

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №33

ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

старший преподаватель

должность, уч. степень, звание

подпись, дата

К.А. Жиданов

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

по курсу: ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

СТУДЕНТ ГР. № 3331

номер группы

подпись, дата

В.В. Узлова

инициалы, фамилия

Санкт-Петербург
2025

Цель работы: создать приложение, представляющее из себя To-Do List с функциями:

- Регистрация/вход в систему
- Добавление/редактирование/удаление задач
- Интеграция с Telegram

А также провести анализ взаимодействия с ИИ во время создания.

Основные компоненты

1. Веб-интерфейс (index.html):
 - Страница авторизации и регистрации.
 - Интерфейс для управления задачами (добавление, редактирование, удаление).
2. Серверная часть (index.js):
 - Реализована на Express.js.
 - Обрабатывает запросы на регистрацию, вход, управление задачами.
 - Интегрирована с Telegram API для обновления меню бота.
 - Работа с базой данных SQLite
3. Telegram-бот (bot.js):
 - Интерфейс для управления задачами через Telegram
 - Поддержка команд: регистрация, вход, управление задачами

Функциональность

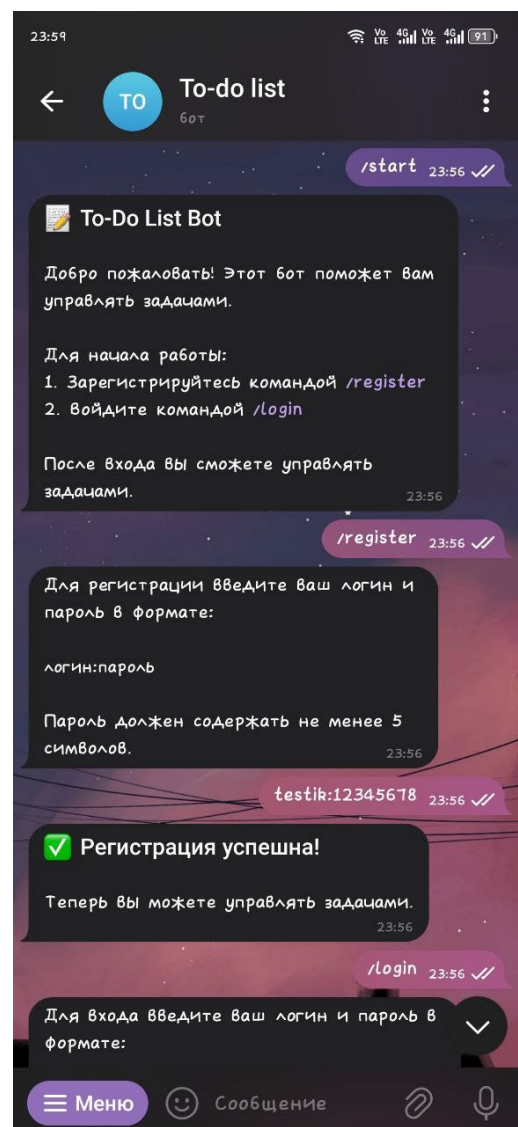
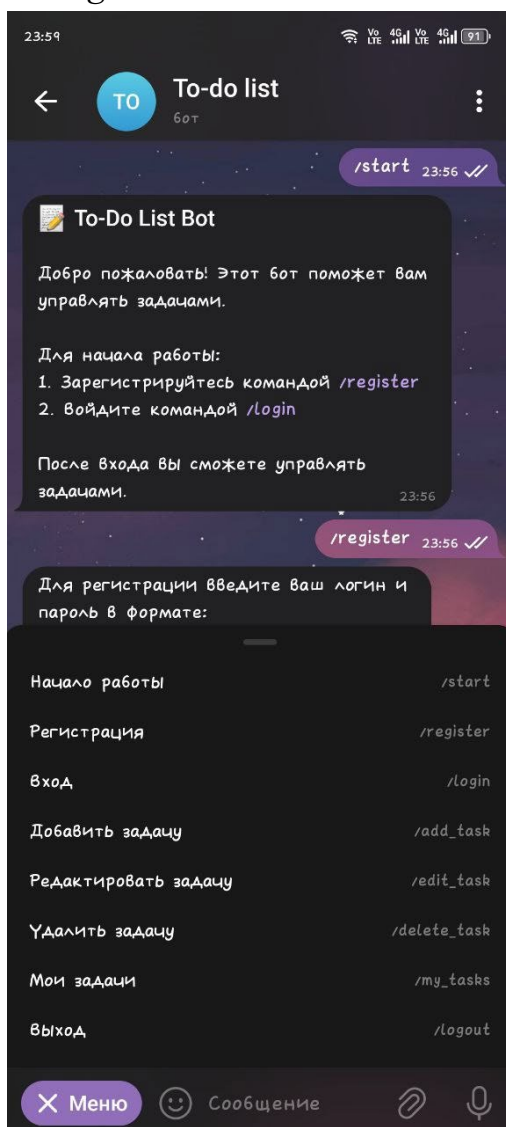
1. Аутентификация:
 - Регистрация и вход через логин и пароль.
 - Сессии пользователей в Telegram.
2. Управление задачами:
 - Добавление, редактирование, удаление задач.
 - Отображение списка задач с нумерацией.
3. Синхронизация:
 - Все изменения синхронизируются между Telegram-ботом и веб-интерфейсом
 - Единая база данных для всех пользователей
4. Интерфейсы:
 - Веб-интерфейс: интуитивно понятный, с подсветкой действий.
 - Telegram-бот: удобные команды и подсказки.

