



## مقدمه

در فاز اول شما به پیاده‌سازی دستورات در محیط command line پرداختید. در این فاز تعدادی دستور به دستورات قبلی اضافه می‌شود که برای اضافه و حذف کردن پیشنهادات خرید انجام می‌شود. همچنین تعدادی از دستورات دچار تغییر می‌شوند. در مجموع این تغییرات برای این است که سامانه‌ی نوشته شده را به واقعیت نزدیک‌تر بکنیم.



## شرح تمرین

در این فاز منطق سامانه خرید و فروش کالای ما با تعدادی دستور جدید تکمیل خواهد شد. این دستورات در ادامه به شما داده خواهند شد. همچنین برای هر دستور تعدادی خطا تعریف شده است که باید توسط منطق شما بررسی شوند.

تعدادی لیست دستور خواهیم داشت که برای هر کدام، آرگومان‌های لازم برای اجرا شدن تعریف می‌شوند. برای مثال برای عضویت در سامانه از یکی دستورهای POST به نام signup بهره خواهیم برد. همچنین به همراه آن آرگومان‌های لازم برای اجرا شدن را پس از ؟ خواهیم آورد.

به این نکته توجه کنید که در قالب دستوراتی که در ادامه می‌آید، مقدار هر آرگومان پس از وارد کردن نام آن وارد خواهد شد. اما ترتیب آرگومان‌ها ثابت نیست و می‌توانند با ترتیب‌های گوناگونی وارد شوند. برای مثال برای دستور signup هر دو حالت زیر را می‌توانیم داشته باشیم:

```
POST signup ? username <username> role <role> address <address>
POST signup ? role <role> username <username> address <address>
```

لیست دستوراتی که با آنها کار خواهیم کرد شامل موارد زیر می‌شود:

POST : برای ارسال اطلاعات و آرگومان به سامانه به منظور ثبت اطلاعات استفاده می‌شود.

GET : برای دریافت اطلاعات و گرفتن لیست کالاهای موجود یا خرید و فروش‌های انجام شده.

PUT : برای تغییر اطلاعات کالاهای ثبت شده استفاده می‌شود.

DELETE : برای حذف کردن برخی اطلاعات و مقادیر ثبت شده در سامانه استفاده می‌شود.

این نام‌گذاری دستورها در فازهای آتی پروژه که برنامه خود را روی وب عرضه خواهید کرد معنای خاص پیدا خواهند کرد. همینطور دقت داشته باشید که تمامی دستورها پس از اجرا شدن دارای خروجی مشخص هستند که منحصرأ ذکر می‌شود. همچنین با توجه به تمامی نکات ذکر شده اعم از عبارت قبل از دستورات و یا آرگومان‌های داده شده به دستورات، اگر در دستورات وارد شده کاربر، خطایی وجود داشته باشد باید با توجه به توضیحاتی که همراه با هر دستور آمده است، خطای آن را خروجی دهید.

خروجی پروژه شما به صورت خودکار آزموده می‌شود، لذا خروجی شما باید دقیقاً همانند خروجی خواسته شده باشد. در غیر این صورت نمره‌ی بخش آزمون را از دست خواهید داد. همچنین این مورد را توجه کنید که به غیر از خطاهای ذکر شده در هر دستور، خطاهای دیگر آزموده نمی‌شوند و خطاهای ترکیبی نیز نداریم.

## پاسخ دستورات

این بخش همانند فاز قبلی خواهد بود و باید خروجی مناسب نمایش داده شود.

## فهرست دستورات

توجه شود که تمامی دستورات فاز قبل، در این فاز هم باید اجرا شوند و در تست‌های نهایی آزموده خواهند شد.

## مجموعه دستورات خریدار

### ثبت نظر

با این دستور کاربر می‌تواند برای کالاهای ثبت شده، نظر خود را بگذارد. اگر شناسه‌ی کالای داده‌شده موجود نبود، خطای Not

Found نمایش داده خواهد شد. همچنین تضمین می‌شود که پیام وارد شده، یک رشته‌ی واحد خواهد بود و فاصله یا اسپیس

ندارد.

