



مدرس: رامتین خسروی

طراحان: الهه خداوردی، نسا عباسی، عرفان عسگری،
کورش علی‌نقی، شهنام فیضیان، عرفان میرشمس،
امیررضا نادی، پریسا یحیی‌پور، محمد امین یوسفی

مهلت تحویل: چهارشنبه ۱۶ خرداد ۱۴۰۳، ساعت ۲۳:۵۵



UTMS

مقدمه

این پروژه به جمع‌بندی آموخته‌های شما در این درس می‌پردازد. انتظار می‌رود مهارت‌هایی را که در تمرین‌های پیشین و سایر بخش‌های درس آموخته‌اید، در پیاده‌سازی این پروژه به کار گیرید. در این فاز شما باید قابلیت‌های جدیدی به سامانه دانشگاهی که در فاز قبل پیاده‌سازی کرده‌اید اضافه کنید. از جمله این قابلیت‌ها می‌توان به اضافه شدن کانال اطلاع‌رسانی به ارائه درس، فرم جذب دستیار آموزشی و عکس پروفایل برای کاربران اشاره کرد. نکته قابل توجه در این پروژه این است که باید سعی کنید از قابلیت‌هایی که در فاز قبل توسعه داده‌اید به خوبی استفاده کنید و تا جای ممکن از کد تکراری پرهیز کنید.

شرح تمرین

در این فاز کاربر جدیدی اضافه نمی‌شود و همچنان ۳ نوع کاربر دانشجو، استاد و کاربر پیش‌فرض سیستم (مدیر سیستم) را خواهیم داشت. اساتید می‌توانند برای ارائه درس خود فرم جذب دستیار آموزشی بسازند و در صفحه شخصی خود آن را به اشتراک بگذارند. دانشجویان در صورت دارا بودن شرایط، می‌توانند این فرم‌ها را پر کنند و داوطلب مشارکت به عنوان دستیار آموزشی در یک ارائه درس بشوند. اساتید می‌توانند مشخصات افرادی که فرم‌های آنها را پر کرده‌اند مشاهده کنند و آنها را تایید یا رد کنند. به هر ارائه درس یک کانال اطلاع‌رسانی تعلق می‌گیرد که استاد و دستیاران آموزشی آن ارائه درس اجازه پست گذاشتن در آن کانال را

دارند. اگر پست جدیدی در کانال ارائه درس منتشر شود برای استاد، دستیاران آموزشی و دانشجویان آن ارائه درس اعلان ارسال می‌شود. همچنین کاربران سامانه می‌توانند برای حساب کاربری خود عکس پروفایل بگذارند.

در ابتدای اجرای برنامه، شما باید یک سری اطلاعات را از روی مجموعه داده¹هایی که در قالب CSV² به شما داده می‌شوند خوانده و در برنامه خود ذخیره کنید. در ادامه توضیح هر کدام از این مجموعه داده‌ها آمده است.

قالب فایل‌های ورودی

اطلاعات رشته‌ها، دانشجویان، دروس و استادان در فایل‌های جداگانه قرار دارد. مسیر این فایل‌ها به ترتیب (ابتدا مسیر فایل رشته‌ها، دانشجویان، دروس و سپس استادان) از طریق آرگومان خط فرمان به برنامه داده می‌شود. برای آشنایی با این نوع آرگومان‌ها، می‌توانید [این لینک](#) را مشاهده کنید.

نمونه ورودی
<code>./UTms.out </path/to/majors/file> </path/to/students/file> </path/to/courses/file> </path/to/professors/file></code>

فایل رشته‌ها

این فایل شامل دو ستون mid,major است که نشان دهنده کد رشته و نام رشته است.

نمونه فایل رشته‌ها
<code>mid,major 1,Computer Engineering 2,Electrical Engineering</code>

فایل دانشجویان

این فایل شامل پنج ستون sid, name, major_id, semester, password است که به ترتیب نشان دهنده شماره دانشجویی، نام، کد رشته تحصیلی، ترم فعلی دانشجو و رمز ورود است.

