

Telegram bot.

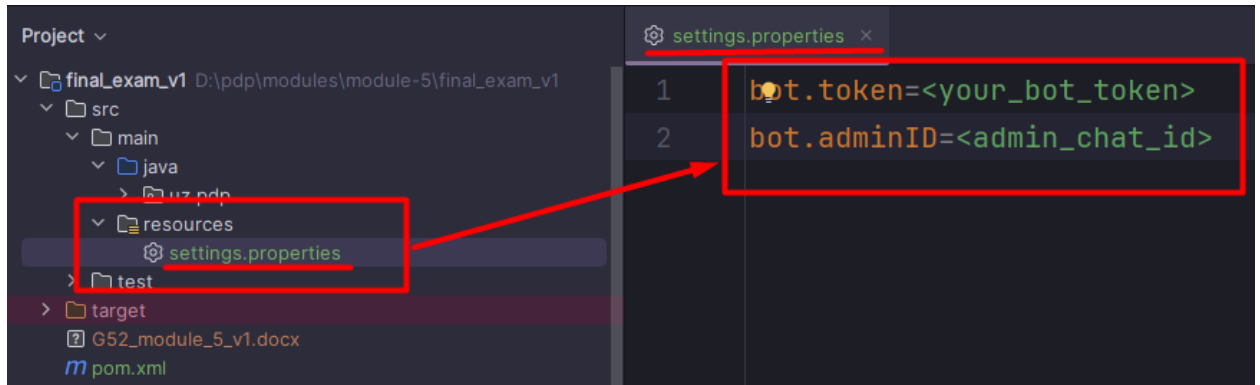
Variant-1

Mavzu: BookShopBot

1. User entity bo`lishi va kamida chatId si saqlanishi kerak;
2. UserState enum yaratilinishi kerak va barcha statelar uchun alohida value ga ega bo`lishi kerak;
3. Book entity bo`lishi va u o`zida:
 - Id (UUID);
 - Name (O`tkan kunlar);
 - Price (100\$);
 - Quantity (10 do`Konda qancha kitob bor ekanligi);
 - BookPhoto (kitobning tashqi rasmi, .jpeg, .png va h.k);
 - BookDocument (kitobning materiali: .pdf, .txt, .epub va h.k);Field larga ega bo`lishi kerak.
4. Order entity – bu history uchun kerak, ya`ni user xarid qilgan kitoblar ro`yxatini ko`rish uchun. U
 - Id (UUID);
 - userChatId (Long)
 - bookId(UUID)
 - Name (O`tkan kunlar);
 - Price (100\$);
 - Quantity (10 do`Konda qancha kitob bor ekanligi);
 - bookPhotoFileId (String);
 - bookDocumentFileId (String);
 - isSold(Boolean)Field larga ega bo`lishi kerak.
5. Bot dan **admin** va **user** foydalanishiligi kerak; Admin uchun alohida menu va user uchun alohida menu shakllantirilishi kerak;
6. **<dependency>**
<groupId>com.github.pengrad</groupId>
<artifactId>java-telegram-bot-api</artifactId>
<version>8.3.0</version>
</dependency>
<dependency>
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>

```
<version>1.18.38</version>  
</dependency>
```

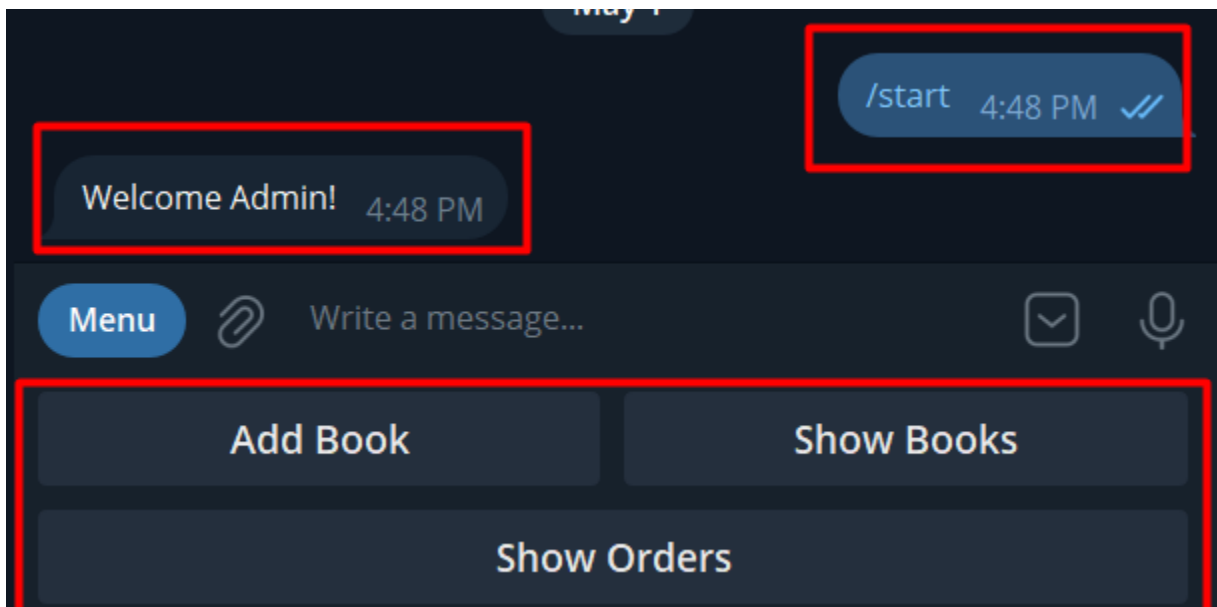
- 7.
8. `/resources` papkani ichida **settings.properties** file yaratilishi va unda bot token va admin chatId si yozilgan bo'lishi kerak.



