Министерство науки и высшего образования РФ

Федеральное государственное автономное образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Базовая кафедра «Интеллектуальные системы управления»

**ОТЧЕТ О КУРСОВОЙ РАБОТЕ**

СФУ, ИКИТ, базовая кафедра ИСУ

создание приложения для переадресации сообщений в мессенджерах  
тема

Руководитель от университета \_\_\_\_\_\_\_\_ А.А. Даничев

подпись, дата

Студент КИ23-03Б, 032326992 \_\_\_\_\_\_\_\_ М.С. Бекетов

номер группы, зачетной книжки подпись, дата

Студент КИ23-03Б, 032322121 \_\_\_\_\_\_\_\_ Д.А. Зверьков

номер группы, зачетной книжки подпись, дата

Студент КИ23-03Б, 032322222 \_\_\_\_\_\_\_\_ Е.А. Баннов

номер группы, зачетной книжки подпись, дата

Студент КИ23-03Б, 032320725 \_\_\_\_\_\_\_\_ М.А. Дедусь

номер группы, зачетной книжки подпись, дата

Красноярск 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc198948437)

[1 Выбор и обоснование выбора структуры приложения 4](#_Toc198948438)

[2 Создание серверного приложения 6](#_Toc198948439)

[2.1 Дублирование сообщений в любимый мессенджер получателя 6](#_Toc198948440)

[2.2 Создание программы стандартизации изображений, аудио и видео файлов 7](#_Toc198948441)

[2.3 Создание логгера ошибок 8](#_Toc198948442)

[3 Создание сайта 9](#_Toc198948443)

[3.1 Создание общих объектов для страниц 9](#_Toc198948444)

[3.2 Написание основных разделов для сайта 9](#_Toc198948445)

[3.3 Создание стилей для сайта 9](#_Toc198948446)

[3.4 Выхлоп 14](#_Toc198948447)

[4. Шифрование данных пользователя 15](#_Toc198948448)

[ЗАКЛЮЧЕНИЕ 16](#_Toc198948449)

# ВВЕДЕНИЕ

В современном мире существует огромное множество мессенджеров, и каждый человек использует не тот мессенджер, который ему удобен, а тот что популярный среди его окружения. Мы выбираем Telegram, потому что там сидят друзья, WhatsApp — потому что на нём настаивает семья, а Viber — потому что его до сих пор используют коллеги. Даже если интерфейс неудобен, а функционал ограничен, мы миримся с этим ради возможности оставаться на связи. Так технологии, призванные объединять, становятся ещё одним поводом для раздражения: бесконечные уведомления из десятка приложений, путаница в чатах и вечная необходимость проверять несколько мессенджеров одновременно. Но выбора нет — ведь цифровая коммуникация давно перестала быть вопросом личного удобства, а превратилась в социальное обязательство. Но почему бы это не исправить?

Цель работы: создать приложения для переадресации сообщений в мессенджерах

Задачи:

- выбрать и обосновать выбор структуры приложения

- создать серверное приложения для

- создать сайта

- создать базу данных

**1) хранения токенов, никнеймов в БД**

**2) добавления/удаления токенов в БД**

- обеспечить надежное шифрование данных пользователей

# 

# Выбор и обоснование выбора структуры приложения

Пользователь нашего приложения должен писать в свой любимый мессенджер, а другой пользователь должен получать его в уже своем любимом мессенджере. Желательно чтобы юзер не устанавливал на устройство сторонних программ, так как это существенно упростит поддержку кроссплатформенности продукта. Приложение должно быть реализовано на сервере что позволит обеспечивать его бесперебойную работу.

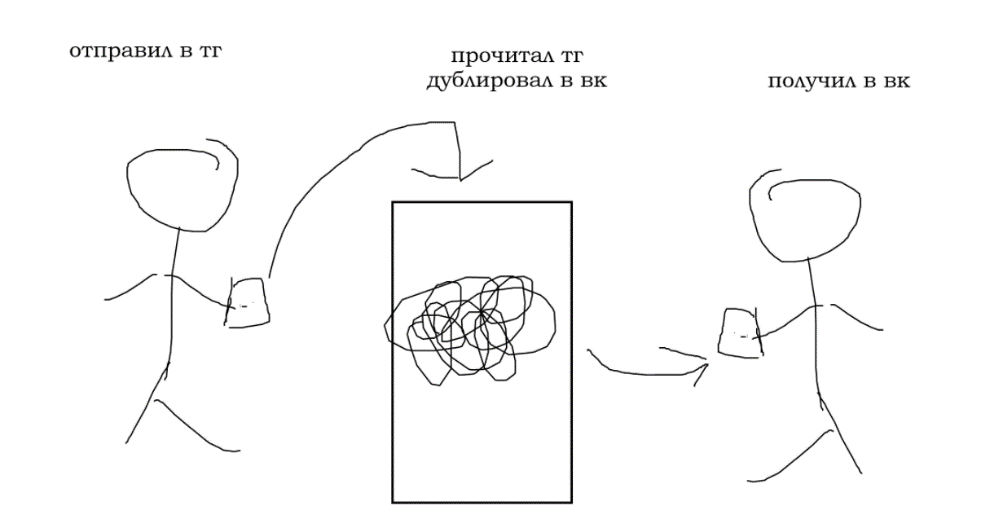


Рисунок 1 ‒ схема работы приложения

Для достижения данного результата можно использовать userbot-ов. Их поддерживают большинство современных мессенджеров, и они позволяют писать сообщения от лица человека, не прибегая к чрезмерному усложнению.

Для хранения токенов и данных пользователей необходимо создать 2 соединённых БД. Хранение информации будет производиться в MySQL, так как это СУБД с высокой надёжностью, производительностью и поддержкой современных технологий. Примерная структура приведена ниже на рисунке 2.

