

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

Факультет Информационных технологий
Кафедра «Информатика и информационные технологии»

Направление подготовки/ специальность: 09.03.02 АСОИУ

ОТЧЕТ
по проектной практике

Студент: Журавлев Кирилл Дмитриевич Группа: 241-331

Место прохождения практики: Московский Политех, кафедра ИиИТ

Отчет принят с оценкой _____ Дата _____

Руководитель практики: Привалов Вячеслав Анатольевич

Москва 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	2
1. ОБЩАЯ ИНФОРМАЦИЯ О ПРОЕКТЕ.....	3
2. ОБЩАЯ ХАРАКТЕРИСТИКА ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИИ	3
3. ОПИСАНИЕ ЗАДАНИЯ ПО ПРОЕКТНОЙ ПРАКТИКЕ	4
4. ОПИСАНИЕ ДОСТИГНУТЫХ РЕЗУЛЬТАТОВ	5
5. ЗАДАНИЯ НА ПРОЕКТНУЮ ПРАКТИКУ	7
5.1 ОБЯЗАТЕЛЬНАЯ ЧАСТЬ.....	7
5.2 ВАРИАТИВНАЯ ЧАСТЬ.....	11
ЗАКЛЮЧЕНИЕ	18
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	18
ПРИЛОЖЕНИЯ.....	19

ВВЕДЕНИЕ

Все файлы проекта находятся в данном репозитории:

<https://github.com/yourthful/practice-2025-1/>

В ходе проектной практики была реализована система автоматизации анкетирования для Московского Политеха. Актуальность проекта обусловлена необходимостью оптимизации процессов сбора и анализа данных при проведении регулярных опросов среди студентов и преподавателей. Существующие решения (Google Forms, MS Forms) не всегда удовлетворяют требованиям университета по обработке данных в специфических форматах (XML, Excel) и их интеграции с внутренними системами.

Разработанное решение на Python позволяет:

- Сократить время создания анкет за счет использования шаблонов

- Автоматизировать обработку результатов с минимальным вмешательством пользователя
- Обеспечить совместимость с существующей ИТ-инфраструктурой вуза

1. ОБЩАЯ ИНФОРМАЦИЯ О ПРОЕКТЕ

Название проекта:

"Автоматизация процесса анкетирования"

Цель и задачи проекта:

Цель: Создание программного комплекса для автоматизации процессов создания, распространения и анализа анкет в учебной среде.

Задачи:

1. Разработка модуля генерации анкет на основе XML-шаблонов
2. Создание системы обработки ответов с поддержкой форматов Excel и XML
3. Обеспечение интеграции с внутренними сервисами университета

2. ОБЩАЯ ХАРАКТЕРИСТИКА ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИИ

Наименование заказчика

ФГБОУ ВО "Московский Политехнический Университет"

Организационная структура

Проект реализовывался под руководством кафедры "Информатика и информационные технологии". В разработке участвовали:

- Научный руководитель
- Технические специалисты кафедры
- Группа тестировщиков из числа студентов

Описание деятельности

Университет активно внедряет цифровые решения для оптимизации образовательного процесса. В рамках данной инициативы особое внимание уделяется:

- Автоматизации административных процессов
- Сбору и анализу данных для принятия управленческих решений
- Развитию ИТ-компетенций студентов через практико-ориентированные проекты

3. ОПИСАНИЕ ЗАДАНИЯ ПО ПРОЕКТНОЙ ПРАКТИКЕ

В рамках проектной практики я участвовал в разработке системы автоматизации анкетирования для Московского Политехнического Университета. Моя роль заключалась в реализации функционала создания анкет — я отвечал за генерацию шаблонов в XML-формате и их программную обработку. Работа велась в команде, где задачи были распределены между четырьмя направлениями:

- **Создание анкет** – разработка генератора шаблонов и обработчика ответов.
- **Тестирование** – проверка корректности работы системы, выявление ошибок.
- **Архитектура баз данных** – проектирование структуры хранения анкет и результатов.
- **Дизайн интерфейса** – разработка визуальной части статического сайта для отображения статистики.

Взаимодействие в команде:

Мы использовали **чат в Telegram** для постановки задач и контроля их выполнения. Каждая подгруппа работала над своим модулем, но регулярно синхронизировалась на еженедельных встречах, где обсуждались прогресс и возможные изменения в архитектуре. Например, при разработке XML-шаблонов мне требовалось согласовывать структуру данных с командой, отвечающей за базы данных, чтобы обеспечить корректное хранение анкет.

