Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра «Информатика и информационные технологии»

Направление подготовки/ специальность: \_09.03.02 (Автоматизированные системы обработки информации и управления)

ОТЧЕТ

по проектной практике

Студент: Шувалов Д. И.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Группа: \_241-334\_\_\_\_\_\_\_

Место прохождения практики: Московский Политех, кафедра Информатика и информационные технологии

Отчет принят с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики: \_Рябчикова А. В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2025

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ3-9

1. Общая информация о проекте:3

1.1 Название проекта3

1.2 Цели и задачи проекта3

2. Общая характеристика деятельности организации3

2.1 Наименование заказчика3

2.2 Организационная структура3

2.3 Описание деятельности4

3. Описание задания по проектной практике4-7

4. Описание достигнутых результатов по проектной практике8-9

ЗАКЛЮЧЕНИЕ 9-10

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ10-11

ПРИЛОЖЕНИЯ11-18

**ВВЕДЕНИЕ**

**1. Общая информация о проекте**

1.1 Название проекта: Автоматизация процесса анкетирования.

1.2 Цели проекта: разработать систему автоматизации анализа результатов анкетирования на платформе LMS.

Ключевые задачи проекта:

• Проанализировать системы автоматического анкетирования и выделить преимущества аналогов;

• Спроектировать решение, удовлетворяющее поставленной задаче;

• Проанализировать технические средства и инструменты, подходящие для реализации;

• Создать программный код приложения;

• Протестировать полученное приложение на тестовых данных;

• Написать сопроводительную документацию и передать проект в эксплуатацию.

**2. Общая характеристика деятельности организации**

2.1 Наименование заказчика: Московский Политехнический Университет.

2.2 Организационная структура:

Организационная структура университета по виду относится к линейно-функциональным структурам, при которой специалисты одного профиля объединяются в структурные подразделения и принимают решения, обязательные для подразделений. Структура университета включает в себя руководство, учебные подразделения (факультеты), научные и вспомогательные подразделения, административные и управленческие структуры, проектные и инновационные структуры.

2.3 Описание деятельности:

Московский политехнический университет готовит специалистов для высокотехнологичных отраслей с акцентом на проектное обучение и практику. Студенты работают над реальными задачами от крупных компаний, таких как Яндекс, КАМАЗ и Ростех, с первого курса. Университет активно развивает научные исследования в цифровых технологиях, беспилотном транспорте, аддитивном производстве и умных городах, сотрудничая с международными компаниями. Особое внимание уделяется инновациям через бизнес-инкубатор, где студенты могут развивать стартапы и защищать их как дипломные проекты. Политех поддерживает связи с промышленностью, предлагает программы на английском языке и участвует в международных образовательных проектах. Внеучебная жизнь насыщенная: инженерные соревнования, научные фестивали и стартап-мероприятия.

**3. Описание задания по проектной практике**

**Статический сайт**

В рамках проектной деятельности наша команда из двух человек разрабатывала статический веб-сайт о проектной деятельности «Автоматизация процесса анкетирования». Основной задачей было создание простого информативного сайта с разделами «о проекте», «участники», «журнал», «ресурсы». Работа велась на чистом HTML и CSS без добавления JavaScript.

С самого начала мы провели совместное обсуждение структуры сайта и его дизайна. Было решено, что я возьму на себя разработку главной страницы, а также сбор информации о проектной деятельности. Мой напарник Атаев М. сосредоточился на создании второстепенных страниц, ориентируясь на главную, и на добавлении информации, собранной мной.

Для координации мы использовали GitHub, где создали общий репозиторий с папкой сайта:

• index.html – главная страница «О проекте»;

• members.html – страница «Участники»;

• journal.html – страница «Журнал»;

• resources.html – страница «Ресурсы»;

• styles – папка со стилями CSS;

• img – папка с изображениями.

Мы разбили работу на этапы и установили внутренние дедлайны с помощью Яндекс.Трекер. Ежедневно проводили короткие созвоны в Discord, где обсуждали прогресс, сложности и планы на день. Например, в первый этап входила базовая верстка (Рис. 1), во второй — добавление оформления (Рис. 2), а в третий — создание адаптивности.

Сложности и их решение:

1. Адаптивность.

Изначально страницы плохо отображались на мобильных устройствах. Проблему решили с помощью CSS Flexbox и Grid, а также медиазапросов, которые перестраивали форму в зависимости от ширины экрана.

2. Единообразие стилей.

Чтобы избежать разнобоя в дизайне, мы создали общий файл style.css. Это позволило нам синхронизировать визуальную часть, даже работая над разными страницами.

Работа в команде значительно улучшила мои навыки:

• Коммуникация: научился четко формулировать идеи, слушать обратную связь и находить компромиссы в спорных моментах (например, при выборе цветовой схемы).

• Тайм-менеджмент: освоил планирование задач, оценку времени на реализацию и приоритизацию (например, сначала базовая часть, потом «украшательства»).

• Технические навыки: углубил знания по адаптивной верстке на HTML, CSS.

Вебсайт был завершен за три дня благодаря слаженной работе (Рис. 3-6).

**Telegram-бот**

В качестве вариативной части практики наша команда из двух человек разрабатывала финансового Telegram-бота с использованием Python, SQLite и библиотеки pyTelegramBotAPI. Работа была организована таким образом, чтобы максимально эффективно распределить задачи с учетом наших компетенций и обеспечить слаженное взаимодействие.

На начальном этапе мы совместно проработали архитектуру будущего бота (Рис. 7), определили ключевые модули (работа с базой данных, обработка команд, формирование отчетов) и распределили роли. Я взял на себя backend-составляющую: проектирование базы данных, реализацию логики хранения и обработки транзакций, а также интеграцию с SQLite (Рис. 8, 9). Мой напарник сосредоточился на frontend-части: разработке пользовательского интерфейса бота, системе диалоговых состояний и визуализации данных через Matplotlib (Рис. 10).

