



fUTball fantasy

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 یک سایت بازی **فوتبال فانتزی** طراحی کنید که دارای امکانات مختلفی است.

در فاز اول باید منطق برنامه‌ی خود برای این برنامه‌ی رقابتی را پیاده‌سازی کنید؛ به طوری که با استفاده از رابط کاربری خطفرمان^۱ می‌توان از آن استفاده کرد. کاربران می‌توانند جدول‌های مختلفی را در همان ابتدا مشاهده کنند و اگر جذب شدند، بعد از ایجاد اکانت تیم فانتزی خود را ایجاد کنند و با جمع‌آوری امتیازات با دیگران به رقابت بپردازند.

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، این است تمرین حتماً به صورت بخش به بخش، تست و پیاده‌سازی شود؛ بدین شکل که یک قالب کلی برای پروژه ایجاد شود، سپس دستورات به آن اضافه شوند.

^۱ Command Line Interface

فوتبال فانتزی

فوتبال فانتزی یک بازی آنلاین است که در آن شما به عنوان یک مدیر تیم، بازیکنان واقعی فوتبال را برای تشکیل تیم خود انتخاب می‌کنید. هر بازیکن در بازی فوتبال فانتزی، با اعطای امتیاز براساس عملکرد واقعی در بازی‌های واقعی، به شما امتیاز خواهد داد. شما باید با انتخاب بهترین ترکیب بازیکنان، تلاش کنید که بیشترین امتیاز را کسب کنید. هر هفته با بررسی نتایج بازی‌های واقعی و به‌روزرسانی ترکیب تیم، می‌توانید تیم بهتری برای هفته‌ی آتی درست کنید و بررسی کنید که در کدام حالت‌ها می‌توانید امتیاز بیشتری به دست آورید. هدف شما در بازی فوتبال فانتزی، کسب امتیاز بیشتر از بازیکنان دیگر است. دقت کنید که در این بازی کاربر A می‌تواند یک بازیکن خاص را داشته باشد و کاربر B هم همان بازیکن را در تیم خودش داشته باشد و بازیکنان یکتا متعلق به یک تیم فانتزی نیستند.

شرح تمرین

در این فاز منطق اولیه برنامه توسط دستوراتی پیاده‌سازی می‌شود. 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 فایل CSV در اختیار شما قرار خواهد گرفت که شامل اطلاعاتی مانند بازی‌های انجام‌شده خواهد بود. در ابتدای شروع برنامه، باید محتویات این فایل‌ها را بخوانید و به شکل مناسبی آنها را ذخیره کنید. در ادامه به طور دقیق‌تر جزئیات مربوط به این فایل‌ها توضیح داده خواهد شد.

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. ترتیب آرگومان‌ها ثابت است و می‌توانید این مورد را در نظر بگیرید که دستورات با ترتیب گفته شده در صورت پروژه، در آزمون‌ها خواهند آمد. برای مثال دستور signup فقط به حالت زیر خواهد آمد:

```
POST signup ? team_name <team_name> password <password>
```

دستوراتی که با آنها کار خواهیم کرد شامل موارد زیر می‌شوند:

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

GET: برای دریافت اطلاعات استفاده می‌شود.

PUT: برای تغییر اطلاعات استفاده می‌شود.

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

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.

همچنین توجه کنید که در برخی دستورات از علامت «+» استفاده شده است. این بدین معنی است که این آرگومان می‌تواند وارد شود و می‌تواند نشود. در صورت وارد نشدن، گفته می‌شود که چه مقداری در نظر گرفته می‌شود.

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 نشان‌دهنده‌ی وضعیت انجام دستور یا شامل یک‌سری اطلاعات مانند لیست بازیکنان 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 دستوری از لیست دستورات دیگر کاربران را وارد کرد (مثلاً کاربر عادی دستورات ادمین را وارد کند) 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 نام بازیکن یا لیگ یا تیم مربوطه وجود نداشته باشد (در هر قسمت که جستجویی انجام می‌شود اما نتیجه‌ای 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 |



fUTball fantasy

دستورات عمومی

این دستورات توسط تمامی کاربران می‌تواند اجرا شود (چه کاربرانی که ثبت نام کرده‌اند و چه آنهایی که ثبت نام نکرده‌اند).

