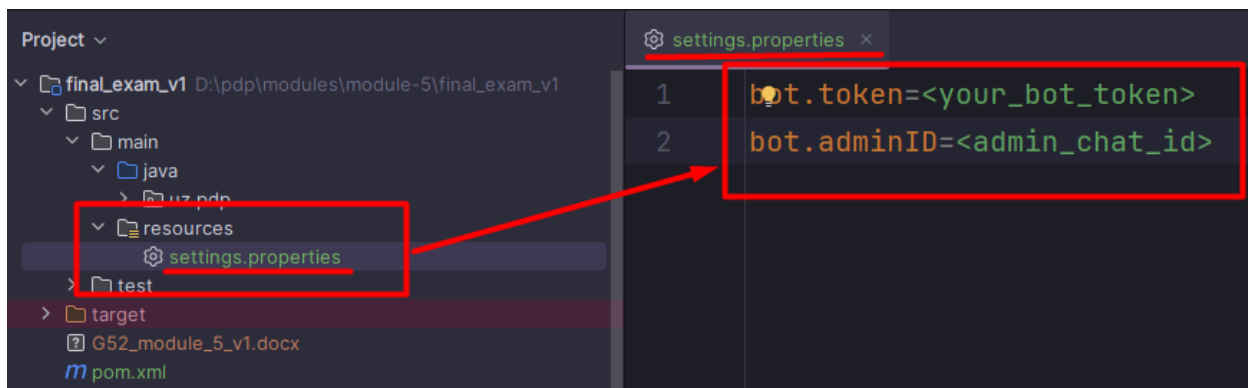


Telegram bot.

Variant-2

Mavzu: HomeworkCheckBot

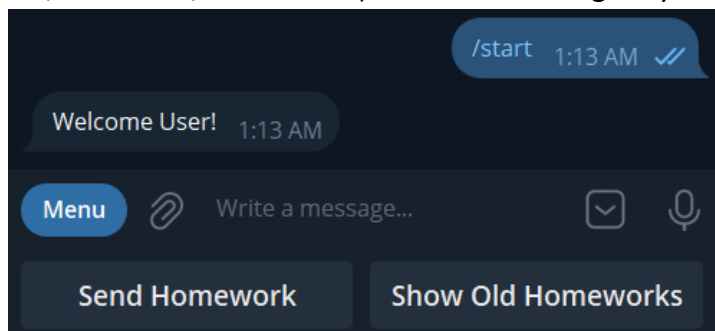
1. User entity bo`lishi va o`zida:
  - chatId;
  - firstName;
  - lastName;
  - username; fieldlari bo`lishi kerak.
2. UserState enum yaratilinishi kerak va barcha statelar uchun alohida value ga ega bo`lishi kerak;
3. Homework entity bo`lishi va u o`zida:
  - Id (UUID);
  - userChatId(Long);
  - themeOrDescription(String mavzu nomi yozilishi kerak);
  - zipFileId(String)
  - ball (int 1-5 – teacher ball beradi );
  - teacherFeedback (String – teacher ball bergandan keyin qiymat beradi);
  - sendTime (Time – student homework ni teacher ga jo`natga vaqt);
  - checkTime (time – teacher homework ga ball berib, feedback yozib bo`lgan vaqt);Field larga ega bo`lishi kerak.
4. Bot dan **admin(teacher)** va **user(student)** foydalanishliligi kerak; Admin uchun alohida menu va user uchun alohida menu shakllantirilishi kerak;
5. **<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>**
- 6.
7. **/resources** papkani Ichida **settings.properties** file yaratilishi va unda bot token va admin chatId si yozilgan bo`lishi kerak.



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

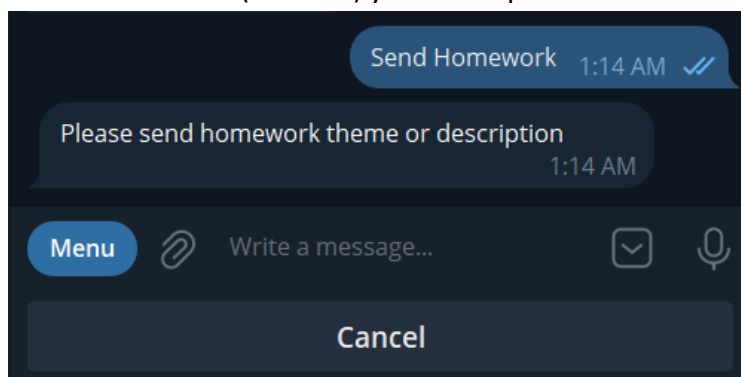
## User(Student) Menu

- Oddiy user bot ga **/start** buyrug`ini bersa, database ga yangi user yaratilishligi kerak ( firstName, lastName, username, chatId bilan) va user menu unga keyboard button lar bilan



jo`natilishi kerak:

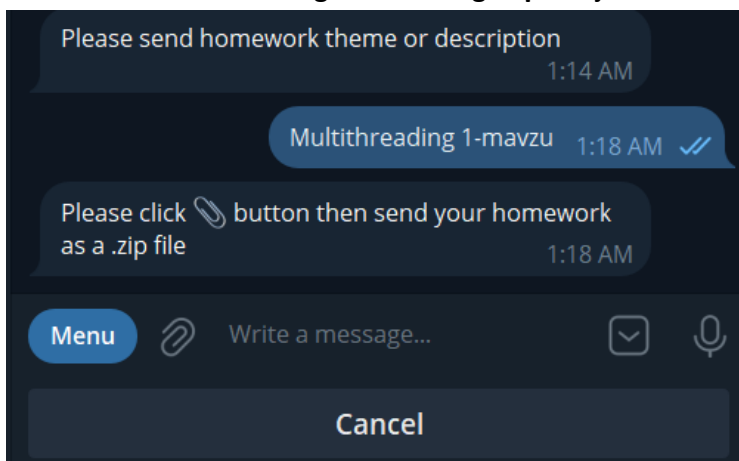
- Send Homework** button bosilsa yangi homework yaratish jarayoni boshlanishi kerak va homework theme(mavzusi) yoki descriptionni kiritilishi kerak:



, bu yerda **Cancel** button bosilsa

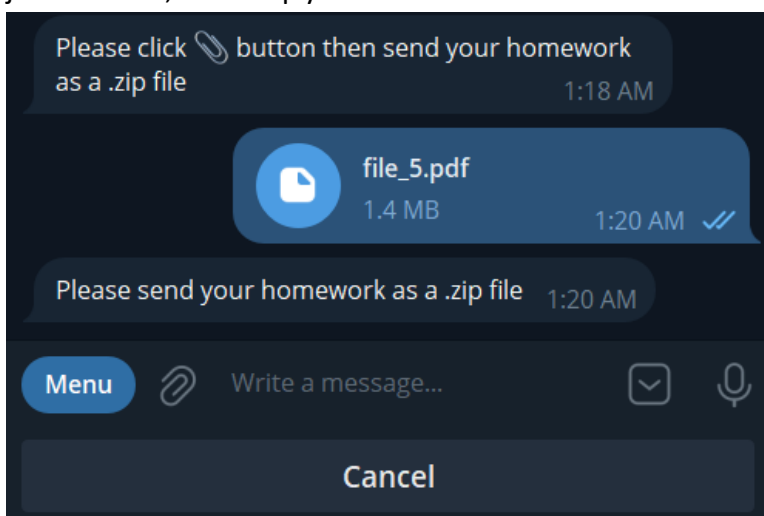
User Menu ga qaytish kerak;

3. Homework theme kiritilgandan so`ng **.zip** file jo`natilishi kerak:



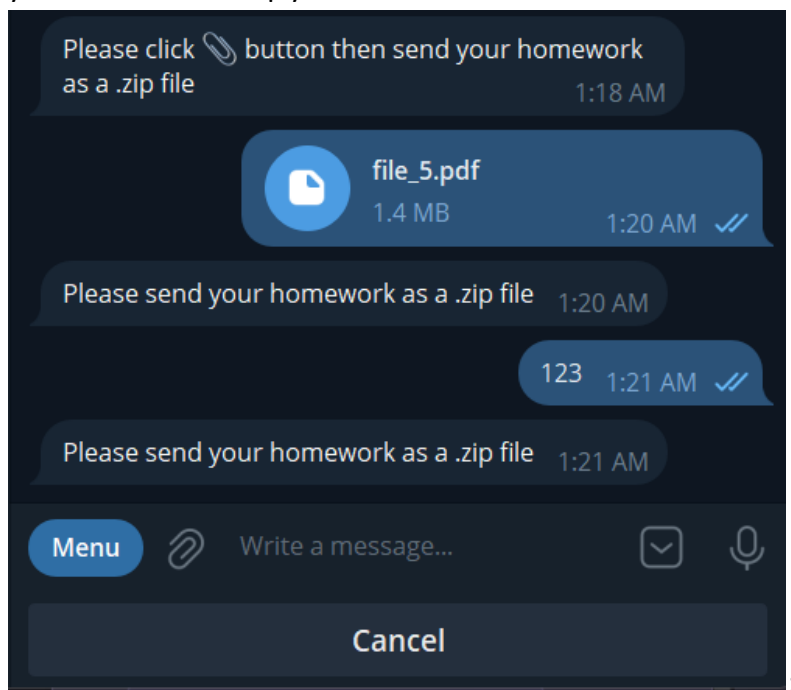
; bu yerda agar **.zip** type da file

jo`natilmasa, xatolik qaytarilishi kerak:

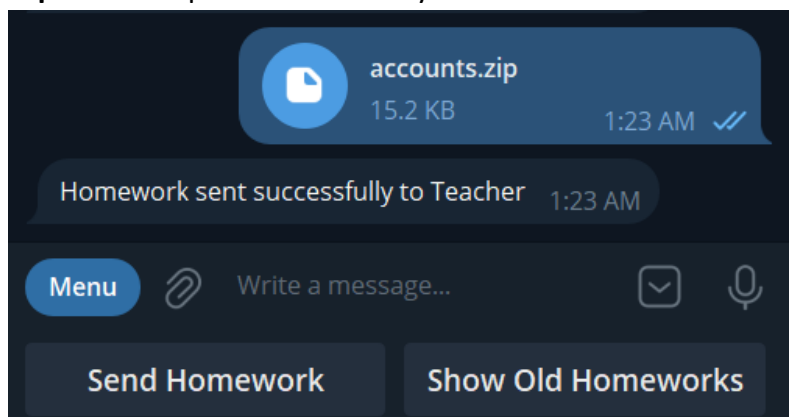


; yoki **.zip** file emas message

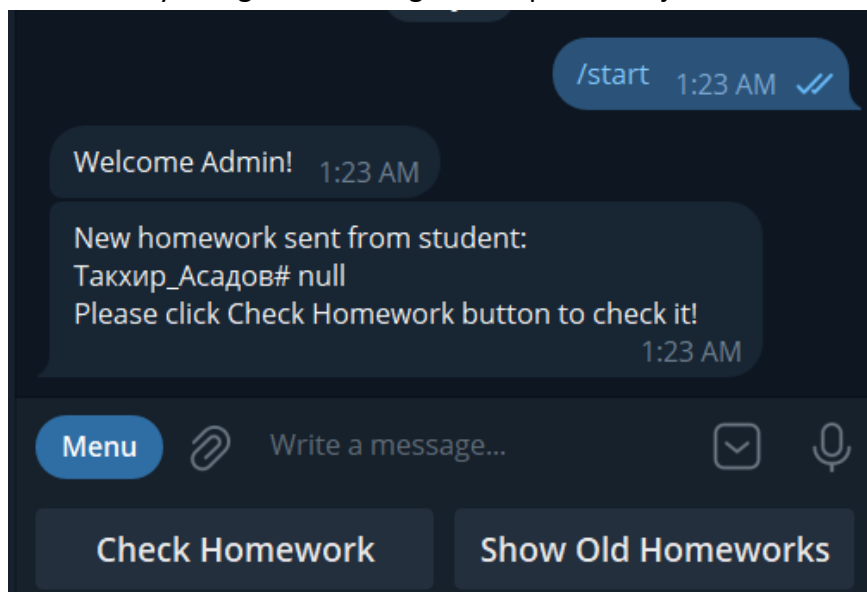
yozihsa ham xatolik qaytarilishi kerak:



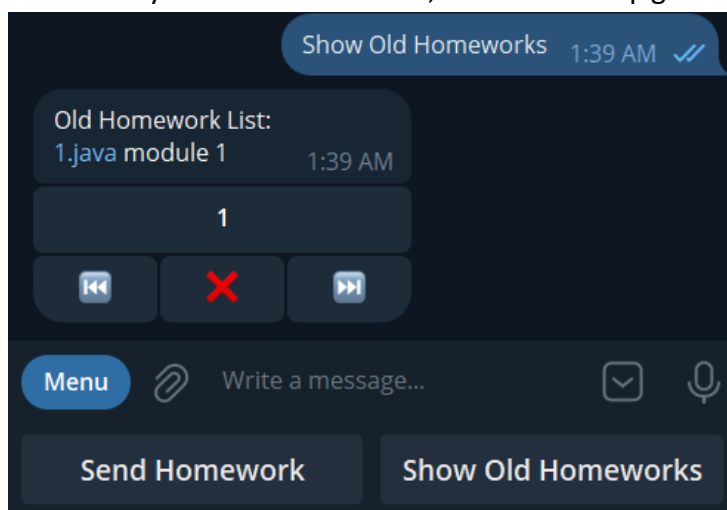
.zip file send qilinsa homework yaratilishi kerak:



Homework yaratilganda Adminga bu haqida xabar jo`natilishi kerak:



4. **Show Old Homeworks** keyboard button bosilsa, student send qilgan homework lar listi

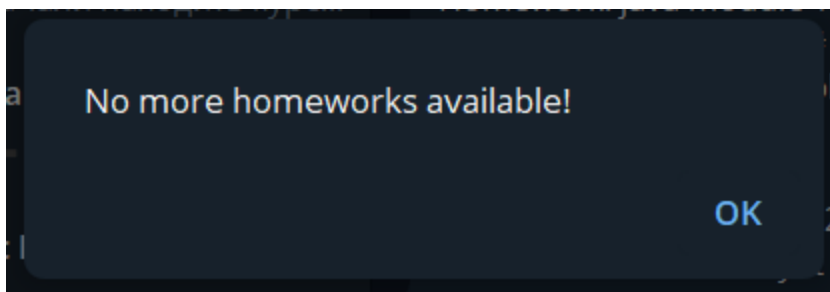


chiqishi kerak:

bu yerda previous, reject, next button lar bosilganda mos ravishda avvalgi page ga qaytishi , reject bo`lishi, keying page ga o`tishi kerak: har bir page da 10 tadan list chiqishi kerak. next




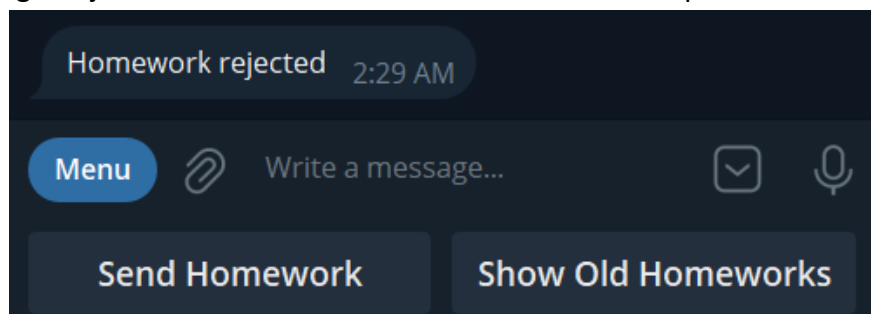
button bosilsa agar keyingi sahifa mavjud bo`lsa keyingi homework lar listi qaytarilishi kerak (bunda oldingi list edit qilinishi kerak, ya`ni oldingi list delete qilinib qayta jo`natilmasligi kerak); agar keyingi list mavjud bo`lmasa qoyidagi message chiqishi kerak:



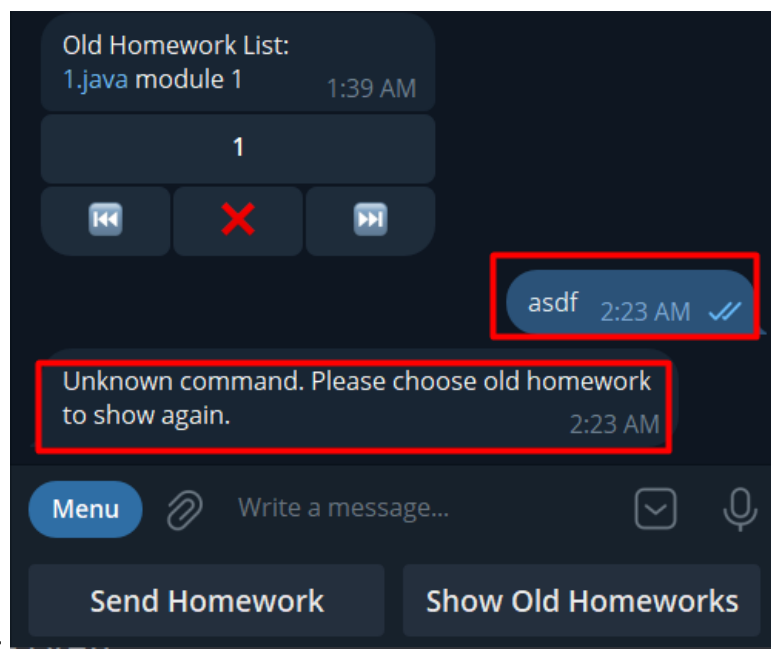
; previous  button

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



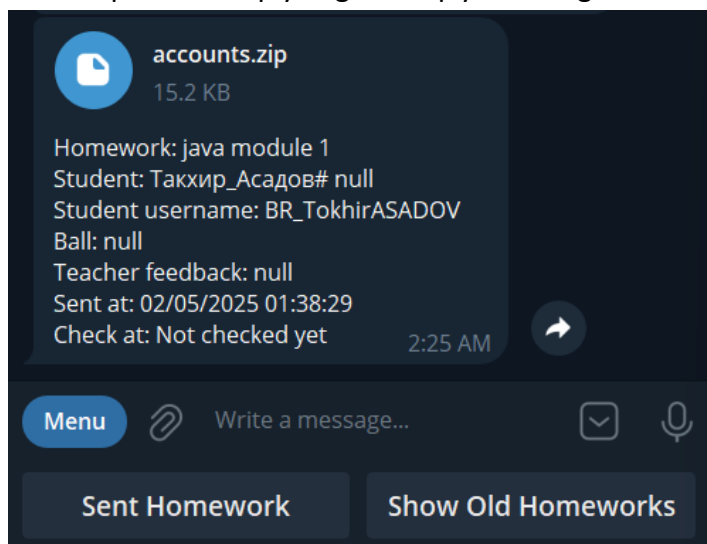
ga o'tishi kerak:



inline button lar bosilmasa:

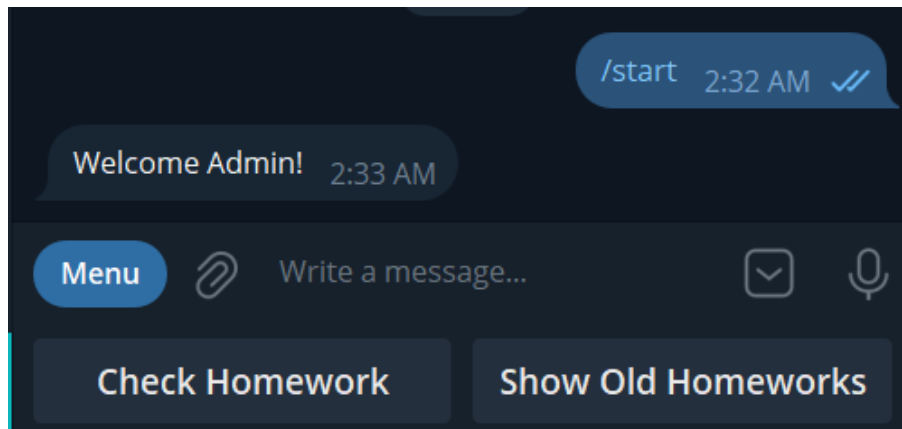
quyidagi javob qaytishi kerak; agar  button bosilib, old homework ni user

ko`rmoqchi bo`lsa quyidagi card qaytarilishligi kerak:

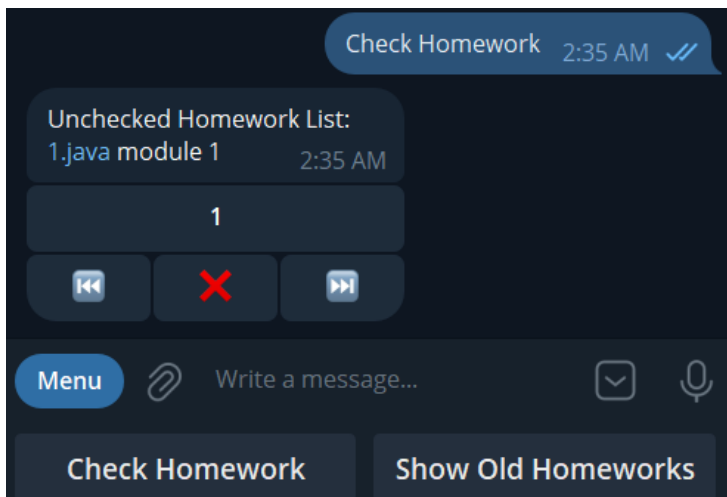


## Admin menu

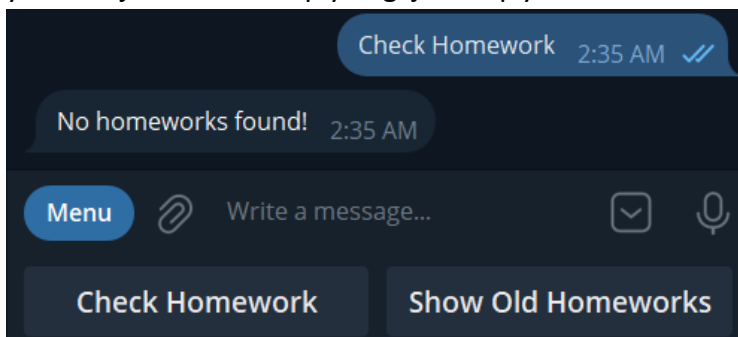
1. Admin **/start** buyrug`ini berganda rasmda ko`rsatilgan response va buttonlar chiqishi kerak