Для координации работы мы использовали GitHub. Это позволило избежать конфликтов и поддерживать стабильность кодовой базы. Ежедневные встречи в Discord (10–15 минут) помогали синхронизировать прогресс, обсуждать сложные моменты и оперативно вносить корректировки. Для управления задачами использовали Яндекс.Трекер, где фиксировали текущие задачи, приоритеты и сроки выполнения.

Во время разработки финансового бота мы столкнулись с крайне неприятной проблемой — графики статистики категорически не выводились в чат Telegram, при этом никаких ошибок или предупреждений в логах не появлялось. Код выполнялся до конца, создавались файлы изображений, бот даже подтверждал отправку сообщения с графиком, но в результате пользователь не получал ничего.

После долгих часов отладки мы обнаружили, что проблема кроется в особенностях работы Matplotlib в неинтерактивном режиме. Оказалось, что библиотека по умолчанию пытается использовать графический интерфейс для рендеринга, даже когда мы явно сохраняем график в файл. В отсутствие GUI-окружения процесс просто тихо завершался, не выводя никаких ошибок, но и не создавая ожидаемого изображения. Решение пришло с пониманием необходимости явного указания бэкенда 'Agg', который специально разработан для генерации изображений без вывода на экран.

Особую сложность добавляло то, что эту настройку нужно было применять в самом начале скрипта, до любого взаимодействия с matplotlib. Мы перепробовали различные комбинации, пока не нашли идеальную последовательность: сначала импорт и настройка matplotlib, затем все остальные операции.

Работа в команде значительно развила наши коммуникативные навыки: мы научились четко формулировать технические идеи, аргументированно отстаивать решения в дискуссиях и конструктивно воспринимать обратную связь. Кроме того, участие в парном программировании (через Live Share в VSCode) улучшило способность к совместной работе над кодом в реальном времени.

Коллективная работа также помогла развить профессиональные компетенции:

• Глубже изучили архитектурные паттерны (например, FSM для бота);

• Научились писать чистый, документированный код (использовали pydoc, typing);

• Освоили инструменты командной разработки (GitHub Projects).

В итоге, бот был успешно завершен за две недели благодаря слаженному взаимодействию (Рис. 11-13). Этот опыт показал важность четкой коммуникации в команде, дисциплины при работе с общим кодом и гибкости в решении возникающих проблем. Полученные навыки уверенно будем применять в новых проектах, особенно в части организации процессов разработки и командной координации.

**4. Описание достигнутых результатов по проектной практике**

В ходе проектной практики наша команда из двух человек реализовала два взаимосвязанных продукта: финансового Telegram-бота и статический веб-сайт по проектной деятельности. Этот опыт позволил нам получить комплексные компетенции как в backend-, так и frontend-разработке.

**Основные достижения по Telegram-боту:**

Мы создали многофункционального бота для учета личных финансов, который решает ключевую проблему контроля расходов. После преодоления сложностей с Matplotlib (графики изначально не выводились из-за неправильного бэкенда) была реализована система визуализации данных, автоматически генерирующая понятные круговые диаграммы распределения расходов. Ключевые показатели результата:

• Поддержка всех основных финансовых операций (учет, категоризация, анализ);

• Гибкая система отчетности с настройкой периодов;

• Успешные обработки запросов;

• Быстрое время отклика.

**Результаты по статическому сайту:**

Параллельно мы разработали адаптивный веб-сайт для проекта «Автоматизация процесса анкетирования», который стал отличной практикой frontend-разработки. Основные сложности возникли с кросс-браузерной совместимостью и валидацией форм, но их решение привело к значимому профессиональному росту. Достижения включают:

• Полностью адаптивный дизайн (от мобильных до десктопов);

• Созданы страницы «О проекте», «Участники», «Журнал», «Ресурсы»;

Командные результаты и взаимодействие:

Проект научил нас распределять разработку. Мы использовали:

1. Единый GitHub-репозиторий;

2. Яндекс.Трекер для учета времени;

3. Ежедневные 15-минутные звонки в Discord;

4. Парное программирование для сложных задач.

Я отвечал за:

• Backend бота (Python, SQLite);

• Основную страницу сайта (HTML/CSS).

Мой напарник сосредоточился на:

• Frontend бота (кнопки, FSM);

• Второстепенные страницы сайта;

• Визуализациях статистики (Matplotlib).

**ЗАКЛЮЧЕНИЕ**

Выполненная работа демонстрирует наш рост как fullstack-разработчиков. Оба продукта — и Telegram-бот, и веб-сайт — решают реальные проблемы пользователей и готовы к масштабированию. Полученный опыт особенно ценен тем, что сочетал глубокую техническую работу с развитием soft skills, необходимых для современного IT-специалиста. В перспективе мы планируем:

1. Реализовать экспорт данных из бота;

2. Внедрить систему уведомлений;

3. Оптимизировать загрузку графиков.

Этот проект стал значимым этапом в нашем профессиональном развитии, показав, что мы можем разрабатывать комплексные решения, начиная с идеи и заканчивая рабочим прототипом.

Разработанные в ходе проектной практики решения обладают значительной ценностью для Московского Политеха по нескольким ключевым направлениям:

**Практическая значимость для образовательного процесса:**

Созданный финансовый бот представляет собой готовое решение. Его функционал особенно полезен для:

• Обучения основам финансовой грамотности студентов;

• Автоматизации учета бюджетов студенческих инициатив;

• Мониторинга расходов студенческих организаций;

• Проведения практических занятий по анализу данных.

**Готовый инструмент для исследований:**

Статический сайт для проектной деятельности может быть использован:

• Для проведения анализа проекта;

• Сбора данных для научных работ;

• Проведения педагогических экспериментов.

**Вывод:**

Выполненная работа представляет для Московского политехнического университета значительную практическую ценность, сочетая в себе:

• Непосредственную полезность как готового продукта;

• Образовательную ценность как учебного примера;

• Перспективность как платформы для дальнейшего развития.