مشاهده تیم هفته

با انجام این دستور، کاربر می‌تواند تیم هفته را مشاهده کند. تیم هفته شامل یک دروازه‌بان، دو دفاع، یک هافبک و یک مهاجم است که در هفته مشخص شده بیشترین امتیاز را کسب کرده‌اند. دقت کنید که لازم نیست همه بازیکنان نمایش داده شده از یک تیم باشند و باید به صورت کلی در نقش خود بهترین بازی را کرده باشند. در صورتی که هفته وارد نشود، تیم هفته فعلی نمایش داده می‌شود. در صورتی که شماره هفته وارد شده وجود نداشته باشد، خطای Bad Request نمایش داده می‌شود.

در تمامی بخش‌ها امتیاز بازیکنان یک عدد بین 0 تا 10 است و همواره با یک رقم اعشار گزارش می‌شود.

ورودی

```
GET team_of_the_week ? +week_num <week_num>
```

خروجی

```
team of the week:  
GoalKeeper: <name> | score: <score>  
Defender 1: <name> | score: <score>  
Defender 2: <name> | score: <score>  
Midfielder: <name> | score: <score>  
Forward: <name> | score: <score>  
| Bad Request
```

ورودی نمونه

```
GET team_of_the_week ? week 3
```

خروجی نمونه

```
GoalKeeper: Mario de Luis | score: 9.2  
Defender 1: Gerard Pique | score: 8.1  
Defender 2: Carles Puyol | score: 9.0  
Midfielder: Zinedine Zidane | score: 9.9  
Forward: Samuel Chukwueze | score: 7.1
```

مشاهده اطلاعات یک تیم

با انجام این دستور، کاربر می‌تواند اطلاعات یک تیم را مشاهده کند. در صورتی که کاربر یکی از کلمات کلیدی gk(دروازه بان)، df(دفاع) md(هافبک)، fw(مهاجم) را وارد کند، تنها اطلاعات بازیکنان نقش مشخص شده نمایش داده می‌شود. همچنین با وارد کردن ranks، بازیکنان مدنظر به ترتیب امتیاز و به صورت نزولی مرتب می‌شوند. در غیر این صورت حروف الفبا. امتیاز هر بازیکن، میانگین امتیازات آن بازیکن در طول فصل می‌باشد (بازی‌هایی که بازی نکرده در میانگین‌گیری حساب نمی‌شوند و اگر بازیکنی کلاً بازی نکرده بود، نمره آن 0.0 در نظر گرفته می‌شود).

در صورتی که نقش وارد شده وجود نداشته باشد، خطای Bad Request نمایش داده می‌شود.

در صورتی که تیم وارد شده وجود نداشته باشد، خطای Not Found نمایش داده می‌شود.

- توجه: در ورودی اسم تیم‌هایی که به صورت چندبخشی هستند مانند Bayern Munich با استفاده از "_" به هم

متصل می‌شوند: "Bayern_Munich"

ورودی

```
GET players ? team_name <team_name> +gk/df/md/fw +ranks
```

خروجی

list of players:

1. name: <name> | role: <role> | score: <score>

2. name: <name> | role: <role> | score: <score>

...

| Bad Request | Not Found

ورودی نمونه

```
GET players ? team Bayern_Munich fw
```

خروجی نمونه

list of players:

1. name: Jamal Musiala | role: fw | score: 6.9

2. name: Leroy Sane | role : fw | score: 9.4

3. name: Sadio Mane | role: fw | score: 6.7

4. name: Thomas Muller | role: fw | score: 7.3

مشاهده رتبه تیم‌های لیگ

با اجرای این دستور، کاربر می‌تواند اطلاعاتی شامل امتیاز، تعداد گل‌های زده شده، تعداد گل‌های خورده و رتبه تیم‌های لیگ را مشاهده کند. در اطلاعات نمایش داده شده، GF مخفف goals for نشان‌دهنده تعداد گل‌های زده شده و GA مخفف goals against نشان‌دهنده تعداد گل‌های خورده شده است. تیم‌های نمایش داده شده بر اساس امتیاز مرتب شده‌اند. در صورتی که دو تیم امتیاز یکسانی داشته باشند، بر اساس تفاضل گل و در صورتی که تفاضل گل آنها هم یکی بود، بر اساس

تعداد گل زده شده مرتب می‌شوند. در تمامی بخش‌ها این پروژه در صورت تساوی امتیازات بر اساس حروف الفبا نمایش می‌دهیم.

| ورودی |
|--|
| GET <code>league_standings</code> |
| خروجی |
| league standings: 1. <team_name>: score: <score> GF: <num_goals_for> GA: <num_goals_against> 2. <team_name>: score: <score> GF: <num_goals_for> GA: <num_goals_against> ... |
| ورودی نمونه |
| GET <code>league_standings</code> |
| خروجی نمونه |
| league standing: 1. Barcelona: score: 85 GF: 64 GA: 13 2. Real Madrid: score: 71 GF: 70 GA: 32 3. Atletico Madrid: score: 69 GF: 60 GA: 27 ... |

مشاهده رتبه کاربران

با اجرای این دستور کاربر می‌تواند امتیاز کلی تیم‌های فانتزی را مشاهده کند. تیم‌های فانتزی بر اساس امتیاز و به صورت نزولی مرتب شده‌اند (امتیاز کلی در اینجا جمع امتیاز هر تیم در کل فصل است). در صورت تساوی امتیازات بر اساس حروف الفبا نمایش می‌دهیم.

| ورودی |
|---|
| GET <code>users_ranking</code> |
| خروجی |
| 1. team_name: <team_name> point: <point> 2. team_name: <team_name> point: <point> ... |
| ورودی نمونه |


```
GET users_ranking
```

خروجی نمونه

```
1. team_name: noob_master24 | point: 396.7
2. team_name: MEHdi131 | point: 387.9
3. team_name: Mohammad_messi | point: 386.5
```

مشاهده اطلاعات بازی های یک لیگ

با اجرای این دستور کاربر می تواند بازی های تیم های لیگ را در هفته مشخص شده مشاهده کند. در صورتی که هفته داده نشود، اطلاعات بازی های هفته فعلی نمایش داده می شود.
در صورتی که که نام تیم وجود نداشته باشد، خطای Not Found نمایش داده می شود.
در صورتیکه هفته وارد شده وجود نداشته باشد، خطای Bad Request نمایش داده می شود.

ورودی

```
GET matches_result_league ? +week_num <week_num>
```

خروجی

```
<team1> <team1_goals> | <team2> <team2_goals>
<team1> <team1_goals> | <team2> <team2_goals>
...
| Bad Request | Not Found
```

ورودی نمونه

```
GET matches_result_league ? week_num 2
```

خروجی نمونه

```
Elche 2 | Real Betis 3
Espanyol 2 | Mallorca 1
Real Madrid 1 | Atletico Madrid 1
...
```

احراز هویت

این دستورات برای ورود به مسابقات و ایجاد تیم فانتزی خود برای کسب امتیازات می باشد.