Планирование и организация работы:

Я разбил свои задачи на этапы:

1. **Анализ требований** – изучение форматов анкет и возможных типов вопросов.
2. **Разработка генератора XML-шаблонов** – реализация кода на Python с использованием ElementTree.
3. **Создание обработчика ответов** – парсинг данных из Excel (openpyxl) и XML, валидация.
4. **Интеграция с другими модулями** – взаимодействие с командами тестирования и дизайна.

Для управления временем я применял **ежедневное планирование** в Obsidian.

Основные сложности возникли при работе с нестандартными типами вопросов в XML, но они были решены через дополнительное тестирование и правки логики обработки данных.

Приобретённые навыки:

- **Командная работа** – согласование задач, распределение ролей, совместное решение проблем.
- **Технические компетенции** – углублённое изучение Python, XML, Excel-парсинга.
- **Самоменеджмент** – соблюдение сроков, адаптация к изменениям в требованиях.
- **Коммуникация** – презентация своих решений, аргументация выбора технологий.

4. ОПИСАНИЕ ДОСТИГНУТЫХ РЕЗУЛЬТАТОВ

В результате совместной работы команды было разработано решение для автоматизации анкетирования. Основным достижением стало создание работоспособной системы, состоящей из нескольких взаимосвязанных модулей.

Я, как участник команды по созданию анкет, разработал ключевой функциональный модуль системы. Этот модуль позволяет автоматически создавать анкеты различной сложности в XML-формате, поддерживая все необходимые типы вопросов: одиночный и множественный выбор, текстовые поля, шкалы оценок. Для обработки ответов был реализован отдельный компонент, который корректно работает с данными в форматах Excel и XML, выполняет их валидацию и предварительный анализ.

Особую ценность представляет интеграция разработанного нашей командой модуля с другими компонентами системы. Анкеты, созданные генератором, успешно сохраняются в структурированную базу данных, спроектированную командой архитекторов. Полученные результаты удобно отображаются на интуитивно понятном веб-интерфейсе, разработанном дизайнерами.

Тестировщики провели комплексную проверку всей системы, что позволило выявить и устранить критические ошибки на этапе разработки.

Для обеспечения удобства использования системы наша команда подготовила подробную техническую документацию.

Реализованное решение имеет значительную практическую ценность для университета. Оно позволяет:

- В несколько раз сократить время на подготовку и проведение опросов
- Минимизировать ручной труд при обработке результатов
- Получать аналитику в наглядном виде
- Легко масштабировать систему под новые виды анкет

В процессе работы я не только усовершенствовал свои технические навыки, но и получил ценный опыт командной разработки. Особенно важным стало понимание того, как отдельные модули интегрируются в единую систему, и как важно учитывать требования смежных команд при разработке своего компонента.

5. ЗАДАНИЯ НА ПРОЕКТНУЮ ПРАКТИКУ

5.1 ОБЯЗАТЕЛЬНАЯ ЧАСТЬ

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.

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.

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).

4. Взаимодействие с организацией-партнёром:

- Организуйте 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.

Создание статического сайта:

Описание кода страницы сайта на примере index.html:

1. `<!DOCTYPE html>`

Объявление типа документа, указывающее браузеру на использование HTML5.

2. `<html lang="ru">`

Корневой элемент документа с указанием русского языка содержимого.

3. Секция `<head>`

Содержит метаданные и подключение ресурсов:

```
<head>
    <meta charset="UTF-8"> <!-- Кодировка UTF-8 для
корректного отображения кириллицы -->
    <title>Главная - Проект</title> <!-- Заголовок
вкладки браузера -->
    <link rel="stylesheet" href="style.css"> <!--
Подключение внешнего CSS-файла -->
</head>
```

4. Секция `<body>`

Основное содержимое страницы включает:


Навигационное меню (`<nav>`):

```
<nav>
    <a class="active" href="index.html">Главная</a>
    <a href="about.html">О проекте</a>
    <a href="members.html">Участники</a>
    <a href="journal.html">Журнал</a>
    <a href="resources.html">Ресурсы</a>
</nav>
```

Примечание: Класс “active” выделяет текущую страницу в меню.

