторонних сервисов.

## Алгоритм работы приложения.

1. Инициализация: загрузка конфигурационных параметров из файла, проверка наличия интернет-соединения.
2. Авторизация: установка соединения с SMTP-сервером, аутентификация с помощью логина и пароля.
3. Подготовка письма: чтение списка получателей (из файла или ручного ввода), формирование темы и текста письма, установка кодировки сообщения.
4. Отправка: последовательная отправка писем всем получателям, логирование результатов каждой отправки.
5. Завершение: корректное закрытие SMTP-соединения, формирование итогового отчета.

## Реализация работы приложения.

Для начала работы, мы импортируем библиотеки:

1. **smtplib** - библиотека для работы с SMTP. Она позволяет авторизоваться и отправлять письма.

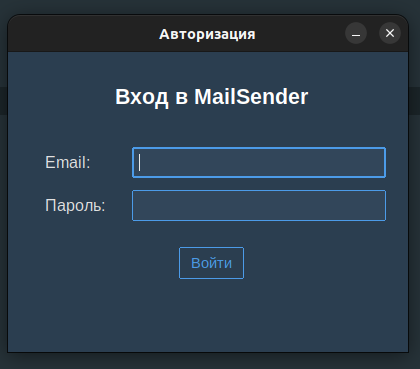
Сначала происходит setup настройки логгеров и запуск функции start\_gui\_app где создается окно с авторизацией. После этого мы ждем ввод от пользователя логина и пароля от SMTP. Если введены не все данные и была нажата кнопка отправки, то будет выведено сообщение о том, что не все поля заполнены. Если поля заполнены, но данные некорректны, будет выведено сообщение о том, что данные некорректны. Это реализовано за счет базовой конструкции try-except в python. Далее мы создаем новое окно для выбора файлов с почтами получателей и шаблоном письма. Они так-же валидируются. После нажатия кнопки отправки происходит попытка отправки писем. Если отправка не удалась, то процесс не прервется и отправка на другие почты продолжится. В конце будет выведено окно с информацией об отправке.

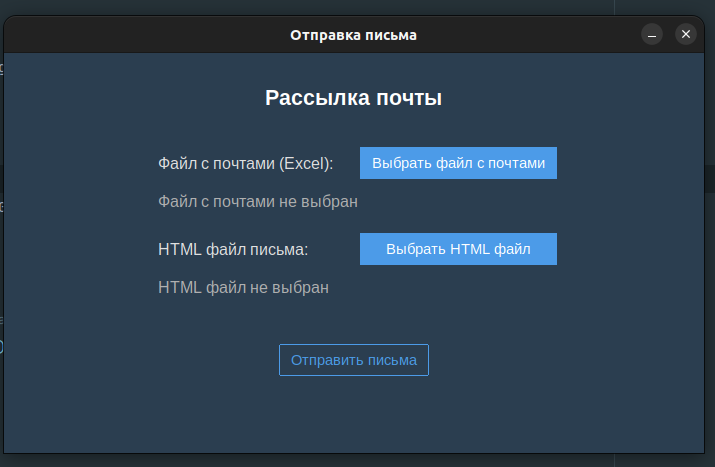
## Отклик приложения при нажатии на кнопки.

1. **"Войти"** — происходит попытка авторизации на SMTP сервере почты.
2. **"Отправить"** — попытка отправить письмо на указанные в файле почты.

Обработка нажатий реализована через функции-команды в tkinter.

## Пример использования приложения “Mail Sender”.





# Заключение

В ходе выполнения проектной работы мы разработали приложение Mail Sender для автоматизированной отправки электронных писем через SMTP-сервер. Программа успешно интегрирует библиотеку smtplib, позволяя пользователям настраивать параметры рассылки текст, список получателей и отправлять письма.

**Достигнутые результаты:**

* Реализована отправка писем через SMTP с поддержкой TLS/SSL.
* Добавлена загрузка списка получателей из файла.
* Создан интуитивный интерфейс для пользователя.
* Настроено логирование статусов отправки (успех/ошибка).

**Приобретенные навыки:**

* Работа с smtplib для взаимодействия с SMTP-серверами.
* Обработка пользовательского ввода и файловых операций на Python.
* Оптимизация кода для массовой рассылки без зависаний.

Эта работа позволила углубить знания в сетевых технологиях (SMTP) и Python-разработке, а также освоить принципы создания автономных утилит для автоматизации рутинных задач. Полученный опыт может быть применен в более сложных проектах, например, для разработки корпоративных систем email-нотификаций.

# Литература

1. Официальная документация Python URL: <https://docs.python.org/3/> (дата обращения: 02.05.2025).
2. Real Python: Sending Emails with Python URL: <https://realpython.com/python-send-email/> (дата обращения: 23.04.2025).
3. Stack Overflow: Python SMTP Questions URL: <https://stackoverflow.com/questions/tagged/python+smtp> (дата обращения: 25.04.2025).
4. Книга: "Automate the Boring Stuff with Python" (Al Sweigart) URL: <https://automatetheboringstuff.com/> (дата обращения: 23.04.2025).
5. Документация библиотеки email URL: https://docs.python.org/3/library/email.html (дата обращения: 03.05.2025).

# Приложение. Код программы

|  |
| --- |
| https://github.com/mikhail-sannikov/mail-sender |