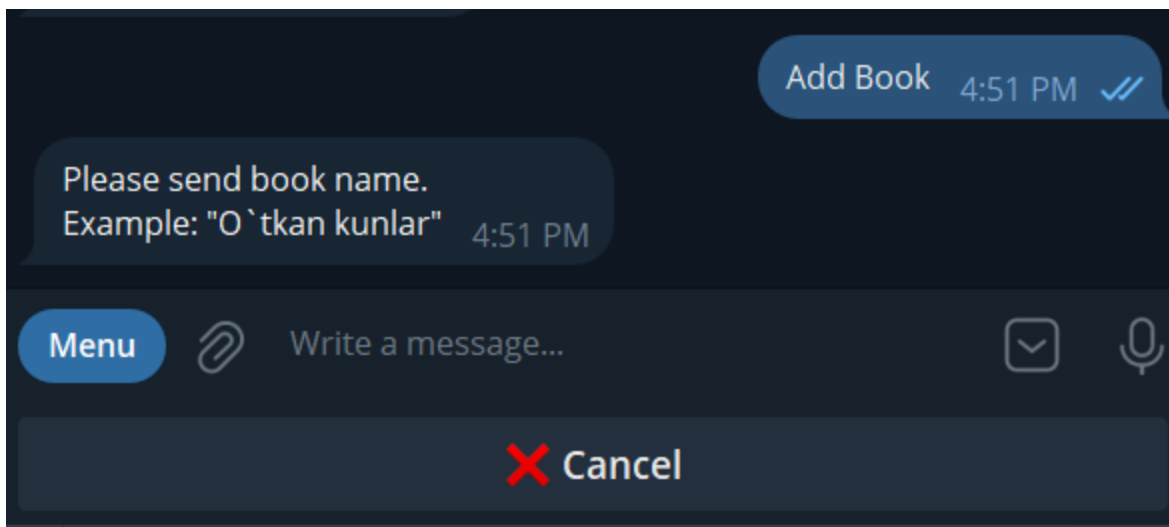
9. Project run qilinganda database shakllantirilib(List dan foydalaning), *settings.properties* dagi **bot.adminID** orqali admin database ga avtomatik qo'shilishi kerak;

Admin menu

1. Admin `/start` buyrug'ini berganda rasmda ko'rsatilgan response va buttonlar chiqishi kerak

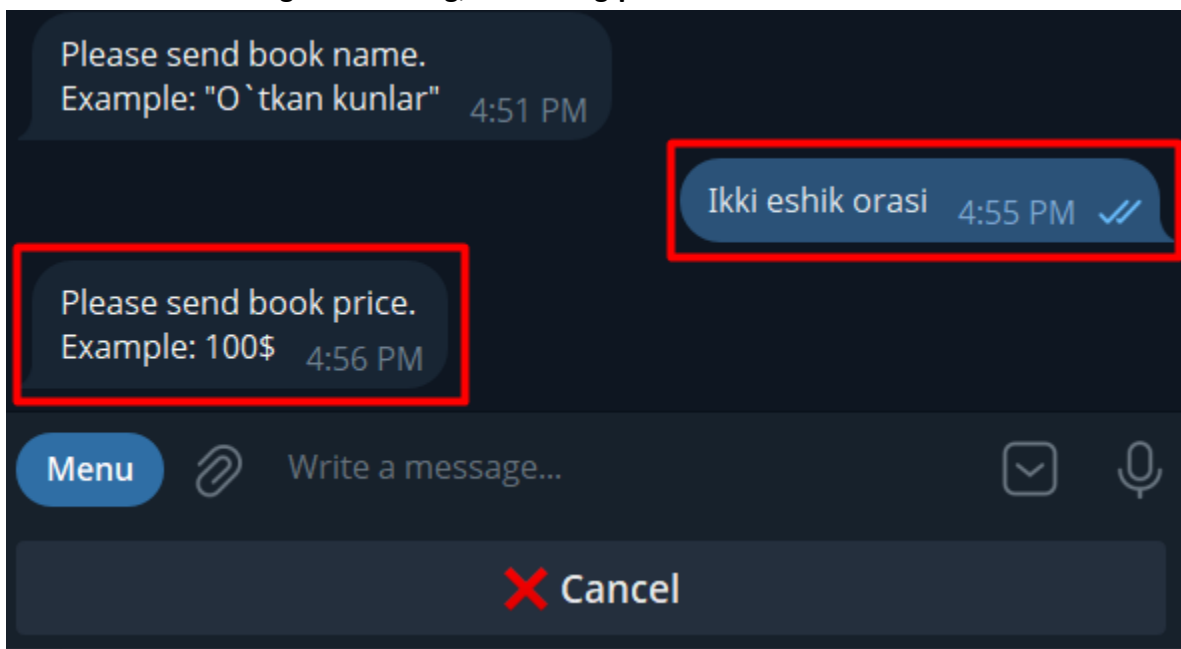


2. **Add Book** button bosilsa , quyidagi rasmda ko'rsatilgan response va **Cancel** buttoni chiqishi kerak.



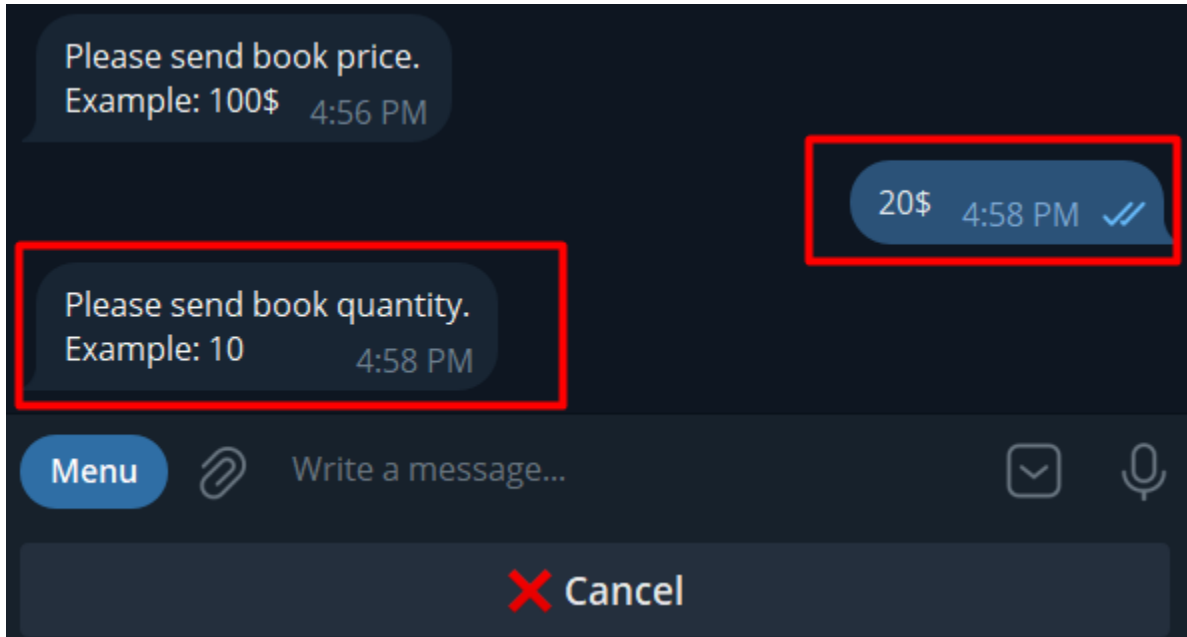
Cancel button bosilga admin **Admin Menu** ga qaytishi va amalyot bekor qilinishi kerak. Aks holda add book jarayoni davom etishi va keying state ga o`tishi kerak.

