



UTAXI!

مقدمه

این پروژه به جمع بندی 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 یک سایت درخواست تاکسی طراحی کنید. ابتدا لیست مناطق شهر از فایل خوانده می‌شود که می‌توانند مبدا و مقصد سفرها باشند. سپس مسافران می‌توانند مبدا و مقصد خود را انتخاب کنند و رانندگان نیز می‌توانند مسافر را قبول کنند. در فاز اول این پروژه، شما منطق برنامه‌ی این سایت را پیاده‌سازی می‌کنید که با استفاده از رابط کاربری خط فرمان¹ می‌توان از آن استفاده کرد. نکته قابل توجه در این پروژه این است که بهتر است پروژه به صورت قسمت به قسمت پیاده‌سازی و تست شود. به طوری که ابتدا یک ساختار کلی از پروژه پیاده‌سازی شود و سپس دستورات مختلف به آن اضافه گردد.

¹ CLI (Command Line Interface)

شرح تمرین

در این فاز، منطق برنامه در قالب تعدادی دستور در ادامه توضیح داده شده است. روند استفاده از برنامه به این شکل است که کاربر در برنامه شما با استفاده از رابط خط فرمان، دستوری همراه با آرگومان‌های لازم برای اجرای آن در ورودی استاندارد وارد می‌کند. به عنوان مثال، برای گرفتن یک لیست از اطلاعات موجودیت‌ها از دستور GET و در صورت گرفتن یک عنصر خاص از موجودیت مورد نظر بعد از دستور، شناسه² موجودیتی که مایل به گرفتن اطلاعات آن است را نیز وارد می‌کند تا دستور مورد نظرش اجرا شود.

همینطور در نظر داشته باشید که این اطلاعات یعنی آرگومان‌های هر دستور پس از علامت ؟ در دستور می‌آیند و نیز ترتیب خاصی برای آن‌ها وجود ندارد؛ به این معنا که لزومی ندارد آرگومان‌ها به همان ترتیبی که در توضیحات هر دستور گفته شده، وارد شوند. برای مثال، نیازی نیست که در دستور درخواست سفر آرگومان مبدا حتما قبل از مقصد وارد شود.

نکته دیگری که برای دستورات وجود دارد این است که قبل از وارد شدن دستور، عبارت GET یا POST یا DELETE وارد می‌شود که به این شکل دستورات به نحوی از هم جدا می‌شوند به این ترتیب که دستوراتی که بیشتر برای گرفتن اطلاعات از سیستم استفاده می‌شوند، از GET و دستوراتی که بیشتر برای فرستادن اطلاعات به سیستم استفاده می‌شوند، از POST استفاده می‌کنند، همچنین دستوراتی که قصد حذف یک منبع را دارند از DELETE استفاده می‌کنند. 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 منحصرا ذکر می‌شود.

دقت داشته باشید 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. به عنوان مثالی از خطاهایی که ممکن است رخ دهد، می‌توانید وارد کردن دستور مخصوص راننده توسط یک کاربر مسافر را در نظر بگیرید.

² ID

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.



● پاسخ دستورات

به ازای هر دستوری که اقدام به اجرای آن می‌کنیم، پاسخی از سمت سیستم دریافت می‌کنیم. این پاسخ می‌تواند اطلاعاتی باشد که از سیستم خواسته شده است مانند اطلاعات یک سفر یا لیست همه سفرهای کاربر. اما گونه‌های دیگری از پاسخ نیز وجود دارد.

○ پاسخ درخواست موفقیت‌آمیز

اگر دستوری که کاربر وارد می‌کند (قسمت پس از POST یا GET یا DELETE)، به درستی انجام پذیرد ولی نیازی به بازگرداندن اطلاعاتی نباشد، این پاسخ برای کاربر نمایش داده می‌شود.

