



UTkala

مقدمه

این پروژه به جمع‌بندی آموخته‌های شما در درس می‌پردازد. انتظار می‌رود مهارت‌هایی را که در تمرین‌های پیشین و سایر بخش‌های درس آموخته‌اید، در پیاده‌سازی این پروژه به کار گیرید. در نهایت شما باید سایت خرید و فروش کالا را طراحی کنید که دارای امکانات مختلفی می‌باشد.

در فاز اول باید منطق برنامه‌ی خود برای یک سامانه خرید فروش کالا به نام UTkala را پیاده‌سازی کنید، به طوری که با استفاده از رابط کاربری خط فرمان¹ می‌توان از آن استفاده کرد. ابتدا کاربران مختلف می‌توانند در سیستم به عنوان خریدار یا فروشنده عضو شوند. سپس دستورات مخصوص خود را وارد می‌کنند.

علاوه بر درستی برنامه، تمیزی کد و جدا کردن برنامه به توابع مختلف اهمیت زیادی دارند. سعی کنید که قبل از انجام تمرین حتماً ویدیوهای بخش تمیزی کد موجود در صفحه‌ی درس را مشاهده کنید.

نکته‌ای که باید در این پروژه حتماً به آن توجه کنید آن است که این پروژه باید به صورت قسمت به قسمت تست و پیاده‌سازی بشود. بدین شکل که یک قالب کلی برای پروژه ایجاد شود، سپس دستورات به آن اضافه شوند.

¹ Command Line Interface

شرح تمرین

در این فاز منطق سامانه خرید و فروش کالا توسط چند دستور پیاده‌سازی می‌شود. این دستورات در ادامه به شما داده خواهند شد. همچنین برای هر دستور تعدادی خطا تعریف شده‌است 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 آمده‌است، 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

به ازای هر دستوری که وارد می‌شود، پاسخی از جانب سیستم دریافت می‌کنیم که می‌تواند نشان دهنده وضعیت انجام دستور یا شامل یک سری اطلاعات مانند لیست تراکنش‌های انجام شده باشد. در ادامه پاسخ‌های کلی در نظر گرفته شده، برشمرده می‌شوند (در بخش دستور مربوطه، برای هر دستور پاسخ‌های احتمالی نیز گفته می‌شود).

● پاسخ موفقیت‌آمیز بودن درخواست

اگر دستوری که کاربر وارد می‌کند به درستی انجام شود و به اتمام برسد، این پاسخ نمایش داده می‌شود.

خروجی
OK

● پاسخ درخواست اشتباه

اگر اولین قسمت ورودی کاربر، هیچ کدام از لیست دستورهای GET, POST, DELETE, PUT نباشد، این پاسخ نمایش داده می‌شود. همچنین اگر قالب دستور وارد شده، با هیچ کدام از دستوراتی که در ادامه می‌آیند مطابقت نداشته باشد (آرگومان‌های دستور به درستی داده نشده باشند)، این پاسخ نمایش داده می‌شود.

خروجی
Bad Request

- پاسخ دسترسی غیر مجاز (عدم دسترسی)

اگر یک کاربر دسترسی از لیست دستورات دیگر کاربران را وارد کرد یا درخواست تغییر اطلاعات کالایی را داشت که برای فروشنده دیگری بود، این پیغام نمایش داده می‌شود.

خروجی
Permission Denied

- پاسخ عدم وجود

در صورتی که دستور وارد در لیست دستورات GET, POST, DELETE, PUT وجود نداشت، این پیغام نمایش داده می‌شود. همچنین در صورتی که شناسه کالا یا نام کاربری وارد شده وجود نداشته باشد (در هر قسمت که جستجویی انجام می‌شود اما نتیجه ای در بر ندارد).

خروجی
Not Found

- پاسخ خالی بودن

در صورتی که لیست درخواست شده از سامانه هیچ مورد قابل نمایشی نداشته باشد، این پاسخ به کاربر نمایش داده خواهد شد.

