

Федеральное государственное автономное образовательное учреждение высшего
образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра «Информационные системы и технологии»

Направление подготовки/ специальность: 09.03.02 Автоматизированные системы обработки
информации и управления

ОТЧЕТ

по проектной практике

Студент: Захарова Кира Игоревна Группа: 241-334

Место прохождения практики: Московский Политех, кафедра Информационные
системы и технологии

Отчет принят с оценкой _____ Дата _____

Руководитель практики: Рябчикова Анна Валерьевна

Москва 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. Общая информация о проекте	4
2. Общая характеристика деятельности организации	6
3. Описание задания по проектной практике	8
4. Описание достигнутых результатов по проектной практике	
4.1. Базовая часть	10
4.2. Вариативная часть	15
ЗАКЛЮЧЕНИЕ	24
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	25

Введение

Данный отчет содержит итоги проектной практики, реализованной в процессе обучения по направлению «Информационные системы и технологии». Практическая деятельность осуществлялась на кафедре «Информационных систем и технологий» Московского Политехнического университета и включала два основных направления: разработку веб-сайта согласно проекту по дисциплине «Проектная деятельность» и создание Telegram-бота, который станет удобным инструментом для управления личными финансами.

Общая информация о проекте

Наименование проекта: Помощь людям с ОВЗ: сайт для оказания экстренной помощи (I курс).

Актуальность проекта: Люди с ограниченными возможностями часто сталкиваются с множеством препятствий в повседневной жизни. Ограничения в мобильности, сложности в коммуникации и недостаточная социальная интеграция приводят к ощущению изоляции и зависимости от окружающих. Проект способен решить данную проблему и наладить быструю связь между волонтерами и людьми с ОВЗ, как для бытовых задач, психологической помощи, так и для экстренных ситуаций.

Актуальность проекта обусловлена следующими факторами:

1. Специализированный сайт обеспечит прямую связь между волонтерами и нуждающимися, делая помощь более быстрой и эффективной.
2. Сайт поможет людям с ОВЗ получить доступ к информации и ресурсам, улучшая их качество жизни, способствуя интеграции в общество.
3. Платформа будет источником информации о потребностях людей с ОВЗ, что поможет организовать необходимую помощь.
4. Проект соответствует современным тенденциям создания общества, где все граждане имеют равные права и возможности

Проблематика: Сложность в выборе организации, в которую при необходимости экстренной помощи могут обращаться лица с ОВЗ. Разные интерфейсы сайтов, в которых сложно ориентироваться.

Цель проекта: Создание сайта для взаимодействия волонтеров и лиц с ОВЗ в одном из административных округов г. Москва.

Задачи проекта:

1. Создание дизайна главных страниц.
2. Разработка интерактивных элементов для главной страницы.
3. Верстка сайта.
4. Создание базы данных
5. Создание политики конфиденциальности
6. Адаптация сайта под мобильные устройства
7. Запустить продвижение сайта в соцсетях
8. Составление диаграммы Ганта
9. Составление проектно-технической документации
10. Защита проекта

Общая характеристика деятельности организации

Несмотря на отсутствие указания заказчика в паспорте проекта, разработка веб-сайта осуществляется под поддержкой и рекомендациями ресурсного центра Мосволонтер. Таким образом, в качестве заказчика проекта можно рассматривать ресурсный центр добровольчества в России Мосволонтер.

Структура организации:

Мосволонтер — это организация, которая объединяет волонтеров Москвы и занимается координацией и поддержкой их деятельности. Структура Мосволонтера представляет из себя следующие отделы:

Центральный офис: руководящий и координирующий центр, который отвечает за стратегическое планирование, разработку программ и инициатив, а также взаимодействие с государственными органами и партнерами.

Отделы и секции: включают в себя различные подразделения, которые занимаются конкретными аспектами волонтерской работы, такими как набор и обучение волонтеров, организация мероприятий, работа с молодежью, экологические проекты и т.д.

Координаторы волонтеров: в каждом районе Москвы назначаются координаторы, которые отвечают за работу с волонтерами на местах, взаимодействуют с местными организациями и общественными учреждениями, а также обеспечивают реализацию проектов на локальном уровне.

Команды волонтеров: состоят из активных участников, которые занимаются непосредственной волонтерской деятельностью. Команды могут формироваться для участия в конкретных проектах или быть постоянными группами, работающими в определенной сфере (например, помощь пожилым людям, экологические инициативы и т.д.).

Тренинговые и образовательные программы: направлены на повышение квалификации и развитие навыков волонтеров через проведение семинаров, мастер-классов и тренингов.

Совет волонтеров: консультативный орган, который помогает в принятии решений и разработке новых инициатив, учитывая мнение активных участников движения.



Описание деятельности ППО Московского политеха:

Ресурсный центр «Мосволонтёр» является крупнейшей в России организацией, посвящённой развитию и поддержке волонтёрского движения в Москве. Основные направления его работы включают создание волонтёрских команд для городских и всероссийских мероприятий, таких как Международная выставка-форум «Россия» и Всемирный фестиваль молодёжи.