3. Book name kiritilgandan so`ng, book ning **price** kiritilishi so`ralishi kerak:



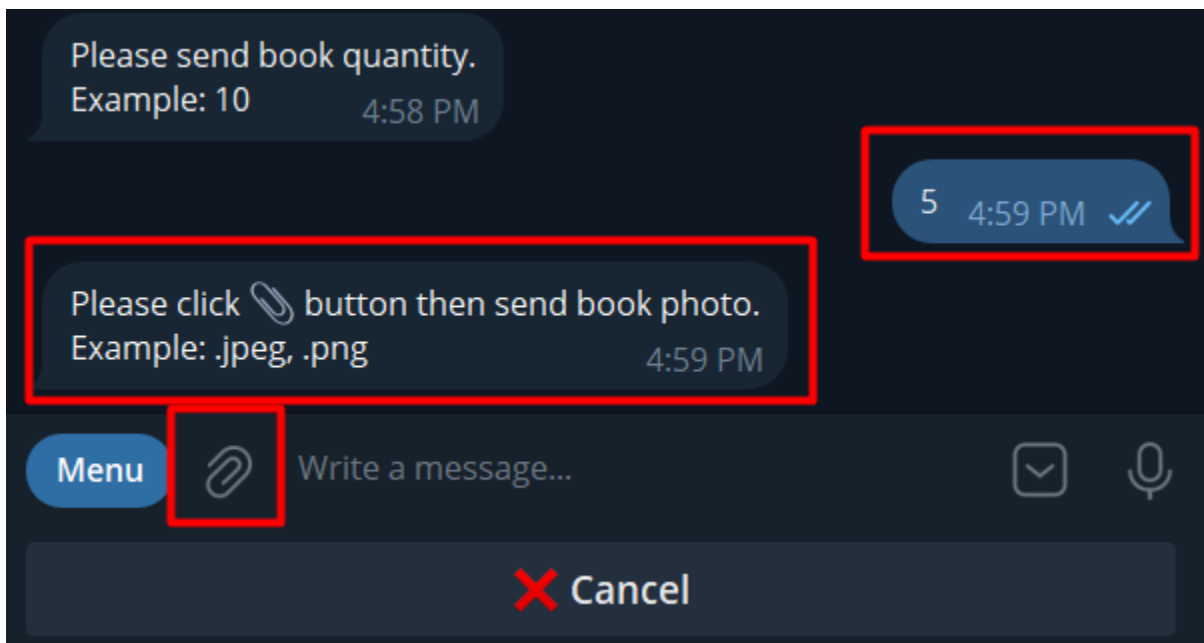
Cancel button bosilga admin **Admin Menu** ga qaytishi va amalyot bekor qilinishi kerak. Aks holda add book jarayoni davom etishi va keying state ga o`tishi kerak.

4. Book price kiritilgandan so`ng book ning **quantity** (do`Konda kitobdan qancha mavjud ekanligi) si so`ralishi kerak:



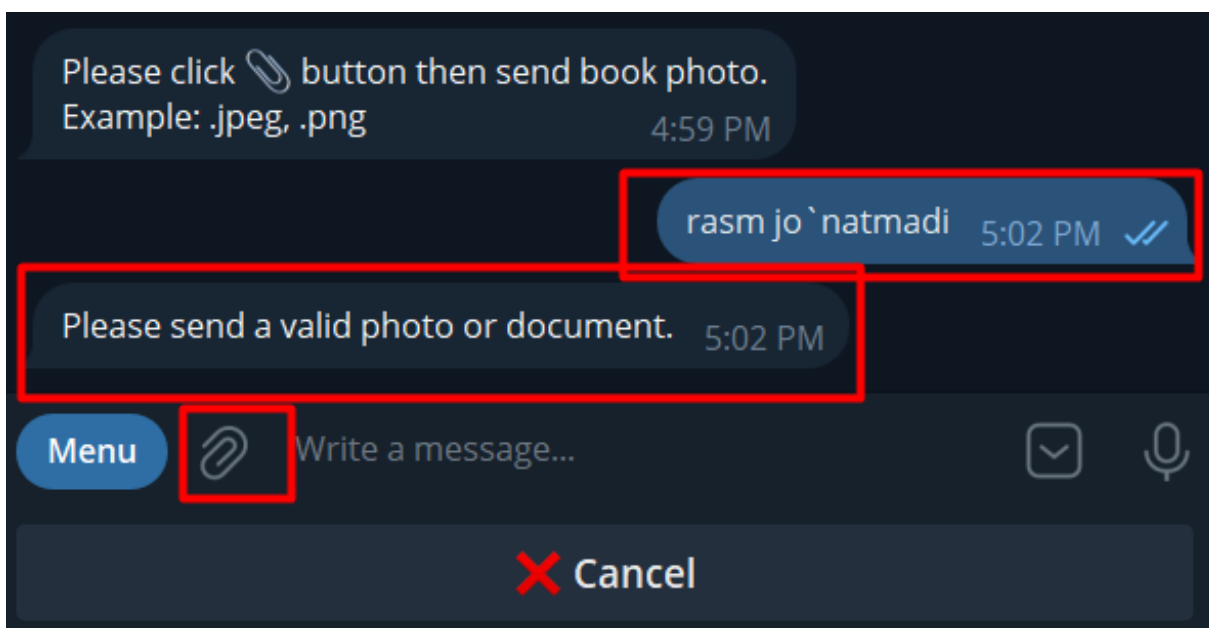
Cancel button bosilga admin **Admin Menu** ga qaytishi va amalyot bekor qilinishi kerak. Aks holda add book jarayoni davom etishi va keying state ga o`tishi kerak.

5. Book quantity kiritilgandan so`ng book ning **tashqi yuz sahifasining rasmi** jo`natilishi kerak:

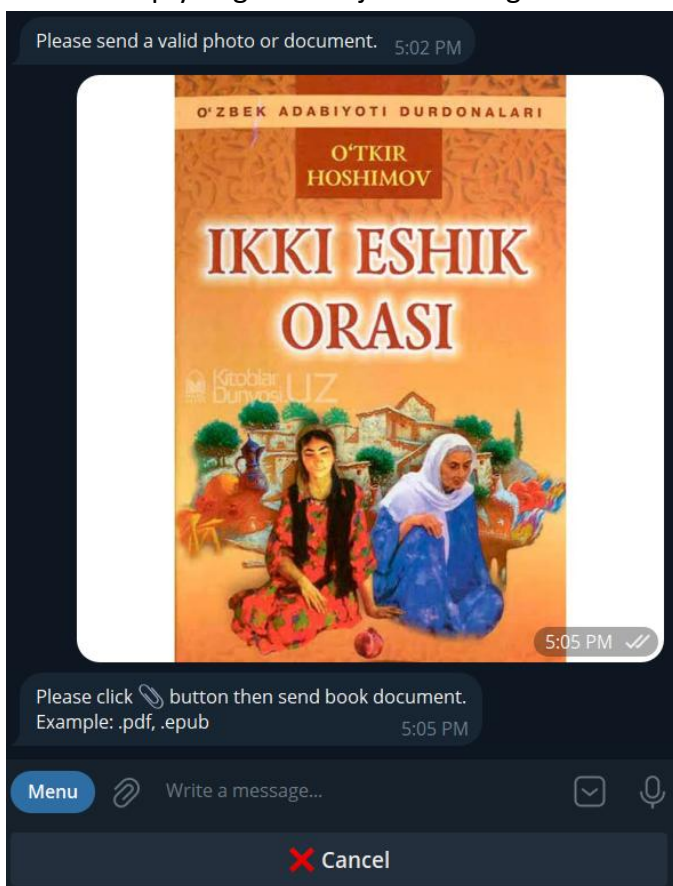


agar file jo`natmasdan biror message yozilga xatolik qaytarilishi va qayta rasm jo`natilishligi