¹ Dataset

² Comma-Separated Values

نمونه فایل دانشجویان

```
sid,name,major_id,semester,password  
810102612,Elahe,1,2,ImtheproblemItsme  
810102342,Nesa,2,4,meoow
```

فایل دروس

این فایل شامل پنج ستون cid, name, credit, prerequisite, majors_id که به ترتیب کد درس، نام درس، تعداد واحد، پیشنیاز درس (حداقل ترم دانشجو) و کد رشته‌هایی است که مجاز به اخذ و ارائه آن درس هستند را نشان می‌دهند. در صورتی که دانشجویان و اساتید چند رشته مجاز به اخذ و ارائه درس باشند کد رشته‌ها با سمی‌کالن (;) از هم جدا می‌شوند.

نمونه فایل دروس

```
cid,name,credit,prerequisite,majors_id  
1,Advanced Programming,3,2,1;2  
2,Operating System,3,5,1
```

فایل استادان

این فایل شامل 5 ستون pid,name,major_id,position,password است که به ترتیب شناسه، نام، کد رشته، درجه هر استاد و رمز ورود را مشخص می‌کنند. مقادیر قابل قبول برای ستون position شامل موارد زیر است:
instructor, assistant professor, associate professor, professor

نمونه فایل استادان

```
pid,name,major_id,position,password  
810420432,Kambiz Afrooz,1,professor,halalalaylalalay
```

تضمین می‌شود major_id های استفاده شده در تمامی فایل‌ها در فایل رشته‌ها وجود دارند.

انواع دستورات

در این فاز، منطق برنامه در قالب تعدادی دستور که در ادامه توضیح داده شده است پیاده‌سازی می‌شود. روند استفاده از برنامه به این شکل است که کاربر در برنامه شما با استفاده از رابط خط فرمان³، 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، شناسه⁴ 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. توجه کنید که برای دستوراتی که آرگومان نداریم نیز علامت ؟ می‌آید.

نکته دیگری که برای دستورات وجود دارد این است که قبل از وارد شدن دستور، عبارت GET یا POST یا DELETE یا PUT وارد می‌شود که به این شکل دستورات به نحوی از هم جدا می‌شوند. دستوراتی که برای دریافت اطلاعات از سیستم استفاده می‌شوند در دسته GET قرار می‌گیرند؛ برای وارد کردن اطلاعات از دسته POST، برای تغییر اطلاعات موجود از دسته PUT و برای حذف اطلاعات از دسته DELETE استفاده می‌کنیم. این نام‌گذاری دستورها در فازهای آتی پروژه که برنامه خود را روی وب عرضه خواهید کرد معنای خاص پیدا خواهند کرد. همچنین دقت کنید که ممکن است دو دستور با نام‌های مشابه وجود داشته باشند اما در دسته‌های متفاوتی قرار بگیرند؛ در این صورت ماهیت این دو دستور متفاوت بوده و در صورت فراخوانی آن‌ها، نتایج متفاوتی را مشاهده خواهیم کرد.

همینطور دقت داشته باشید که تمامی دستورها پس از اجرا شدن دارای خروجی مشخص هستند که منحصراً ذکر می‌شود. اگر در دستورات وارد شده کاربر، خطایی وجود داشته باشد، چه در دستورات چه در آرگومان‌ها، 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؛ بنابراین 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 که از سیستم خواسته شده است، باشد. اما گونه‌های دیگری از پاسخ نیز وجود دارد که در ادامه توضیح داده خواهد شد (رنگ‌های نمونه صرفاً برای خوانایی می‌باشد و نباید از آن‌ها در خروجیتان استفاده کنید).

³ command line

⁴ ID

پاسخ درخواست موفقیت‌آمیز

اگر دستوری که کاربر وارد می‌کند به درستی انجام شود و به اتمام برسد، این پاسخ نمایش داده می‌شود (در برخی از حالات ممکن است خود دستور خروجی مفصل‌تری داشته باشد که در این صورت این پاسخ نمایش داده نمی‌شود؛ این حالات در ادامه و در توضیح هر بخش توضیح داده شده‌اند).