5. Основной контент (<main>)

Содержит аннотацию проекта:

```
<main>
  <section>
    <h2 class="center-text">Аннотация проекта</h2>
    <p>Современные образовательные учреждения всё
чаще используют...</p>
    <ul>
      <li> Быстро формировать анкетные...</li>
      <!-- Другие элементы списка -->
    </ul>
  </section>
</main>
```

Особенности:

- Использование семантических тегов (<section>, <main>)
- Класс “center-text” для центрирования заголовка
- Структурированный список преимуществ проекта

6. Подвал сайта (<footer>)

```
<footer>
  <p>Автоматизация процесса анкетирования (I курс),
2025.</p>
</footer>
```

Ключевые технологии:

- Внешнее подключение CSS через <link>
- Семантическая разметка для улучшения доступности
- Адаптивная навигационная панель (реализуется в CSS)

Соответствие требованиям:

- Включает все обязательные разделы (Главная, О проекте, Участники и др.)

В ходе разработки был получен статический сайт с использованием CSS стилей. Сайт содержит всю необходимую информацию о проекте.

5.2 ВАРИАТИВНАЯ ЧАСТЬ

Для вариативной части практической работы был разработан Telegram-бот на языке Python. Суть бота заключается в генерации случайного пароля по критериям пользователя. Сгенерированные пароли можно удобно хранить в одном месте.

Этапы создания Telegram-бота на примере main.py:

1. Получение токена от BotFather

Для создания Telegram-бота необходимо получить уникальный API-токен:

1. Откройте Telegram и найдите официального бота **@BotFather**.
2. Отправьте команду /start, затем /newbot.
3. Укажите:
 - Имя бота (отображается у пользователей, например: "Генератор паролей").

- Юзернейм бота (должен заканчиваться на bot, например: zhuravlev241331_bot).

2. Подключение библиотек

Установите необходимые библиотеки через терминал:

```
pip install python-telegram-bot
```

Импортируйте их в коде:

```
import os
import random
import string
import json
from datetime import datetime
from telegram import Update, ReplyKeyboardMarkup
from telegram.ext import (
    ApplicationBuilder,
    CommandHandler,
    MessageHandler,
    CallbackContext,
    filters
)
```

3. Настройка бота

Конфигурация:

```
TOKEN = "8004240677:AAF8SUKjOA5cZ8Z92XGyT6N2yRppxPnS3Ns" #
Замените на реальный токен
PASSWORDS_FILE = "src/user_password.json" # Файл для
хранения паролей
```

```
DEFAULT_SETTINGS = {'length': 12, 'use_special': True} #  
Настройки по умолчанию
```

Загрузка/сохранение данных:

```
def load_passwords():  
    if not os.path.exists(PASSWORDS_FILE):  
        return {}  
    with open(PASSWORDS_FILE, 'r') as f:  
        return json.load(f)  
  
def save_passwords(passwords):  
    with open(PASSWORDS_FILE, 'w') as f:  
        json.dump(passwords, f, indent=2)
```

4. Основные функции

Генерация пароля:

```
def generate_password(settings):  
    characters = string.ascii_letters + string.digits  
    if settings['use_special']:  
        characters += string.punctuation  
    return ''.join(random.choice(characters) for _ in  
range(settings['length']))
```

Команда /start:

```
async def start(update: Update, context: CallbackContext):  
    keyboard = [
```

```

        ["Сгенерировать пароль"],
        ["Мои сохранённые пароли"],
        ["Справка"]
    ]
    reply_markup = ReplyKeyboardMarkup(keyboard,
resize_keyboard=True)
    await update.message.reply_text(
        "🔒 *Генератор паролей*\nВыберите действие:",
        reply_markup=reply_markup,
        parse_mode='Markdown'
    )

```

5. Обработка сообщений

Логика диалога:

```

async def handle_message(update: Update, context:
CallbackContext):
    text = update.message.text
    user_id = str(update.message.from_user.id)

    if text == "Сгенерировать пароль":
        # Показывает меню настроек
    elif text == "Изменить длину":
        # Запрашивает новую длину
    elif text == "Сохранить пароль":
        # Сохраняет пароль в JSON
    # ... и другие команды