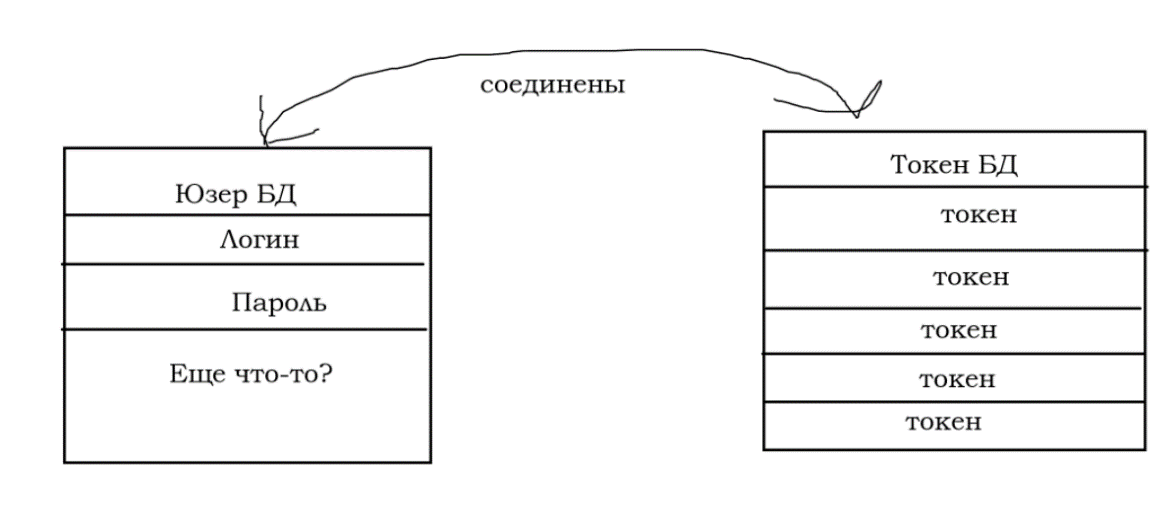


Рисунок 2 ‒ схема БД

Для первого знакомства пользователя с продуктом и его дальнейшей регистрации в нем хорошо подойдет вебсайт. Это быстрый, доступный и удобный способ для первого взаимодействия с потребителем, который не требует излишних действий с обеих сторон. Предполагаемый макет сайта приведен на рисунке ниже.



Рисунок 3 ‒ макет сайта

# Создание серверного приложения

Для создания серверного приложения необходимо продумать его структуру. Сервер должен: дублировать сообщения в любимый мессенджер получателя, иметь возможность отправки изображений, аудио, видео и т.д., а также иметь логгер ошибок. Это все ставит перед нами конкретные задачи:

- дублирование сообщений в любимый мессенджер получателя

- создание программы стандартизации изображений, аудио, видео и т.д. файлов

- создание логгера ошибок

## Дублирование сообщений в любимый мессенджер получателя

Дублирование сообщений в мессенджерах будет происходит по следующему алгоритму

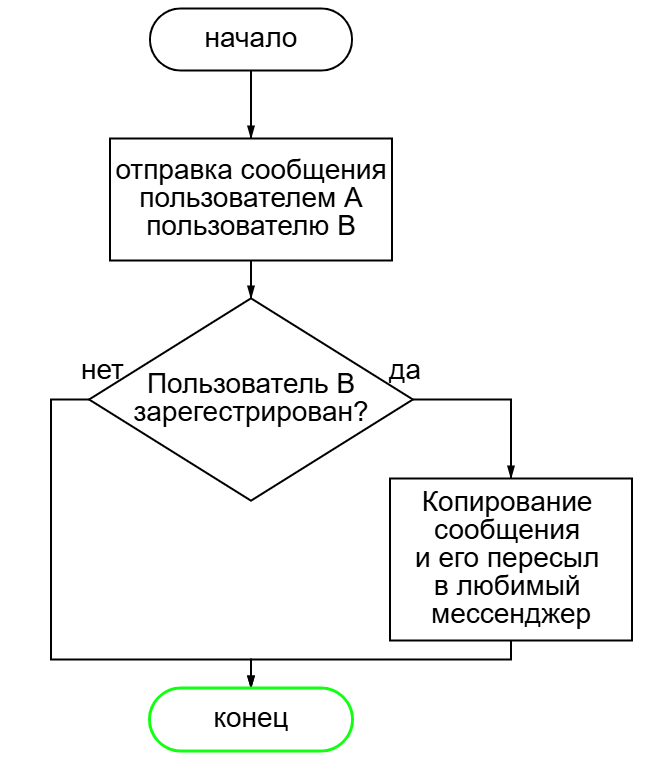


Рисунок 4 ‒ блок-схема работы серверного приложения

Для реализации напишем код на ЯП python. Для telegram будем использовать библиотеку telethon, а для vk – vk\_api. Реализуем класс Bridge для отправки сообщения в мессенджеры. В инициализации создадим: клиент telegram, сессию VK с использованием токена, а также инициализируем переменные для хранения ID последнего сообщения VK и управления потоками. Напишем методы класса:

а) tg\_to\_vk(event) (асинхронный):

1) Обрабатывает новые сообщения из Telegram

2) Проверяет, что сообщение приватное и исходящее

3) Проверяет, что чат соответствует целевому пользователю

4) Если сообщение не пустое, пересылает его в VK целевому пользователю

б) vk\_polling():

1) Запускается в отдельном потоке

2) Постоянно проверяет новые входящие сообщения в VK (каждые 3 секунды)

3) При получении нового сообщения (не исходящего и с новым ID) пересылает его в Telegram

4) Использует asyncio.run\_coroutine\_threadsafe для безопасной отправки в Telegram из другого потока