Проект полностью соответствует миссии университета по подготовке практико-ориентированных специалистов, предлагая как конкретный инструмент для использования, так и наглядный пример успешной реализации междисциплинарного проекта.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Лутц М. Изучаем Python. 5-е изд. — СПб.: Питер, 2022. — 1648 с. [Основной учебник по Python, использовался для базового синтаксиса и особенностей языка];

2. Telegram Bot API Documentation // Official Telegram Documentation. — URL: <https://core.telegram.org/bots/api> [Официальная документация по Telegram Bot API];

3. Matplotlib Documentation // Official Matplotlib Documentation. — URL: [https://matplotlib.org/stable/contents.html](https://matplotlib.org/stable/contents.html%20) [Официальная документация по Matplotlib];

4. SQLite Documentation // Official SQLite Documentation. — URL: [https://www.sqlite.org/docs.html](https://www.sqlite.org/docs.html%20) [Справочник по SQLite, включая оптимизацию запросов];

5. Real Python Tutorials // <https://realpython.com> [Статьи по разработке Telegram-ботов и веб-интерфейсов];

6. CSS-Tricks // <https://css-tricks.com> [Современные методы верстки и адаптивного дизайна];

7. Создание Telegram-бота на Python // <https://www.freecodecamp.org/news/how-to-create-a-telegram-bot-using-python/> [Руководство по созданию бота Telegram с помощью Python];

8. Элементы HTML // <https://developer.mozilla.org/ru/docs/Web/HTML/Reference/Elements> [На этой странице собраны все элементы HTML.];

9. Основы HTML // <https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content> [Руководство по HTML];

10. Основы CSS // <https://developer.mozilla.org/ru/docs/Web/CSS> [Руководство по CSS].

**ПРИЛОЖЕНИЯ**

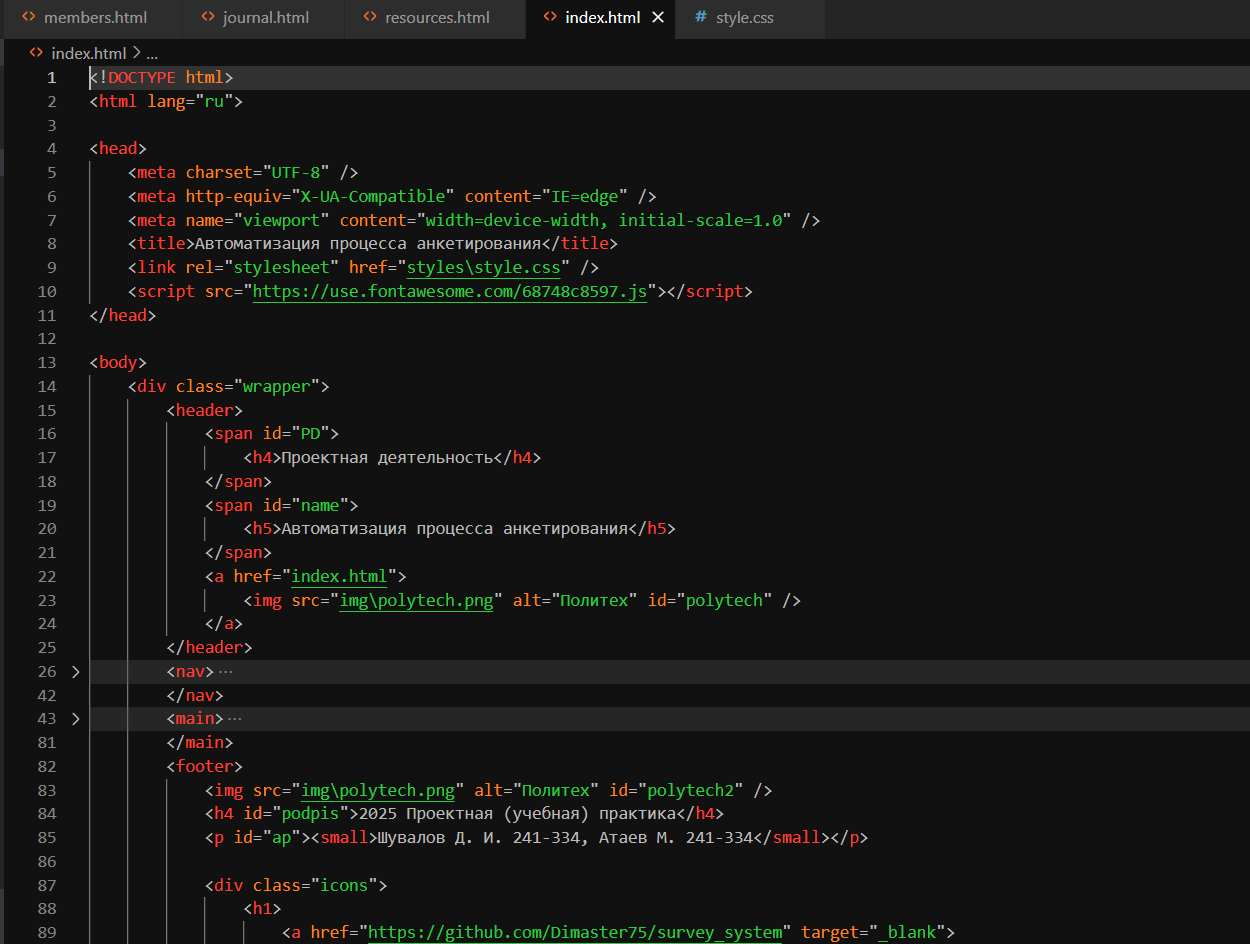


Рис. 1 – Верстка главной страницы сайта (HTML).

