### **1. Жалпы ақпарат**

**Боттың атауы:** [@alimansur\_bot]  
 **Боттың мақсаты:** [Сервер калай жумыс аткаратынын билу, телеграмм боттын ишки курлысымен танысу]  
 **Даму тілі:** Python 3.13.2  
 **Дерекқор:** (жок тикелей ноутбук аркылы)  
 **Орнатылған кітапханалар:**

aiogram python-dotenv

pyTelegramBot

### **2. Жүйенің архитектурасы**

**Бот келесі негізгі модульдерден тұрады:**

1. Қолданушы интерфейсі (UI) модулі

• aiogram негізінде құрылған Telegram интерфейсі

• Пайдаланушы хабарламаларын өңдеу

• Кері байланыс үшін жауаптар жіберу

**2. Бизнес-логика модулі**

• Командаларды және батырмаларды өңдеу

• Қолданушыларға байланысты әрекеттерді орындау

• Құжаттар мен презентацияларды жіберу

3. **Дерекқор модулі (егер қажет болса)**

**•** SQLite/PostgreSQL/MongoDB қолдау

• Қолданушы деректерін сақтау (мысалы, ботты қолдану статистикасы)

**4. Файл менеджмент модулі**

• FSInputFile арқылы файлдарды жүктеу және жіберу

• Құжаттар мен презентацияларды сақтау

**5. Асинхронды жұмыс модулі**

• asyncio және aiogram арқылы асинхронды сұраныстарды орындау

• Көп пайдаланушы режимінде тұрақты жұмыс

### **3. Командалар мен функционал**

### **4. Боттың жұмыс істеу логикасы**

**Қолданушының әрекеті:**

1. /start командасын жібереді
2. Бот басты мәзірді ұсынады
3. Қолданушы "📝 Құжат жасау" батырмасын басады
4. Бот қажетті мәліметтерді сұрайды
5. Мәліметтерді жинап, құжатты PDF/DOCX форматында жасайды
6. Дайын құжатты қолданушыға жібереді

### **5. Орнату және іске қосу**

**1. Репозиторийді клондау:**

git clone https://github.com/your-repo.git

cd your-repo

**2. Қажетті кітапханаларды орнату:**

pip install -r requirements.txt

**3. .env файлын жасау және токенді енгізу:**

BOT\_TOKEN=your\_telegram\_bot\_token

**4. Ботты іске қосу:**

python main.py

### **6. Қосымша ақпарат**

Егер ботта қателер болса, logging арқылы журнал жүргізу ұсынылады:

import logging

logging.basicConfig(level=logging.INFO)

Осылайша, боттың жұмысын бақылау жеңіл болады.

Жумыс истеу коды  
**import** asyncio

**from** aiogram **import** Bot, Dispatcher, types, F

**from** aiogram.types **import** ReplyKeyboardMarkup, KeyboardButton, Message

**from** aiogram.types.input\_file **import** FSInputFile # <-- Важно!

TOKEN = "вв"

bot = Bot(token=TOKEN)

dp = Dispatcher()

# Создаем клавиатуру

kb = ReplyKeyboardMarkup(

keyboard=[

[KeyboardButton(text="Студенты 👨‍🎓"), KeyboardButton(text="Тех документация")],

[KeyboardButton(text="Презентация")]

],

resize\_keyboard=**True** # Делаем кнопки компактными

)

@dp.message(F.text == "/start") # Новый синтаксис для фильтрации команд

**async** **def** start(message: Message):

**await** message.answer("Привет!", reply\_markup=kb)

@dp.message(F.text == "Студенты 👨‍🎓")

**async** **def** students(message: Message):

students\_list = ["Толебек Алимансур", "Декабрь Еркежан"]

**await** message.answer("\n".join(students\_list))

@dp.message(F.text == "Тех документация")

**async** **def** send\_tech\_doc(message: Message):

file\_path = "/Users/mackbook/my\_telegram\_bot/aaa.docx"

**try**:

document = FSInputFile(file\_path) # <-- Используем FSInputFile

**await** message.answer\_document(document, caption="Техническая документация")

**except** Exception **as** e:

**await** message.answer(f"Ошибка при отправке файла: {e}")

@dp.message(F.text == "Презентация")

**async** **def** send\_presentation(message: Message):

file\_path = "/Users/mackbook/my\_telegram\_bot/files/presentation.pdf"

**try**:

document = FSInputFile(file\_path) # <-- Используем FSInputFile

**await** message.answer\_document(document, caption="Презентация")

**except** Exception **as** e:

**await** message.answer(f"Ошибка при отправке файла: {e}")

**async** **def** main():

**await** dp.start\_polling(bot)

**if** \_\_name\_\_ == "\_\_main\_\_":

asyncio.run(main())