خروجی
Empty

دستورات

ثبت نام

این دستور مخصوص کاربرانی است که به سامانه وارد نشده‌اند. با انجام این دستور کاربر وارد سامانه می‌شود. نام کاربری هر فردی که ثبت نام انجام می‌دهد باید یکتا باشد. نقش فرد نیز فقط buyer یا seller است. اگر هر یک از مواردی که گفته شد رعایت نشود، کاربر خطای Bad Request خواهد گرفت. کاربر بعد از ثبت نام، مستقیم وارد سامانه خواهد شد.

ورودی
POST signup ? username <username> password <password> role <role> city <city>
خروجی
OK Bad Request
ورودی نمونه
POST signup ? username ali password 123123 role seller address Karaj
خروجی نمونه
OK

ورود

با انجام این دستور، کاربری که قبلاً ثبت نام انجام داده است، وارد سامانه می شود و می تواند دستورات خود را وارد کند. اگر رمز خود را اشتباه وارد کند، Permission Denied دریافت می کند. اگر نام کاربری که وارد می کند در سامانه وجود نداشته باشد، Not Found دریافت خواهد کرد.

ورودی
POST login ? username <username> password <password>
خروجی
OK Bad Request Permission Denied Not Found
ورودی نمونه
POST login ? username ali password 123123
خروجی نمونه
OK

خروج

این دستور مخصوص کاربران عضو شده است. با وارد کردن از سیستم خارج می شوند و باید دوباره وارد شوند.

ورودی
POST <code>logout</code>
خروجی
OK

مجموعه دستورات خریدار

افزایش کیف پول

این دستور موجودی کیف پول خریدار را افزایش می دهد. مقدار وارد شده به عنوان آرگومان باید عدد صحیح و بزرگتر از صفر باشد. در غیر این صورت خروجی، خطای Bad Request خواهد بود.

ورودی
POST <code>increase_credit ? amount <amount></code>
خروجی
OK <code>Bad Request</code>
ورودی نمونه
POST <code>increase_credit ? amount 10</code>
خروجی نمونه
OK

گرفتن موجودی کیف پول

این دستور موجودی کیف پول خریدار را نشان می دهد.

ورودی
GET <code>wallet_balance</code>
خروجی
<code>wallet balance : amount</code>

نمایش کالاها

این دستور لیست تمام کالا های موجود در سامانه را چاپ می کند. به ازای هر کالا: شناسه، نام، نام کاربری فروشنده، دسته بندی، قیمت، تاریخ ثبت در سامانه، تعداد ثبت شده در سامانه، و تعداد خرید هایی که از آن کالا انجام شده نمایش داده می شوند. اگر هیچ کالایی در سامانه وجود نداشت، Empty به کاربر نمایش داده می شود. همچنین این دستور می تواند دو آرگومان اختیاری دیگر نیز داشته باشد.

دز صورتی که آرگومان seller_username فراهم شده باشد، فقط لیست کالاهایی که توسط فروشنده ای با آن username در سامانه ثبت شده باشد، نمایش داده می شود. اگر username در سامانه موجود نباشد، Not Found نمایش داده می شود. در صورتی که آرگومان sort_by فراهم شده باشد، فقط دو مقدار مشخص highest_price و lowest_price را می تواند بگیرد که لیست را بر اساس قیمت (صعودی/نزولی) به صورت مرتب شده چاپ می کند. مقداری غیر از این دو، خطای Bad Request را در پی خواهد داشت.

ورودی

```
GET list_items ? +sort_by <highest_price/lowest_price> +seller_username
<seller_username>
```

خروجی

Bad Request | Not Found | Empty |

```
id : <id>
name : <name>
seller : <seller_username>
category : <category>
date : <date>
price : <price>
quantity : <quantity>
purchased_count : <count>
```

```
id : <id>
...
```