Центр предоставляет бесплатные образовательные программы для волонтёров всех уровней, ведёт электронные книжки волонтёров, в которых фиксируются их достижения и количество часов участия. Кроме того, «Мосволонтёр» занимается сбором и распределением гуманитарной помощи через штабы «Москва Помогает».

Организация активно взаимодействует с партнёрами, предоставляя им консультации и информационную поддержку, а также способствует развитию навыков волонтёров. «Мосволонтёр» охватывает широкую аудиторию — от школьников до пожилых людей, объединяя более 285 000 зарегистрированных волонтёров и ежегодно организуя более 2 100 волонтёрских команд.

Описание задания по проектной практике

В рамках проектной практики, я занималась созданием видеоролика, подготовкой проектной документации, а также составлением диаграммы Ганта для проекта, направленного на социальную интеграцию людей с ограниченными возможностями здоровья (ОВЗ). Наш проект, «Поддержка рядом», имел целью разработку сайта для облегчения взаимодействия между волонтерами и лицами с ОВЗ, что способствовало бы их интеграции в общество и упрощению коммуникаций.

Планирование и организация работы

Проектная деятельность осуществлялась в составе команды, где роли и задачи распределялись с учётом компетенций и интересов каждого участника. Я в основном сосредоточилась на визуальном контенте и документации проекта. Распределение ролей было организовано в начале проекта, что позволило каждому члену команды сосредоточиться на своих сильных сторонах. Для эффективной организации работы я использовала диаграмму Ганта, которая помогала визуализировать сроки выполнения задач и координировать командные усилия.

Сложности и их преодоление

В процессе работы возникали трудности, связанные с необходимостью синхронизации различных аспектов проекта, таких как дизайн и программирование. Для их преодоления мы организовывали регулярные встречи, где обсуждали текущие проблемы и находили совместные решения. Важной частью процесса стало гибкое планирование, позволяющее адаптироваться к изменениям и корректировать сроки.

Развитие профессиональных и коммуникативных навыков

Участие в коллективной работе способствовало развитию профессиональных и коммуникативных навыков. В процессе выполнения проектных задач я улучшила навыки тайм-менеджмента, научилась эффективнее расставлять приоритеты и координировать действия в команде. Благодаря взаимодействию с различными специалистами, я повысила свою способность к адаптации и быстрому обучению новым технологиям и инструментам.

Описание достигнутых результатов по проектной практике

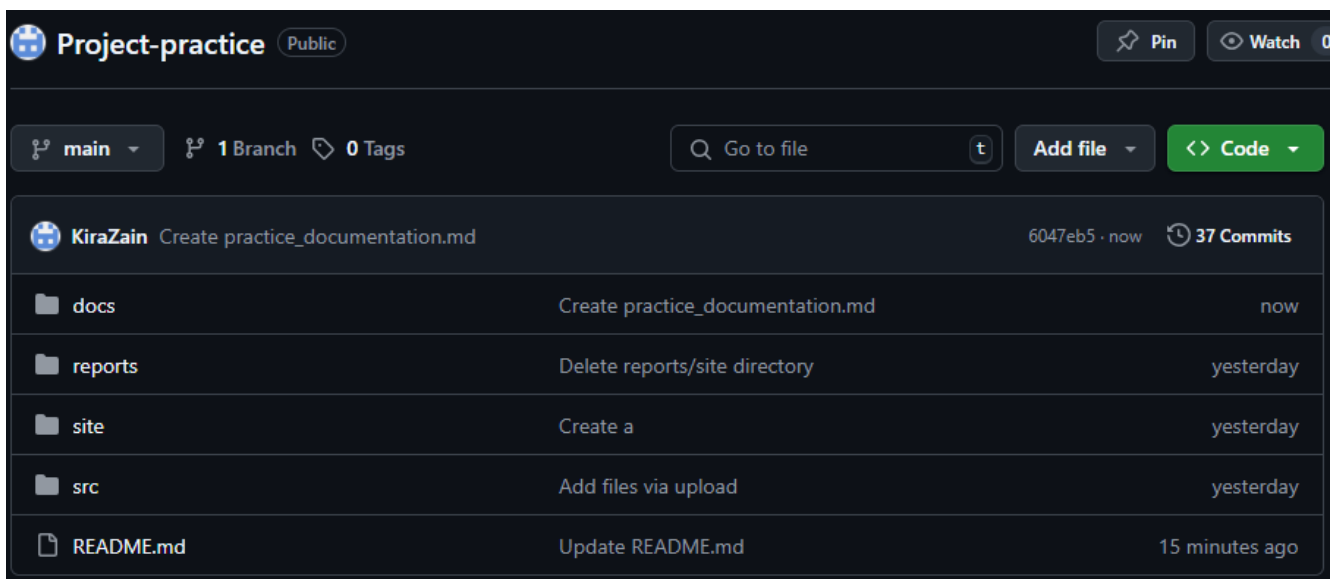
Базовая часть

Настройка Git и репозитория.

