

**Федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Факультет информационных технологий
Кафедра «Информатика и вычислительная техника»**

Направление подготовки/ специальность: системная и программная инженерия

ОТЧЕТ

по проектной практике

Студент: Ортиков Рамиз Умидович Группа: 241-326

Место прохождения практики: Московский Политех, кафедра «Инфокогнитивные
технологии»

Отчет принят с оценкой _____ Дата _____

Руководитель практики: Баринова Наталья Владимировна

Москва 2025

1 Оглавление

ВВЕДЕНИЕ	3
1 ОБЩАЯ ИНФОРМАЦИЯ О ПРОЕКТЕ	4
1.1 Название проекта.....	4
1.2 Цели и задачи проекта.....	4
2 ОБЩАЯ ХАРАКТЕРИСТИКА ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИИ.....	5
2.1 Наименование заказчика.....	5
2.2 Организационная структура	Ошибка! Закладка не определена.
2.3 Описание деятельности.....	5
3 ОПИСАНИЕ ЗАДАНИЯ ПО ПРОЕКТНОЙ ПРАКТИКЕ	6
4 ОПИСАНИЕ ДОСТИГНУТЫХ РЕЗУЛЬТАТОВ ПО ПРОЕКТНОЙ ПРАКТИКЕ.	9
ЗАКЛЮЧЕНИЕ	22
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	23

ВВЕДЕНИЕ

Современная IT-индустрия предъявляет высокие требования к начинающим специалистам, ожидая от них не только теоретических знаний, но и практических навыков работы с ключевыми инструментами разработки. В рамках проектной практики мы целенаправленно осваивали три фундаментальных технологии, составляющих основу профессиональной деятельности в сфере веб-разработки: систему контроля версий Git, язык разметки Markdown, а также технологии HTML и CSS для создания веб-интерфейсов.

Работа с Git и платформой GitHub предоставила нам бесценный опыт командной разработки. Мы научились не только фиксировать изменения в коде, но и грамотно разрешать конфликты слияния веток, возвращаться к предыдущим версиям проекта и организовывать эффективное взаимодействие между участниками команды. Эти навыки особенно важны в условиях реальной разработки, где над проектами одновременно работают несколько специалистов.

Использование Markdown значительно упростило процесс создания технической документации. Освоив его интуитивно понятный синтаксис, мы смогли быстро оформлять README-файлы, инструкции по установке и другие сопроводительные материалы. Этот опыт подчеркнул важность качественной документации в IT-проектах любого масштаба.

Практика разработки статического веб-сайта стала для нас важной вехой в освоении frontend-технологий. Создавая страницу проекта, мы не только применили на практике принципы семантической верстки HTML, но и освоили основные приемы стилизации при помощи CSS. Этот опыт заложил прочную основу для дальнейшего изучения более сложных веб-технологий и фреймворков.

В данном отчете мы подробно расскажем о нашем проекте в рамках дисциплины "Проектная деятельность", а также подведем итоги работы над базовой частью практики. Нам удалось не только освоить ключевые инструменты современного IT-специалиста, но и получить ценный опыт командной работы, который обязательно пригодится нам в будущей профессиональной деятельности.

2 ОБЩАЯ ИНФОРМАЦИЯ О ПРОЕКТЕ

1.1 Название проекта

Полное наименование проекта - Информационная система для автоматизации работы пожарной охраны «Lifeguard»

Краткое наименование – «Lifeguard».

1.2 Цели и задачи проекта

Цель: создать приложение для автоматизации процессов пожарной охраны: учёт дежурных смен, мероприятий, инвентаризации и документооборота.

Задачи проекта:

- реализовать разграничение доступа
- реализовать систему управления снаряжением
- реализовать систему управления дежурными сменами
- презентовать проект на конференции рсс
- реализовать систему управления мероприятиями
- реализовать шаблонизатор документов
- построить серверную инфраструктуру
- выпустить готовый продукт

3 ОБЩАЯ ХАРАКТЕРИСТИКА ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИИ

3.1 Наименование заказчика

Заместитель начальника ОАСФ «СПСО Феникс» Смирнов Александр Сергеевич.

3.2 Описание деятельности

ОАСФ «СПСО Феникс» - студенческий пожарно-спасательный отряд создан для обучения правилам действий в ЧС, предупреждения и проведения аварийно-спасательных и других работ в очагах поражения при возникновении производственных аварий, катастроф и стихийных бедствий на территории Москвы.