so`ralishi kerak:

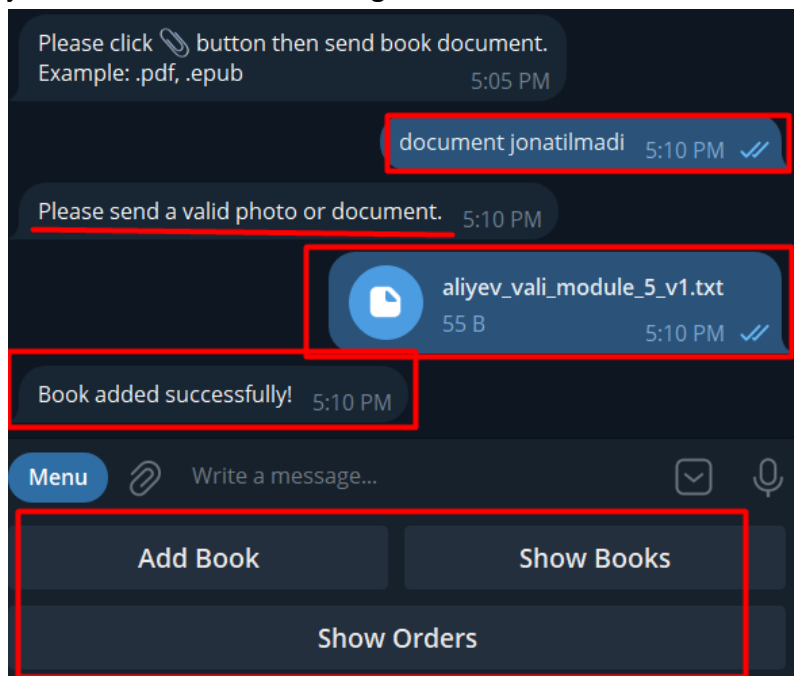


aks holda quyidagicha file jo`natilishligi kerak:



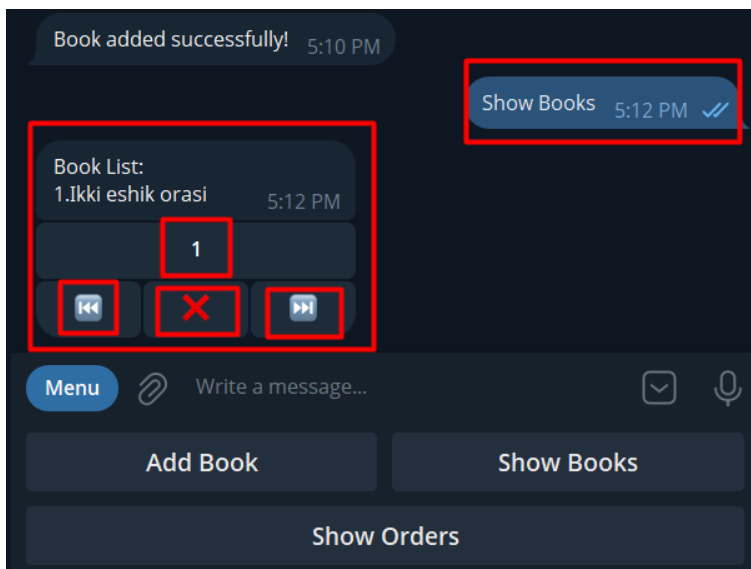
Cancel button bosilga admin **Admin Menu** ga qaytishi va amalyot bekor qilinishi kerak. Aks holda add book jarayoni davom etishi va keying state ga o'tishi kerak.

6. Book ning **tashqi yuz sahifasining rasmi** kiritilgandan so'ng book ning electron formati (.pdf, .txt, .epub va h.k) document jo'natilishi kerak va bu jarayon ham **tashqi yuz sahifasining rasm** ni olish kabi xatoliklardan tekcrishilgan bulishi va document junatilsa u saqlanib **Add book** jarayoni yakunlanishi va book muvaffaqiyatli saqlanganligi to'g'risida message adminga jo'natilishi va admin menu ga o'tishi kerak:



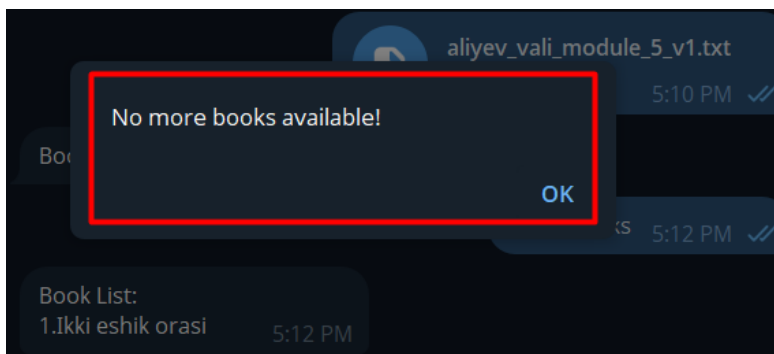
Cancel button bosilga admin **Admin Menu** ga qaytishi va amalyot bekor qilinishi kerak. Aks holda add book jarayoni davom etishi va keying state ga o'tishi kerak.