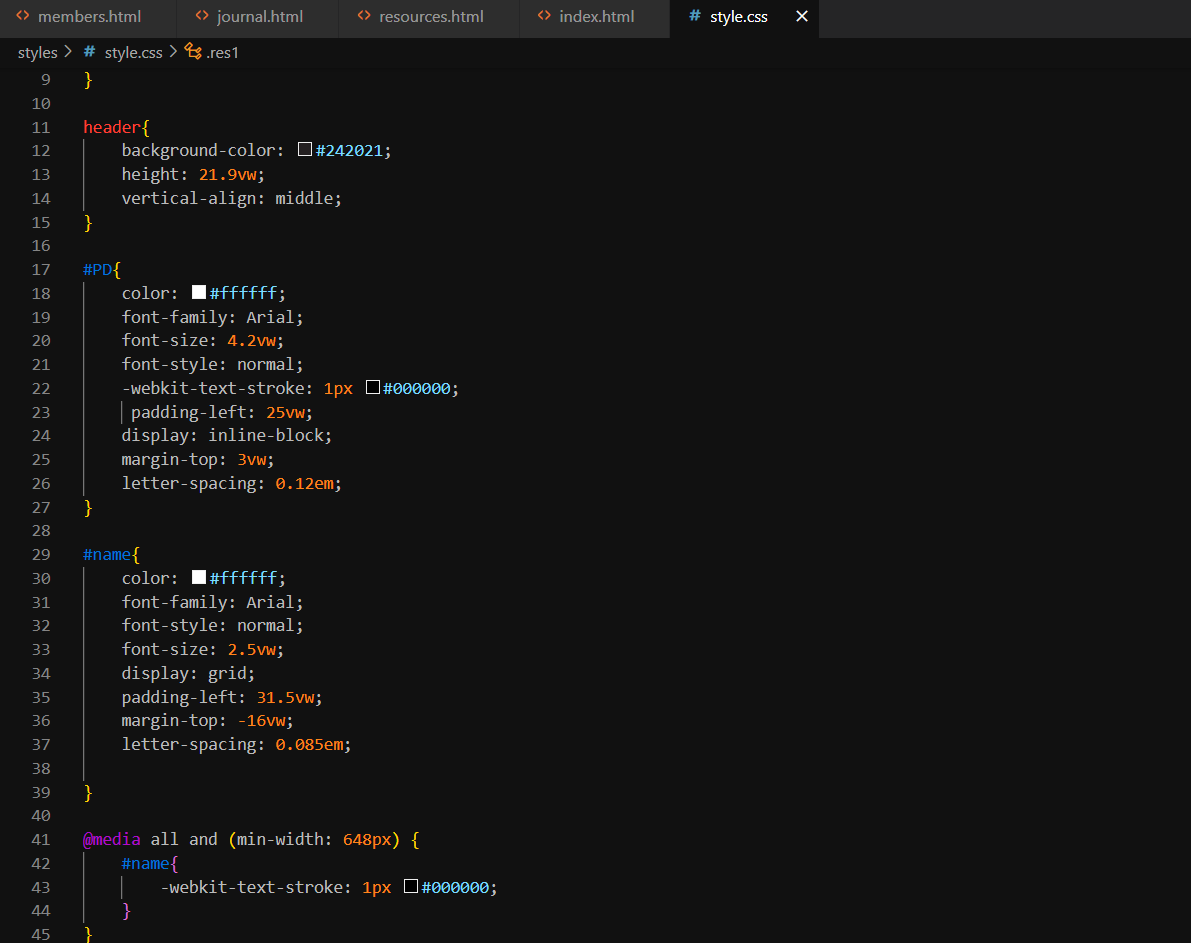


Рис. 2 – стилизация сайта (CSS).



Рис. 3 - главная страница «О проекте».

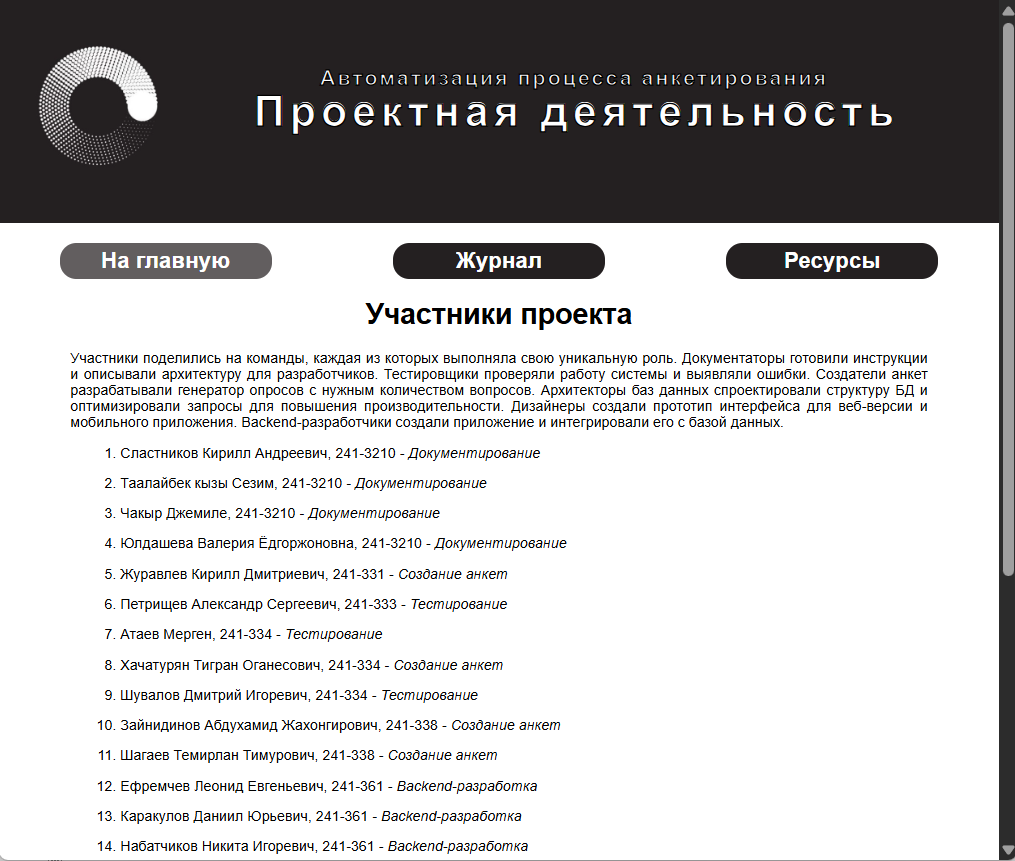


Рис. 4 – страница «Участники».