в) send\_to\_telegram(text) (асинхронный):

1) Отправляет полученный текст целевому пользователю в Telegram

г) start() (асинхронный):

1) Запускает клиент Telegram

2) Добавляет обработчик новых сообщений (tg\_to\_vk)

3) Выводит сообщение о успешном запуске

д) run():

1) Основной метод для запуска всего моста

2) Настраивает event loop asyncio

3) Запускает Telegram клиент

4) Создает отдельный поток для опроса VK

5) Обрабатывает остановку по Ctrl+C, корректно закрывая соединения

**Сюда надо скрин кода?**

## Создание программы стандартизации изображений, аудио и видео файлов

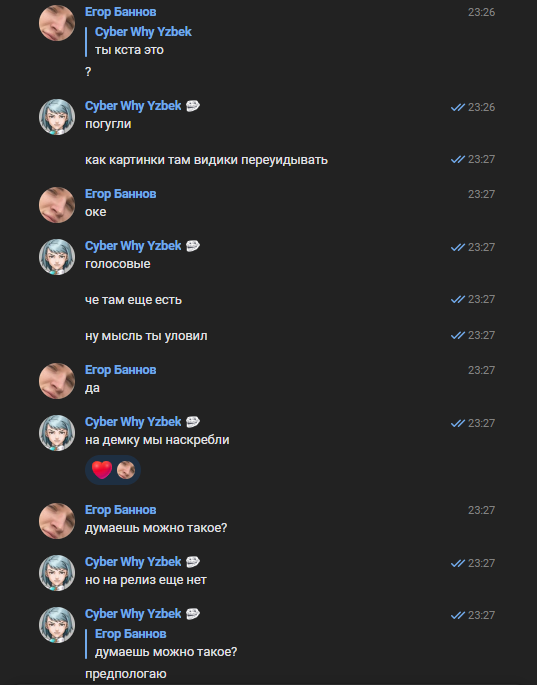


Рисунок 5 ‒ Егор поработает над этим

## Создание логгера ошибок

Егор крепись

# Создание сайта

Так как сайт — это место, где пользователь знакомится с продуктом на нем должна располагаться вся интересующая информация. Через сайт удобно будет регистрироваться, добавлять токены, а также выбирать свой любимый мессенджер. Исходя из вышесказанного обозначим задачи:

- создание раздела для информирования пользователей о продукте;

- регистрация пользователя;

- создание личного кабинета;

- выбор любимого мессенджера пользователем, подключение и отключение токенов.

## 3.1 Создание общих объектов для страниц

Для каждого сайта в интернете характерны 3 основных раздела header, footer и main. На нашем сайте мы будет использовать одинаковый header и footer для всех страниц. В header будут помещены: навигационное меню, название и логотип нашего сайта. В footer поместим копирайт и кнопку для смены темы. Также реализуем дизайн при помощи CSS

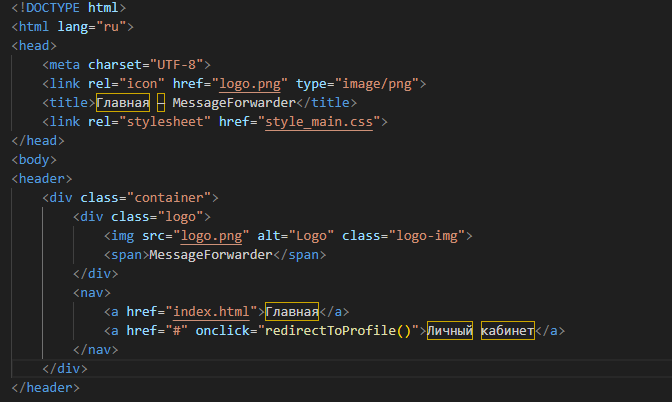


Рисунок 6 ‒ код header-а сайта

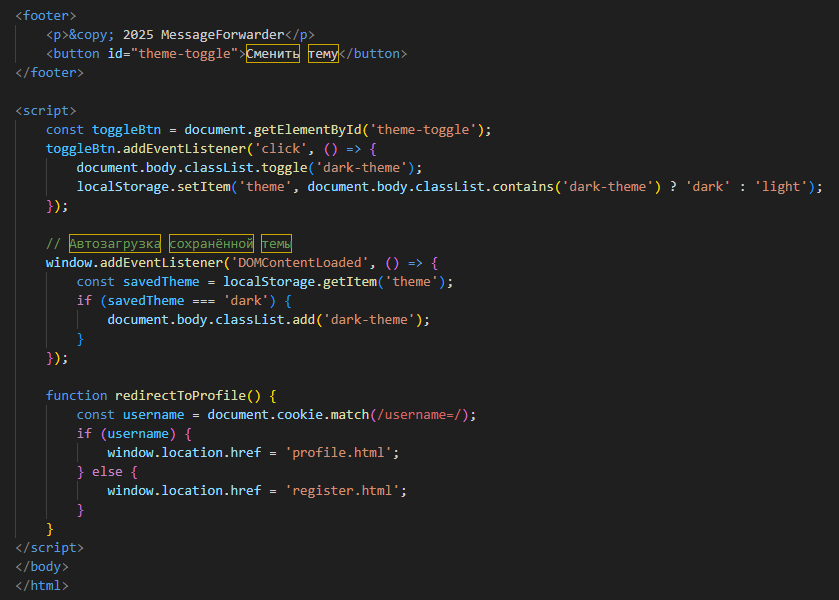


Рисунок 7 ‒ код footer-а сайта

## Написание основных разделов для сайта

На главной странице сайта разместим ключевые преимущества нашего приложения и добавим кнопку для перехода к регистрации/логину.