ثبت نام

این دستور مخصوص کاربرانی است که به سامانه وارد نشده‌اند. با انجام این دستور کاربر وارد سامانه می‌شود. نام تیم فانتزی هر فردی که ثبت نام انجام می‌دهد باید یکتا باشد و نشان‌دهنده‌ی آن فرد در سامانه خواهد بود. 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 ? team_name <team_name> password <password> |
| خروجی |
| OK Bad Request |
| ورودی نمونه |
| POST signup ? team_name Sharks password 021021 |
| خروجی نمونه |
| OK |

ورود

با انجام این دستور، کاربری که قبلاً ثبت نام انجام داده‌است، وارد سامانه می‌شود و می‌تواند دستورات خود را وارد کند. اگر رمز خود را اشتباه وارد کند، Permission Denied دریافت می‌کند. اگر نام تیم فانتزی که وارد می‌کند در سامانه وجود نداشته باشد، Not Found دریافت خواهد کرد.

| ورودی |
|---|
| POST login ? team_name <team_name> password <password> |
| خروجی |
| OK Bad Request Permission Denied Not Found |
| ورودی نمونه |
| POST login ? team_name Sharks team_name 021023 |
| خروجی نمونه |
| Permission Denied |

ورود ادمین

با انجام این دستور، ادمین می‌تواند وارد حساب خود شود و دستورات مربوط به ادمین را اجرا کند. اگر نام کاربری ادمین یا رمز او غلط باشد، خطای Bad Request داده می‌شود.

نکته: در کد خود به صورت پیش‌فرض نام کاربری ادمین را admin و رمز او را 123456 در نظر بگیرید.

| |
|--|
| ورودی |
| POST register_admin ? username <username> password <password> |
| خروجی |
| OK Bad Request |
| ورودی نمونه |
| POST register_admin ? username admin 123456 |
| خروجی نمونه |
| OK |

خروج

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

| |
|--------------------|
| ورودی |
| POST logout |
| خروجی |
| OK |

مجموعه دستورات کاربر

ساختار تیم

هر تیم فانتزی شامل 5 بازیکن می‌شود: یک دروازه‌بان، دو مدافع، یک هافبک و یک مهاجم. تمام این 5 بازیکن در محاسبه‌ی امتیاز نهایی تیم لحاظ می‌شوند و کاربر نمی‌تواند بازیکن ذخیره بخرد. همچنین یک تیم تا زمانی که تکمیل نشده وارد بازی نمی‌شود و از هفته‌ای که هر 5 بازیکن را داشته باشد، برایش امتیاز لحاظ می‌شود. همچنین اگر تیم کاملی در یک هفته بازیکن بفروشد ولی جای آن را با بازیکن دیگری پر نکند، در آن هفته امتیازی برایش لحاظ نمی‌شود و صفر امتیاز می‌گیرد.

فروش بازیکن

این دستور یک بازیکن خاص را از تیم فانتزی کاربر حذف می‌کند. دقت کنید که اسم کامل بازیکن با space می‌تواند بیاید. اگر بازیکن خواسته‌شده در تیم فانتزی کاربر وجود نداشت، باید خطای Not Found دریافت شود. اگر پنجره‌ی نقل‌وانتقالات در آن زمان باز نبود، باید خطای Permission Denied دریافت شود. هر کاربر در هر هفته اجازه دارد که 2 بازیکن را از تیم بفروشد و بازیکن دیگری بخرد. اگر کاربر در آن هفته تعداد نقل‌وانتقالات مجازش تمام شده باشد، باید خطای Permission Denied دریافت شود. دقت کنید که محدودیت تعداد نقل‌وانتقالات برای کاربریست که تیم خود را تکمیل کرده. اگر کاربر جدیدی وارد بازی شود، تا هفته‌ای که تیم خود را تکمیل نکند، می‌تواند به شکل نامحدود بازیکن بفروشد و بخرد زیرا هنوز وارد بازی نشده است.

| ورودی |
|---|
| POST sell_player ? name <name> |
| خروجی |
| OK Bad Request Permission Denied Not Found |
| ورودی نمونه |
| POST sell_player ? name Ruben Dias |
| خروجی نمونه |
| OK |

خرید بازیکن

این دستور یک بازیکن خاص را به تیم فانتزی کاربر اضافه می‌کند. دقت کنید مانند دستور فروش، اسم کامل بازیکن با space می‌تواند بیاید.