ورودی نمونه

```
GET list_items ? seller_username mmd
```

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
```

خرید کالا

با این دستور می‌توان کالایی که شناسه آن برای خریدار مشخص است را خریداری کرد. 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 شناسه کالا در سامانه موجود نباشد، خطای Not Found رخ می‌دهد. اگر مقدار آرگومان count از quantity کالا در آن زمان بیشتر باشد، خطای Bad Request رخ می‌دهد. همچنین اگر خریدار پول کافی در کیف پول خود برای پرداخت هزینه کل نداشته باشد، خطای Bad Request رخ می‌دهد. برای محاسبه هزینه کل خرید هم بدین شکل محاسبات را انجام می‌دهیم:

$$\text{هزینه ارسال} + \text{تعداد خریداری شده} \times \text{هزینه کالا} = \text{هزینه کل}$$

دو نوع هزینه ارسال خواهیم داشت. اول آنکه ارسال داخل شهری باشد. یعنی شهر وارد شده هنگام خرید برابر با شهر فروشنده باشد. نوع دوم آنکه خارج شهری باشد. یعنی اینکه شهر وارد شده برابر شهر فروشنده نباشد. هزینه ارسال داخل شهری را ۱۰۰۰ و هزینه ارسال خارج شهری را ۲۰۰۰ در نظر می‌گیریم.

در صورت وجود هیچ گونه خطا و پس از تغییر مناسب وضعیت سیستم، خرید تکمیل می‌شود و هزینه خرید چاپ می‌شود.

ورودی

```
POST buy_item ? id <id> count <count> +city <city>
```

خروجی

```
total_cost : <cost> | Not Found | Bad Request
```

ورودی نمونه

```
POST buy_item ? id 1 count 5 city Karaj
```

خروجی نمونه

```
total_cost : 12000
```


نمایش لیست خرید های قبلی

با این دستور خریدار می تواند به لیست تمام خریدهایی که تا آن زمان انجام داده است، دست یابد. به ازای هر خرید قبلی: شناسه، تاریخ ثبت، فروشنده کالا، شناسه کالا، نام کالا، قیمت کالا، تعداد خریداری شده، هزینه ارسال، و هزینه کل خرید باید نمایش داده شوند. اگر خریدی وجود نداشته باشد، Empty چاپ می شود.

ورودی

GET `list_purchased`

خروجی

```
Empty |  
id : <id>  
date : <date>  
seller : <seller_username>  
item_id : <item_id>  
item_name : <item_name>  
item_price : <price>  
purchased_count : <count>  
delivery_cost : <price>  
total_cost : <cost>  
  