4 ОПИСАНИЕ ЗАДАНИЯ ПО ПРОЕКТНОЙ ПРАКТИКЕ

4.1 Настройка Git и репозитория:

Создайте личный или групповой репозиторий на [GitHub](#) или [GitVerse](#) на основе предоставленного [шаблона](#).

Освойте базовые команды Git: клонирование, коммит, пуш и создание веток.

Регулярно фиксируйте изменения с осмысленными сообщениями к коммитам.

4.2 Написание документов в Markdown:

Все материалы проекта (описание, журнал прогресса и др.) должны быть оформлены в формате Markdown.

Изучите синтаксис Markdown и подготовьте необходимые документы.

4.3 Создание статического веб-сайта:

Вы можете использовать **только HTML и CSS** для создания сайта, если освоение более сложных инструментов представляется трудным. Это делает задание доступным для студентов с базовым уровнем подготовки.

Создайте новый сайт об основном проекте по дисциплине «Проектная деятельность», выберите тему и добавьте контент. Оформление и наполнение сайта должны быть уникальными (не совпадать с работами других студентов) более, чем на 50%.

Сайт должен включать:

- **Домашнюю страницу** с аннотацией проекта.
- **Страницу «О проекте»** с описанием проекта.
- **Страницу или раздел «Участники»** с описанием личного вклада каждого участника группы в проект по «Проектной деятельности».
- **Страницу или раздел «Журнал»** с минимум тремя постами (новостями, блоками) о прогрессе работы.
- **Страницу «Ресурсы»** со ссылками на полезные материалы (ссылки на организацию-партнёра, сайты и статьи, позволяющие лучше понять суть проекта).

Оформите страницы сайта графическими материалами (фотографиями,

схемами, диаграммами, иллюстрациями) и другой медиа информацией (видео).

Ожидаемое время: изучение и настройка — 10–14 часов, дизайн и наполнение — 4–8 часов.

4.4 Взаимодействие с организацией-партнёром:

Организируйте взаимодействие с партнёрской организацией (визит, онлайн-встреча или стажировка).

Участвуйте в профильных мероприятиях по тематике проекта и профилю организации-партнёра (конференции, выставки, митапы, семинары, хакатоны и др.).

Напишите отчёт в формате Markdown с описанием опыта, полученных знаний и связи с проектом. Отчёт добавьте в репозиторий и на сайт.

4.5 Отчёт по практике

Составьте отчёт по проектной (учебной) практике.