```

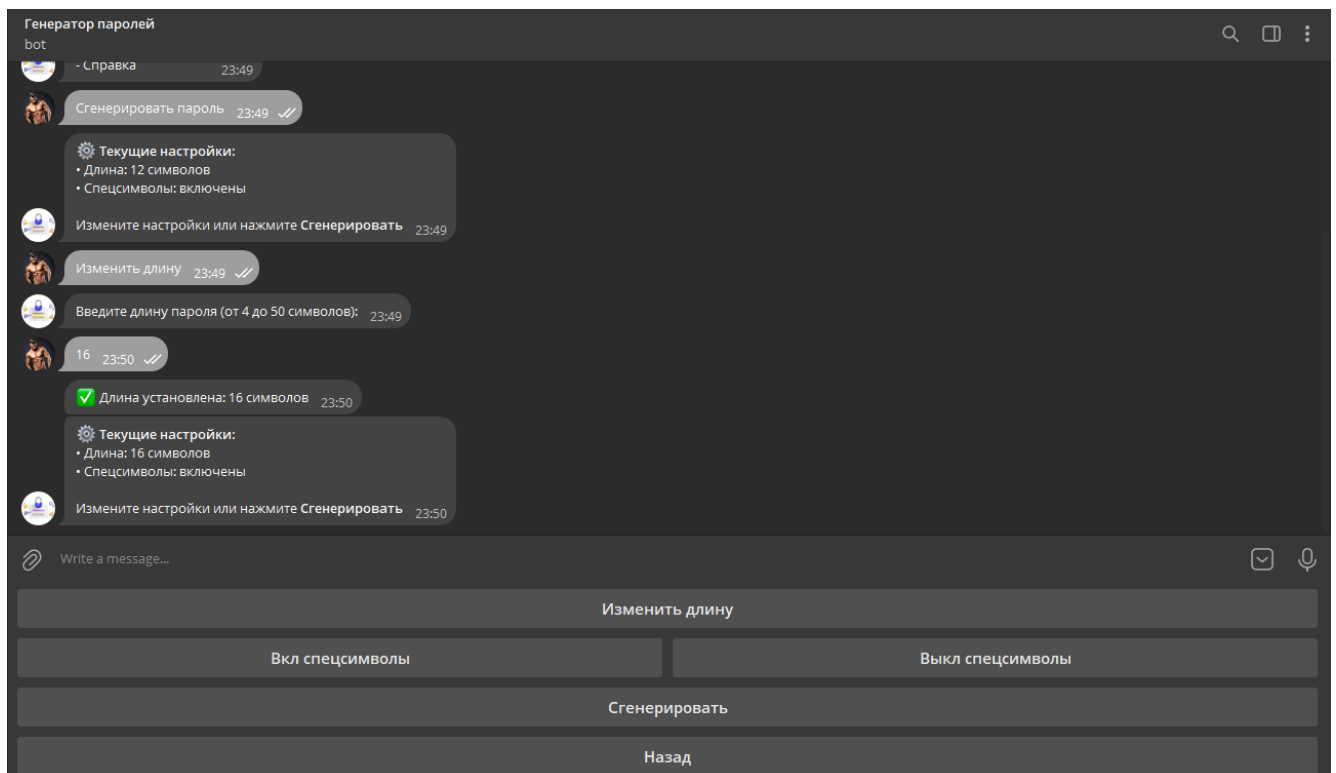
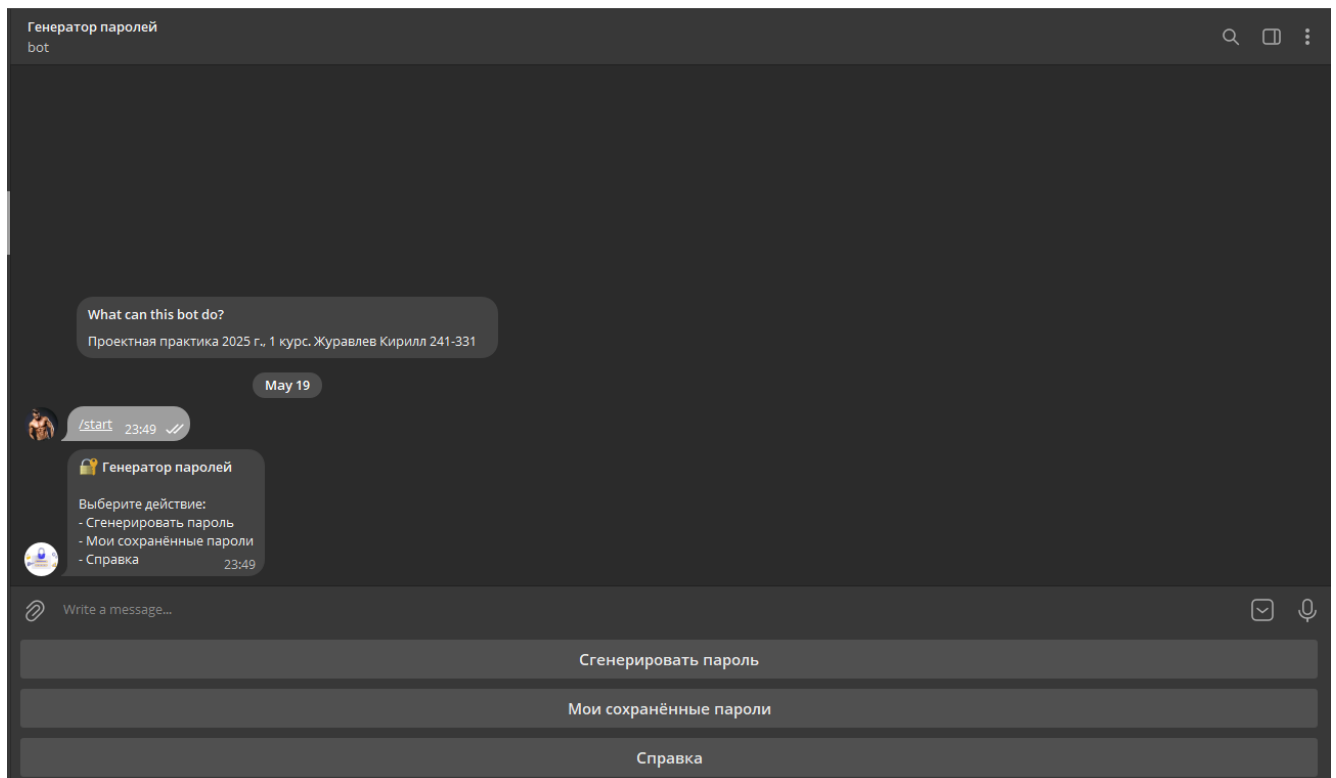
6. Запуск бота