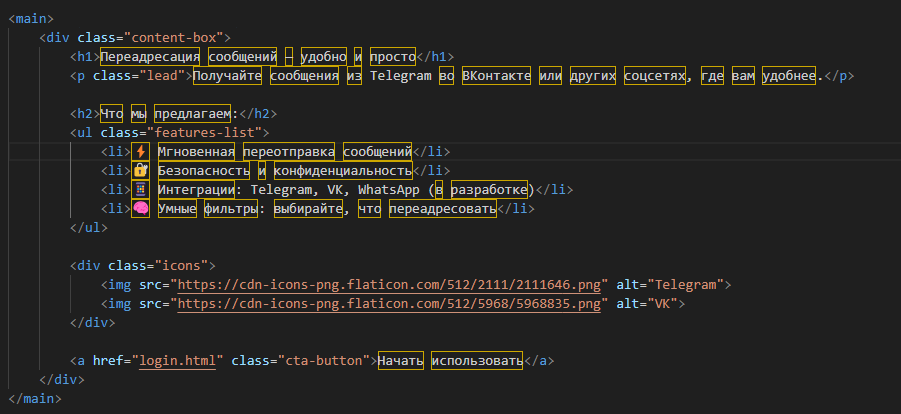


Рисунок 8 ‒ код основного раздела заглавной страницы сайта

В разделе с логином расположим плейсхолдеры для логина и пароля, а также кнопки для регистрации нового пользователя и восстановления пароля.

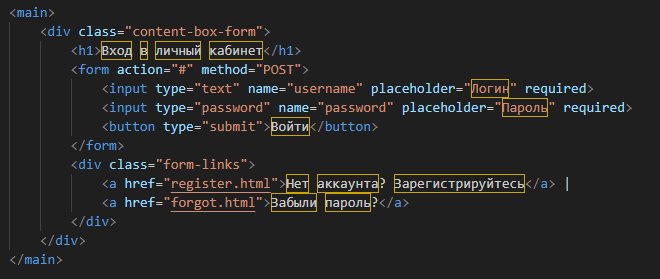


Рисунок 9 ‒ код основного раздела страницы с логином

В разделе с регистрацией расположим плейсхолдеры для логина, пароля и email пользователя, также необходимо добавить кнопку для входа уже зарегистрированных пользователей.

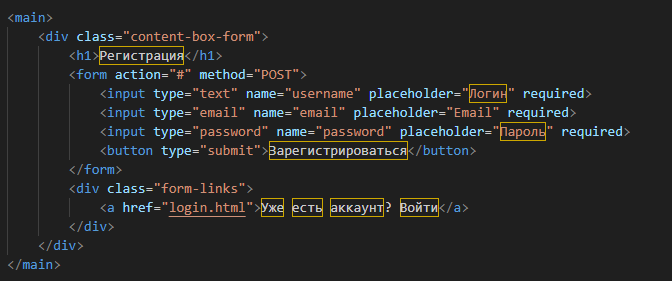


Рисунок 10 ‒ код основного раздела страницы с регистрацией

В разделе для восстановления пароля создадим форму для запроса почты пользователя.

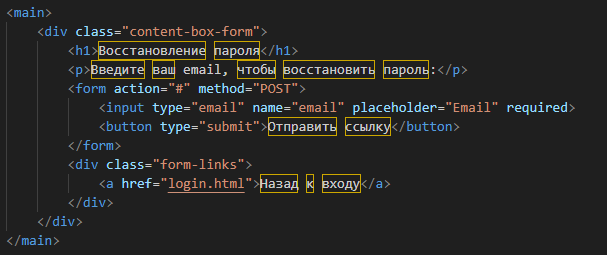


Рисунок 11 ‒ код основного раздела страницы с восстановлением пароля

В личном кабинете пользователя будет выводится его имя и добавленные токены.

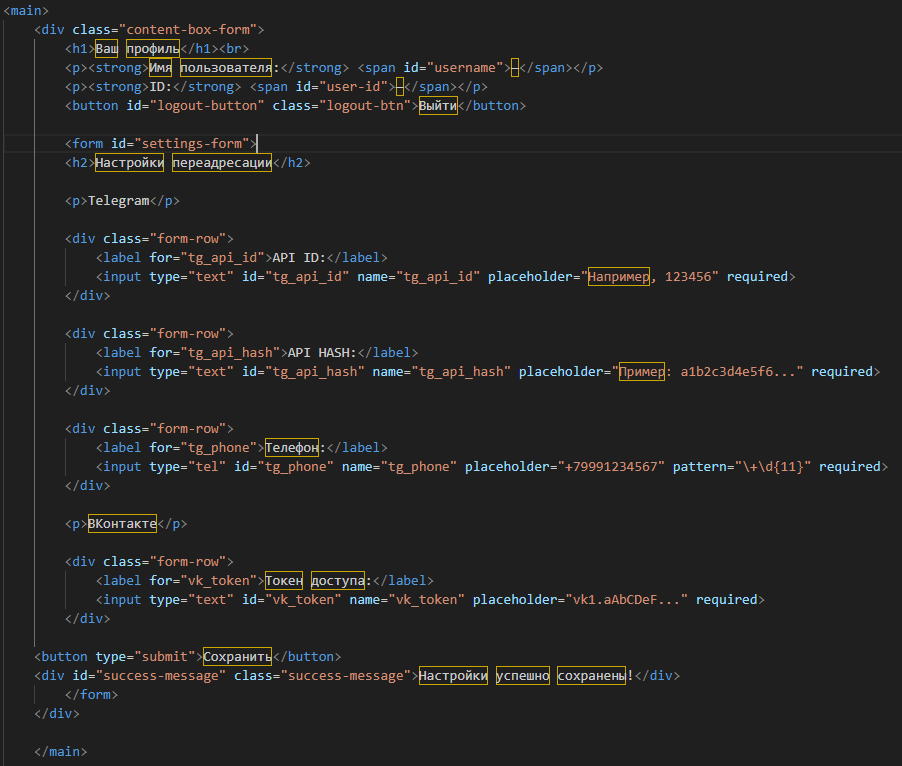


Рисунок 12 ‒ код основного раздела личного кабинета

## Создание стилей для сайта

В CSS-файле реализована стилизация для светлой и тёмной темы интерфейса с переключаемыми переменными (CSS Custom Properties).