ورودی
POST comment ? id <id> message <message>
خروجی
OK   Bad Request   Not Found
ورودی نمونه
POST comment ? id 1 message this_is_bad_food
خروجی نمونه
OK

### ثبت امتیاز

با این دستور کاربر می‌تواند پس از خرید یک کالا، برای آن ثبت امتیاز بکند. دقت شود که اگر کاربر برای کالایی که تا حالا نخریده‌است نمی‌تواند ثبت امتیاز کند و اگر این کار را کرد باید خطای Permission Denied نمایش داده‌شود. همچنین امتیازی که کاربر وارد می‌کند، یک عدد صحیح بین صفر تا ده خواهد بود که اگر این‌گونه نبود ارور Bad Request نمایش داده می‌شود.

ورودی
POST <b>rate</b> ? id <id> rate <0-10>
خروجی
OK   Bad Request   Not Found   Permission Denied
ورودی نمونه
POST <b>rate</b> ? id 1 rate 4
خروجی نمونه
OK

## نمایش کالاها

دقت شود که در دستور نمایش کالاها (که فاز قبل توضیح داده شد) نیز باید میانگین امتیاز داده شده به کالا (با تقریب یک رقم اعشار)، همراه دیگر اطلاعات کالا نشان داده شود. همچنین حالا که هر کالا می تواند نظر نیز داشته باشد، برای نمایش اطلاعات کامل کالا اعم از نظرهای های مربوط به آن یک دستور جدید نیز در نظر می گیریم

ورودی
GET <b>list_items</b> ? id <id>
خروجی
Bad Request   Not Found   Empty
id : <id>
name : <name>
seller : <seller_username>
category : <category>
date : <date>
price : <price>
quantity : <quantity>
purchased_count : <count>
average_rate : <avg_rate>
comments :

```
"<comment1>" by <username1>
"<comment2>" by <username2>
...
```

ورودی نمونه

```
GET list_items ? id 1
```

خروجی نمونه

```
id : 1
name : shampoo
seller : mmd
category : sanitary
date : Fri Dec 23 10:40:56 2022
price : 11000
quantity : 4
purchased_count : 3
average_rate : 2.5
comments :
"good" by ali
"imperfect" by reza
```

## فیلترها

در این بخش کاربر با وارد کردن چند فیلتر می‌تواند خروجی خود برای دستور نمایش کالاها (list\_items) را شخصی‌سازی کند (در حالتی که بعد از دستور id نمی‌دهیم و چند کالا را نمایش می‌دهیم). نکته‌ای که باید توجه شود این است که هنگامی که فیلتری اعمال شود و دوباره دستور همان فیلتر وارد شود، فیلتر جدید جایگزین فیلتر قبلی از همان نوع خواهد شد. همچنین اگر خریدار از حساب خود خارج شود، تمامی فیلترها از بین می‌روند. در ادامه دستورات مربوط به فیلترها را می‌بینیم.

### ● فیلتر امتیاز

با این فیلتر، کالاهایی که میانگین امتیاز ثبت شده برای آنها بین بازه داده شده‌است، نمایش داده می‌شوند. دقت شود کالاهایی که امتیازی برای آنها ثبت نشده امتیاز صفر برای آنها در نظر گرفته می‌شود (بازه‌ی داده شده یک بازه بسته است. یعنی اگر 2-4 بازه ما باشد و کالایی دارای میانگین 2 یا 4 باشد، در خروجی نمایش داده خواهد شد).

ورودی
POST <code>rate_filter</code> ? min_rating <0-10> max_rating <0-10>
خروجی
OK   <code>Bad Request</code>
ورودی نمونه
POST <code>rate_filter</code> ? min_rating 2 max_rating 4
خروجی نمونه
OK

### ● فیلتر قیمت

کالاهایی که قیمت ثبت شده برای آنها در بازه‌ی داده شده باشد، نمایش داده خواهند شد.

ورودی
POST <code>price_filter</code> ? min_price <min_price> max_price <max_price>
خروجی
OK   <code>Bad Request</code>
ورودی نمونه
POST <code>price_filter</code> ? min_price 0 max_price 3000
خروجی نمونه
OK

### ● فیلتر دسته بندی

صرفاً کالاهای ثبت شده در دسته بندی داده شده نمایش داده خواهند شد. اگر دسته بندی وارد شده در لیست دسته بندی های موجود وجود نداشت، خطای `Not Found` نمایش داده خواهد شد.  
دسته بندی ها شامل این موارد خواهند شد : `sports` , `digital` , `clothing` , `food` , `cosmetic`

ورودی
POST <code>category_filter</code> ? category <category>

خروجی
OK   Bad Request   Not Found
ورودی نمونه
POST category_filter ? category sports
خروجی نمونه
OK

## ● حذف فیلتر

با این دستور تمامی فیلترهای ثبت شده، حذف خواهند شد.