خروجی

OK

پاسخ خالی بودن

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

خروجی

Empty

پاسخ عدم وجود

در صورتی که دستور وارد شده در لیست دستورات GET, POST, DELETE, PUT وجود نداشت، این پیغام نمایش داده می‌شود. همچنین در صورتی که شناسه وجود نداشته باشد و به طور کلی در هر قسمت که جستجویی انجام می‌شود اما نتیجه‌ای در بر ندارد، این پاسخ داده می‌شود.

خروجی

Not Found

پاسخ درخواست اشتباه

اگر اولین قسمت ورودی کاربر، هیچ کدام از لیست دستورهای GET, POST, PUT و DELETE نباشد، این پاسخ نمایش داده می‌شود. همچنین اگر دستور واردشده، اطلاعات کافی برای اجرا را در خود نداشته باشد و یا قالب دستور واردشده، با هیچ کدام از دستوراتی که در ادامه می‌آیند مطابقت نداشته باشد (آرگومان‌های دستور به درستی داده نشده باشند یا مقادیر آرگومان‌ها مطابق انتظار نباشند)، این پاسخ نمایش داده می‌شود.

خروجی

Bad Request

پاسخ عدم دسترسی (دسترسی غیرمجاز)

اگر یک کاربر دسترسی از لیست دستورات دیگر کاربران را وارد کرد این پیغام نمایش داده می‌شود.

خروجی

Permission Denied

سنجش خطاها

اولیت‌بندی سنجش خطاها در اجرای برنامه به صورت زیر می‌باشد:

1. ابتدا بررسی می‌شود که دستور با یکی از متدهای GET، POST، PUT یا DELETE شروع می‌شود. در صورتی که در ابتدای دستور وارد شده یکی از این چهار کلمه نباشد خطای Bad Request نمایش داده می‌شود.
2. پس از آن بررسی می‌شود که دستور وارد شده در لیست دستورات وجود دارد یا خیر؛ برای مثال دستور GET something_non_existant در دستورات برنامه نیست. در این حالت باید پاسخ Not Found نمایش داده شود.
3. سپس برای هر دستور اجازه دسترسی بررسی شود که در صورت عدم دسترسی با دستور Permission Denied مواجه شوند. در صورتی که کاربر هنوز لاگین نکرده باشد، یا در صورتی که یک کاربر دستورات یک نوع کاربر دیگر را وارد کند، از مصادیق این خطا هستند.
4. پس از آن حالات خاص هر دستور بررسی می‌شود، تضمین می‌شود در این حالت صرفاً با یکی از حالات خاص مواجه هستیم (چند خطا در اینجا رخ نمی‌دهد).

دستورات

دستورات مشترک استاد و دانشجو

دستور ارسال پست

با این دستور هر کاربر می‌تواند محتوای مورد نظر خود را در صفحه شخصی خود قرار دهد. با این کار برای کسانی که با کاربر کانکت هستند یک اعلان ارسال می‌شود که به شخص نشان می‌دهد که کاربر یک پست جدید در صفحه شخصی خود به اشتراک گذاشته است. تفاوت این دستور با فاز قبل اضافه شدن عکس به پست‌ها است. به این صورت که اگر شخص می‌خواهد به پست خود عکس اضافه کند هنگام وارد کردن دستور، آدرس عکس مربوطه را نیز وارد می‌کند. توجه کنید که پست‌ها می‌توانند عکس نداشته باشند.

توجه کنید که پست ها حتما دارای متن هستند و اگر دارای عکس بودند در چاپ اطلاعات پست ها که دستوراتش در فاز قبل عنوان شد عکس ها نمایش داده نمی شوند و خروجی دستورات مربوط به نمایش اطلاعات پست ها تفاوتی نخواهد کرد. این عکس ها در فاز بعدی که پروژه روی وب پیاده سازی می شود، نمایش داده می شوند.

ورودی
<code>POST post ? title <title> message <message> image <image_address></code>
خروجی
<code>OK Bad Request</code>
نمونه ورودی
<code>POST post ? title "Good Day" message "Ye rooz joonane dige" image images/itsFine.png</code>
نمونه خروجی
<code>OK</code>
نمونه ورودی
<code>POST post ? title "Good Day" message "Ye rooz joonane dige"</code>
نمونه خروجی
<code>OK</code>

دستور اضافه کردن عکس پروفایل

با این دستور هر کاربر می تواند برای خود عکس پروفایل مشخص کند. این دستور روی خروجی دیگر دستورات تاثیری نمی گذارد و در فاز بعدی مورد استفاده قرار می گیرد.