Первый шаг для работы с Git через веб-интерфейс — регистрация на платформе GitHub. Была создана учетная запись. На главной странице GitHub с помощью кнопки "New" создаем репозиторий. В открывшейся форме указывается имя репозитория. Выбирается тип репозитория: публичный или приватный, для проекта использовался публичный. Отменяем опцию "Initialize this repository with a README", чтобы автоматически создать файл README.md. Далее с помощью кнопки "Create repository" создается репозиторий.

Для управления версиями проекта в процессе изучения был создан репозиторий на платформе GitHub. Структура репозитория организована следующим образом:

- README.md: основной файл, содержащий описание практики.
- docs: папка, в которой хранится описание работы над Telegram-ботом.
- reports: папка, в которой хранится отчет по практике .md.
- site: директория, содержащая исходный код веб-сайта.
- src: директория, содержащая исходный код бота.



Документация в Markdown.

Все текстовые файлы проекта, включая .md файлы, написаны в формате Markdown. Этот формат широко используется для создания различных текстовых документов, таких как статьи, отчёты и документация, благодаря своей простоте и удобочитаемости. В рамках проекта Markdown помогает структурировать и организовать информацию, делая её доступной и понятной для всех участников команды. Файл README.md является ключевым компонентом любого репозитория. Он служит визитной карточкой проекта и содержит основную информацию, такую как:

- Описание проекта: Краткое пояснение целей и задач, которые решает проект.
- Использование: Примеры использования и основные команды для взаимодействия с проектом.
- Вклад: Рекомендации и правила для тех, кто хочет внести свой вклад в развитие проекта или создать что-то похожее.

Этот файл играет важную роль в понимании структуры и назначения проекта, и является первым документом, с которым знакомятся новые участники или пользователи.

Создание статического веб-сайта.

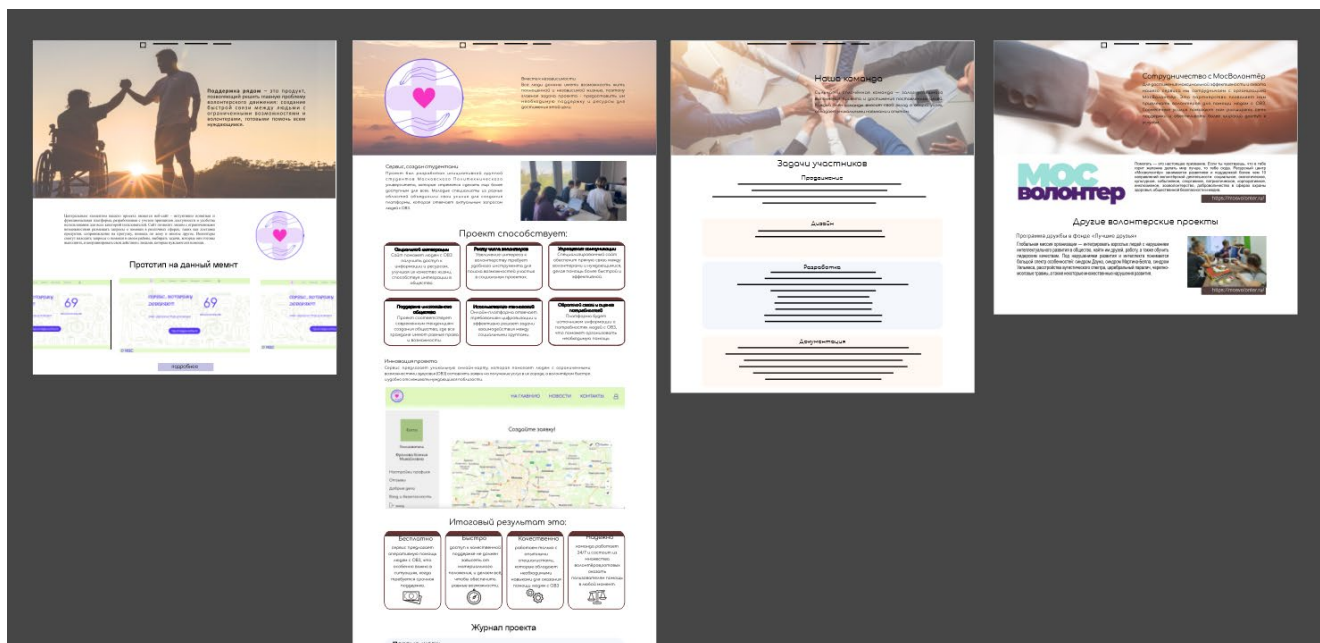
Для создания сайта были выделены следующие задачи:

- Разработка макета сайта
- Сбор информации и написание текстовой части
- Создание стилей
- Разработка сайта

Разработка макета сайта

Сайт включает 4 страницы: поддержка рядом, о проекте (с разделом журнал), участники и ресурсы.

При создании макета использовался шрифт Comfortaa, 20пт для общего текста и 25-30 для заголовка. Были выбраны тёплые, спокойные оттенки и подобраны соответствующие фотографии для шапок сайта. Отступы от краев 200px



Сбор информации и написание текстовой части

Был проведен анализ похожих проектов для помощи людям с ОВЗ. Данные организации были представлены в разделе Ресурсы, чтобы все желающие могли сильнее проникнуться данной темой. Также прошла работа по выделению ключевых моментов и преимуществ нашего проекта, которые дают главное понимание о целях платформы.

