



مقدمه

در فاز اول شما به پیاده‌سازی منطق برنامه و استفاده از رابط خط فرمان^۱ برای تعامل با آن پرداختید. در این فاز امکاناتی جهت توسعه برنامه به آن اضافه خواهید کرد.

شرح تمرین

در این فاز پروژه، تعدادی دستور در راستای گسترش منطق و افزایش هوشمندی برنامه به آن افزوده و پیاده‌سازی می‌شوند. قالب این دستورات مانند فاز قبلی می‌باشد و می‌توانید برای بررسی دقیق‌تر به فاز قبل مراجعه کنید. همچنین باید تمام دستورات فاز قبلی در این فاز نیز قابل اجرا باشد (به جز دستورات فاز یک که در این فاز دستخوش تغییر شده‌اند، این دستورات باید به صورت جدید اجرا شوند). در این دستورات برخی ویژگی‌های جدید به سامانه اضافه خواهند شد تا کاربران بتوانند میزان علاقه‌مندی خود به آهنگ‌هایی که می‌شنوند را ثبت کنند و بتوانند آهنگ‌های مورد علاقه دیگر کاربران را مشاهده کنند. همچنین می‌توانند لیست‌هایی از آهنگ‌های مورد علاقه خود ایجاد کنند و آنها را نگه دارند.

پاسخ دستورات

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

^۱ Command Line Interface

دستورات جدید

دستورات مشترک کاربر عادی و هنرمند

نمایش یک کاربر در سامانه

در این فاز، کاربر اطلاعات بیشتری را در خروجی باید مشاهده کند.

- اطلاعات نمایش داده شده شامل شناسه، نوع کاربر، اسم، لیست افرادی که آن‌ها را دنبال کرده و لیست افرادی که توسط آن‌ها دنبال شده است و همچنین برای کاربران عادی، پلی‌لیست‌های آن‌ها و برای هنرمندان، آهنگ‌های آن‌ها می‌باشد.
- اسامی دنبال‌کنندگان و دنبال‌شوندگان به ترتیب زمانی که کاربر را دنبال کرده‌اند و یا توسط کاربر دنبال شده‌اند نمایش داده می‌شود (هر که زودتر دنبال کرده باشد یا دنبال شده باشد در لیست زودتر نمایش داده می‌شود).

ورودی

```
GET users ? id <{id}>
```

خروجی

ID: <id>

Mode: <mode>

Username: <username>

Followings: <following_1>, <following_2>, ...

Followers: <follower_1>, <follower_2>, ...

Playlists/Songs: <playlist_1_name>, <playlist_2_name>,

.../<song_1_title>, <song_2_title>, ...

| Not Found

نمونه ورودی

```
GET users ? id <11>
```

نمونه خروجی

ID: 11
Mode: user
Username: FabulousMatin
Following: low mist, Mahdi, Dante, ...
Followers: Dante, Mahdi, Alumpish, ...
Playlists: Jazz playlist, Road playlist

دنبال و عدم دنبال کردن کاربران و هنرمندان

با دستور follow کاربر می‌تواند یکی دیگر از کاربران یا یکی از هنرمندان را دنبال کند. و با دستور unfollow می‌توانیم از دنبال کردن کسی دست بکشیم.

- در صورتی که از قبل دنبال شده بود و دوباره می‌خواستیم دنبال کنیم و یا در حالتی که دنبال نشده بود و می‌خواستیم از دنبال کردن آن دست بکشیم، پیام Bad Request چاپ می‌شود.
- در صورتی که کاربر بخواهد خودش را دنبال کند با پیام Bad Request مواجه می‌شود.
- در صورت نبود شناسه پیغام Not Found چاپ می‌شود.

ورودی

```
POST follow ? id <{id}>
```

خروجی

OK | Not Found | Bad Request

نمونه ورودی اول

```
POST follow ? id <5>
```

نمونه خروجی اول

Bad Request

ورودی

```
POST unfollow ? id <{id}>
```

خروجی

OK | Not Found | Bad Request

نمونه ورودی

```
POST unfollow ? id <5>
```

نمونه خروجی

OK

دستورات کاربر عادی

حذف پلی لیست

کاربر با استفاده از این دستور می‌تواند با وارد کردن نام یکی از پلی لیست‌های متعلق به خود، آن پلی لیست را حذف کند.

