**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***Факультет Информационных технологий  
Кафедра Информатики и информационных технологий***

**Направление подготовки  
09.03.03 «Информационные системы и технологии»**

**Проектная практика**

**Выполнили: студенты группы 241-337  
  
Бобров Денис, Езерская Маргарита, Лазарев Савва** (Фамилия И.О.)

**Дата, подпись 15.05.2025 \_\_\_\_\_\_\_\_\_\_\_\_  
 (Дата) (Подпись)**

**Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_  
 (Фамилия И.О., степень, звание) (Оценка)**

**Дата, подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_** **(Дата) (Подпись)**

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Москва  
2025**

***Отчёт по работе Telegram-бота для конвертации аудио в текст***

***Сравнение кода бота с инструкцией из freeCodeCamp***

1. ***Общая структура бота***

***По инструкции:***

- Используется библиотека python-telegram-bot

- Бот создаётся через класс Application

- Обработчики команд добавляются через декораторы

***В предоставленном коде:***

- Используется библиотека aiogram (более современная альтернатива)

- Бот создаётся через экземпляры Bot и Dispatcher

- Обработчики также добавляются через декораторы, но с другим синтаксисом

2. Обработка команд

***По инструкции:***

async def start(update: Update, context: ContextTypes.DEFAULT\_TYPE):

await update.message.reply\_text('Hello!')

application.add\_handler(CommandHandler("start", start))

***В предоставленном коде:***

@dp.message(CommandStart())

async def start\_command(message: types.Message):

await message.answer('Привет! Отправь мне аудиофайл и я конвертирую его в текст!')

***Различия:***

- Вместо CommandHandler используется декоратор @dp.message(CommandStart())

- Более современный синтаксис с аннотацией типов

- Ответ на русском языке и с уточнением функционала

3. ***Обработка медиафайлов***

***По инструкции:*** не рассматривается работа с аудиофайлами

***В предоставленном коде:*** реализована полноценная обработка аудио:

@dp.message(F.audio)

async def converting\_audio\_to\_text(message: types.Message):

# Скачивание файла

# Конвертация в WAV через ffmpeg

# Распознавание речи через Google Speech Recognition

# Отправка результата

# Удаление временных файлов

4. ***Дополнительные зависимости***

***По инструкции***: только python-telegram-bot

***В предоставленном коде:***

- aiogram (альтернативная библиотека для работы с Telegram API)

- speech\_recognition для распознавания речи

- ffmpeg для конвертации аудиоформатов

- asyncio для асинхронной работы

***5. Безопасность:***

- В коде присутствует реальный токен бота (что является плохой практикой)

- В инструкции рекомендуется хранить токен в переменных окружения

***Выводы:***

Предоставленный код:

1. Использует более современную библиотеку aiogram вместо python-telegram-bot

2. Реализует конкретный полезный функционал (конвертация аудио в текст), а не просто демонстрацию возможностей

3. Добавляет сложную логику работы с медиафайлами и сторонними сервисами

4. Требует больше зависимостей и более сложной настройки окружения

5. Недостаточно обрабатывает возможные ошибки и содержит токен в коде