ورودی
DELETE filters
خروجی
OK

## سبد خرید

در فاز قبل، در هر خرید فقط یک نوع کالا می توانست خریداری شود. اکنون می خواهیم با معرفی چند دستور، مفهوم سبد خرید را به سامانه اضافه کنیم. دقت شود که دیگر دستور خرید کالا (buy\_item) که در فاز قبل معرفی شد، معتبر نیست.

### اضافه کردن کالا به سبد خرید

با این دستور، کالای مورد نظر را به سبد خرید خود اضافه می کنیم. اگر مقدار وارد شده برای آرگومان count باعث شود که مجموعاً تعداد کل کالایی با id مشخص از موجودی کل آن کالا بیشتر شود، دستور نباید موفقیت آمیز باشد و با خطای Bad Request مواجه خواهیم شد. اگر کالا نیز در سیستم موجود نباشد با خطای Not Found روبه رو خواهیم شد.

ورودی
POST add_to_cart ? id <id> count <count>
خروجی
OK   Bad Request   Not Found
ورودی نمونه

POST **add\_to\_cart** ? id 1 count 2

خروجی نمونه

OK

## تکمیل خرید

در این دستور، پرداخت کالاهای موجود در سبد خرید انجام می‌شود. آرگومان city اختیاری است و اگر در دستور نیامده باشد، از همان city که در زمان ثبت نام از کاربر پرسیده شده است، استفاده می‌شود. اگر خریدار پول کافی در کیف پول خود برای پرداخت هزینه نداشته باشد، خطای Bad Request رخ می‌دهد. هزینه قابل پرداخت هم از جمع هزینه قابل پرداخت برای هر نوع کالای سبد خرید به دست می‌آید.

هزینه ارسال + تعداد خریداری شده  $\times$  (مجموع تخفیف اعمال شده روی کالا - هزینه کالا) = هزینه قابل پرداخت برای کالا

هزینه ارسال مانند فاز اول محاسبه می‌شود. در صورت وجود هیچ گونه خطا و پس از تغییر مناسب وضعیت سیستم، خرید تکمیل می‌شود و هزینه کل، مجموع تخفیف‌های اعمال شده و هزینه پرداخت شده چاپ می‌شود.

توضیح درباره تخفیف‌ها : سه نوع تخفیف در ادامه معرفی می‌شود. در صورتی که کالا شرایط تخفیف را دارا بود، تخفیف روی آن کالا اعمال می‌شود. برای مثال اگر روی یک کالا یک تخفیف ۲۰ درصدی از نوع package\_offer و یک تخفیف ۱۰ درصدی از نوع category\_offer و یک تخفیف ۵ درصدی از نوع total\_cost\_offer اعمال شود و قیمت واحد کالا ۱۰۰۰۰ باشد، هزینه قابل پرداخت واحد آن بدون هزینه ارسال برابر  $6500 = (1 - 0.2 - 0.1 - 0.05) * 10000$  خواهد بود. تضمین می‌شود مجموع تخفیف‌های اعمال شده روی کالا تاثیر کل قیمت کالا را از بین نمی‌برند و باعث منفی شدن قیمت قابل پرداخت نمی‌شوند. همچنین در صورت اعشاری شدن محاسبات، به عدد صحیح تقریب بزنید.

ورودی

POST **purchase\_cart** ? +city <city>

خروجی

Bad Request |  
total\_cost : <cost>  
total\_discount : <discount>  
paid : <cost>

ورودی نمونه

POST **purchase\_cart** ? city Karaj

خروجی نمونه



```
total_cost : 10000
total_discount : 1000
paid : 9000
```

## حذف سبد خرید

با این دستور، سبد خرید از بین رفته و تمام کالاهای موجود در آن از سبد خرید حذف می‌شوند. دقت شود که با خروج خریدار از حساب خود نیز، سبد خرید حذف می‌شوند.