7. **Show Books** button bosilsa, database da bor kitoblar 10 talik list va inline buttonlar bilan birga previous, reject, next button lar bilan response qaytishi kerak:




next  button bosilsa

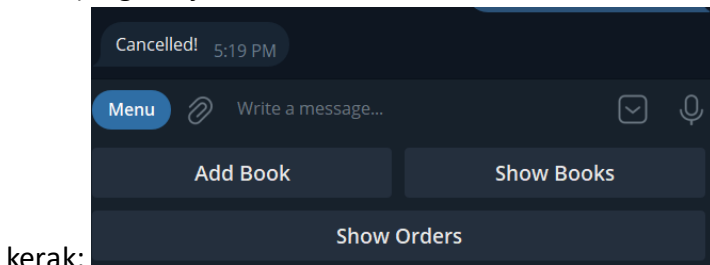
agar keyingi sahifa mavjud bo'lsa keyingi book lar listi qaytarilishi kerak (bunda oldingi list edit qilinishi kerak, ya'ni oldingi list delete qilinib qayta jo'natilmasligi kerak); agar keyingi list mavjud bo'lmasa quyidagi message chiqishi kerak:



previous  button bosilsa

ortqa qaytish amali bajarilishi kerak (xuddi next buttoni bosilgani kabi list update qilinishi

kerak); agar reject  button bosilsa bu list delete qilinishi va admin menu ga o'tishi



kerak: . Agar book raqami mavjud bo'lgan

button  bosilsa o'sha kitobning ma'lumotlari quyidagicha

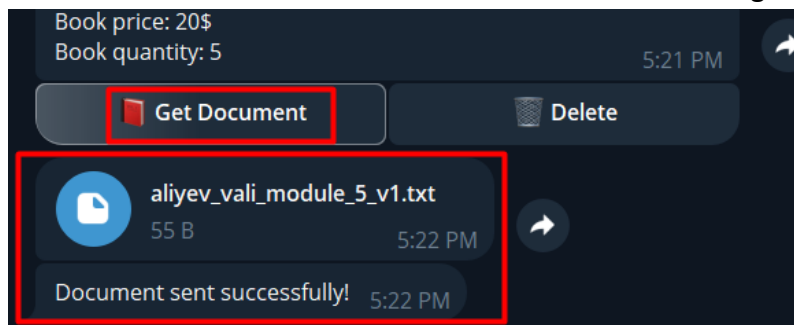


qaytarilishi kerak:

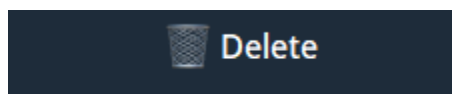
. Va



button bosilsa kitobning documenti qaytarilishi kerak:



yoki



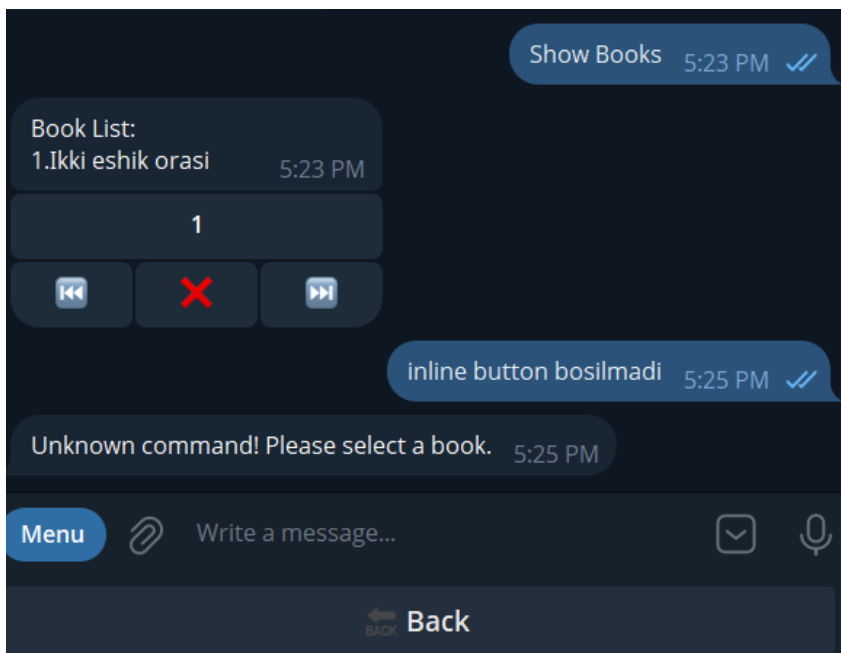
button bosilsa kitob databasedan delete qilinishi kerak.



Show Books button bosilganda inline button lar bosilmasa

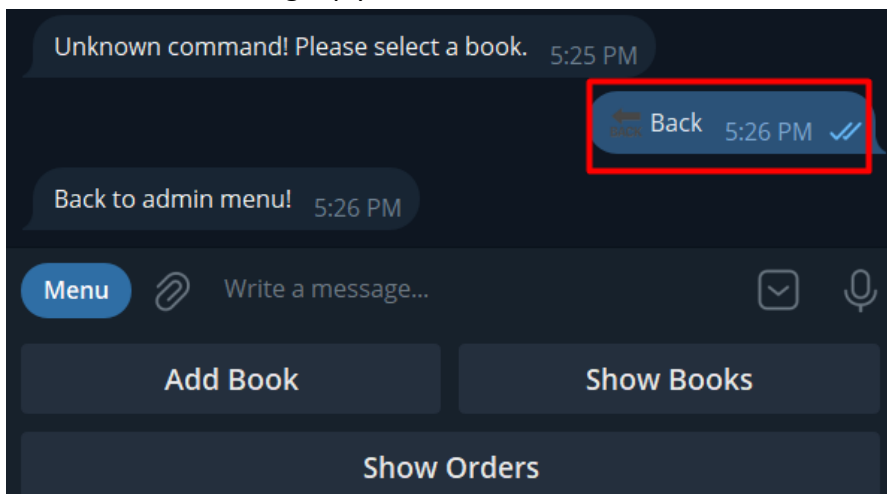


, hamda biror boshqa keyboard button yoki message yozilsa quyidagicha xatolik chiqishi kerak:

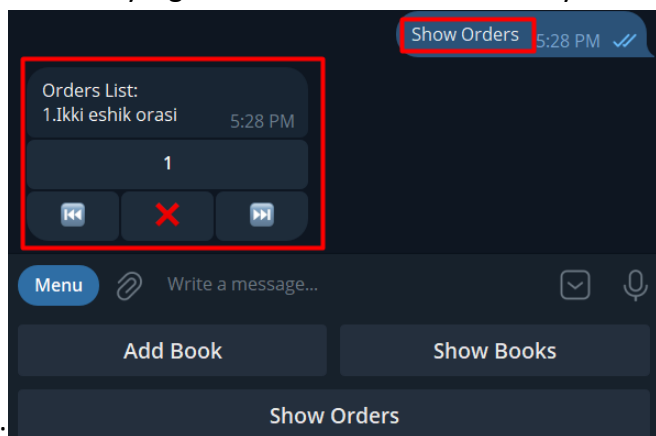


bu yerda **Back** button

bosilsa **admin menu** ga qaytarilishi kerak:



8. **Show Orders** buyrug'i berilsa barcha orderlar ro'yxati Book list kabi button lar bilan



chiqishi kerak:



book list dagi kabi ishlashi kerak;