خروجی

OK

○ پاسخ خالی بودن

اگر دستوری که کاربر وارد می کند برای نمایش لیستی باشد که در آن لیست هیچ داده ای وجود ندارد، این خروجی نمایش داده می شود. در ادامه مواقعی که نیاز به نمایش این پاسخ وجود دارد توضیح داده شده است.

خروجی
EMPTY

○ پاسخ عدم وجود

اگر دستوری که کاربر وارد می کند (قسمت پس از POST یا GET یا DELETE)، در لیست دستورهای تعریف شده نباشد یا کاربر با وارد کردن شناسه ی ناموجود در سیستم قصد دسترسی به موجودیتی را داشته باشد، این پاسخ در خروجی چاپ می شود.

خروجی
Not Found

○ پاسخ درخواست اشتباه

اگر اولین قسمت خط ورودی کاربر یکی از سه مقدار POST، GET، DELETE نباشد یا این دستور اطلاعات کافی برای اجرا را در خود نداشته باشد، سیستم خروجی زیر را چاپ می کند.

خروجی
Bad Request

○ پاسخ عدم دسترسی

اگر کاربری از دستوراتی که دسترسی ندارد استفاده کند، این خروجی به کاربر نشان داده می‌شود. برای مثال در صورتی که کاربر مسافر از دستور مربوط به راننده استفاده کند، این پیام نشان داده می‌شود.

خروجی
Permission Denied

لیست مناطق شهر

فایل اطلاعات مناطق در اختیارتان قرار گرفته است. در این فایل لیست تمام مناطقی که می‌توان به عنوان مبدا و مقصد مورد انتخاب قرار داد به همراه مختصات جغرافیایی آمده است. از مختصات که از دو بخش `latitude` و `longitude` تشکیل شده است در فاز بعد استفاده خواهد شد. در این فایل نام هر منطقه و مختصات در یک خط آمده است. این بخش‌ها با استفاده از و از یکدیگر جدا شده‌اند. نمونه‌ای از ورودی در کنار صورت پروژه در سامانه درس در اختیار شما قرار گرفته است.

وارد کردن لیست مناطق

در ابتدای شروع برنامه، باید لیست مناطق از یک فایل csv که نام آن به عنوان آرگومان خط فرمان به برنامه‌ی شما داده می‌شود، خوانده شده و در حافظه‌ی برنامه ذخیره شود. دستور آغاز برنامه به شکل زیر خواهد بود.

توجه کنید نام فایل اجرایی برنامه شما باید **utaxi.out** باشد.

ورودی

```
./utaxi.out <csv_file_relative_address>
```

ورودی نمونه

```
./utaxi.out folder1/folder2/info.csv
```

دستورات

ثبت نام

با این دستور افراد می‌توانند در سامانه به عنوان مسافر یا راننده ثبت نام کنند. نام کاربری هر کاربر یک رشته یکتا بین تمام کاربران چه راننده و چه مسافر در سیستم است. اگر نام کاربری در سیستم موجود باشد درخواست معتبر نیست و خروجی Bad Request چاپ می‌شود. تضمین می‌شود اطلاعات ورودی، درون خودشان کاراکتر فاصله ندارند. آرگومان role می‌تواند driver برای کاربر راننده و یا passenger برای مسافر باشد. در غیر این دو حالت خروجی Bad Request چاپ می‌شود.

ورودی

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

خروجی

OK | Bad Request

درخواست سفر

این دستور مخصوص کاربران مسافر است و با آن سفری با مبدا و مقصد مشخص در سیستم ثبت می‌کند. توجه کنید هر درخواستی که ثبت می‌شود که شناسه (id) به آن تخصیص پیدا می‌کند. شناسه‌ها از ۱ شروع می‌شوند و به ترتیب زیاد می‌شوند. مسافرها قبل از پایان سفر قبلی خود نمی‌توانند سفر جدیدی درخواست کنند. اگر نام مبدا، مقصد یا کاربر در سیستم نباشد خطای Not Found چاپ می‌شود. اگر آرگومان‌های لازم داده نشود نیز پیام Bad Request چاپ می‌شود. همانطور که قبلا ذکر شد اگر مسافری هنگامی که هنوز سفر فعلی تمام نشده است درخواست سفر کند نیز باید Bad-Request چاپ شود.