2. **Check Homework** button bosilsa , agar database (List dan foydalaning) da, checkTime mi null bo`lgan homework lar mavjud bo`lsa, o`shalarining listi:

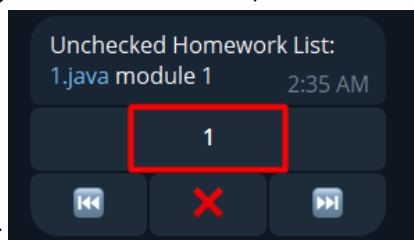


yoki mavjud bo`lmasa quyidagi javob qaytishi kerak:



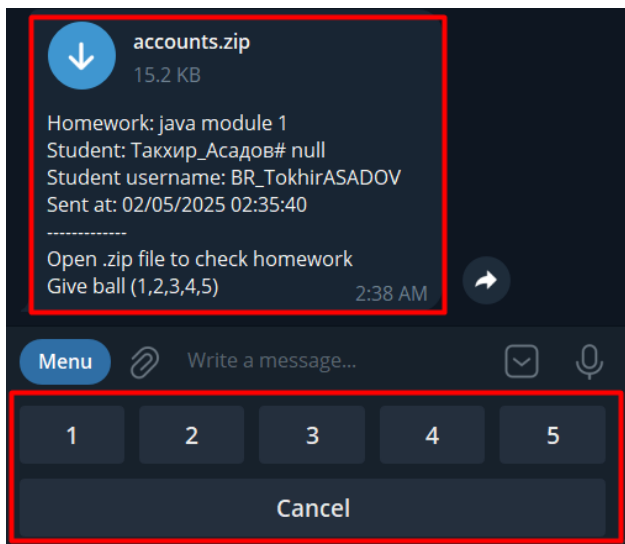
; previous, reject, next buttonlar

user dagi kabi ishlashi kerak;

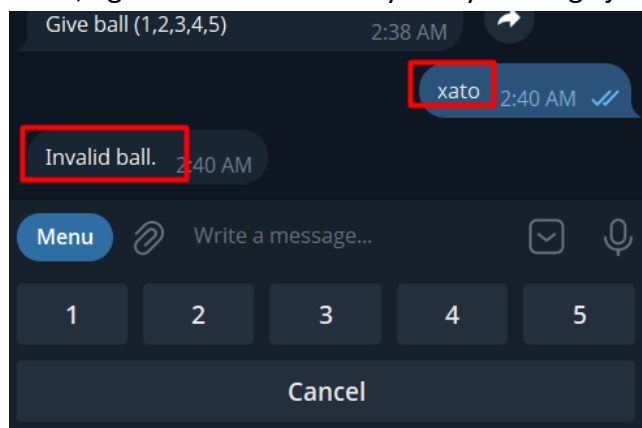


3. Agar button bosilsa, quyidagi card jo`natilishi kerak:



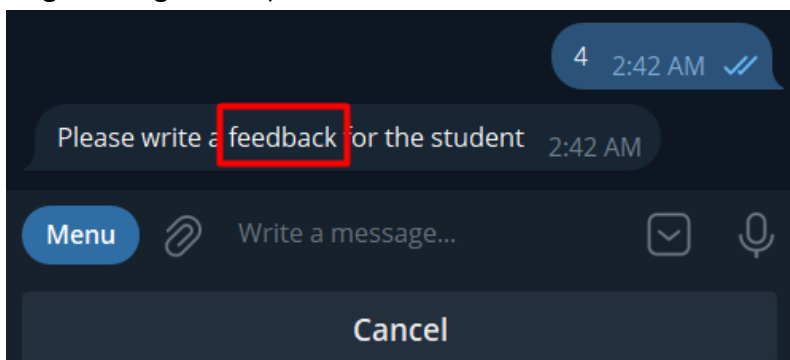


bu yerda 1-5 button lar bosilsa shu qiymat homework uchun ball bo`lishi kerak, **Cancel** button bosilga admin **User Menu** ga qaytishi va amalyot bekor qilinishi kerak. Aks holda check homework jarayoni davom etishi va keying **feedback** state ga o`tishi kerak; agar buttonlar bosilmay oddiy message jo`natilsa, quyidagi