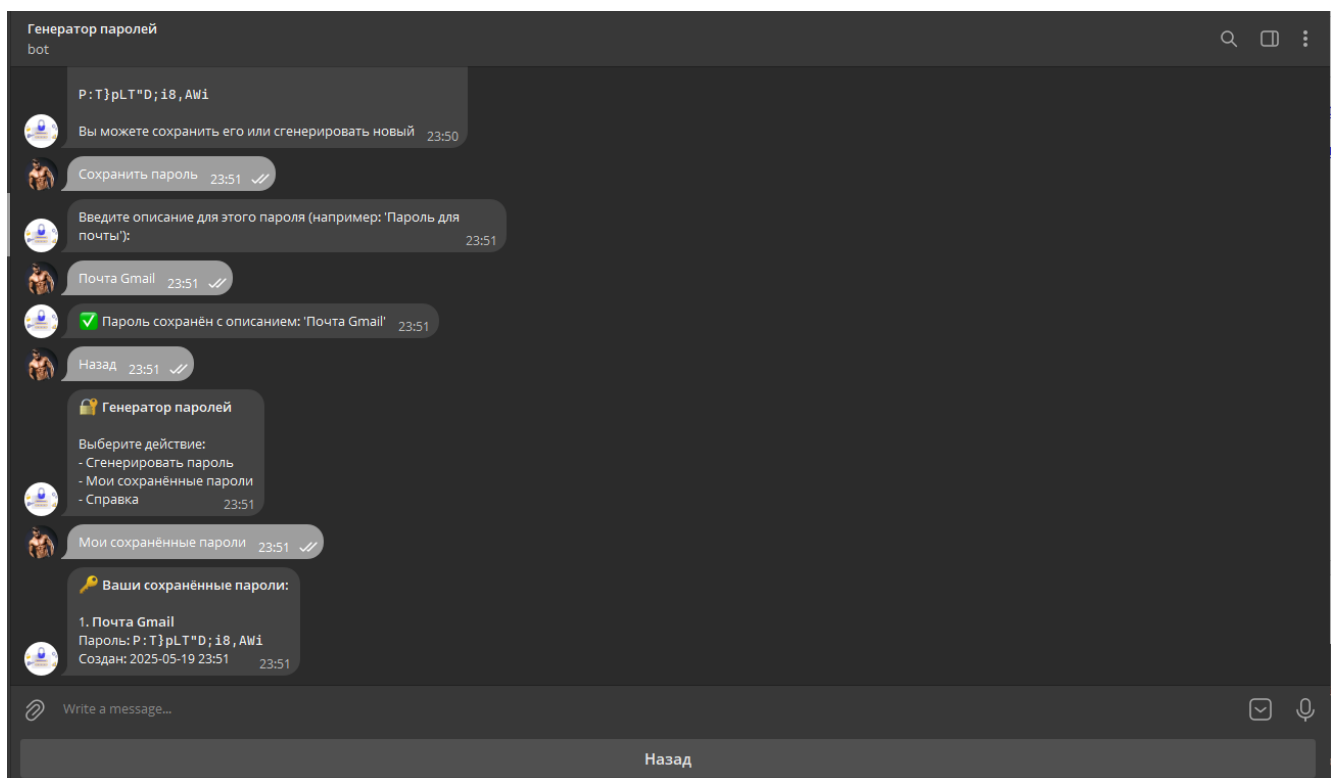
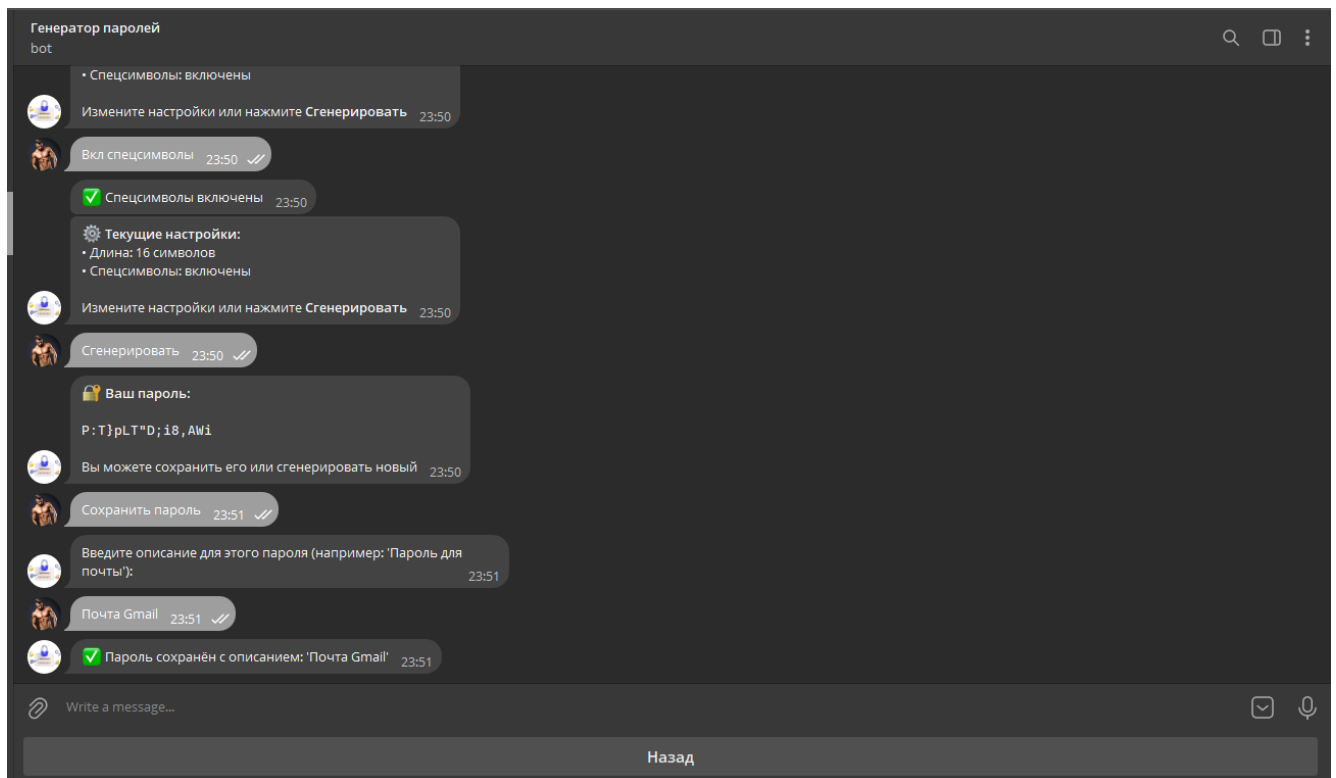
```
def main():  
    app = ApplicationBuilder().token(TOKEN).build()  
    app.add_handler(CommandHandler('start', start))  
    app.add_handler(MessageHandler(filters.TEXT,  
handle_message))  
    print("Бот запущен...")  
    app.run_polling()  
  
if __name__ == '__main__':  
    main()
```