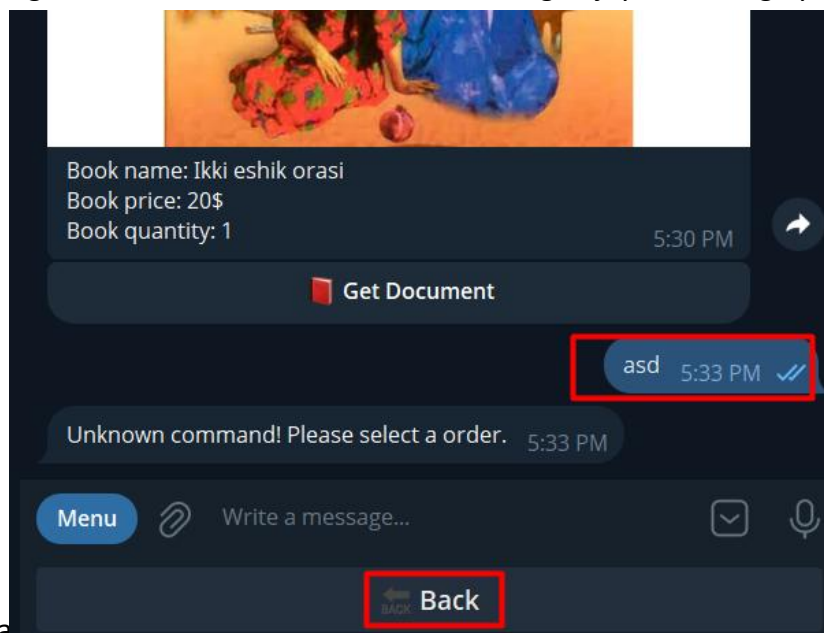
button bosilsa quyidagi response qaytishi kerak:



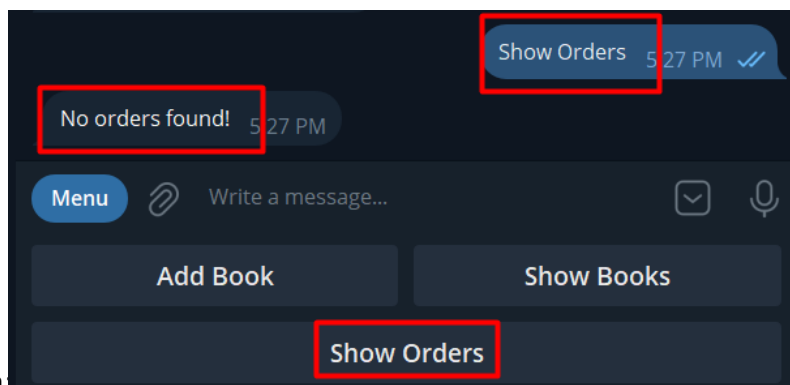
bunda

 **Get Document**

button book list da qanday vazifani bajarsa bu yerda ham shu vazifani bajarishi kerak; agar inline button lar bosilishi kerak bulgan joyda message yoki



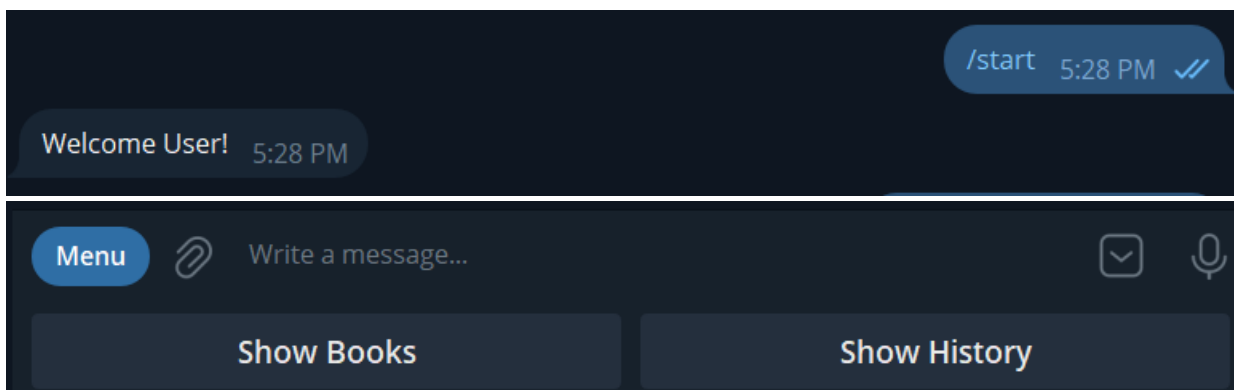
keyboard button bosilsa quyidagi response va **Back** jo`natilinishi kerak; va **Back** button bosilsa admin menu ga qaytishi kerak;



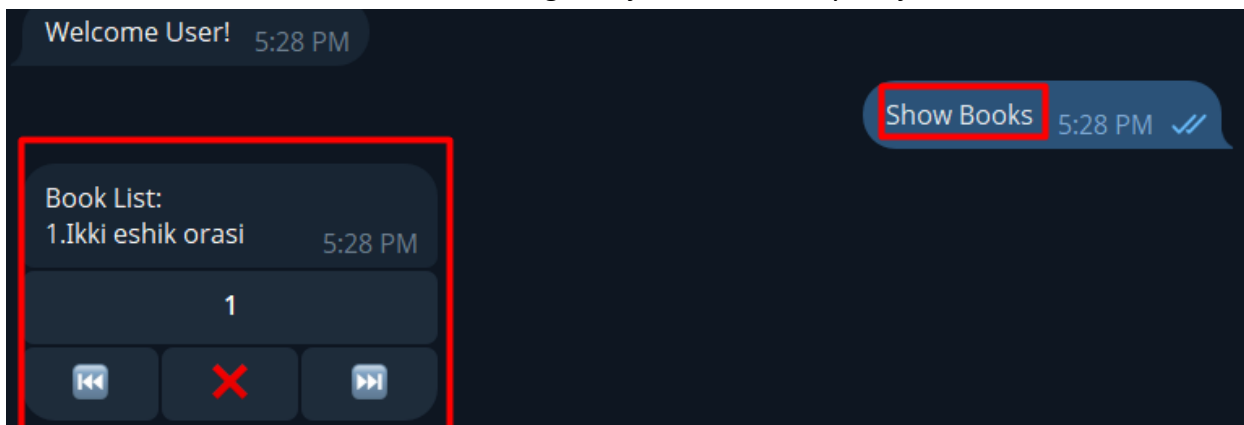
agar order mavjud bo`lmasa quyidagi javob qaytishi kerak.