اگر بازیکن خواسته‌شده وجود نداشت، باید خطای Not Found دریافت شود. اگر پنجره‌ی نقل‌وانتقالات در آن زمان باز نبود، باید خطای Permission Denied دریافت شود. اگر پست آن بازیکن در تیم فانتزی کاربر از قبل کامل اشغال شده باشد، باید خطای Bad Request دریافت شود (برای مثال کاربر بخواهد یک مدافع بخرد درحالی‌که از قبل دو مدافع در تیم خود دارد). اگر بازیکن مورد نظر در هفته‌ی آتی نمی‌تواند بازی کند باید پیغام "This player is not available for next week" دریافت شود.

یک بازیکن در 3 شرایط برای هفته‌های بعدی در دسترس نیست:

1. اگر در هفته‌ی قبل کارت قرمز دریافت کرده باشد.
2. اگر تعداد کارت زردهای دریافتی بازیکن به ۳ برسد. (بعد از اعمال محرومیت، تعداد کارت‌های زرد دوباره از اول حساب می‌شود).

3. اگر در 3 هفته‌ی اخیر مصدوم شده باشد.

| ورودی |
|--|
| POST buy_player ? name <name> |
| خروجی |
| OK This player is not available for next week Bad Request Permission Denied Not Found |
| ورودی نمونه |
| POST buy_player ? name Virgil Van Dijk |
| خروجی نمونه |
| OK |

مشاهده‌ی تیم فانتزی

این دستور بازیکنان تیم فانتزی خواسته شده و میزان کل امتیازهای تیم (مجموع در کل هفته‌ها) را برای کاربر نشان می‌دهد. کاربر می‌تواند اسم تیم فانتزی که می‌خواهد ببیند را وارد کند اما اگر چیزی وارد نکرد، اطلاعات تیم خودش را دریافت می‌کند. اگر تیم خواسته شده وجود نداشت، باید خطای Not Found دریافت شود. اگر تیم خواسته شده وجود داشت اما تکمیل نبود، باید خطای Empty دریافت شود. دقت کنید که مدافعان به ترتیب حروف الفبا نمایش داده می‌شوند.

| ورودی |
|---|
| GET squad ? +fantasy_team <fantasy_team> |
| خروجی |
| Permission Denied Not Found Empty fantasy_team: <fantasy_team> Goalkeeper: <goalkeeper_name> Defender1: <defender1_name> Defender2: <defender2_name> Midfielder: <midfielder_name> Striker: <striker_name> |

| |
|---|
| Total Points: <points> |
| ورودی نمونه |
| GET squad ? fantasy_team Liverpool_Iran |
| خروجی نمونه |
| fantasy_team: Liverpool_Iran Goalkeeper: Alisson Becker Defender1: Virgil Van Dijk Defender2: Trent Alexander-Arnold Midfielder: Jude Bellingham Striker: Divock Origi Total Points: 2023 |

مجموعه دستورات ادمین

بستن پنجره نقل و انتقالاتی

این دستور پنجره نقل و انتقالاتی را می‌بندد و بعد از آن دیگر کاربران نمی‌توانند برای تیمشان خرید و فروش داشته باشند.

| |
|-----------------------------------|
| ورودی |
| POST close_transfer_window |
| خروجی |
| OK |
| ورودی نمونه |
| POST close_transfer_window |
| خروجی نمونه |
| OK |

بازکردن پنجره نقل و انتقالاتی

این دستور پنجره نقل و انتقالاتی را باز می‌کند و بعد از آن کاربران می‌توانند برای تیمشان خرید و فروش داشته باشند.

| |
|---------------------------|
| ورودی |
| POST open_transfer_window |
| خروجی |
| OK |
| ورودی نمونه |
| POST open_transfer_window |
| خروجی نمونه |
| OK |