Были созданы:

- Цветовые темы

- Основная структура

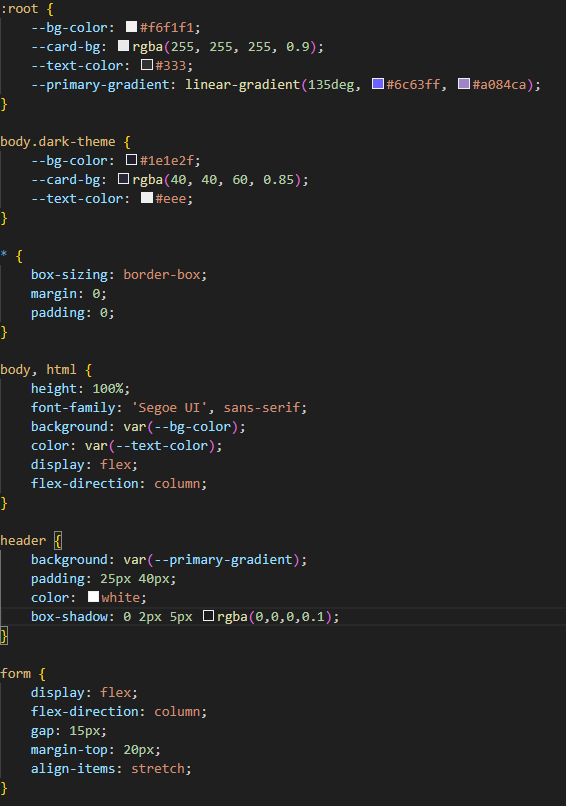
- Формы

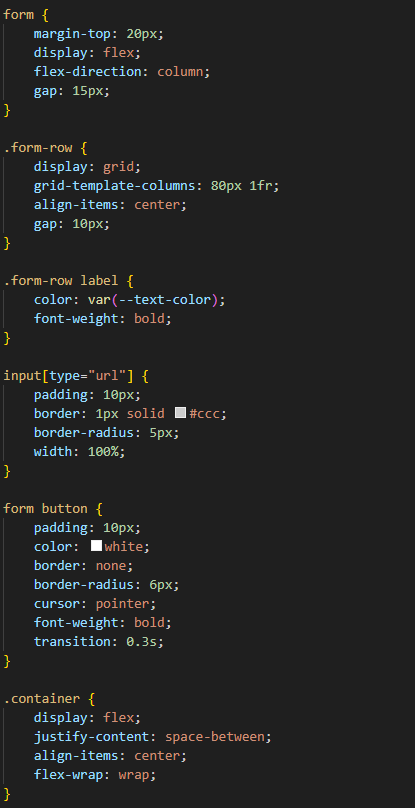
- Визуальные элементы

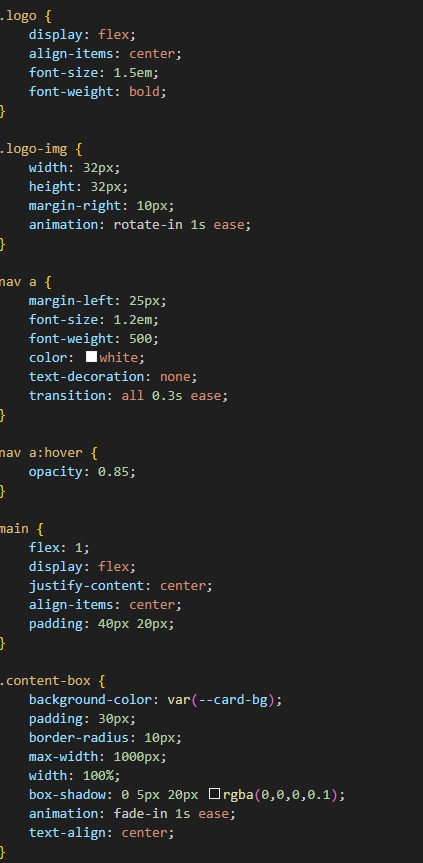
- Адаптивность

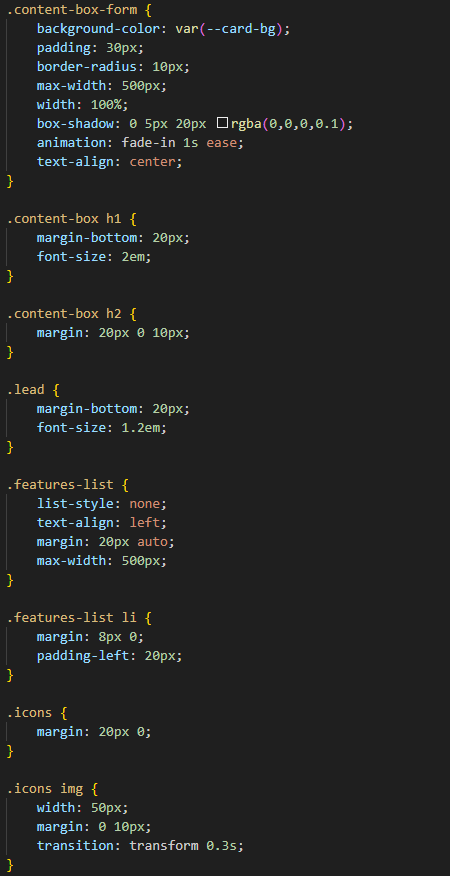
- Дополнительные компоненты

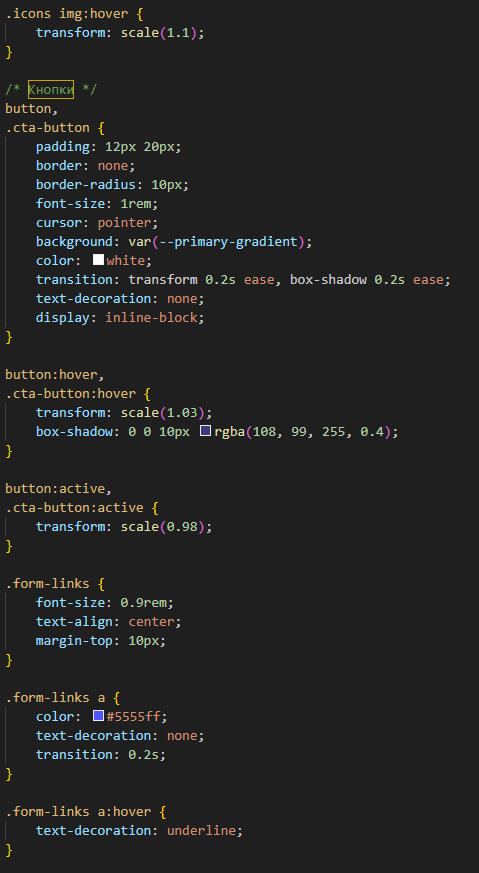
**Выкинуть картинки в приложение**



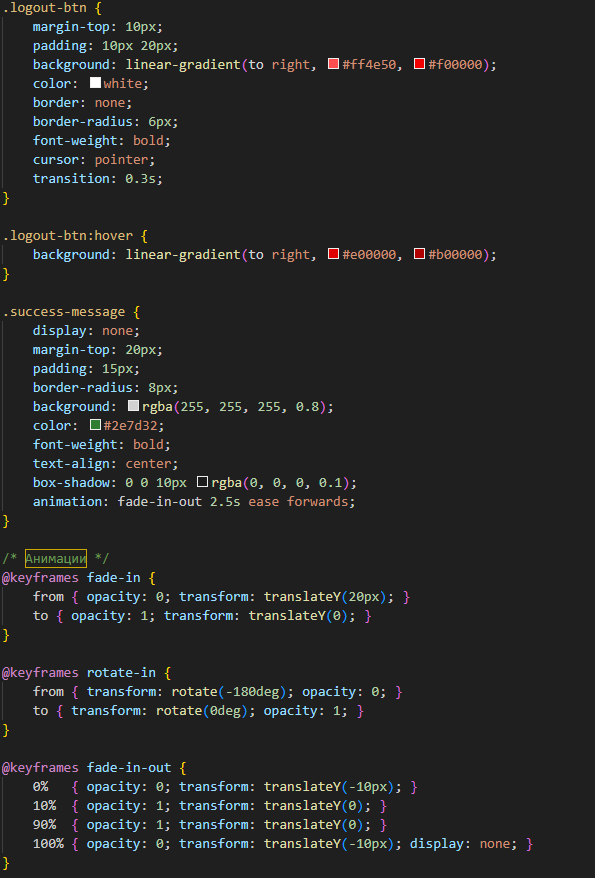








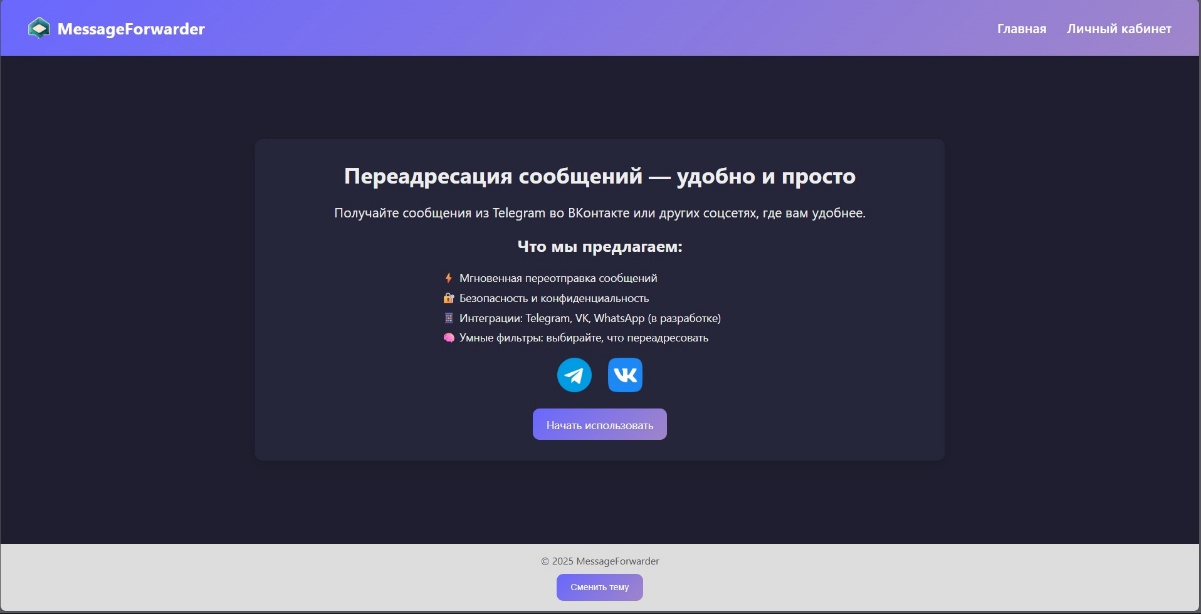


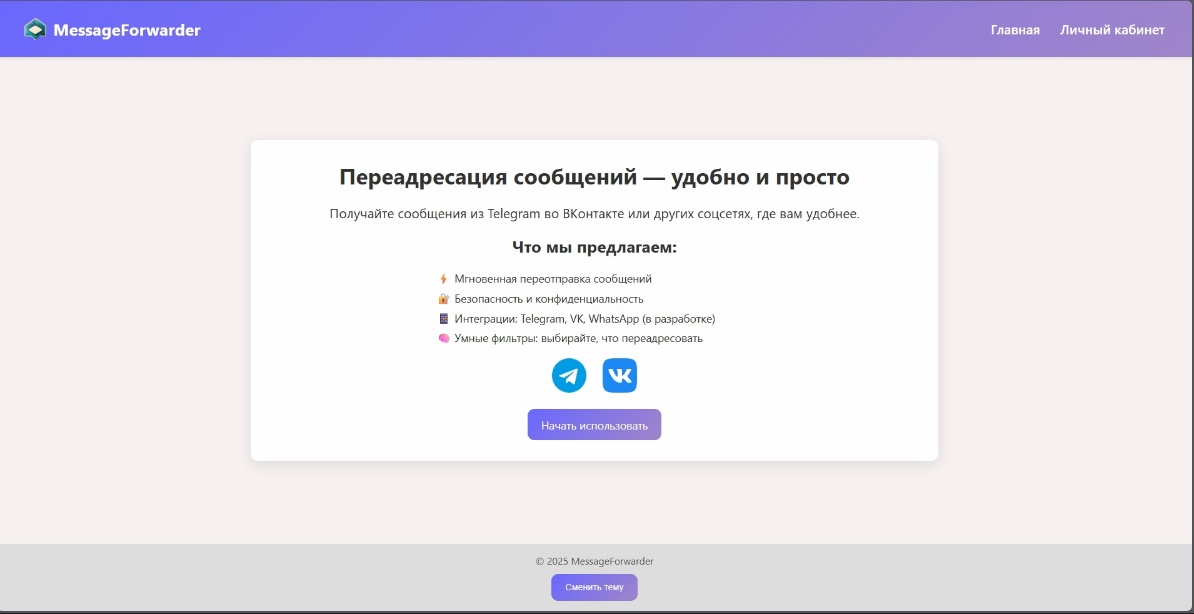


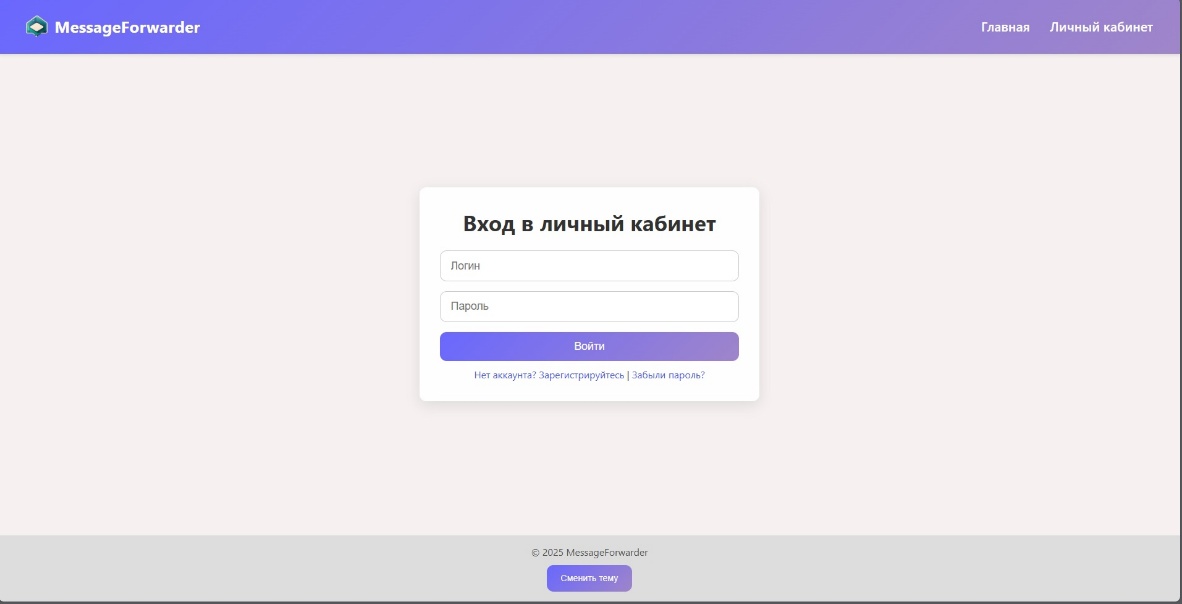


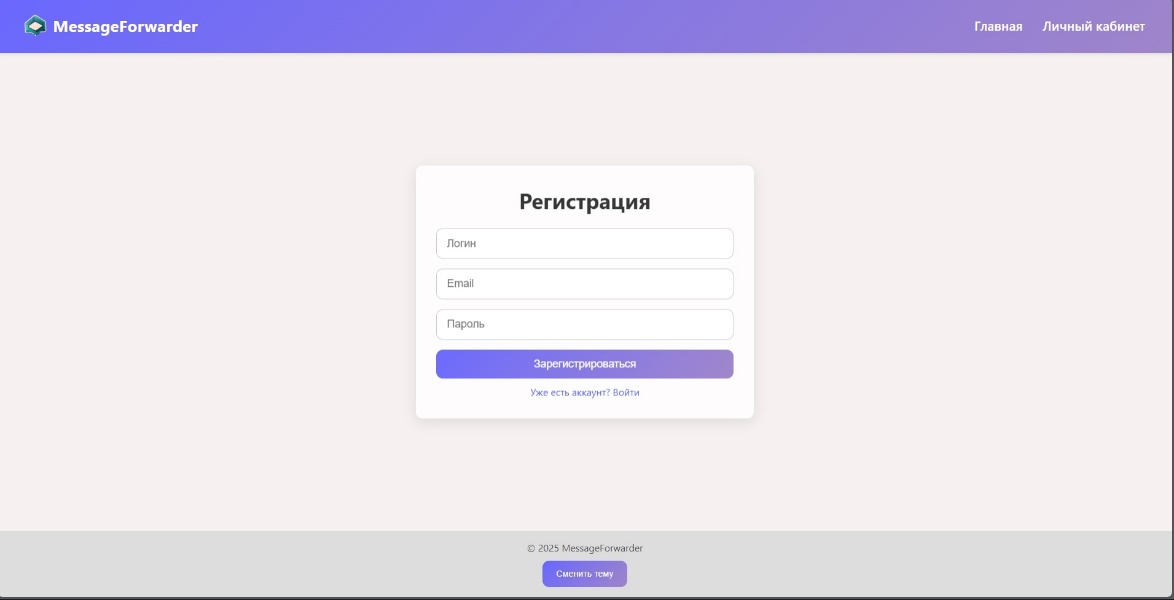
## Вывод

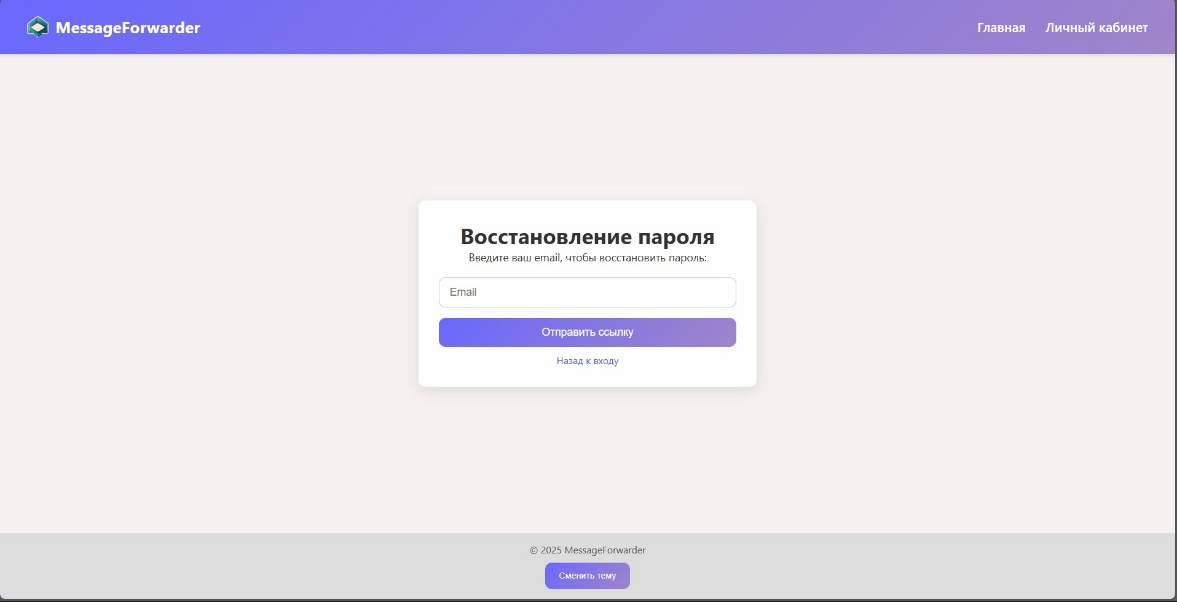