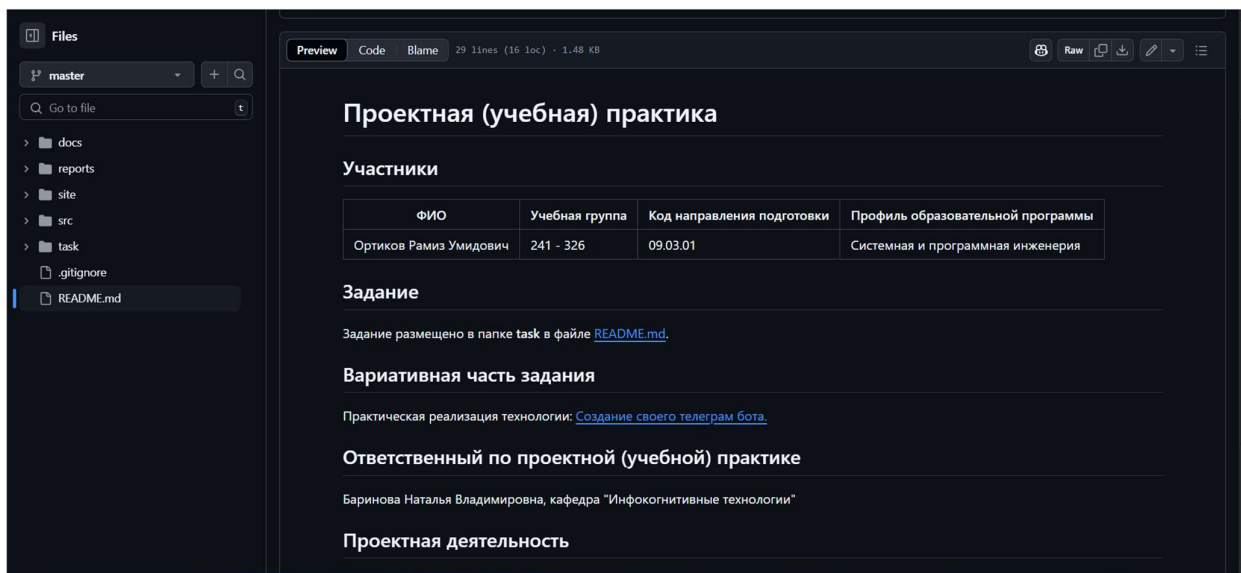
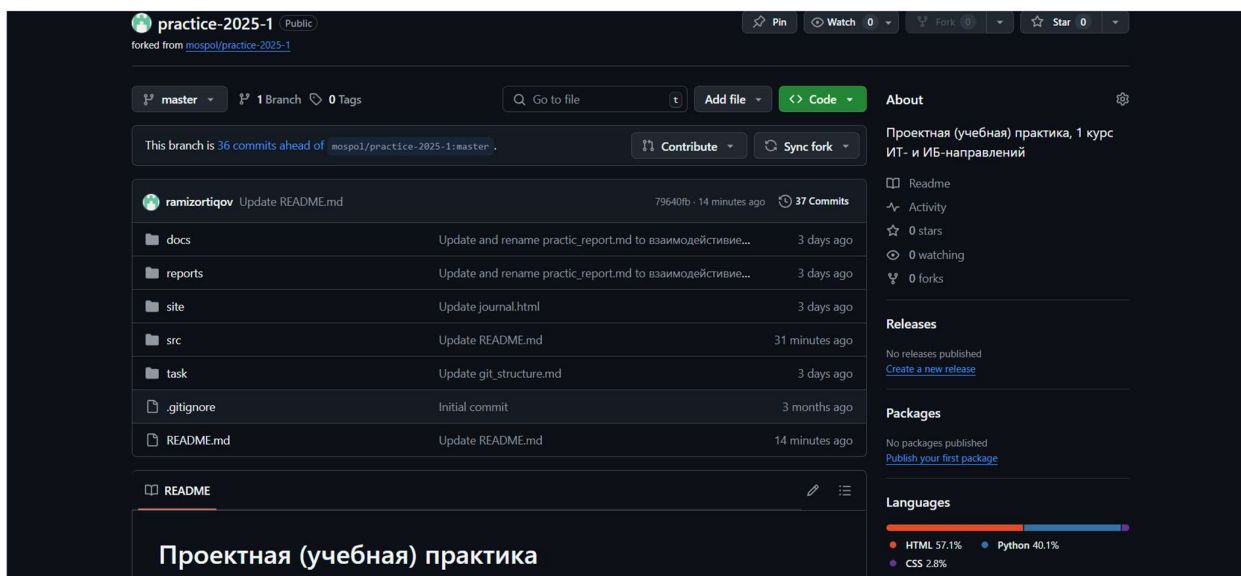
Вариативная часть задания по проектной практике представляет собой:

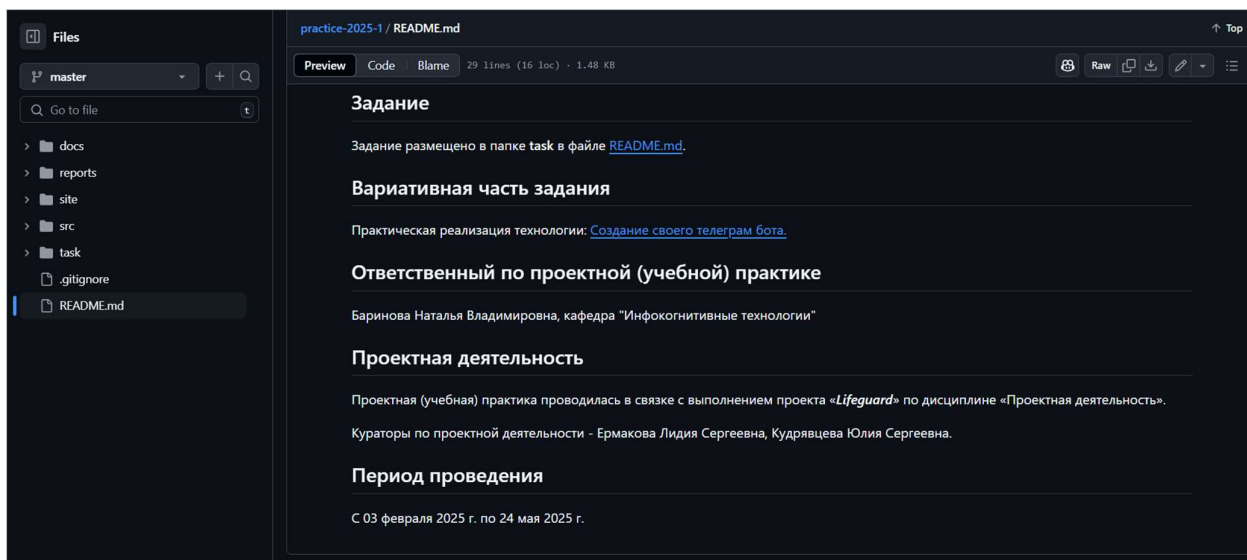
1. Выбор любой технологии (тематике) из списка, представленного в репозитории [codecrafters-io/build-your-own-x](https://github.com/codecrafters-io/build-your-own-x).
2. Согласовать внутри команды выбранную тему. Выбрать стек технологий (подсказки также есть в репозитории).
3. Провести исследование: изучите, как создать выбранную технологию с нуля, воспроизведите практическую часть.
4. Создать подробное описание в формате Markdown.
5. Создать техническое руководство или tutorial по созданию проекта на выбранную тему. Для визуализации архитектуры, процессов и прочего используйте разные типы диаграмм UML, схемы, графики, таблицы.
6. Сделать модификацию проекта согласно полученным знаниям и навыкам в течение года (творческий пункт, самостоятельно выбираете в какой части модифицировать). Описать в технической документации модификации
7. Сделать видео презентацию выполненной работы (цель, задачи, как решали, демонстрация работоспособного результата).
8. Подготовить финальный отчет (в хронологической последовательности опишите этапы работы, отдельно должны быть представлены индивидуальные планы каждого участника).

5 ОПИСАНИЕ ДОСТИГНУТЫХ РЕЗУЛЬТАТОВ ПО ПРОЕКТНОЙ ПРАКТИКЕ

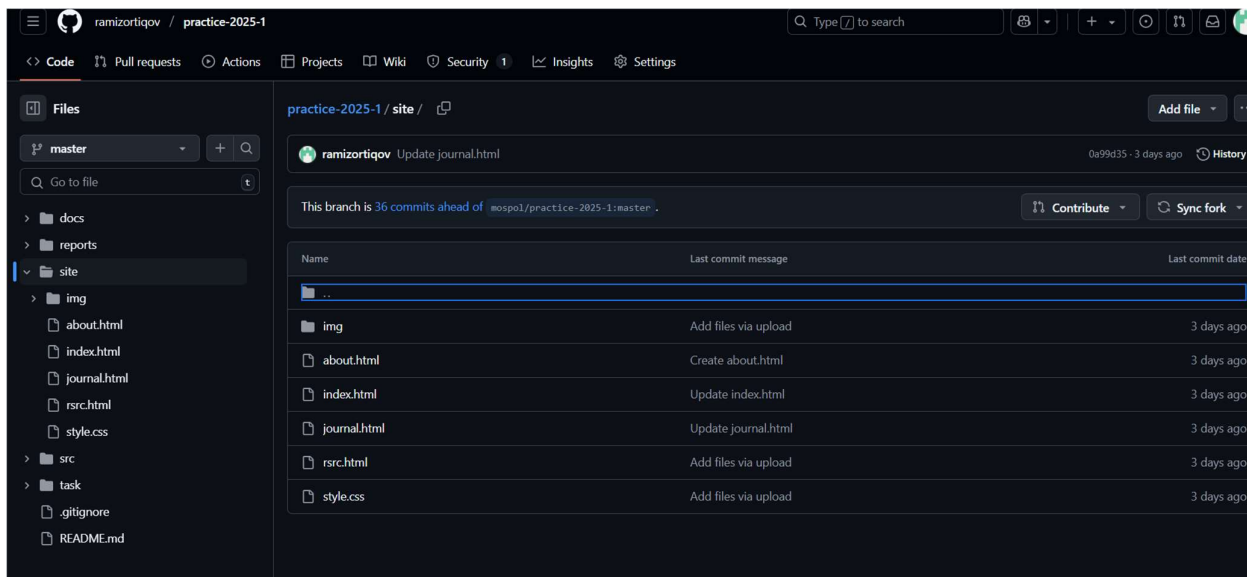
5.1 Базовая часть

Репозиторий на GitHub был успешно создан на основе предоставленного шаблона и заполнен в соответствии с требованиями базовой части проектной практики