ورودی
DELETE <b>cart</b>
خروجی
OK

از آنجا که در این فاز، هر خریدار می‌تواند در هر خرید، چند کالا خریداری کند، فرمت دستورات `list_purchased` و `list_transactions` نیز باید تغییر کنند.

## امتیازی: نمایش لیست خرید های قبلی

ورودی
GET <b>list_purchased</b>
خروجی
<b>Empty</b>   id : <id> date : <date> items : 1) seller : <seller_username> item_id : <item_id> item_name : <item_name> item_price : <price> purchased_count : <count>

```
delivery_cost : <price>
2)
seller : <seller_username>
item_id : <item_id>
...

total_cost : <cost>
paid : <cost>

id :
...
```

ورودی نمونه

GET **list\_purchased**

خروجی نمونه

```
id : 1
date : Fri Dec 23 10:40:56 2022
items :
1)
seller : mmd
item_id : 1
item_name : shampoo
item_price : 11000
purchased_count : 3
delivery_cost : 2000

total_cost : 35000
paid : 30000
```

مجموعه دستورات فروشنده

امتیازی: نمایش خرید های انجام شده از فروشنده

### ورودی

GET `list_transactions`

### خروجی

```
Empty |
id : <id>
date : <date>
buyer : <buyer_username>
items :
1)
item_id : <item_id>
item_name : <item_name>
item_price : <price>
purchased_count : <count>
delivery_cost : <price>
2)
item_id : <item_id>
item_name : <item_name>
...

total_cost : <cost>
paid : <cost>

id :
...
```

### ورودی نمونه

GET `list_purchased`

### خروجی نمونه

```
id : 1
date : Fri Dec 23 10:40:56 2022
seller : ahmad
items:
1)
```

```
item_id : 1
item_name : shampoo
item_price : 11000
purchased_count : 3
delivery_cost : 2000

total_cost : 35000
paid : 30000
```

## دستورات مربوط به پیشنهادات فروشنده

در این قسمت فروشنده می‌تواند تعدادی پیشنهاد یا آفر برای کالاهای خود تعریف کند. این پیشنهادات به شیوه های مختلفی روی کالاهای درون سبد اعمال خواهند شد که در توضیح هر پیشنهاد آمده‌اند. دقت شود که پیشنهادات فروشنده فقط روی کالاهایی در سبد خرید اثر دارند که از خود فروشنده خریده شده باشند. دقت شود که از تمام پیشنهادات می‌توانیم چند عدد داشته باشیم به جز پیشنهاد هزینه کلی. مثلاً می‌توانیم سه یا چهار پیشنهاد مختلف از وجود کالاهایی خاص در سبد (package\_offer) داشته باشیم، ولی در نهایت یک پیشنهاد از هزینه کلی (total\_cost\_offer) داریم.

### ● وجود کالاهایی خاص در سبد

این پیشنهاد در صورتی اعمال خواهد شد که تعدادی کالای خاص که شناسه آنها مشخص شده در سبد خرید موجود باشند. سپس روی تمام این کالاها (که شناسه آنها مشخص شده) تخفیف اعمال می‌شود. اگر شناسه‌ی وارد شده وجود نداشته باشد، خطای Not Found نمایش داده می‌شود.

ورودی
POST <b>package_offer</b> ? id <1,2,3,...> discount <1-100 percentage>
خروجی
OK   <b>Bad Request</b>   <b>Not Found</b>
ورودی نمونه
POST <b>package_offer</b> ? id 2,3 discount 20
خروجی نمونه
OK

- وجود کالاهایی از دسته بندی خاص در سبد

با این پیشنهاد اگر کالاهای خاصی در سبد وجود داشته باشند، روی این کالاها (از آن دسته بندی خاص) تخفیف تعلق می گیرد. در صورتی که دسته بندی وارد شده، در کالاهای ثبت شده وجود نداشته باشد خطای Not Found نمایش داده می شود.

ورودی
POST <b>category_offer</b> ? category <category> discount <1-100 percentage>
خروجی
OK   Bad Request   Not Found
ورودی نمونه
POST <b>category_offer</b> ? category sports discount 10
خروجی نمونه
OK

- بیشتر بودن هزینه کلی خرید از رقمی خاص

در صورتی که میزان خرید کالا در یک سبد از یک فروشنده از رقم خاصی بالاتر باشد(بدون در نظر گرفتن تخفیف ها)، این تخفیف روی تمام آن کالاها (که از فروشنده مدنظر گرفته شده) اعمال خواهند شد.