- در صورتی که اسم پلی لیست در بین پلی لیست‌های این کاربر نبود پاسخ Not Found نشان داده شود.

ورودی

```
DELETE playlist ? name <{name}>
```

خروجی

OK | Not Found | Bad Request | Permission Denied

نمونه ورودی

```
DELETE playlist ? name <Rock n Roll>
```

نمونه خروجی

OK

نمایش یک پلی لیست

کاربر با استفاده از این دستور و با وارد کردن شناسه کاربر مورد نظر و نام پلی لیست او می‌تواند آهنگ‌های موجود در یک پلی لیست را مشاهده کند.

- ترتیب نمایش آهنگ‌ها بر اساس زمان اضافه شدن به لیست است (آخرین آهنگ اضافه شده در انتهای لیست خواهد بود).
- در خط اول خروجی نام ستون‌ها آمده و در ادامه شناسه، نام و هنرمند آهنگ‌ها می‌آید.
- دقت کنید که همچنان در صورتی که پارامتر name وارد نشده بود، خروجی مانند فاز اول خواهد بود و لیست پلی لیست‌ها نمایش داده می‌شود، و در صورتی که پارامتر name وارد شده بود اطلاعات آن پلی لیست به خصوص (مانند نمونه) نمایش داده می‌شود.

- در صورتی که شناسه کاربری وارد شده یا پلی لیست مشخص شده در سامانه وجود نداشته باشند، پیام Not Found در پاسخ نمایش داده می شود.
- در صورتی که شناسه کاربر متعلق به یک هنرمند بود با پاسخ Bad Request مواجه می شویم.

ورودی

```
GET playlist ? name <{name}> id <{user id}>
```

خروجی

```
ID, Name, Artist
{id_1}, {name_1}, {artist_1}
{id_2}, {name_2}, {artist_2}
... | Not Found | Bad Request | Permission Denied
```

نمونه ورودی

```
GET playlist ? name <loneliness> id <9>
```

نمونه خروجی

```
ID, Name, Artist
13, Epitaph, Antimatter
2, Nothing Else Matters, Metallica
```

افزودن یک آهنگ به علاقه مندی ها

با استفاده از این دستور می توانیم یک آهنگ را به لیست علاقه مندی های خود اضافه کنیم. برای اضافه کردن آهنگ از شناسه یکتا آن استفاده می کنیم.

- در صورت نبود شناسه پیام Not Found دیده می شود.
- در صورتی که از قبل آهنگ در لیست مورد علاقه ها بود، با پیام Bad Request مواجه می شویم.

ورودی

```
POST like ? id <{id}>
```

خروجی

```
OK | Not Found | Bad Request | Permission Denied
```

نمونه ورودی

POST like ? id <9>

نمونه خروجی

OK

نمایش آهنگ‌های پرطرفدار

کاربر با استفاده از این دستور می‌تواند فهرست پرطرفدارترین آهنگ‌های موجود در سامانه را به عنوان آهنگ‌های پیشنهادی سامانه به کاربر مشاهده کند.

- در خروجی این دستور حداکثر ۵ آهنگ نمایش داده می‌شوند. ممکن است کمتر از ۵ آهنگ برای نمایش در این بخش موجود باشد که در این صورت همان تعداد نمایش داده خواهد شد.
- پرطرفدار بودن آهنگ‌ها بر اساس تعداد علاقه‌مندی‌های آن‌ها مشخص می‌شود. همچنین آهنگ‌ها برای نمایش داده شدن در این بخش باید حداقل توسط یک نفر به لیست علاقه‌مندی‌ها اضافه شده باشند.
- دقت کنید که در صورت وجود آهنگی در فهرست علاقه‌مندی‌های کاربر، آن آهنگ نباید در فهرست خروجی این دستور نمایش داده شود.
- ترتیب نمایش آهنگ‌ها در این بخش بر اساس تعداد لایک‌های آهنگ‌ها به صورت نزولی است. به این صورت که آهنگ با بیشترین تعداد لایک در ابتدای لیست نمایش داده می‌شود.
- در صورت برابر بودن تعداد لایک‌های دو آهنگ، آهنگ با شناسه کوچکتر بالاتر نمایش داده می‌شود.
- در خروجی این دستور، خط اول ترتیب ستون‌ها (به ترتیبی که در مثال‌ها نشان داده شده) بوده و بعد از آن در هر خط اطلاعات مربوط به یک آهنگ نمایش داده می‌شود.
- در صورت وجود نداشتن آهنگ واجد شرایط در خروجی پیام Empty نمایش داده می‌شود.