7. Пример сессии работы

1. Пользователь: /start → бот показывает меню.
2. Пользователь: "Сгенерировать пароль" → бот запрашивает настройки.
3. Пользователь: Устанавливает длину 16, включает спецсимволы.
4. Бот: Генерирует пароль P:T}pLT"D;i8,AWi.
5. Пользователь: "Сохранить пароль" → вводит описание ("Почта Gmail").
6. Бот: Сохраняет в файл user_password.json.

8. Наглядный пример сессии работы





```
src > {} user_password.json > ...
1  {
2    "5796444935": [
3      {
4        "description": "\u041f\u043e\u0447\u0442\u0430 Gmail",
5        "password": "P:T}pLT\"D;i8,Awi",
6        "date": "2025-05-19 23:51"
7      }
8    ]
9  }
```

ЗАКЛЮЧЕНИЕ

В результате выполнения проектной практики все поставленные перед командой задачи были успешно выполнены. Разработанная система готова к внедрению в учебный процесс. Полученный опыт будет полезен в моей дальнейшей профессиональной деятельности. Работа имеет значимую ценность для заказчика (Московского Политеха), так как проект может быть внедрён как внутренняя система опросов или часть образовательных решений.

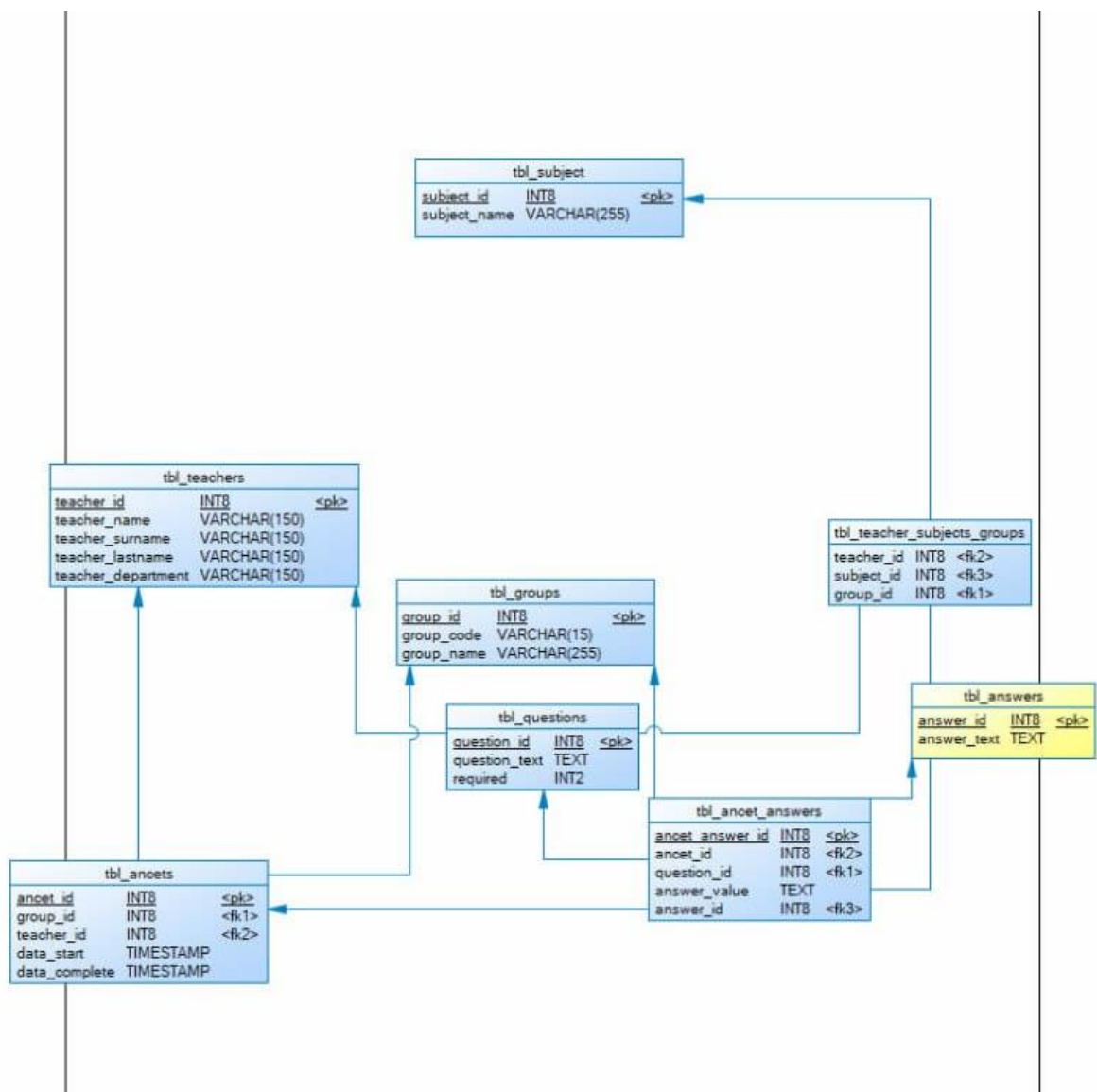
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Документация Python: <https://docs.python.org/3/>
2. Описание модуля xml.etree.ElementTree:
<https://docs.python.org/3/library/xml.etree.elementtree.html>
3. Документация по библиотеке BeautifulSoup:
<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
4. Документация по библиотеке openpyxl: <https://openpyxl.readthedocs.io/>
5. Документация HTML: <https://developer.mozilla.org/en-US/docs/Web/HTML>