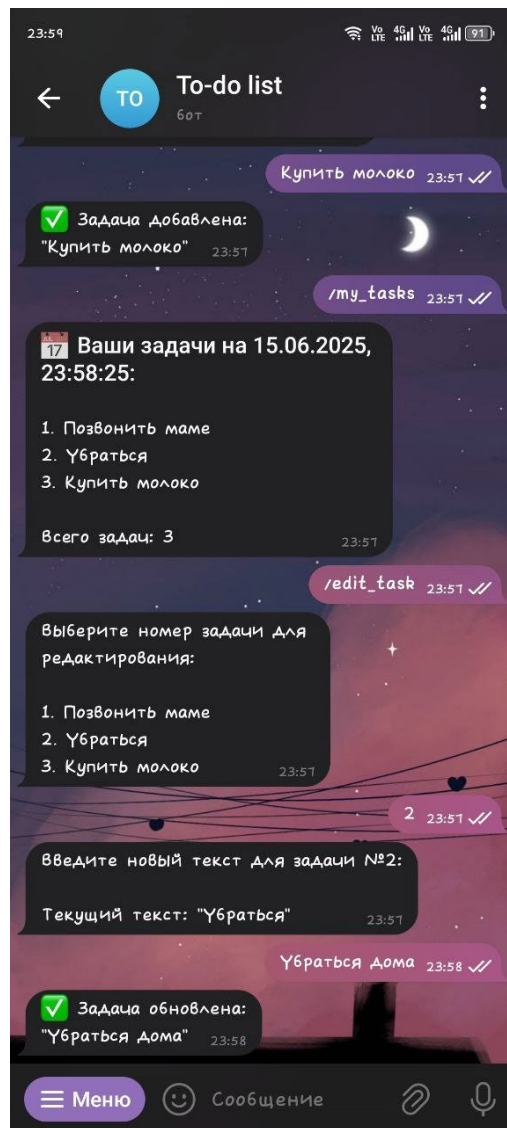
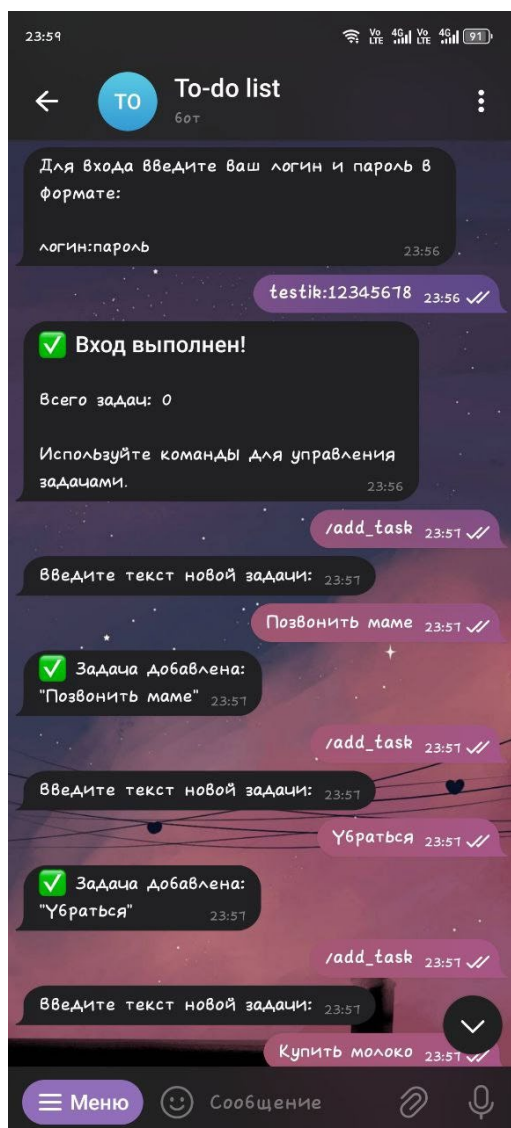
Технические детали