Создание стилей

Стили сайта были написаны с использованием CSS. Основные этапы разработки:

1. Базовая настройка: сбрасываются стандартные отступы (margin, padding) для всех элементов и используется box-sizing: border-box через универсальный селектор *, что помогает в корректном расчете размеров.
2. Макет и выравнивание: используется flexbox для выравнивания элементов в навигации и других контейнерах, что обеспечивает гибкость и простоту адаптивной верстки.

3. Эстетика и визуальные элементы: были добавлены тени (box-shadow) и скругления (border-radius) для карточек, что придает современный и объемный вид элементам интерфейса.
4. Цветовая схема: используются нейтральные цвета для текста и фона, а также акцентные цвета для украшения карточек, которые выделяются благодаря цветным полосам в верхней части.
5. Изображения: изображения оптимизированы с помощью max-width: 100%, чтобы они корректно масштабировались на всех устройствах, сохраняя пропорции.
6. Контейнеры и отступы: контейнеры выровнены и центрированы с помощью отступов и маргинов, что создает аккуратный и организованный макет.
7. Типографика: задаются конкретные стили для заголовков и текстовых блоков, включая размеры шрифтов и межстрочные интервалы (line-height), что улучшает визуальное восприятие и читабельность текста.

Взаимодействие с организацией-партнёром

18 апреля наша команда проекта провела встречу с представителями партнера «МосВолонтер», целью которой было обсуждение стратегии и возможностей нашего дальнейшего сотрудничества.

Во-первых, мы сосредоточились на развитии нашего проекта, обсуждая, как сделать помощь людям с ограниченными возможностями здоровья (ОВЗ) еще более доступной и эффективной. Были предложены различные идеи и инициативы, которые могут существенно улучшить оказываемую поддержку и доступ к ресурсам.

Во-вторых, мы обсудили потенциал нашего сайта, который предоставляет полезные ресурсы для людей с ОВЗ. Стало очевидно, что наш ресурс имеет значительную ценность не только для некоммерческих организаций, но и для государственных структур. Представители «МосВолонтер» подтвердили, что такой

сайт крайне востребован и может существенно способствовать повышению эффективности координации помощи при должном результате проекта.

Третьим важным моментом стало обсуждение формализации нашего сотрудничества. Хотя «МосВолонтер» не может напрямую выступить заказчиком нашего проекта, они предоставили нам ценный совет обратиться в префектуру. Префектура обладает полномочиями стать официальным заказчиком, что открывает перед нами новые возможности для получения официальной поддержки и признания. В дополнение к этому, волонтеры выразили готовность поддержать нас в качестве партнеров, что укрепляет наше сотрудничество и дает уверенность в устойчивом развитии проекта.

Подводя итоги, встреча продемонстрировала высокий потенциал нашего сотрудничества и открыла новые пути для развития. Мы получили ценные рекомендации и поддержку, которые помогут нам в дальнейшем продвижении нашего проекта.



Вариативная часть

Разработка Telegram-бота Финансовый помощник

Задачи, которые необходимо выполнить в ходе работы:

- Изучить различные типы ботов, их классификацию и ключевые компоненты архитектуры.
- Проанализировать, какие боты в данной среде уже существуют и активно используются.
- Изучить создание Telegram-ботов, архитектуру построения кода для их написания
- Сопоставить всю полученную информацию, на основе анализа расписать функции бота и выполнить пути реализации

Изучение типов ботов, использующихся на рынке.

Существует множество типов ботов, каждый из которых имеет свои уникальные функции и возможности. В процессе изучения я выделила несколько наиболее популярных типов:

Информационные боты. Информационные боты предназначены для предоставления пользователям актуальной информации и новостей. Они могут охватывать различные темы, такие как политика, экономика, спорт, технологии и многое другое.

- **Развлекательные боты.** Эти боты созданы для того, чтобы развлекать пользователей и предоставлять им способы расслабления и отдыха.
- **Образовательные боты.** Образовательные боты помогают пользователям в обучении и развитии новых навыков.
- **Торговые боты.** Торговые боты облегчают процесс покупки и продажи товаров и услуг.
- **Финансовые боты.** Финансовые боты предназначены для управления личными финансами и получения информации о финансовых рынках.

- Поддержка и обслуживание клиентов. Эти боты используются компаниями для улучшения взаимодействия с клиентами.

Однако в моём случае особое внимание привлекли финансовые боты. Они предназначены для того, чтобы помогать в управлении личными финансами, отслеживании расходов, составлении бюджета и планировании накоплений. Это всегда было актуальным и действительно способно упростить жизнь.

Анализ существующих экономических ботов.