User Menu

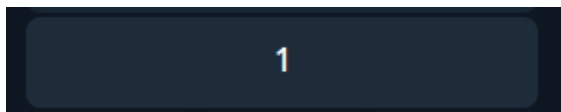
1. Oddiy user bot ga /start buyrug`ini bersa, database ga yangi user yaratilishligi kerak va user menu unga keyboard button lar bilan jo`natilishi kerak:



2. **Show Books** button bosilsa database dagi mavjud kitoblar ro'yxati jo'natilishi kerak:



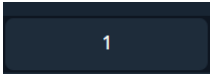
, bu yerdagi previous, reject, next button lari yuqoridagi tartibda ishlashligi kerak,



button bosilsa, user uchun o'sha book ning malumotlari quyidagi ko'rinishda jo'natilinishi kerak:




bu yerda  button

bosilganda agar  button ning qiymat **1** bo'lsa, quyidagi xabar chiqishi kerak:



, 1 dan katta bo'lsa, uning qiymati bittaga

kamaytirilishi kerak,  button bosilganda shu kitobdan database da nechta bo'lsa, o'shangacha qiymatini 1 ga oshirish imkoni berilishi kerak:

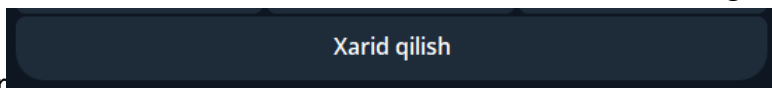


bosilganda database da mavjud kitoblar sonidan qiymat oshib ketmasligi va



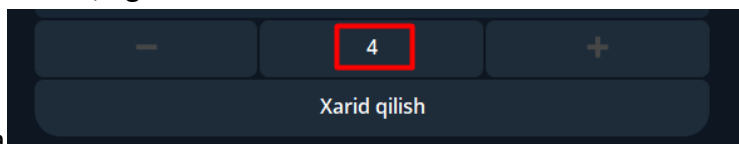
shu kabi message userga

chiqishi kerak; agar



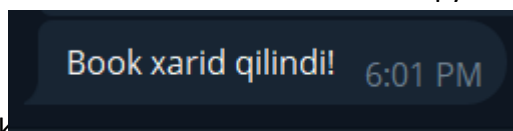
button

bosilsa



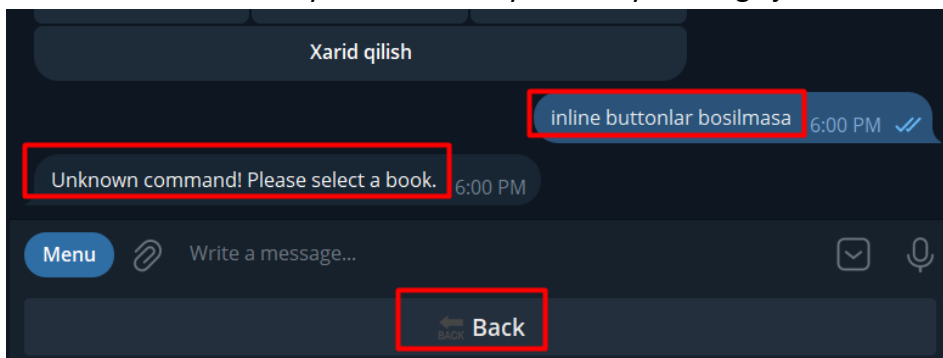
shu qiymatcha kitob xarid

qilinishi va kard delete bo'lishi kerak



va database

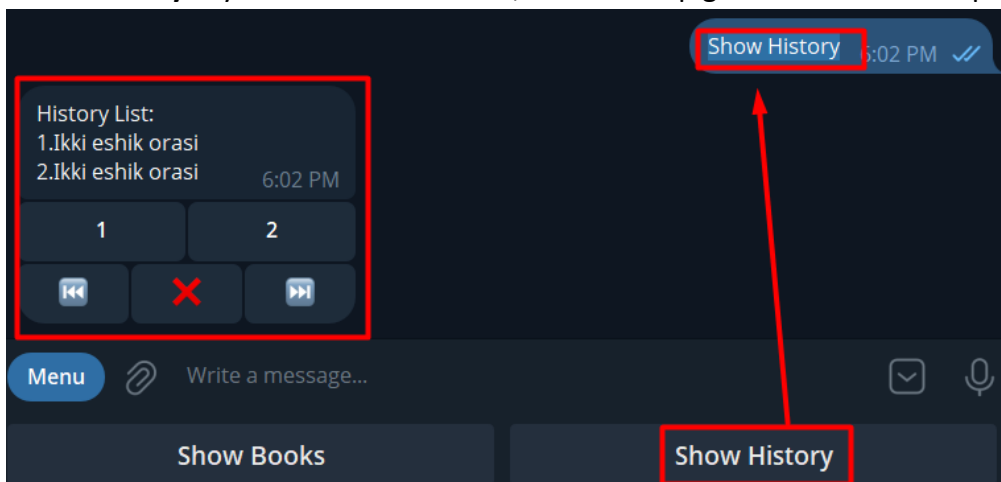
da book quantity dan shu miqdorch qiyamat kamayishi kerak; bu jarayonda inline button lardan bosilmasdan keyboard button yoki oddiy message junatilsa



mana shu javob

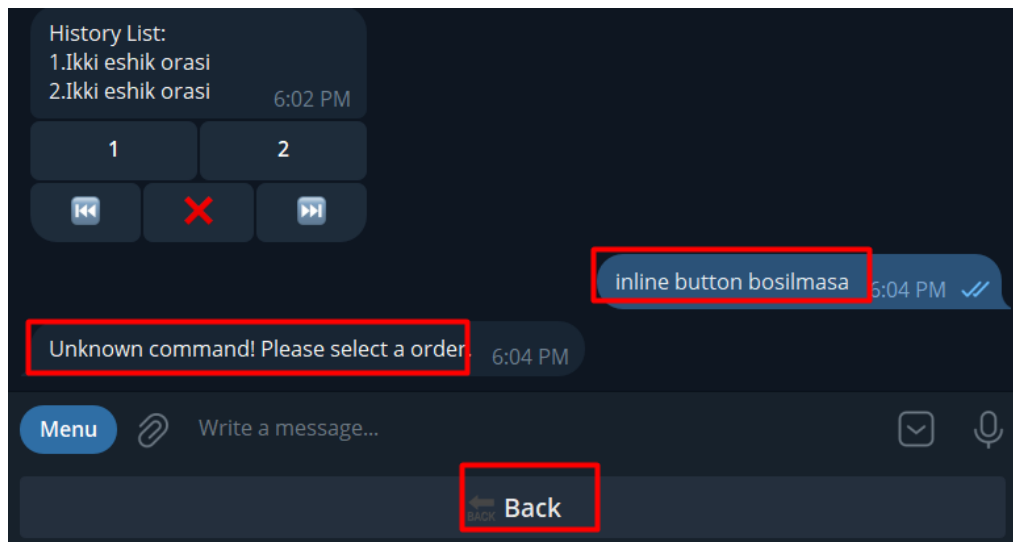
qaytishi va **Back** bosilsa user menu ga qaytarilishi kerak.