1. Стек технологий

- Backend: Node.js, Express
 - Database: SQLite
 - Telegram API: node-telegram-bot-api
 - Frontend: HTML, CSS, Vanilla JavaScript
 - Дополнительные библиотеки: Axios, Body-parser, CORS, Crypto, Dotenv
2. Запуск приложения:
 - Установка зависимостей: `npm install`.
 - Запуск сервера и бота: `npm run dev`
 - Требуется файл `.env` с токеном бота (`TELEGRAM_BOT_TOKEN`).
 3. Хранение данных:
 - Данные пользователей и задач хранятся в `database.sqlite`.

Примеры использования *Telegram*





Рисунки 1-4. Пример работы telegram бота.

Сайт

Авторизация

| | | | |
|--------|-------|-------|--------------------|
| testik | | Войти | Зарегистрироваться |
|--------|-------|-------|--------------------|

Рисунок 5. Вход на сайте

Мой To-Do List

Новая задача

Добавить

Позвонить маме

Убраться дома

Купить молоко

Рисунок 6. Список задач, который был составлен в telegram боте.

Мой To-Do List

JavaScript с "http://localhost:3000"

Введите новый текст задачи:

Купить молоко и хлеб

OK

Отмена

Новая задача

Добавить

Позвонить маме

Убраться дома

Купить молоко

Рисунок 7. Редактирование задачи.

Мой To-Do List

Новая задача

Добавить

Позвонить маме

Купить молоко и хлеб

Рисунок 8. Список после удаления задачи

Мой To-Do List

JavaScript с "http://localhost:3000"

Задача успешно добавлена!

OK

Купить подарок

Позвонить маме

Купить молоко и хлеб

Мой To-Do List

Новая задача

Добавить

Позвонить маме

Купить молоко и хлеб

Купить подарок

Рисунки 9-10. Список задачи на сайте после добавления новой задачи.

Запросы для ИИ и возникшие проблемы:

1. Запросы для функций добавление/удаление/редактирование элементов
 1. Перепиши функцию добавления элемента в файл index.html по-другому
 2. Добавь функцию редактирования элементов
 3. Отредактируй код в файле index.js в соответствии с изменениями в файле index.html
2. Даны файлы. Измени их так, чтобы на сайте добавилась авторизация пользователя.
3. Интеграция с telegram, разработка бота
 1. Теперь добавь интеграцию этого сайта с телеграмом
 2. Расскажи подробнее, как создать бота в телеграмме (на этом этапе возникли проблемы, так как нейросеть мне говорила делать через доменное имя привязку к сайту, но я собиралась через `http://localhost:3000`)
 3. Как без домена использовать бота в телеграмме, используя только токен? (здесь выяснилось, что можно через Telegram Bot API)
 4. Неоднократные запросы были о возникающих ошибках
 5. Нужно ли что-то менять в package-lock в соответствии с другими прикрепленными файлами?
 6. Всё равно мой бот в телеграмме не работает
 7. Бот стал работать, но не отвечает на команды из меню
 8. Не работает команда /login
 9. Сделай так, чтобы бот работал (продолжалась проблема с запуском бота, хотя все возможные причины были устранены)
 10. После `npm install` всё было хорошо, но в конце выдало вот это: `npm error code EINTEGRITY npm error sha512-...` (после этого выяснилось, что был скачан не весь пакет node.js)
 11. У меня работает, теперь пиши мне файлы, чтобы работала интеграция с моим сайтом
 12. Бот не работает
 13. Как мне сделать, чтобы бот работал именно таким способом? (здесь я прикрепила диалог, как хочу, чтобы работал бот. На этом моменте выяснилось, что нейросеть всё это время неправильно понимала, что конкретно я хочу от неё видеть и она полностью переработала логику бота)
 14. Теперь сделай так, чтобы 1) Можно было добавить задачу; 2) Можно было удалить задачу; 3) Выйти из системы
 15. Добавь редактирование задач и отредактируй прикрепленные файлы под новый bot.js

16. В соответствии с файлом bot.js измени остальные файлы так, чтобы ещё изменился список в кнопке меню у бота
17. Теперь измени эти файлы, чтобы любой человек, использующий бот, мог зарегистрироваться в нём
18. Когда после регистрации хочу снова войти систему, он мне пишет, что такой логин уже занят, из-за этого не получается войти в свою же систему
4. Интеграция с telegram, синхронизация с сайтом
 1. Что мне теперь делать с этими файлами, чтобы я могла также использовать сайт?
 2. Теперь у меня не работает бот и на сайте выдаёт ошибку регистрации
 3. Теперь бот работает, а сайт нет
 4. Я смогла зайти на сайте, но не работают кнопки добавить задачу и синхронизировать с телеграмом
 5. Добавь на сайт возможность редактировать задачи
 6. Осталось только подправить кнопку синхронизации с телеграмом
 7. При попытке на сайте "синхронизировать telegram" пишет "сначала войдите в систему. Для исправления ошибки напиши код полностью
 8. Сделай мне общую базу пользователей и их задач у телеграмма и сайта
 9. На сайте не работают кнопки добавить и синхронизация с телеграмом
 10. Напиши для сайта функцию редактирования, просто отдельно функцию, и скажи, куда её вставить
5. Добавление базы данных (всё это время приложение работало без базы данных, то есть пользователи и их задачи сохранялись в файл и после перезапуска сервера всё терялось, поэтому нужно было добавить SQLite, чтобы после перезапуска всё сохранялось)
 1. Как мне подключить базу данных
 2. Перепиши мои файлы для использования SQLite
 3. Объясни с самого начала, как работать с SQLite
 4. Не работает сохранение задач на сайте и ошибка в синхронизации
 5. Эти файлы работают с SQLite, но на сайте пишет ошибку: данные не сохранены
 6. Cannot access 'data' before initialization что за ошибка на сайте при попытке входа
 7. На сайте теперь не работают кнопки "войти" и "зарегистрироваться"
 8. Теперь на сайте не работают кнопки добавления, редактирования и удаления элементов

После всех этих запросов у меня стали работать и бот, и сайт, а данные пользователей и их задач стали сохраняться в базу данных, благодаря которой после перезапуска сервера все они сохраняются.

Процесс работы:

Запускаем командную строку. Пишем «cd (директория, где лежат файлы)», затем `npm run dev`, чтобы запустить и сайт, и бот. После этого бот будет работать по команде `/start`, а сайт по ссылке `http://localhost:3000`. Данные пользователей и задач сразу синхронизируются между ботом и сайтом из-за общей базы данных.

Файлы `index.html` и `telegram-web-app.js` должны лежать в папке `public` в папке со всеми файлами, но GitHub не позволяет их сохранить с папкой.

Вывод:

В ходе лабораторной работы успешно разработано приложение To-Do List с аутентификацией, управлением задачами и интеграцией с Telegram. Решены проблемы с синхронизацией, базой данных (SQLite) и работой бота. Результат: стабильная система с сохранением данных после перезапуска.