Money Lover	Бот для управления личными финансами и бюджетом.	Отслеживание расходов, планирование бюджета, напоминания.
CoinMarketCap Bot	Бот для отслеживания криптовалют и финансовых рынков.	Котировки криптовалют, новости рынка, оповещения о ценах.
Splitwise Bot	Бот для разделения расходов между друзьями или коллегами.	Создание групп, отслеживание долгов, уведомления о платежах.
Personal Finances	Бот для учета личных финансов и планирования бюджета.	Управление доходами и расходами, создание отчетов.
Expense Manager	Бот для учета ежедневных расходов и финансового планирования.	Ведение записей о расходах, создание бюджетов, напоминания.
Cryptocurrency Bot	Бот для отслеживания и анализа криптовалютных рынков.	Графики цен, новости, прогнозы, уведомления о рыночных изменениях.

Создание Telegram-ботов, архитектура построения кода.

Основные шаги в создании Telegram-бота

Регистрация бота: для начала необходимо создать бота в Telegram. Это делается с помощью официального бота Telegram — BotFather. После создания бот получает уникальный токен, который используется для взаимодействия с Telegram API.

Выбор платформы и языка программирования: Telegram-боты могут быть разработаны на различных языках программирования. Наиболее популярные — Python, JavaScript, PHP, Java и другие.

Настройка окружения: установка необходимых библиотек и зависимостей. Например, для Python часто используют библиотеку python-telegram-bot.

Настройте сервер или используйте облачные платформы, такие как Heroku или AWS, для развертывания бота.

Тестирование и отладка:

Проверка работы бота в различных сценариях. Исправление ошибок и оптимизация кода для улучшения производительности.

Архитектура построения кода

Архитектура Telegram-бота может варьироваться в зависимости от сложности и функционала. Однако основные компоненты остаются неизменными:

- Основной модуль: точка входа в приложение, где происходит инициализация бота и настройка параметров.
- Обработчики (Handlers): обработчики используются для управления различными типами входящих сообщений и команд.
- Логика обработки: здесь реализуются основные функции бота. Например, вычисления, взаимодействие с базами данных или сторонними API.
- Взаимодействие с API: бот может взаимодействовать с различными API для получения данных или выполнения действий (например, получение курсов валют).

Разработка Telegram-бота Финансовый помощник.

На основе полученной информации был выбран Telegram-бот “Финансовый помощник”. Написанный на Python.

Этапы разработки:

Импортирование нужных библиотек для работы бота и его функций.	<pre>import telebot from telebot import types from currency_converter import CurrencyConverter from datetime import datetime</pre>
Инициализация бота с использованием API токена.	<pre>API_TOKEN = '7774198724:AAF6-wR17af0MUiFN0jHLqCtLJkZMB2jC3M' bot = telebot.TeleBot(API_TOKEN)</pre>

Реализация команды /start, которая выводит приветственное сообщение и показывает основные функции бота.	<pre>@bot.message_handler(commands=['start']) def send_welcome(message): markup = types.ReplyKeyboardMarkup() markup.add(*args: 'Курсы валют', 'Бюджет', 'Напоминания о платежах', 'Расчет кредитования') bot.send_photo(chat_id=message.chat.id, photo=open('logo.png', 'rb'), caption="Добро пожаловать в ваш личный финансовый помощник!\n\n" "Я помогу вам быстро перевести нужные валюты, сформировать бюджет, " "напомню о предстоящих платежах и рассчитаю процентную ставку кредита.\n\n" "Выберите услугу:", reply_markup=markup)</pre>
Обработчики команд для различных функций бота.	<pre>def currency_rates(message): markup = types.ReplyKeyboardMarkup(row_width=1, resize_keyboard=True) btn1 = types.KeyboardButton('USD/EUR') btn2 = types.KeyboardButton('EUR/USD') btn3 = types.KeyboardButton('USD/GBP') btn4 = types.KeyboardButton('Другое значение') btn_main_menu = types.KeyboardButton('Главное меню') markup.add(*args: btn1, btn2, btn3, btn4, btn_main_menu) bot.send_message(message.chat.id, text="Введите сумму", reply_markup=markup) bot.register_next_step_handler(message, summa)</pre> <p>Обработчики для кнопки "Курсы валют" реализованы через @bot.message_handler(func=lambda message: ...). - Обработчики для остальных кнопок реализованы аналогично.</p>
Реализация логики обработки команд и взаимодействия с пользователем.	<pre>def currency_rates(message): markup = types.ReplyKeyboardMarkup(row_width=1, resize_keyboard=True) btn1 = types.KeyboardButton('USD/EUR') btn2 = types.KeyboardButton('EUR/USD') btn3 = types.KeyboardButton('USD/GBP') btn4 = types.KeyboardButton('Другое значение') btn_main_menu = types.KeyboardButton('Главное меню') markup.add(*args: btn1, btn2, btn3, btn4, btn_main_menu) bot.send_message(message.chat.id, text="Введите сумму", reply_markup=markup) bot.register_next_step_handler(message, summa)</pre>
Запуск бота в режиме polling для постоянного взаимодействия с пользователями.	<pre>if __name__ == "__main__": bot.polling(none_stop=True)</pre>

Функционал:

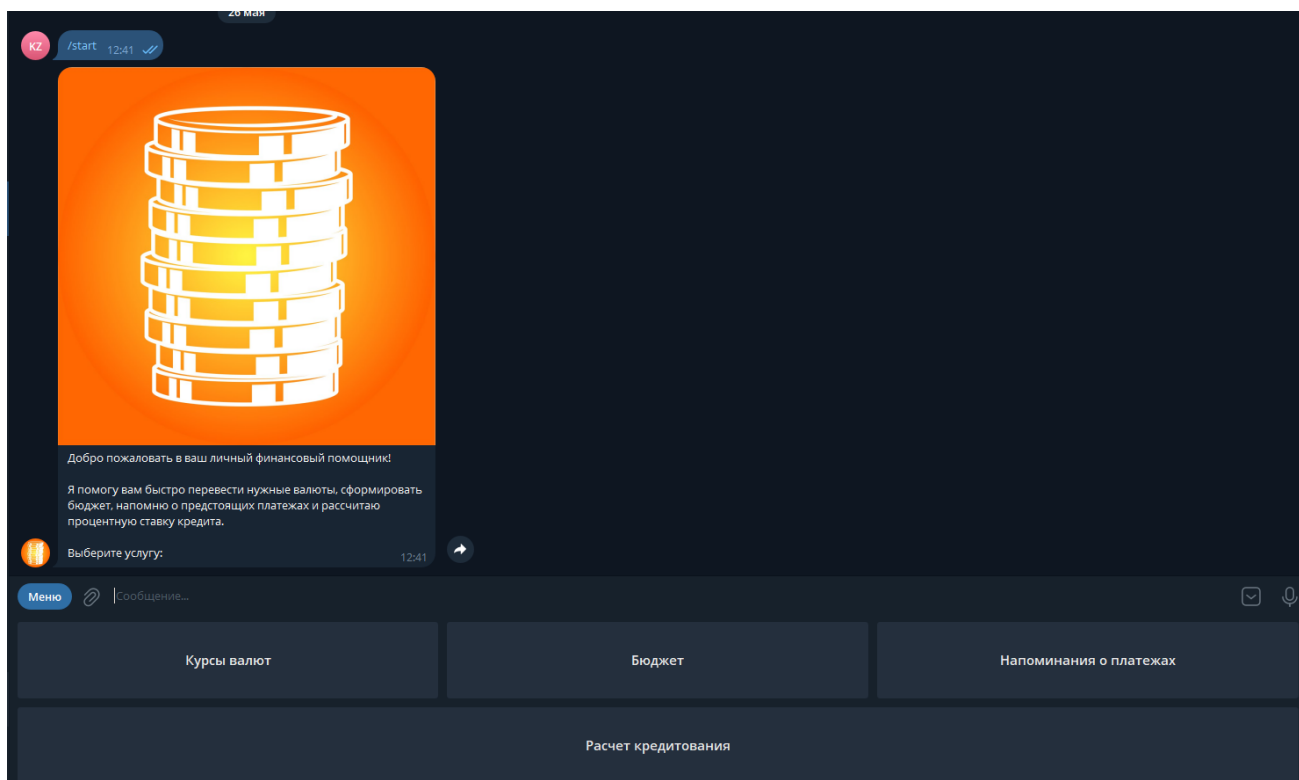
Курсы валют: сначала бот предлагает пользователю ввести сумму для конвертации. После этого пользователь выбирает валютную пару из предложенных вариантов или	Обработка сообщений: Код реализует обработчики сообщений с использованием декораторов. Интерфейс пользователя: используются элементы интерфейса
---	--

<p>вводит свою. Если ввод корректен, бот выполняет конвертацию и отображает результат. В случае ошибок, например, при некорректном вводе суммы или валютной пары, бот запрашивает повторный ввод. Вся логика обработки реализована через обработчиков сообщений, которые реагируют на определенные команды и текстовые входные данные пользователя.</p>	<p>Telegram, такие как клавиатуры (ReplyKeyboardMarkup, KeyboardButton), для упрощения взаимодействия пользователя с ботом.</p> <p>Обработка ошибок: Код включает обработку ошибок ввода, таких как неверный формат суммы.</p> <p>Асинхронное взаимодействие: Использование методов, таких как register_next_step_handler, позволяет боту запоминать текущий шаг взаимодействия с пользователем и продолжать диалог после получения ответа, сохраняя контекст диалога.</p> <p>Конвертация валют: предполагается использование класса или библиотеки для конвертации валют (CurrencyConverter).</p>
<p>Бюджет: бот позволяет пользователям добавлять доходы и расходы, просматривать текущий бюджет и сбрасывать его. Пользователь может выбрать действие из предложенных кнопок: "Добавить доход", "Добавить расход", "Просмотреть бюджет" и "Сбросить бюджет". При добавлении дохода или расхода бот запрашивает сумму и категорию, обновляя соответствующие списки. Для</p>	<p>Управление бюджетом:</p> <p>Добавление дохода и расхода: Пользователь вводит сумму и категорию в формате "сумма, категория". Код разбивает ввод на соответствующие части, преобразует сумму в числовой формат и добавляет её в соответствующий словарь (income или expenses) по категории.</p> <p>Просмотр бюджета: Код суммирует все доходы и расходы и рассчитывает</p>