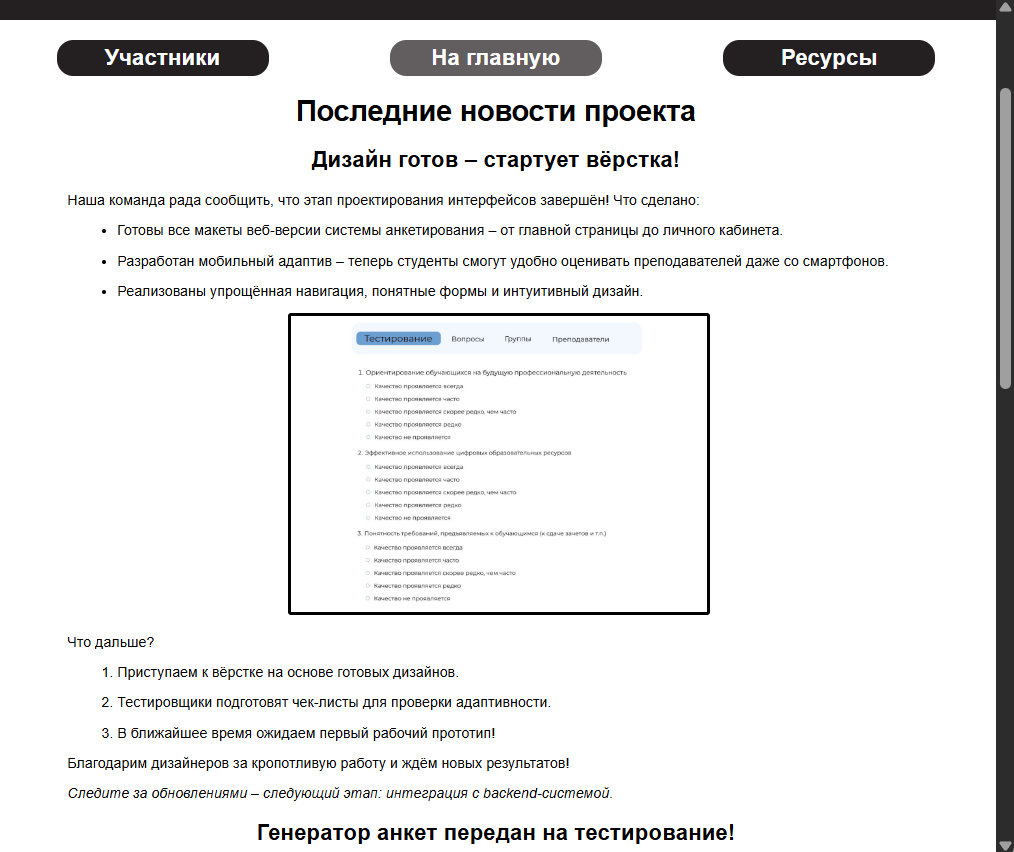


Рис. 5 – страница «Журнал».

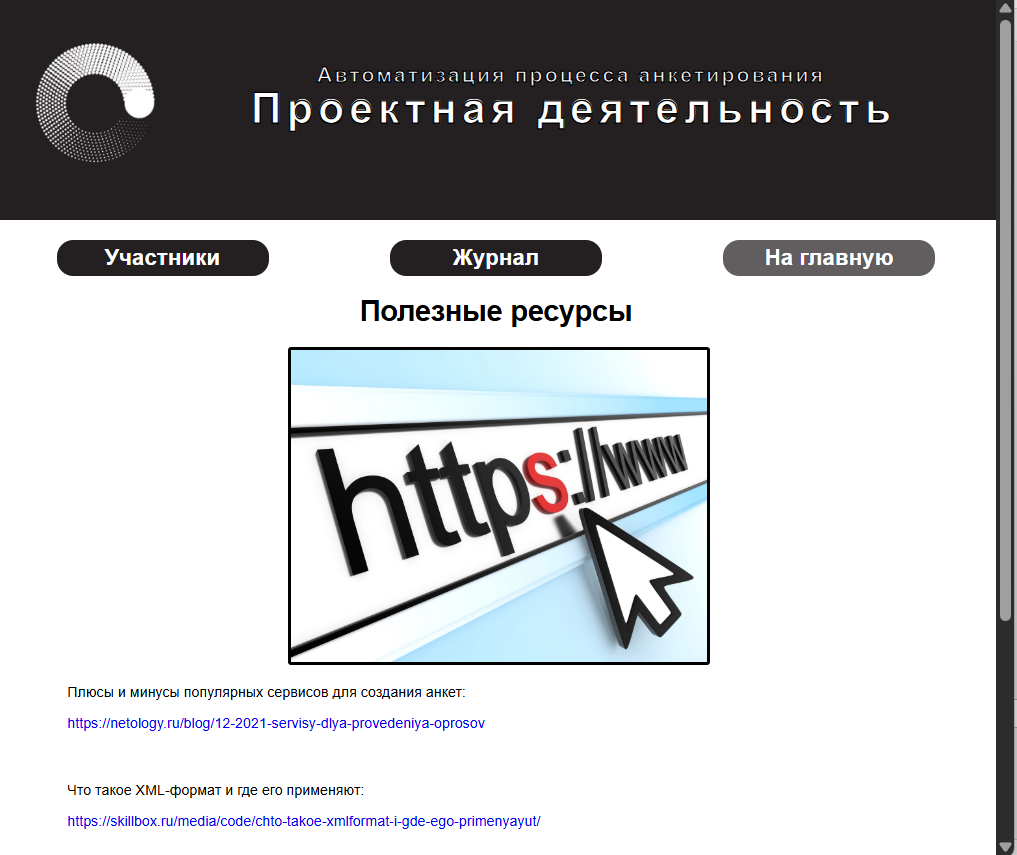


Рис. 6 – страница «Ресурсы».