ورودی
POST <b>total_cost_offer</b> ? cost <cost> discount <1-100 percentage>
خروجی
OK   Bad Request   Not Found
ورودی نمونه
POST <b>total_cost_offer</b> ? cost 1000 discount 15
خروجی نمونه
OK

### ● امتیازی: فهرست کردن پیشنهادات (مخصوص خریدار)

با این دستور خریدار می‌تواند لیست پیشنهادات یک فروشنده‌ی خاص را ببیند. در صورتی که نام فروشنده‌ی وارد شده موجود نبود خطای Not Found نمایش داده می‌شود.

ورودی
GET <b>list_offers</b> ? username <seller_username>
خروجی
Empty   Not Found   <offer_id> <offer_type> ( <ids> or <category> or <cost> ) <discount> ...
ورودی نمونه
GET <b>list_offers</b> ? username ahmad
خروجی نمونه
3 total_cost_offer 330 15

### ● امتیازی: حذف کردن پیشنهاد

با این دستور فروشنده می‌تواند پیشنهادات خود را پاک کند. اگر شناسه خاصی را مشخص کند، آن پیشنهاد پاک می‌شود و اگر مشخص نکند، تمام پیشنهادات او پاک می‌شود. اگر دستور حذف پیشنهاد فروشنده دیگری را وارد کرد، خطای Permission Denied نمایش داده خواهد شد.

ورودی
DELETE <b>offers</b> ? +id <id>
خروجی
Not Found   Permission Denied   Bad Request   OK
ورودی نمونه
DELETE <b>offers</b> ? id 1
خروجی نمونه
OK

## نحوه تحویل

تمام فایل‌های خود را در قالب یک پرونده‌ی زیپ با نام A4-SID.zip در صفحه‌ی Elearn درس بارگذاری کنید که SID شماره دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۰۰۹۹۹ است، نام پرونده شما باید A4-810100999.zip باشد.

○ برای مثال، نمونه فایل مورد قبول در زیر آمده است:

A4-810100999.zip

```
main.cpp — |
makefile — |
... — L
```

- با توجه به حجم نسبتاً زیاد این فاز از تمرین توصیه می‌شود قبل از پیاده‌سازی کد طراحی اولیه‌ای برای منطق برنامه و روندهای آن مثل ثبت نام، درخواست سفر و ... انجام دهید و پس از این طراحی شروع به پیاده‌سازی آن کنید. از آن جایی که در فازهای بعدی شما باید رابط کاربری برنامه‌ی خود را از command line به روش‌هایی دیگر تغییر دهید، بهتر است تا طراحی برنامه‌ی شما طوری باشد که کمترین وابستگی میان منطق برنامه و رابط کاربری آن وجود داشته باشد.
- دقت کنید که پرونده زیپ آپلودی شما باید پس از Unzip شدن شامل پرونده‌های پروژه شما از جمله Makefile باشد و از زیپ کردن پوشه‌ای که داخل آن فایل‌های پروژه‌تان قرار دارد خودداری فرمایید.
- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودیهای آزمون اجرا شود.
- دقت کنید که پروژه شما باید multi-file باشد و Makefile داشته باشد. همینطور در Makefile خود مشخص کنید که از استاندارد c++11 استفاده می‌کنید.
- درستی برنامه‌ی شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین پیشنهاد می‌شود با استفاده از ابزارهایی مانند diff خروجی برنامه خود را با خروجی‌هایی که در اختیارتان قرار داده شده است مطابقت دهید.
- دقت کنید که نام پرونده‌ی اجرایی شما باید utkala.out باشد.
- طراحی درست، رعایت سبک برنامه نویسی درست و تمیز بودن کد برنامه‌ی شما در نمرهی تمرین تأثیر زیادی دارد.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آنها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید.
- توجه داشته باشید که حالتهای خاصی که در صورت پروژه ذکر نشده است در تست‌های خودکار نخواهد بود و می‌توانید به هر شکلی که مد نظر دارید آنها را مدیریت کنید.