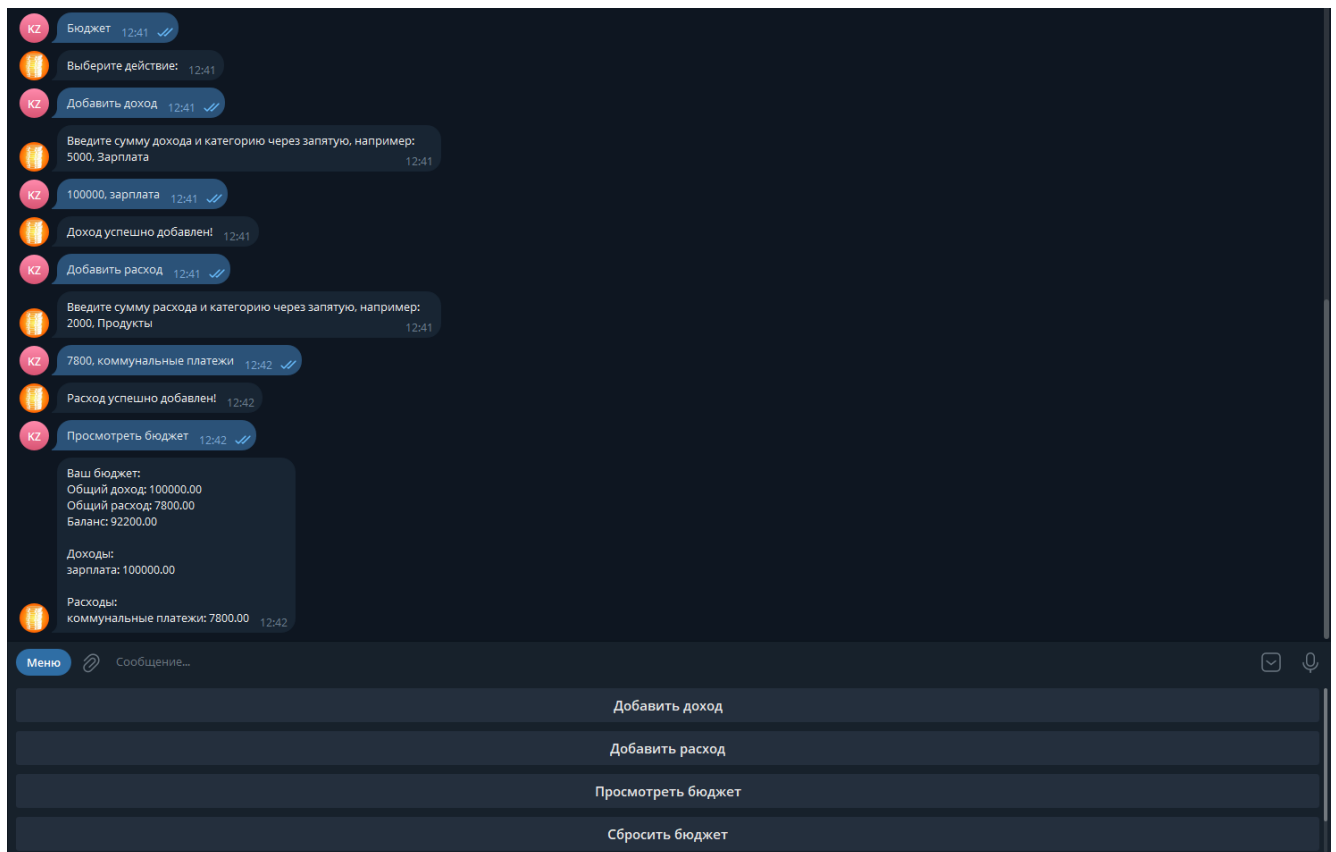
<p>просмотра бюджета бот показывает общий доход, расход и баланс, а также детализированную информацию по категориям. Функция сброса очищает все данные о доходах и расходах.</p>	<p>баланс. Затем он формирует и отправляет пользователю сообщение с детальной информацией о доходах, расходах и общем балансе.</p> <p>Сброс бюджета: Обнуляет словари <code>income</code> и <code>expenses</code>, <code>effectively</code> очищая все данные о бюджете.</p> <p>Хранение данных: Доходы и расходы хранятся в виде словарей, где ключами являются категории, а значениями — суммы. Это позволяет легко управлять и агрегировать данные по категориям.</p>
<p>Напоминания о платежах: бот предлагает пользователям функции добавления и просмотра платежей. Пользователь может выбрать действие через кнопки: "Добавить платеж" или "Просмотреть платежи". При добавлении платежа, бот запрашивает дату и описание с суммой, проверяет корректность формата и сохраняет информацию. В случае ошибки бот предлагает повторный ввод. Для просмотра платежей бот отображает список всех добавленных записей.</p>	<p>Добавление платежей: Пользователь вводит данные о платеже в формате "дд.мм.гггг, описание - сумма". Код разбивает ввод на дату и описание, проверяет правильность формата даты с помощью <code>datetime.strptime</code>, и затем добавляет платеж в список <code>payments</code>.</p> <p>Просмотр платежей: Код проверяет, есть ли платежи в списке. Если нет, отправляется сообщение "Список платежей пуст". Если платежи есть, они перечисляются с указанием даты и описания.</p> <p>Хранение данных: Платежи сохраняются в списке <code>payments</code>, который хранит каждый платеж как словарь с ключами "date" и</p>