ورودی
<code>POST profile_photo ? photo <photo_address></code>
خروجی
<code>OK Bad Request</code>
نمونه ورودی
<code>POST profile_photo ? photo photos/ponio.png</code>

نمونه خروجی

OK

دستور ارسال پست در کانال اطلاع رسانی

با این دستور استاد و دانشجو می‌توانند در صفحه مربوط به درسی که ارائه می‌دهند و یا دستیار آموزشی آن هستند پست اضافه کنند.

این پست‌ها همانند پست‌هایی است که تاکنون گفته شد. اعلان این پست‌ها برای همه افراد مربوط به صفحه درس از جمله استاد، دستیاران آموزشی و دانشجویان که آن ارائه درس را اخذ کرده‌اند ارسال خواهد شد. توجه کنید که اعلان پست جدید برای شخصی که آن پست را اضافه کرده ارسال نخواهد شد.

نمونه اعلان

7 Advanced Programming: New Course Post

- اگر ارائه درسی با شناسه وارد شده وجود نداشت با خطای Not Found مواجه می‌شویم.
- اگر ارائه درسی با شناسه وارد شده در بین دروسی که استاد ارائه می‌کند یا دانشجو دستیار آموزشی آن‌ها است وجود نداشت با خطای Permission Denied مواجه می‌شویم.
- توجه داشته باشید منظور از آرگومان id در این دستور شناسه ارائه درس است.

ورودی

```
POST course_post ? id <id> title <title> message <message> image <image_address>
```

خروجی

OK | Bad Request | Not Found | Permission Denied

نمونه ورودی

```
POST course_post ? id 4 title "Homework 6" message "Phase 2"
```

نمونه خروجی

OK

دستور مشاهده کانال اطلاع‌رسانی یک ارائه درس

با این دستور استاد و دانشجو می‌تواند خلاصه پست‌هایی که در یک کانال اطلاع‌رسانی گذاشته شده را مشاهده کنند.

در خط اول خروجی این دستور مشخصات ارائه درس نوشته شده که شامل شناسه، اسم، ظرفیت، نام استاد، زمان ارائه، تاریخ امتحان و شماره کلاس است و بعد از آن خلاصه پست‌های کانال که شامل شناسه پست، اسم نویسنده و موضوع پست است به ترتیب زمان نمایش داده می‌شود (پست جدیدتر بالاتر نمایش داده می‌شود).

- اگر ارائه درسی با شناسه وارد شده وجود نداشت با خطای Not Found مواجه می‌شویم.
- اگر ارائه درسی با شناسه وارد شده در بین دروسی که استاد ارائه می‌کند یا دانشجو دستیار آموزشی آن‌ها است یا در آن شرکت دارد وجود نداشت با خطای Permission Denied مواجه می‌شویم.

ورودی

```
GET course_channel ? id <id>
```

خروجی

```
<id> <name> <capacity> <professor> <time> <exam_date>
<class_number>
<post_id> <author_name> "<title>"
<post_id> <author_name> "<title>" ... |
OK | Bad Request | Not Found | Permission Denied
```

نمونه ورودی

```
GET course_channel ? id 7
```

نمونه خروجی

```
7 Advanced Programming 90 Ramtin Khosravi Tuesday:10-12 1403/4/2 1
4 Shahn timer "Phase2 will not be extended"
3 Nesa "Phase2 uploaded"
2 Elahe "Phase1 extended"
1 Erfan "Phase1 uploaded"
```

دستور مشاهده پست از کانال اطلاع‌رسانی

با این دستور استاد و دانشجو می‌تواند محتوای پستی که شناسه آن عنوان شده را به طور کامل مشاهده کند. ابتدا مشخصات ارائه درس ذکر شده و سپس جزئیات پست که شامل شناسه، نام نویسنده، موضوع و متن پست است نمایش داده می‌شود.