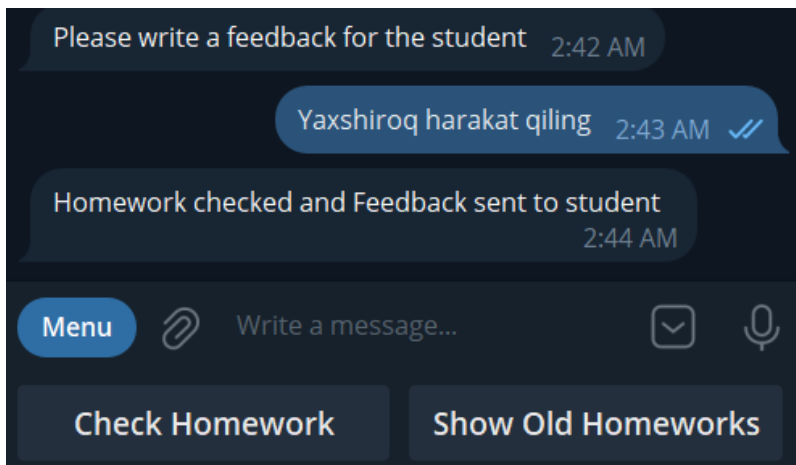


xatolik qaytarilishi kerak:

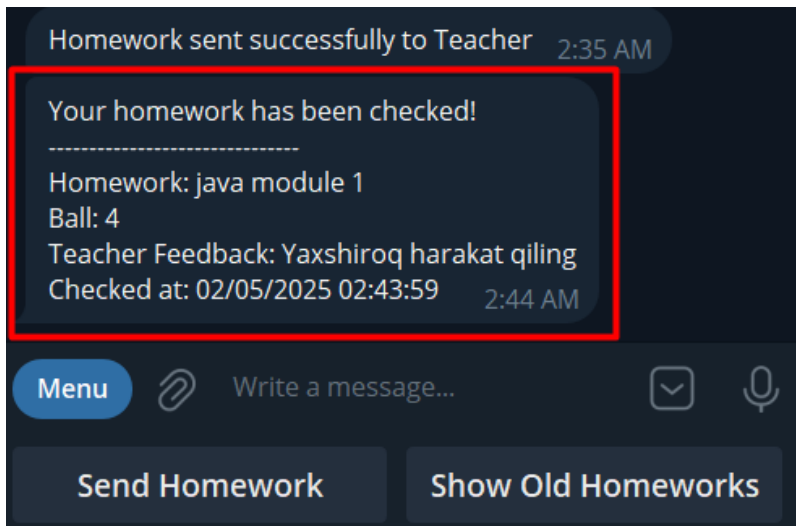
4. Homework ball kiritilgandan so`ng homework ning feedBack (teacher ning vazifa to`g`rishidagi fikrlari) so`ralishi kerak:



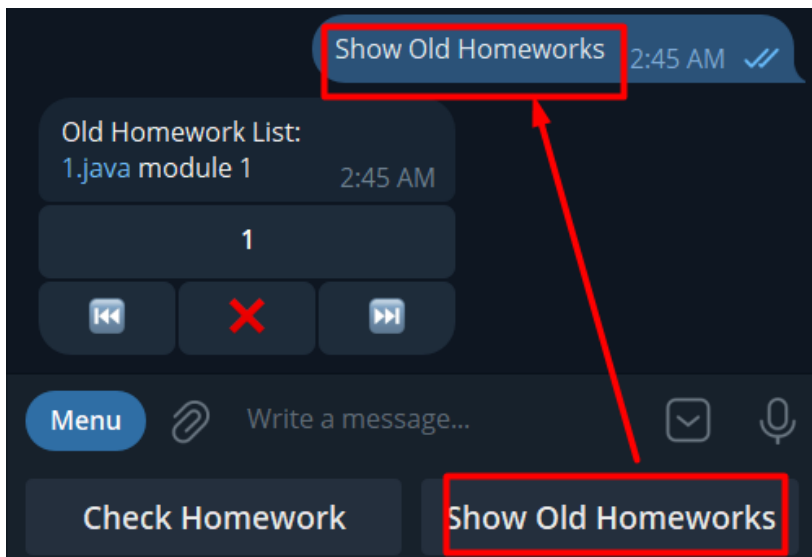
**Cancel** button bosilga admin **User Menu** ga qaytishi va amalyot bekor qilinishi kerak. Aks holda check homework jarayoni yakunlanishi: ,