3. **Show History** keyboard button bosilsa, user xarid qilgan kitoblar listi chiqishi kerak:



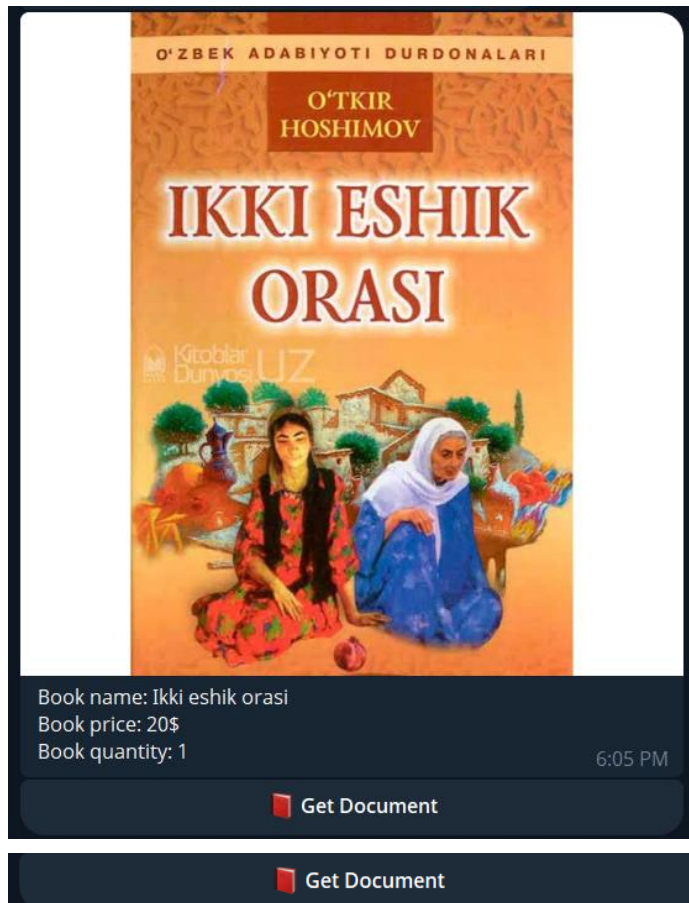
bu yerda

previous, reject, next button lar oldingi holatlar kabi ishlaydi; inline button lar bosilmasa:



quyidagi

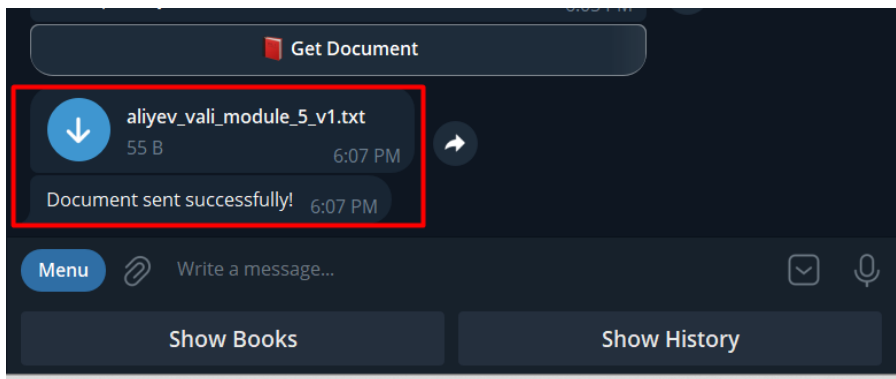
javob qaytishi va **Back** bosilsa, user menuga qaytishi kerak; agar **1** button bosilib, orderni user ko`rmoqchi bo`lsa quyidagi card qaytarilishligi kerak:



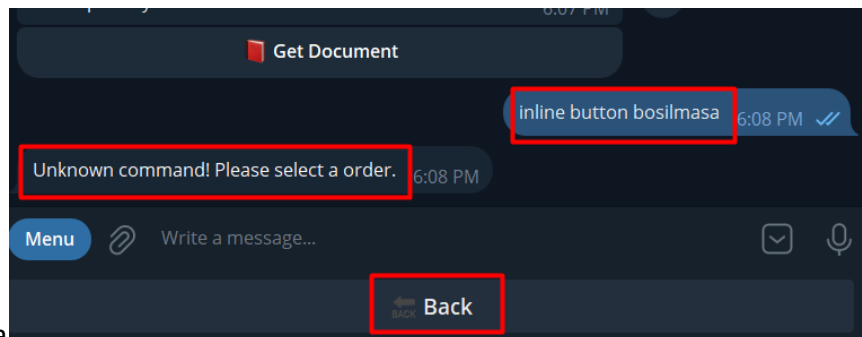
va bu yerda

documentini olish imkoniga ega bo`ladi:

button ni bosish orqali user book ning



; agar inline button



bosilmasa

javob qaytishi

va **Back** bosilganda user menu ga user qaytishi kerak;

Savollar.

1. Thread nima?
2. Multithreading va Multitaskingni farqi nimada?
3. ThreadPool nima?
4. Callable va Runnable farqi va qachon nima uchun ishlatiladi?
5. Future nima va qachon ishlatiladi?
6. Future va CompletableFuture ning farqi?
7. Volatile non-access modifikatori haqida malumot bering. Nima uchun ishlatiladi.
8. Immutable classlar nima uchun kerak va immutable class yaratishimiz uchun qanday shartlar bajarilishi kerak?
9. Asynchrone programming nima?
10. Atomic classlar qanday algoritim orqali race conditionni oldini oladi? Yoki thread-safe?
11. Serialazition/deserialazition nima?
12. Serializable/extrenizeble interfacelarining farqlari nima?
13. ReentrantLock haqida malumot bering.
14. Logging Nima?
15. Daemon thread nima?
16. Jar file nima?
17. Maven nima?
18. Behavior Parameterization nima?
19. Declarative va Imperative programming nima?
20. Stream nima? Stream va Collection larning farqi nimada?

Exam ni yakunlash tartibi:

1. Github da **familiya_ism_module5_v1(2)** ko`rinishida repository yarating va project'ingizni github ga joylang.
2. Readme.md file yarating unda

Aliyev Vali. 1-variant

bot_url : @g52_logger_bot

o`zingizga tegishli malumotlarni yuqorida ko`rsatilgan tartibda ifodalang.

3. Savol-javob.md file yarating va unda barcha savollarga javoblaringizni ifodalagan bo`lishingiz kerak.