6. Документация CSS: <https://developer.mozilla.org/en-US/docs/Web/CSS>
7. Документация по библиотеке python-telegram-bot: <https://docs.python-telegram-bot.org/en/stable/>
8. Документация по библиотеке json: <https://docs.python.org/3/library/json.html>

ПРИЛОЖЕНИЯ

UML-диаграмма базы данных:



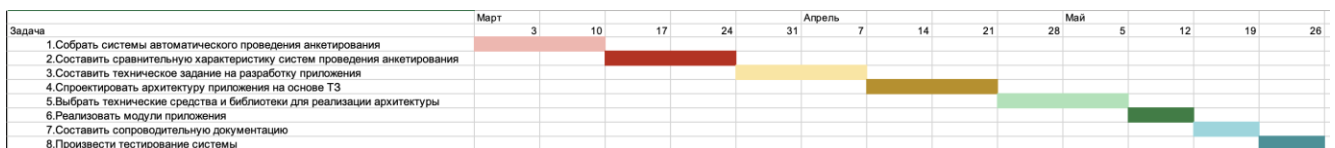
Пример XML-анкеты с вопросами для тестирования:

```
def add_question_items(items, count):
    for i in range(1, count + 1):
        item = ET.SubElement(items, "ITEM", TYPE="multichoicerated", REQUIRED="1")
        ET.SubElement(item, "ITEMID").text = str(i)

        # Текст вопроса
        ET.SubElement(item, "ITEMTEXT").text = f"Текст вопроса {i}"
        ET.SubElement(item, "ITELABEL").text = f"q{i}"
        ET.SubElement(item, "PRESENTATION").text = (
            "r>>>>5#### Качество проявляется всегда|"
            "4#### Качество проявляется часто|"
            "3#### Качество проявляется скорее редко, чем часто|"
            "2#### Качество проявляется редко|"
            "1#### Качество не проявляется"
        )

        ET.SubElement(item, "OPTIONS").text = "ih"
        ET.SubElement(item, "DEPENDITEM").text = "0"
        ET.SubElement(item, "DEPENDVALUE").text = ""
```

Диаграмма Ганта



Структура работы в проекте:



Подтверждаю, что отчет выполнен лично и соответствует требованиям практики.

ФИО:

Журавлев Кирилл Дмитриевич

Дата:

19.05.2025

Подпись:

A handwritten signature in black ink, appearing to be 'Жура', with a long horizontal stroke extending to the right.