, hamda **checkTime** ga checking time berilishi va bu haqida Student ga quyidagicha xabar jo`natilishi kerak:



5. **Show Old Homeworks** button **User Menu** da click qilinganda

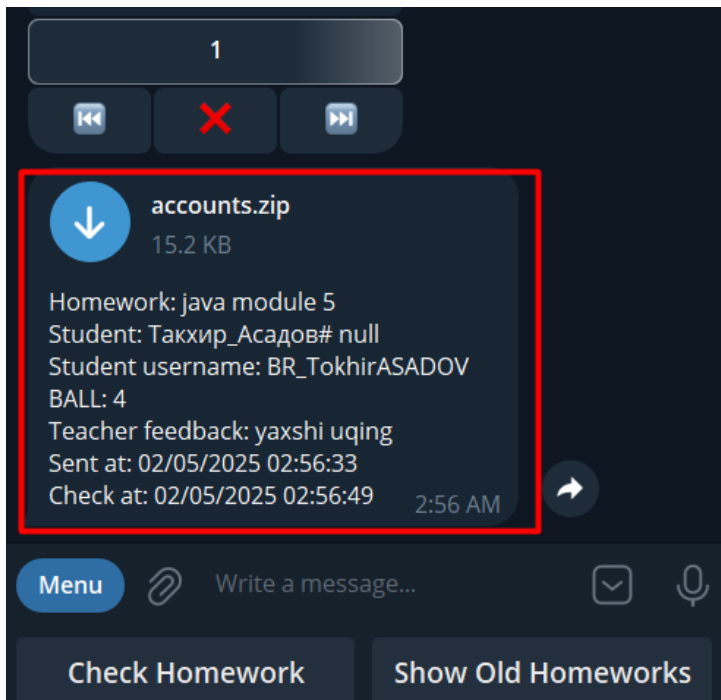


barcha talabalar jo`natgan

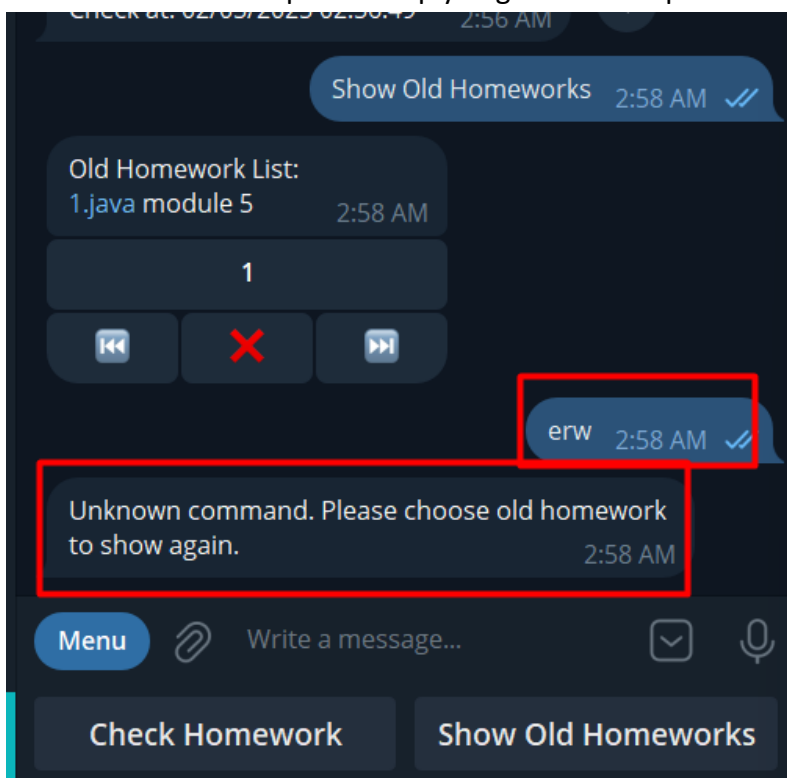
va tekshirilgan homework lar ro`yxati chiqishi kerak list ko`rinishida; bunda previous, reject,

1

next buttonlari yuqoridagidek ishlashi kerak;  
button tanlansa quyidagi card jo`natilishi kerak:



Inline buttonlar click qilinmasa quyidagi xatolik chiqishi 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.