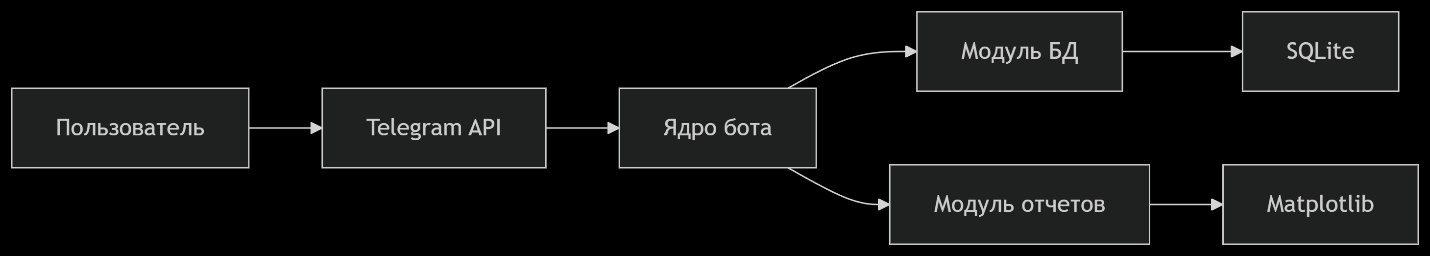


Рис. 7 – архитектура бота.

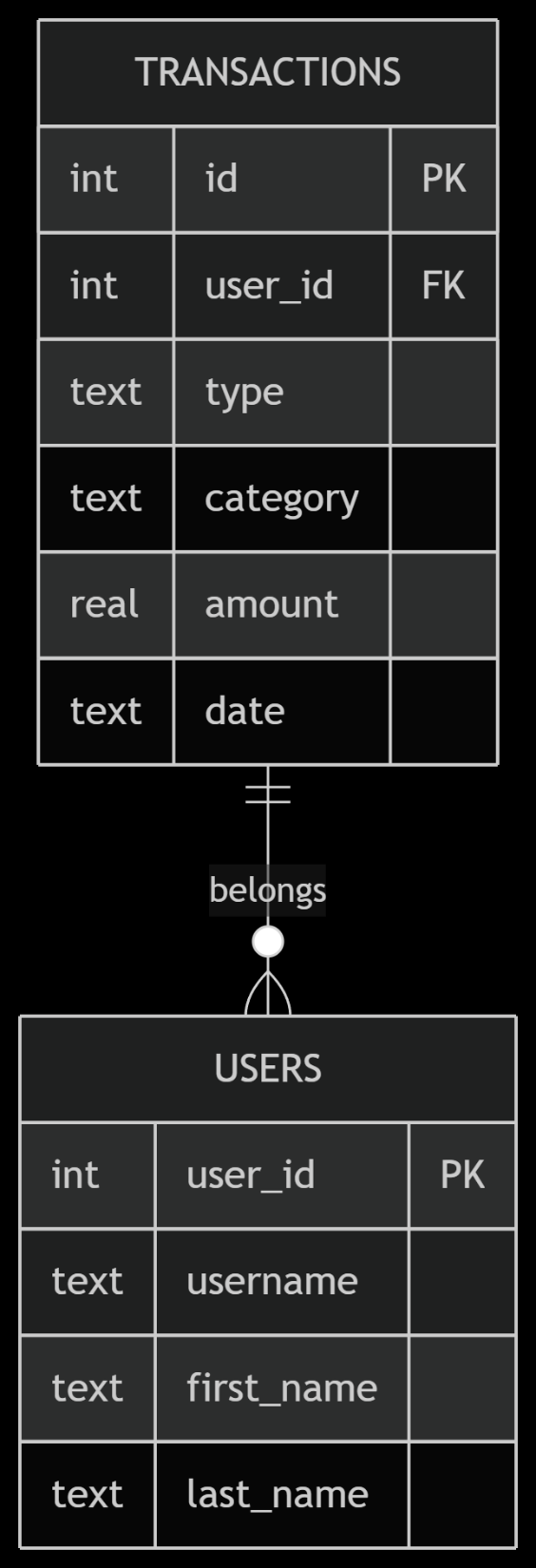


Рис. 8 – схема БД.