ورودی

```
POST trips ? username <username> origin <origin_name> destination  
<destination_name>
```

خروجی

<trip_id> | Bad Request | Not Found | Permission Denied

ورودی نمونه

```
POST trips ? user ali origin Enghelab destination Valiasr
```

خروجی نمونه

لیست سفرها

این دستور مخصوص کاربران راننده است. در خروجی این دستور لیست سفرهای موجود در سیستم همراه با وضعیت آنها چاپ می‌شود. همچنین ترتیب نمایش سفرها در خروجی به ترتیب صعودی شناسه سفر است. وضعیت سفر می‌تواند سه حالت باشد:

- waiting: هنوز راننده‌ای سفر را قبول نکرده.

- traveling: سفر در جریان است.

- finished: سفر به پایان رسیده.

اگر درخواستی وجود نداشت باید پیام EMPTY چاپ شود.

ورودی
GET trips ? username <username>
خروجی
<trip_id> <passenger_name> <origin> <destination> <status> ... Empty Permission Denied
ورودی نمونه
GET trips ? username amin
خروجی نمونه
1 soroush Amirabad Fatemi waiting 2 ali Qods Vesal traveling

مشخصات یک سفر

این دستور مخصوص کاربر راننده است. در خروجی این دستور مشخصات یک سفر چاپ می‌شود.

ورودی
GET trips ? username <username> id <trip_id>
خروجی
<trip_id> <passenger_name> <origin> <destination> <status> ... Not Found Permission Denied
ورودی نمونه
GET trips ? user amin id 2
خروجی نمونه
2 ali Qods Vesal traveling

لغو درخواست سفر

این دستور مخصوص کاربر مسافر است. با این دستور درخواست سفر لغو می‌شود و دیگر در لیست سفرها نشان داده نمی‌شود. توجه داشته باشید که اگر در آینده سفر جدیدی ایجاد شود از شناسه‌ی سفر لغوشده استفاده نخواهد شد. اگر با دستور مشخصات یک سفر، سفر لغو شده درخواست شود خطای Not Found چاپ می‌شود. اگر مسافری قصد لغو سفر مسافر دیگری را داشت باید خطای Permission Denied چاپ شود. اگر سفر آغاز شده بود یا پایان یافته بود باید خطای Bad Request چاپ شود.

ورودی
DELETE trips ? username <username> id <trip_id>
خروجی
OK Not Found Permission Denied Bad Request

قبول سفر

این دستور مخصوص کاربر راننده است. با این دستور یک سفر توسط راننده قبول می‌شود و وضعیت سفر به traveling تغییر می‌کند. توجه کنید راننده‌ای که مسافری دارد تا زمانی که مسافر قبلی را پیاده نکند نمی‌تواند سفر دیگری قبول کند و اگر دستور قبول سفر دیگری وارد کند باید خطای Bad Request چاپ شود.

ورودی
POST accept ? username <username> id <trip_id>
خروجی
OK Not Found Permission Denied Bad Request

اعلام پایان سفر

این دستور مخصوص کاربر راننده است و پایان سفر را اعلام می‌کند و پس از این دستور وضعیت سفر به finished تغییر می‌کند.

ورودی
POST finish ? username <username> id <trip_id>
خروجی
OK Not Found Permission Denied Bad Request

نحوه‌ی تحویل و نکات پایانی

- تمام فایل‌های خود را در قالب یک پرونده‌ی زیپ با نام A7-<SID>.zip در صفحه‌ی Elearn درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۰۰۹۹۹ است، نام پرونده‌ی شما باید A7-810100999.zip باشد.

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

A7-810100999.zip

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

- 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 ثبت‌نام، 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.
- برنامه‌ی 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 **utaxi.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.