ورودی

GET recommendations ?

خروجی

ID, Name, Artist, Likes

<id_1>, <name_1>, <artist_1>, <likes_1>

<id_2>, <name_2>, <artist_2>, <likes_2>

... | Permission Denied | Bad Request | Empty

نمونه ورودی

GET recommendations ?

نمونه خروجی

```
ID, Name, Artist, Likes
1, Flying, Anathema, 10
6, Nothing Else Matters, Metallica, 8
3, Said & Done, Bad Omens, 5
5, Crawling, Linkin Park, 5
8, How I Got Home, SYML, 1
```

نمایش آهنگ‌های مورد علاقه

- کاربر با استفاده از این دستور می‌تواند اطلاعات آهنگ‌های مورد علاقه خود را مشاهده کند.
- ترتیب نمایش بر اساس زمان اضافه شدن به لیست علاقه‌مندی‌ها است (هر چه زودتر به لیست علاقه‌مندی‌ها اضافه شده باشد بالاتر در خروجی نمایش داده می‌شود).
- در خط اول خروجی نام هر ستون و در خطوط بعدی شناسه، نام و هنرمند آهنگ‌ها نمایش داده می‌شود.
- در صورتی که آهنگی مورد علاقه کاربر نباشد، پیام Empty چاپ شود.

ورودی

```
GET likes ?
```

خروجی

```
ID, Name, Artist
<id>, <name>, <artist>
... | Empty | Bad Request | Permission Denied
```

نمونه ورودی

```
GET likes ?
```

نمونه خروجی

```
ID, Name, Artist
16, low mist day 7, Ludovico Einaudi
2, Nothing Else Matters, Metallica
```

نکات و نحوه تحویل

- 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 ۸۱۰۱۰۰۰۰۰ است، نام پرونده‌ی شما باید `A7-810100000.zip` باشد.

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

`A7-810100000.zip`

|— `main.cpp`

|— `makefile`

|— ...

- 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++20` 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++20` استفاده می‌کنید.
- درستی برنامه‌ی شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین از درستی کامل قالب خروجی برنامه خود اطمینان حاصل کنید و از دادن خروجی‌هایی که در صورت پروژه ذکر نشده است اجتناب کنید.
- دقت کنید که نام فایل اجرایی شما باید `sputify.out` باشد.
- 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.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق سیاست درس با آن برخورد خواهد شد.

نمرات

- تمیزی کد

- رعایت کردن نام‌گذاری صحیح و انسجام²
- عدم وجود کد تکراری
- رعایت دندان‌گذاری³
- عدم استفاده از متغیرهای گلوبال
- استفاده صحیح از متغیرهای ثابت⁴ به جای Magic Value-ها
- ساختاردهی کد در قالب توابع کوتاه که فقط یک کار را انجام می‌دهند

- درستی کد

- آزمون‌های خودکار
- پیاده‌سازی صحیح کارکردهای خواسته شده

- طراحی

- طراحی صحیح و منطقی در شی‌گرایی و ارث‌بری
- رعایت Encapsulation
- جداسازی منطق کد از ورودی/خروجی و استفاده از کلاس جداگانه برای مدیریت دستورات
- استفاده مناسب از استثناها برای مدیریت خطا
- میک‌فایل

دقت کنید که موارد ذکر شده لزوماً کل نمره شما را تشکیل نمی‌دهند و ممکن است با تغییراتی همراه باشند.

² Consistency

³ Indentation

⁴ Constant