گذشتن هفته

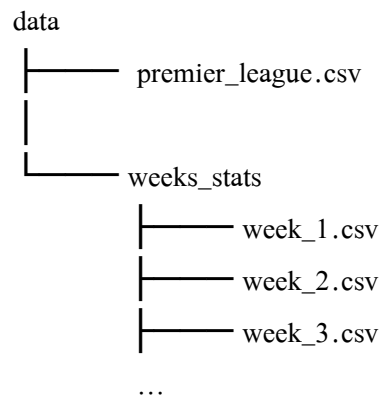
با این دستور امتیازات بازیکنان در هفته‌ی فعلی لود می‌شود و امتیازات تیم‌های فانتزی و تیم‌های واقعی و جدول‌هایشان هم به‌روزرسانی می‌شود.

| |
|----------------|
| ورودی |
| POST pass_week |
| خروجی |
| OK |
| ورودی نمونه |
| POST pass_week |
| خروجی نمونه |
| OK |

توضیحات مربوط به فایل‌های csv

dataset برنامه که شامل لیست تیم‌ها و بازیکنان و نتایج هفته‌ها می‌باشد، در اختیار شما قرار می‌گیرد. توضیح هر یک از فایل‌ها در ادامه آمده است. توجه داشته باشید برای ستون‌هایی که بیشتر از یک مقدار در آن‌ها قرار دارد، مقادیر با سمی‌کالن (;) جدا شده‌اند.

فرض کنید که این dataset در پوشه data در کنار برنامه شما قرار می‌گیرد. ساختار dataset-ای که به شما داده شده به شکل زیر می‌باشد:



فایل premier_league.csv شامل لیست تیم‌ها و اسامی بازیکنان هر تیم به تفکیک پست می‌باشد و فایل‌های درون پوشه weeks_stats شامل اطلاعات بازی‌ها هر هفته به تفکیک هر بازی می‌باشد. اطلاعات هر بازی شامل اسم دو تیم، نتیجه بازی، مصدومیت‌ها، بازیکنانی که کارت‌های زرد و قرمز گرفتند و امتیاز هر یک از بازیکنان می‌باشد (با توجه به اینکه هر تیم تعدادی تعویض دارد تعداد بازیکنان بین 22 تا 28 می‌باشد و تعداد ثابتی ندارد).

با توجه به بزرگ بودن ساینز فایل‌ها، در صورتی که می‌خواهید راحت‌تر محتوای فایل‌های csv را مشاهده کنید، می‌توانید از افزونه‌های موجود در vscode استفاده کنید. همچنین می‌توانید فایل‌ها را روی google drive آپلود کرده و با ابزارهای گوگل (مثل google sheets) به بررسی فایل‌ها بپردازید.

نحوه تحویل

- می‌توانید این تمرین را در قالب گروه‌های دو نفره انجام دهید. گروهی انجام دادن این تمرین، الزامی نیست و می‌توانید به صورت تکی نیز آن را تحویل دهید.
- توجه داشته باشید که در صورت گروهی انجام دادن تمرین، هر دو عضو گروه باید اشراف کامل روی تمرین و کد آپلود شده داشته باشند و این موضوع در تحویل تمرین در نظر گرفته می‌شود.
- تمام فایل‌های خود را در قالب یک پرونده‌ی زیپ با نام A7-SID.zip در صفحه Elearn درس بارگذاری کنید که SID شماره دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۰۱۹۹۹ است، نام پرونده شما باید A7-810101999.zip باشد. همچنین در صورتی که به صورت گروهی این تمرین را تحویل می‌دهید و شماره دانشجویی اعضای تیم، ۸۱۰۱۰۱۹۹۸ و ۸۱۰۱۰۱۹۹۹ باشد، نام این پرونده را A7-810101998-810101999.zip بگذارید.
- در صورت گروهی انجام دادن تحویل، تنها یک نفر از اعضای گروه تمرین را در سامانه آپلود کند.

- با توجه به حجم نسبتاً زیاد این فاز از تمرین توصیه می‌شود قبل از پیاده‌سازی کد طراحی اولیه‌ای برای منطق برنامه انجام دهید و پس از این طراحی شروع به پیاده‌سازی آن کنید. 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.
- دقت کنید که نام پرونده‌ی اجرایی شما باید footballFantasy.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.