id :  
...
```

ورودی نمونه

GET `list_purchased`

خروجی نمونه

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

delivery_cost : 2000

total_cost : 35000

امتیازی: مرجوع کردن کالای خریداری شده

با این دستور کاربر می‌تواند کالای خریداری شده را بازگرداند و بخشی از هزینه کل را با توجه به درصدی که فروشنده برای آن کالا در نظر گرفته از فروشنده پس بگیرد. همچنین توجه کنید که هزینه ارسال بازگردانی نمی‌شود و برای برگرداندن کالا هزینه ای برای برگرداندن در نظر نمی‌گیریم. در صورتی که کالای خریداری شده در خرید، قابلیت بازگردانی نداشته باشد Bad Request نمایش داده می‌شود. اگر کالای شناسه خرید وارد شده در سیستم ثبت نشده باشد Not Found نمایش داده می‌شود. اگر شناسه خرید کاربر دیگری وارد شد Permission Denied نمایش داده می‌شود.

ورودی

POST refund ? purchased_id <purchased_id>

خروجی

OK | Bad Request | Permission Denied | Not Found

مجموعه دستورات فروشنده

اضافه کردن یک کالا به سامانه

با این دستور فروشنده می‌تواند، یک کالا را به لیست کالا های فروشگاه خود اضافه کند. در صورتی که کالا با نام تکراری در لیست کالاهای فروشنده باشد Bad Request نمایش داده می‌شود. اگر مقدار وارد شده در قسمت تعداد و قیمت عدد صحیح و غیر منفی نبود، Bad Request نمایش داده می‌شود. همچنین فروشنده می‌تواند قابلیت بازگردانی را به کالا اضافه کند و درصدی برای آن در نظر بگیرد. اگر این آرگومان را وارد نکند یعنی کالا قابل بازگردانی نیست. اگر دستور با موفقیت انجام شد، شناسه کالای ثبت شده نمایش داده می‌شود.

ورودی

POST add_item ? name <name> price <price> category <category> quantity <quantity> +refund <1 to 100 percentage>

خروجی

id : <id> Bad Request
ورودی نمونه
POST add_item ? name shokolat price 100 category chocolates quantity 10 refund <50>
خروجی نمونه
id : 3

حذف کردن کالا از سامانه

با این دستور، فروشنده می تواند کالایی که قبلا برای فروش ثبت کرده بود را از سامانه حذف کند. اگر شناسه کالا در سامانه وجود نداشته باشد خطای Not Found رخ می دهد. اگر شناسه کالایی وارد شود که مربوط به فروشنده دیگری باشد، خطای Permission Denied رخ می دهد.

ورودی
DELETE item ? id <id>
خروجی
OK Not Found Permission Denied

تغییر قیمت کالا

با این دستور، فروشنده می تواند قیمت کالای از قبل ثبت شده را تغییر دهد. خطاهای Not Found و Permission Denied با شرایط دستور قبل رخ می دهند. همچنین اگر قیمت جدید عدد صحیح نامنفی نباشد، خطای Bad Request رخ می دهد.

ورودی
PUT item_price ? price <price>
خروجی
OK Bad Request Not Found Permission Denied

تغییر تعداد یک کالا

با این دستور، فروشنده می تواند تعداد ثبت شده از یک نوع کالا را تغییر دهد خطای Not Found در صورتی که شناسه کالا در سامانه یافت نشود و خطای Permission Denied در صورتی که کالا مربوط به فروشنده دیگری رخ می دهد. همچنین اگر مقدار آرگومان quantity عدد صحیح نامنفی نباشد، خطای Bad Request رخ می دهد.

ورودی
PUT item_quantity ? quantity <quantity>
خروجی
OK Bad Request Not Found Permission Denied

نمایش کالاهای ثبت شده فروشنده

با این دستور فروشنده می تواند کالاهایی که ثبت کرده است را ببیند. اگر کالایی موجود نبود Empty نمایش داده می شود. در رابطه با sort کردن هم مانند قبل خواهد بود. اگر کالا قابل بازگردانی نبود، مقدار refund_percentage را 0 نمایش می دهیم. همچنین در purchased_count تعداد فروخته شده از کالا را نمایش می دهیم.

ورودی
GET list_items ? +sort_by <highest_price/lowest_price>
خروجی
Empty id : <id> name : <name> category : <category> date : <date> price : <price> quantity : <quantity> purchased_count : <count> refund_percentage : <0-100> id : <id>

...

ورودی نمونه

GET `list_items` ?

خروجی نمونه

```
id : 3
name : shokolat
category : chocolates
date : Fri Dec 23 10:40:56 2022
price : 100
quantity : 10
purchased_count : 0
refund_percentage : 50
```

نمایش خرید های انجام شده از فروشنده

با این دستور فروشنده می تواند کالاهایی که از او خریداری شده اند را ببیند. اگر خریدی انجام نشده بود، Empty نمایش داده می شود. در رابطه با sort کردن هم مانند قبل خواهد بود.

ورودی

GET `list_transactions`

خروجی

```
Empty |
id : <id>
date : <date>
buyer : <buyer_username>
item_id : <item_id>
item_name : <item_name>
item_price : <price>
purchased_count : <count>
delivery_cost : <price>
```

```
total_cost : <cost>
```

```
id :
```

```
...
```

ورودی نمونه

```
GET list_purchased
```

خروجی نمونه

```
id : 1
```

```
date : Fri Dec 23 10:40:56 2022
```

```
seller : ahmad
```

```
item_id : 1
```

```
item_name : shampoo
```

```
item_price : 11000
```

```
purchased_count : 3
```

```
delivery_cost : 2000
```

```
total_cost : 35000
```

امتیازی: نمایش درآمد کسب شده

با این دستور، فروشنده می تواند کل درآمد حاصل از فروش کالاهایی که در سامانه ثبت کرده است را ببیند. دقت کنید که هزینه ارسال هم جزو این درآمد ها حساب و به آن اضافه می شود.

ورودی

```
GET revenue
```

خروجی

```
total revenue : amount
```

ورودی نمونه

```
GET revenue
```

خروجی نمونه

```
total revenue : 20000
```

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.