* В данном разделе была подробно описана разработка веб-сайта, включающая создание структуры, основных разделов и стилизацию интерфейса. Были успешно реализованы ключевые функциональные блоки: общая структура сайта, основные разделы, стилизация и адаптивность. Сам сайт представлен на скриншотах ниже.











# 4. Шифрование данных пользователя

Шифрование данных пользователя является неотъемлемой частью любого проекта. В данной работе лучшим решением является использования алгоритма AES-256. Это современный, быстрый и надежный стандарт шифрования данных.

## 4.1 Написание функций для шифрования и дешифрования

В python существует множество криптографических библиотек. В данной работе будет использоваться cryptography. Данная библиотека поддерживает выбранный ранее алгоритм шифрования. Реализуем две функции для шифрования и дешифрования данных с использованием алгоритма AES-256 в режиме GCM (Galois/Counter Mode). Этот режим обеспечивает не только конфиденциальность данных, но и их аутентификацию, что является важным аспектом информационной безопасности. Функция encrypt\_aes\_256\_gcm выполняет следующие операции:

- преобразует входную строку в байтовое представление с использованием UTF-8 кодировки;

- при необходимости генерирует криптографически стойкий ключ длиной 32 байта (256 бит) и вектор инициализации (IV) длиной 12 байт;

- создает шифр с использованием алгоритма AES-256 в режиме GCM;

- шифрует данные и формирует тег аутентификации;

- возвращает кортеж, содержащий зашифрованные данные, ключ, вектор инициализации и тег аутентификации.

Функция decrypt\_aes\_256\_gcm выполняет обратную операцию:

- создает объект дешифратора с использованием переданных ключа, вектора инициализации и тега аутентификации;

- дешифрует данные;

- преобразует байтовое представление обратно в строку;

- возвращает исходные данные в виде строки.

Для облегчения интеграций написанных нами функций сделаем их пакетами.

# ЗАКЛЮЧЕНИЕ

Все супер ну вообще бомба. Цели достигнуты происшествий не было как говорится.СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Соболев, Н. YouTube. Путь к успеху. Как получать фуры лайков и тонны денег / Н. Соболев. — Москва: АСТ, 2016. — 256 с.