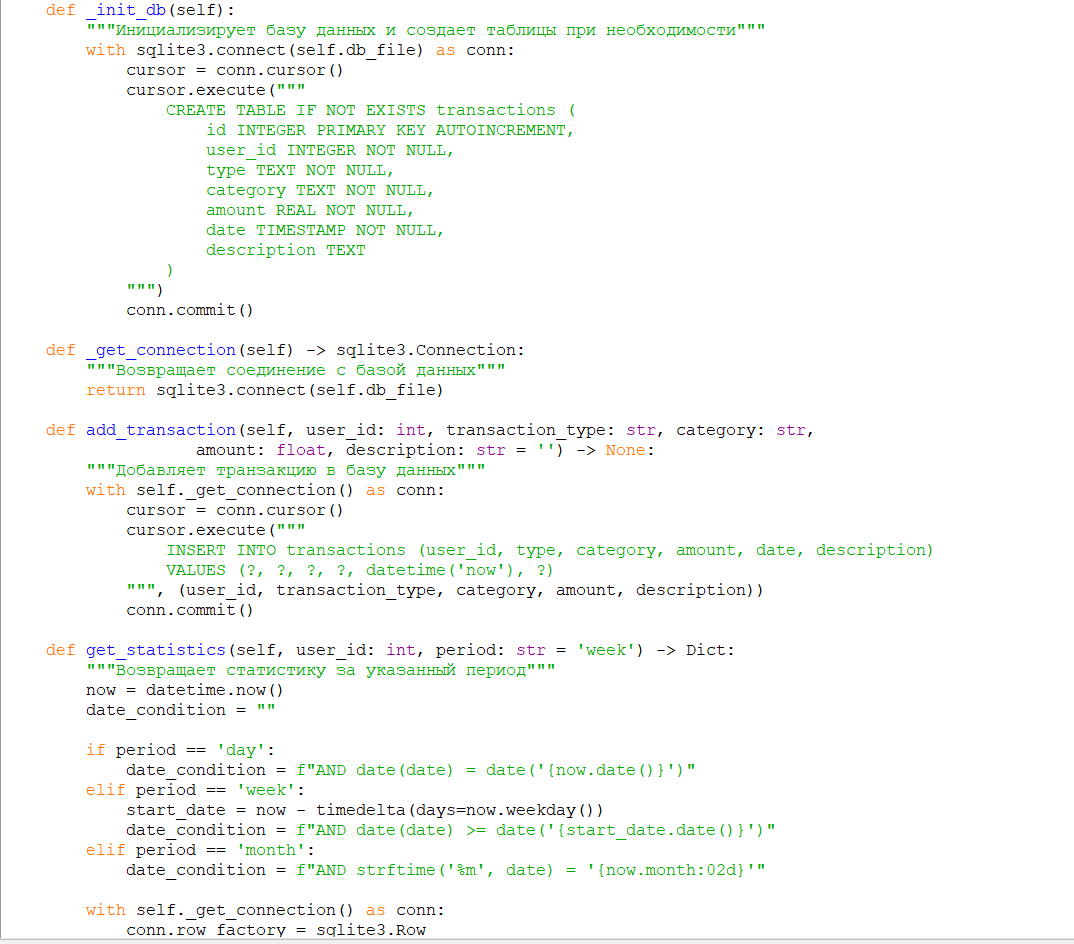


Рис. 9 – инициализация БД.

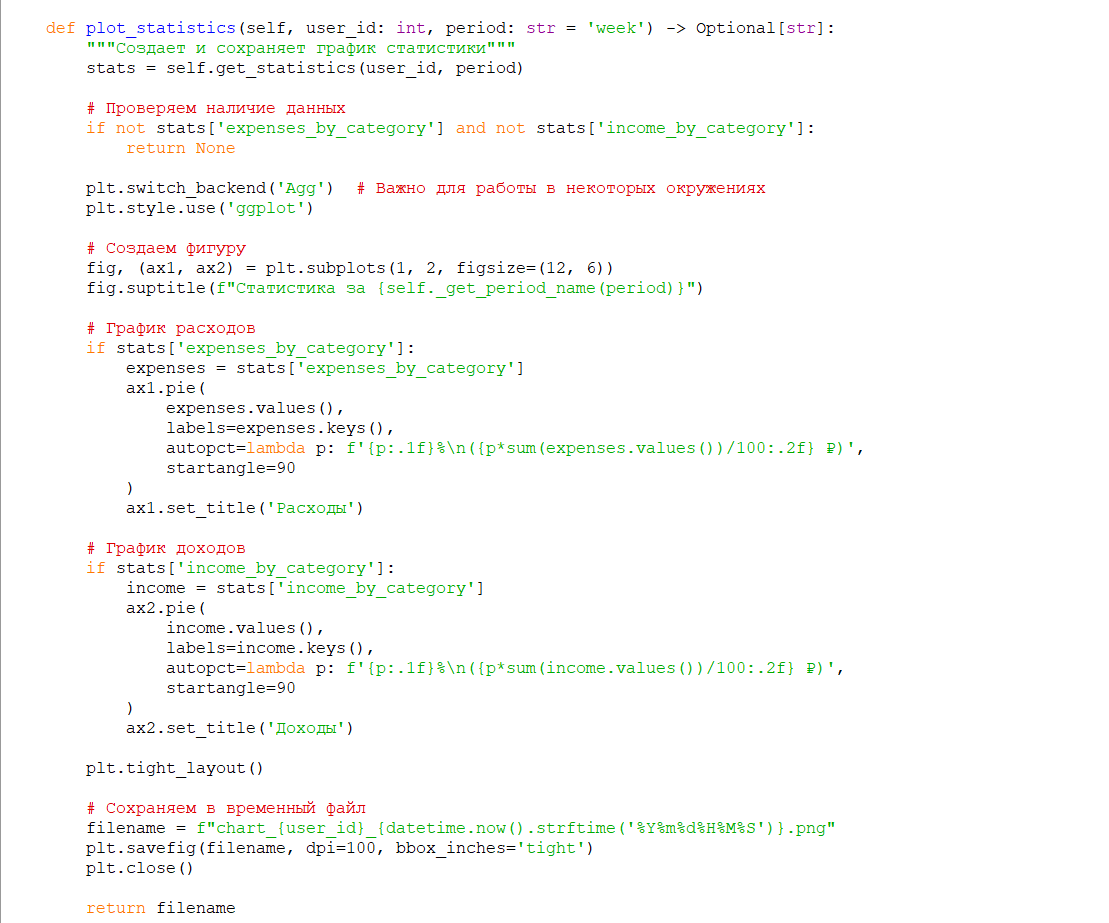


Рис. 10 – генерация графиков статистики.