- اگر ارائه درسی با شناسه وارد شده وجود نداشت با خطای Not Found مواجه می‌شویم.

- اگر در ارائه درس ذکر شده پستی با شناسه وارد شده وجود نداشت با خطای Not Found مواجه می‌شویم.
- اگر ارائه درسی با شناسه وارد شده در بین دروسی که استاد ارائه می‌کند یا دانشجو دستیار آموزشی آن‌ها است یا در آن شرکت دارد وجود نداشت با خطای Permission Denied مواجه می‌شویم.

ورودی

```
GET course_post ? id <id> post_id <post_id>
```

خروجی

```
<id> <name> <capacity> <professor> <time> <exam_date>
<class_number>
<post_id> <author_name> "<title>" "<message>" |
OK | Bad Request | Not Found | Permission Denied
```

نمونه ورودی

```
GET course_post ? id 7 post_id 2
```

نمونه خروجی

```
7 Advanced Programming 90 Ramtin Khosravi Tuesday:10-12 1403/4/2 1
2 Elahe "Phase1 extended" "Phase one was extended for three days"
```

دستورات استاد

دستور ارسال فرم تی‌ای

با این دستور **استاد** برای ارائه درس مشخص فرم درخواست دستیار آموزشی ارسال می‌کند. این فرم برای تمام افراد در سامانه قابل مشاهده است و در صفحه شخصی استاد قرار می‌گیرد. اعلان این پست برای کسانی که **به استاد متصل** هستند نیز ارسال می‌شود.

نمونه اعلان

```
810420432 Kambiz Afrooz: New Form
```

هر استاد فقط برای دروسی که خود ارائه می‌کند می‌تواند فرم دستیار آموزشی ایجاد کند. همچنین هر فرم دستیار آموزشی مانند پست‌ها یک شناسه دریافت می‌کند که این شناسه مانند پست‌های عادی محاسبه می‌شود و باید پست‌های عادی را نیز در شناسه در نظر بگیرید. برای مثال اگر دو پست معمولی منتشر کرده باشیم و بعد یک فرم تی‌ای، دو پست اول شناسه‌های 1 و 2 را دریافت می‌کنند و به فرم شناسه 3 تعلق می‌گیرد.

- اگر شناسه ارائه درس وارد شده در بین دروس ارائه شده نبود با خطای Not Found مواجه می‌شویم.
- اگر شناسه ارائه درس وارد شده در بین دروس ارائه شده استاد نبود با خطای Permission Denied مواجه می‌شویم.

ورودی

```
POST ta_form ? course_id <course_id> message "<message>"
```

خروجی

OK | Bad Request | Not Found | Permission Denied

نمونه ورودی

```
POST ta_form ? course_id 2 message "TA for designing project"
```

نمونه خروجی

OK

خلاصه فرم‌هایی که اساتید منتشر می‌کنند مانند پست‌های معمولی آنها در صفحه شخصی‌شان قابل مشاهده است و همچنین می‌توانیم مانند پست‌های معمولی جزئیات فرم‌ها را نیز با دستور مشاهده پست دریافت کنیم، قالب و مثال برای این نمایش‌ها در زیر آمده است. برای نمایش خلاصه یک فرم تی‌ای در صفحه شخصی استاد ابتدا شناسه فرم و سپس عبارتی که نشان می‌دهد این یک فرم تی‌ای برای درس مورد نظر است را مطابق قالب خروجی زیر چاپ می‌کنید.

ورودی

```
GET personal_page ? id <id>
```

خروجی

```
<name> <major> <position> <courses>
<post_id> "<title>"
<post_id> TA form for <course_name> course ... |
Bad Request | Not Found
```

نمونه ورودی

```
GET personal_page ? id 810420432
```

نمونه خروجی

```
Kambiz Afrooz Computer Engineering Professor
5 "title"
```

2 TA form for Operating System course
1 "title"

حال اگر بخواهیم جزئیات یک فرم را مشاهده کنیم ابتدا شناسه و عبارتی که در بالا مشاهده کردید را چاپ می‌کنیم، در خط بعد مشخصات ارائه درس را چاپ می‌کنیم و در خط بعدی نیز پیامی که استاد روی فرم گذاشته است.