Создана папка **src** для хранения файлов вариативной части задания. В папке **site** был размещен статический сайт:



Сайт был разработан вручную с использованием только HTML и CSS. Все страницы соответствуют требованиям задания. Ниже представлены скриншоты сайта.

Информационная система для автоматизации работы пожарной охраны «Lifeguard»



Автоматизация пожарной охраны

Цель проекта — создать приложение для автоматизации процессов пожарной охраны: учёт дежурных смен, мероприятий, инвентаризации и документооборота.

Lifeguard

[Главная](#) [О проекте](#) [Журнал](#) [Ресурсы](#)

О проекте

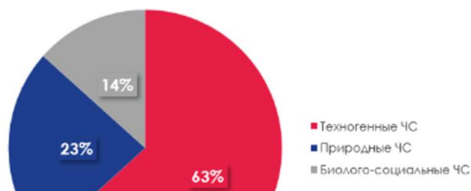
Lifeguard — это кроссплатформенное приложение, созданное для автоматизации процессов пожарной охраны. Оно упрощает учёт смен, мероприятий, инвентаря и документооборота.

Цель

Мобильное, десктопное и web-приложение для автоматизации процессов пожарной охраны таких как: -учёт дежурные смен -учет мероприятий -учёт инвентаризации -автоматизация документооборота Эта информационная система решает несколько проблем: облегчение бюрократических задач пожарной охраны, хранение и обработка актуальных данных по личному составу и инвентарю, предоставление служащим удобного расписания мероприятий. Сейчас некоторые функции выполняют другие приложения, некоторые из них не имеющие спецификации для пожарной охраны, в нашем случае, мы хотим объединить все необходимые для функционирования пожаро-спасательной части аспекты в одном приложении

Актуальность

В Московской области в 2024 г. Зарегистрировано 2227 пожара. В результате пожаров погибло 105 чел. 102 получило травмы, из них 8 детей Ущерб от пожаров составил более 1 млн руб.



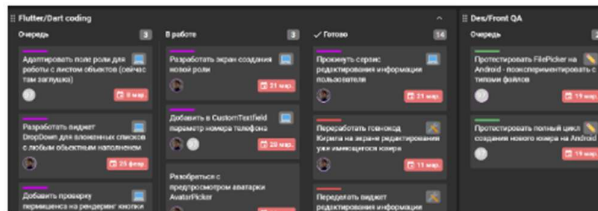
Изучение технологий

На первом этапе я изучил язык программирования **Dart** и фреймворк **Flutter** — основные инструменты для разработки мобильной версии приложения Lifeguard. Ознакомился с основами синтаксиса, принципами работы с виджетами, состоянием, навигацией и подключением стилей. Это дало базу для реализации интерфейса и логики приложения.



Обсуждение задач в Kaiten

Команда использует сервис **Kaiten** для планирования и распределения задач. Все участники активно обсуждают текущие и предстоящие задачи, а также фиксируют прогресс на доске с колонками *Backlog*, *В работе* и *Готово*. Такой подход помогает эффективно организовать рабочий процесс и контролировать сроки выполнения.



Полезные ресурсы

[Главная](#) [О проекте](#) [Журнал](#) [Ресурсы](#)

Организации и партнёры

- [ОАСФ «СПСО Феникс»](#)

Статьи и материалы

- [Habr: Как автоматизировать учёт в силовых структурах](#)
- [Wikipedia: Пожарная охрана](#)
- [Figma — для прототипирования интерфейсов](#)

Технологии, используемые в проекте

- [Flutter](#) — фреймворк для мобильного и веб-приложения
- [Go](#) — язык программирования для серверной части
- [Kaiten](#) — инструмент управления задачами

© 2025 Lifeguard Project Team

Кроме того, в рамках практики я принял участие в университетских мероприятиях. Отчёт о взаимодействии с организацией-партнёром размещён в репозитории, в папке [docs](#).