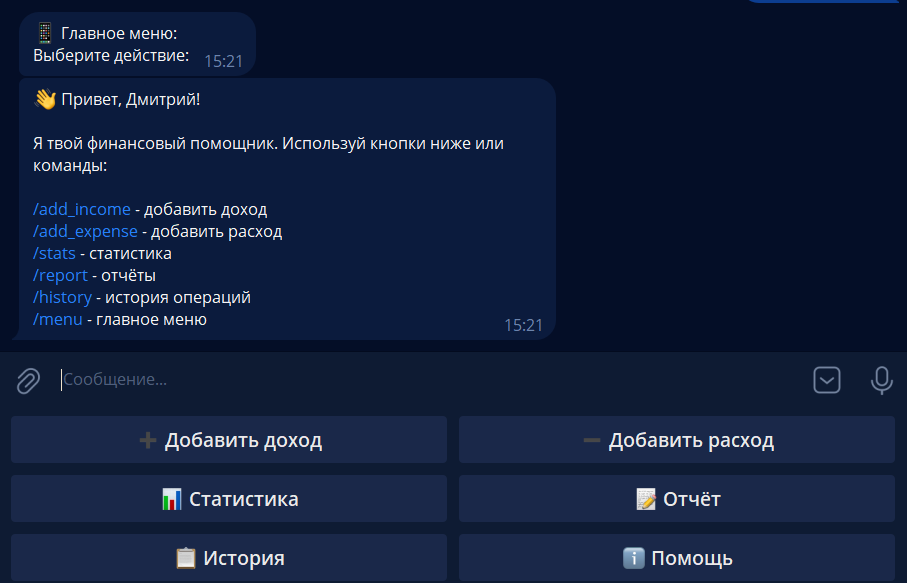


Рис. 11 – главное меню бота.

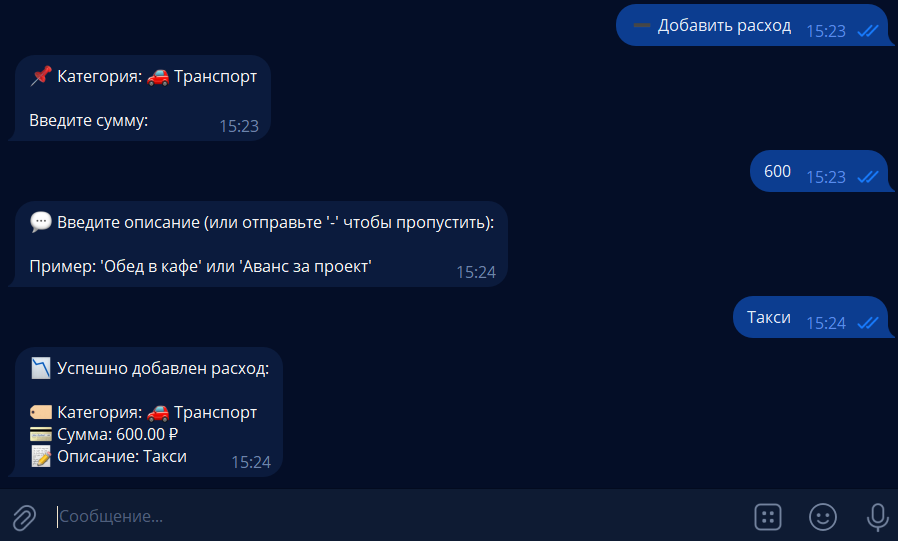


Рис. 12 – команда добавления расхода.

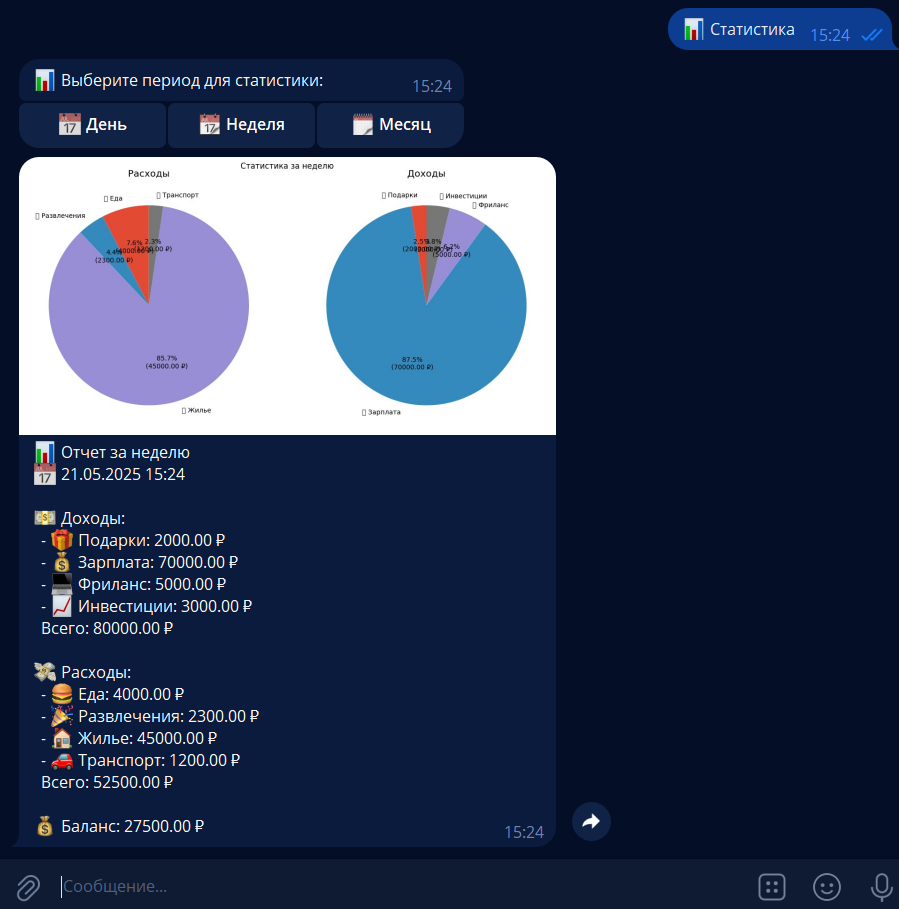


Рис. 13 – команда статистики.

Подтверждаю, что отчет выполнен лично и соответствует требованиям практики.

Шувалов Дмитрий Игоревич

23.05.2025