	<p>"description". Это позволяет легко управлять и отображать данные.</p> <p>Обработка ошибок: используется конструкция try-except для обработки ошибок при разборе данных, таких как неверный формат даты.</p> <p>Хранение данных: Платежи сохраняются в списке payments, который хранит каждый платеж как словарь с ключами "date" и "description". Это позволяет легко управлять и отображать данные.</p>
<p>Расчет кредитования: бот позволяет пользователям выбирать между аннуитетным и дифференцированным платежами. Сначала бот предлагает пользователю выбрать тип кредитования. При выборе аннуитетного платежа пользователь вводит начальную сумму, процентную ставку и срок кредита. Бот рассчитывает ежемесячный платеж, общую сумму выплат и переплату. В случае дифференцированного платежа бот также запрашивает основные параметры и вычисляет первый и последний платежи, общую сумму выплат и переплату. Обработка ошибок осуществляется при вводе</p>	<p>Обработка сообщений: здесь используется обработка команд, чтобы реагировать на текстовые сообщения, такие как "Расчет кредитования" и "Аннуитетный платеж".</p> <p>Обработка ввода данных: пользователи вводят начальную сумму, процентную ставку и срок кредита в виде строки, которую бот разбивает и преобразует в числовые значения для дальнейших расчетов.</p> <p>Код обрабатывает возможные ошибки ввода</p> <p>Расчеты кредитных платежей:</p> <p>Аннуитетный платеж: используется формула для расчета аннуитетных (равномерных) платежей,</p>

<p>неправильного формата данных, с повторным запросом на ввод.</p>	<p>учитывающая процентную ставку и срок кредита.</p> <p>Дифференцированный платеж: расчет дифференцированных платежей предполагает, что основная часть выплаты фиксирована, а проценты начисляются на остаток долга.</p> <p>Циклы и условия: в расчетах дифференцированных платежей используется цикл, чтобы пройти по каждому месяцу и вычислить платежи, что позволяет определить, как выплаты изменяются с течением времени.</p>
--	---



Главное меню



Пример функционала: бюджет

Заключение

В процессе проектной практики были успешно созданы статичный сайт по проекту «Помощь людям с ОВЗ: сайт для оказания экстренной помощи» по дисциплине «Проектная деятельность» и Telegram-бот.

Проектная практика позволила освоить ряд важнейших инструментов и технологий. В частности, были изучены:

- методы организации эффективного репозитория
- навыки создания технической документации с помощью Markdown
- технологии разработки веб-сайтов на базе HTML и CSS

Особое внимание было уделено разработке телеграмм бота: исследована его архитектура, классификация и использование. Проведён анализ существующих решений, их функциональных возможностей, что позволило лучше понять, что сейчас актуально среди пользователей и как автоматизировать процессы.

Встреча с представителем Мосволонтер продемонстрировала высокий потенциал нашего сотрудничества и открыла новые пути для развития личного и командного роста.

Практика также внесла существенный вклад в развитие soft skills: улучшились навыки командной работы, планирования и распределения задач. Все эти компетенции и практический опыт создают прочную основу для дальнейшего профессионального развития в сфере информационных технологий.

Список литературы

1. Официальный сайт. Мосволонтер [Электронный ресурс]. URL: <https://mosvolonter.ru> (дата обращения: 12.04.2025).
2. Учебник по Markdown. Markdown Guide [Электронный ресурс]. URL: <https://www.markdownguide.org> (дата обращения: 18.05.2025).
3. Руководство по GitHub. GitHub Docs [Электронный ресурс]. URL: <https://docs.github.com> (дата обращения: 18.05.2025).
4. Как создать бота в Telegram в 2025 году [Электронный ресурс]. URL: <https://hi-tech.mail.ru/review/117282-kak-sozdat-bota-v-telegram/#anchor1730972355480451> (дата обращения: 14.05.2025).
5. Пошаговое руководство по созданию бота в Telegram. Python Telegram Bot [Электронный ресурс]. URL: <https://elbrusboot.camp/blog/kak-napisat-telegram-bota-na-python-instruktsiia/> (дата обращения: 14.05.2025).
6. Руководство по Python для создания Telegram-ботов. Python Telegram Bot [Электронный ресурс]. URL: <https://python-telegram-bot.readthedocs.io> (дата обращения: 14.05.2025).
7. Изучение HTML: руководства и уроки // MDN Web Docs URL: https://developer.mozilla.org/ru/docs/Learn_web_development/Core/Structuring_content (дата обращения: 17.05.2025).
8. Основы CSS // MDN Web Docs URL: https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Styling_the_content (дата обращения: 17.05.2025).
9. Верстка страницы // Docs URL: <https://habr.com/ru/articles/202408/> (дата обращения: 17.05.2025).

Подтверждаю, что отчет выполнен лично и соответствует требованиям практики.

Захарова Кира Игоревна 26.05.2025 _____ 