5.2 Вариативная часть

Анализ работы Telegram-ботов, включая изучение API, системной архитектуры и связей между её компонентами.

Задачи:

- изучить документацию python и библиотеки telebot.
- проанализировать существующие реализации telegram-ботов на python.
- определить тематику, функционал и основные команды бота.

Результаты:

- получено понимание работы Telegram API и архитектуры ботов.
- выделены ключевые команды для реализации.

5.3 Разработка базовой функциональности

Создание минимально рабочей версии Telegram-бота для получения данных о погоде и курсах валют.

Задачи:

1. Настройка окружения:

- Установка необходимых библиотек (например, python-telegram-bot).
- Регистрация бота через BotFather и получение API-токена.

2. Реализация базового функционала:

- Обработка команд (/start, /weather, /exchange).
- Интеграция с API погодных сервисов (OpenWeatherMap).
- Подключение к API курсов валют.

3. Тестирование:

- Проверка корректности ответов бота на команды.
- Обработка возможных ошибок (неверный ввод, недоступность API).

Результат:

- Рабочий Telegram-бот, способный по запросу выдавать актуальную информацию:
 - Погоду в указанном городе.

- Курсы валют (например, USD, EUR).

5.4 Модификация

Цель:

Улучшение взаимодействия с пользователем за счет удобного интерфейса: кнопок, структурированных сообщений и вариантов ответов.

Задачи:

1. Настройка интерактивного интерфейса:

- Добавление **кнопок** для быстрого выбора команд.
- Реализация **меню**.

2. Оформление сообщений:

- Форматирование текста: выделение, списки, ссылки.
- Добавление **эмодзи** для наглядности.

3. Варианты ответов:

- Предопределенные варианты
- Обработка быстрых ответов через `callback_query` (инлайн-кнопки).

Результат:

- Бот становится удобнее для пользователя.
- Снижается количество ошибок ввода.
- Упрощается навигация по функциям.

5.5 Документирование и создание руководства

Цель:

Разработка понятного технического руководства по созданию Telegram-бота на Python для новичков, включая полное описание процесса разработки, примеры кода и архитектурные решения.

Задачи:

1. Детализация этапов разработки:

- Описание настройки окружения, регистрации бота, работы с API.
- Разбор реализации каждой функции (погода, курсы валют, интерфейс).

2. Структурирование документации:

- Включение исследования предметной области и обоснования выбора технологий.
- Описание внедренных улучшений (кнопки, форматирование сообщений).

3. Практические материалы:

- Пошаговые инструкции с примерами кода (например, обработка команд `/start`, `/weather`).
- Диаграммы архитектуры (блок-схемы, последовательность действий).

4. Оформление и публикация:

- Подготовка документов в формате **Markdown** (README.md).
- Размещение в **Git-репозитории** с разделами:
 - `docs/` — руководство и описание,
 - `src/` — исходный код бота.

Результаты:

1. Техническое руководство для начинающих разработчиков, содержащее:

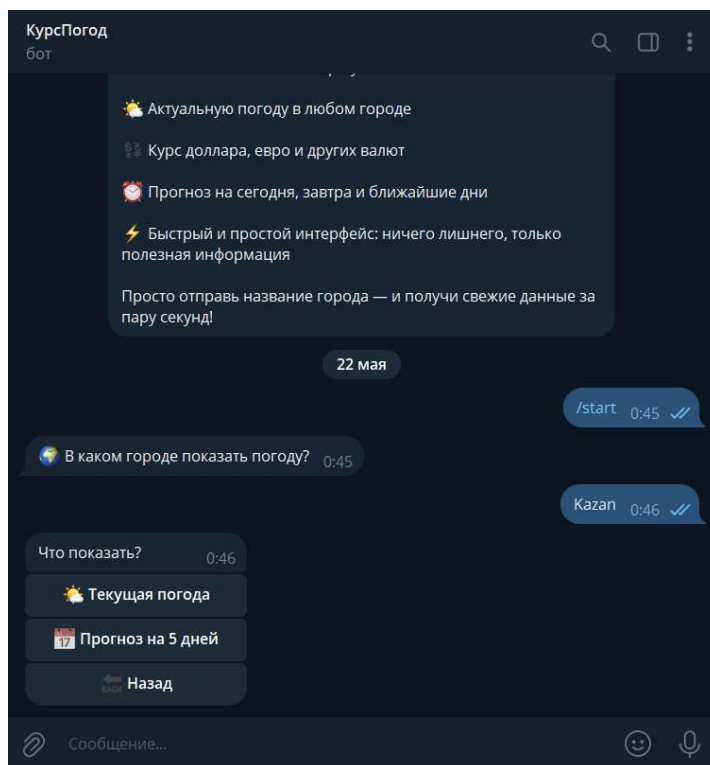
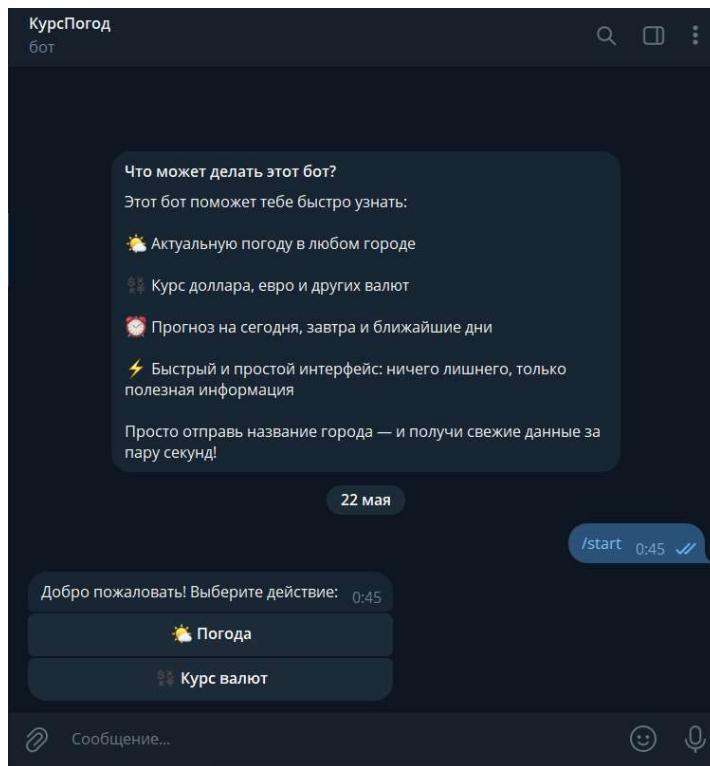
- Пояснения по установке и настройке.
- Примеры реализации функций с кодом.
- Рекомендации по доработке (например, подключение новых API).

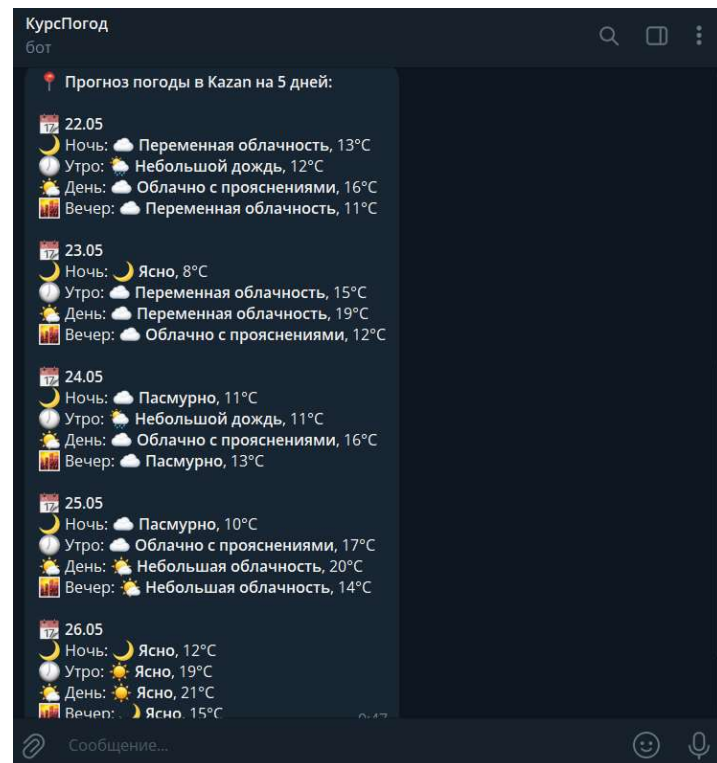
2. Описание технологии с перечнем:

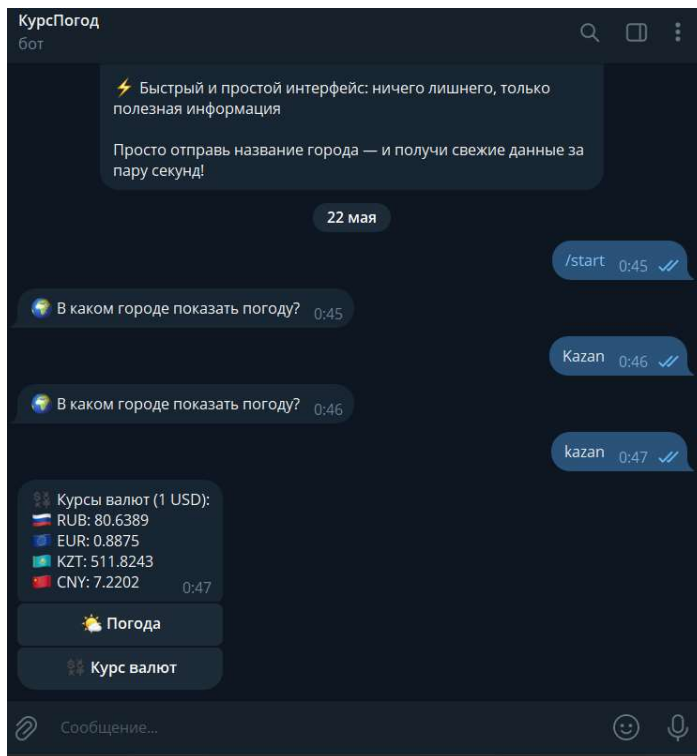
- Используемых библиотек (`python-telegram-bot`, `requests`).
- Списка команд и функций бота.
- Архитектура бота.

3. Документы размещены в **Git** с открытым доступом.

Ниже представлены скриншоты работы бота:







ЗАКЛЮЧЕНИЕ

Базовая часть проектной практики стала ценным этапом в освоении ключевых профессиональных навыков. В ходе работы мы детально изучили платформу GitHub, освоили создание документов в формате Markdown и получили практический опыт разработки статических веб-сайтов с использованием HTML и CSS. Приобретённые компетенции, включая работу с системами контроля версий, оформление технической документации и вёрстку веб-страниц, представляют собой фундаментальные умения, востребованные в IT-сфере, и были успешно применены на практике.

В ходе выполнения вариативной части был успешно реализован и детально проработан Telegram-бот для получения актуальной информации о погоде и курсах валют. Основные достижения включают:

1. Изучение технологий:

- Освоен принцип работы Telegram Bot API
- Изучены библиотеки `python-telegram-bot`

2. Практическая реализация:

- Создана базовая функциональность с обработкой команд
- Реализована интеграция с внешними API (OpenWeatherMap, курсы валют)
- Разработан удобный интерфейс с кнопками и форматированными сообщениями

3. Документирование:

- Составлено подробное техническое руководство
- Создана понятная документация для пользователей и разработчиков
- Проект размещен в открытом доступе на GitHub

4. Полученные навыки:

- Работа с Telegram Bot API
- Интеграция сторонних сервисов
- Обработка и валидация пользовательского ввода
- Создание интерактивного интерфейса
- Написание технической документации

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Как создать Telegram бот с использованием Python – URL: <https://www.freecodecamp.org/news/how-to-create-a-telegram-bot-using-python/> (дата обращения: 20.05.2025).
2. Репозиторий GitHub, созданный в рамках проектной практики – URL: <https://github.com/ramizortiqov/practice-2025-1/tree/master> (дата обращения: 20.05.2025).
3. Оригинальный репозиторий по проектной практике 2025 года. – URL: <https://github.com/mospol/practice-2025-1> (дата обращения: 20.05.2025).
4. Репозиторий GitHub проекта «Lifeguard» – URL: <https://github.com/nakito-own/Lifeguard-app> (дата обращения: 20.05.2025).