ورودی

```
GET post ? id <id> post_id <post_id>
```

خروجی

```
<name> <major> <position> <courses>  
<post_id> TA form for <course_name> course  
<course_id> <name> <capacity> <professor> <time> <exam_date>  
<class_number>  
"<message>" |  
Bad Request | Not Found
```

نمونه ورودی

```
GET post ? id 810420432 post_id 2
```

نمونه خروجی

```
Kambiz Afrooz Computer Engineering Professor  
2 TA form for Operating System course  
4 Operating System 90 Kambiz Afrooz Tuesday:7-9 1403/3/21 5  
"Teaching assistant for designing lab and course projects"
```

توجه داشته باشید که نمایش پست‌های معمولی همچنان **مشابه با فاز قبل** است و تغییری نمی‌کند.

دستور بستن فرم تی‌ای

با این دستور فرم درخواست دستیار آموزشی موجود در صفحه استاد بسته خواهد شد و پس از آن ابتدا تعداد افراد متقاضی نمایش داده می‌شود و سپس اطلاعات افراد متقاضی به ترتیب نمایش داده می‌شود. این اطلاعات به ترتیب شامل شماره دانشجویی فرد، نام فرد و ترم کنونی دانشجو است. استاد پس از نمایش هر شخص با وارد کردن accept یا reject به درخواست افراد پاسخ دهد. پس از رسیدگی به درخواست متقاضیان یک اعلان برای آن‌ها ارسال می‌شود که نتیجه درخواست آن‌ها را نشان می‌دهد. همچنین فرم بسته شده دیگر در صفحه استاد قابل نمایش نخواهد بود و در صورتی که بخواهیم با دستور GET post هم به آن دسترسی پیدا کنیم با خطای Not Found مواجه می‌شویم.

در اعلان ارسال شده شناسه ارائه درس و نام درس نوشته می‌شود و سپس یکی از دو پیام قبول شدن یا رد شدن.

نمونه اعلان

11 Advanced Programming: Your request to be a teaching assistant has been rejected/accepted.

اگر درخواست فردی برای دستیار آموزشی پذیرفته شود فرد به صفحه درس اضافه خواهد شد و می‌تواند به صفحه درس پست اضافه کند و یا پست‌های آن را مشاهده کند.

- اگر فرم با شناسه داده شده در فرم‌های استاد نبود، با خطای Not Found مواجه می‌شویم.

ورودی

```
POST close_ta_form ? id <id>
```

خروجی

We have received <number_of_requests> requests for the teaching assistant position

<first_person_information>:

<second_person_information>:

...

| Bad Request | Not Found

نمونه ورودی

```
POST close_ta_form ? id 4
```

نمونه خروجی

We have received 13 requests for the teaching assistant position

810102345 Mahboube Taheri 6: accept

810103278 Fatemeh Khodashenas 4: reject

810108181 Arman Rahmani 8: blablabla

810108181 Arman Rahmani 8:

...

توجه داشته باشید که عبارت‌های accept/reject توسط استاد وارد می‌شود و با زدن اینتر اطلاعات فرد بعدی نمایش داده می‌شود. اگر عبارت وارد شده چیزی به غیر از accept/reject بود بعد از زدن enter مانند مثال بالا دوباره اطلاعات شخص قبلی نوشته می‌شود.

دستورات دانشجو

دستور ارسال درخواست تی‌ای

با این دستور فرد می‌تواند برای ارائه درسی درخواست دستیار آموزشی پر کند. برای این کار فرد بایستی شناسه استاد و شناسه فرم را وارد کند و سپس درخواست فرد برای دستیار آموزشی ارسال خواهد شد.

- اگر استاد یا فرم با شناسه وارد شده وجود نداشت با خطای Not found مواجه می‌شویم.
- اگر ترم فعلی دانشجو کوچکتر مساوی با حداقل ترم مورد نیاز برای برداشتن درسی بود که برای آن درخواست تی‌ای شدن ارسال کرده با خطای Permission Denied مواجه می‌شویم.

ورودی

```
POST ta_request ? professor_id <professor_id> form_id <form_id>
```

خروجی

OK | Bad Request | Not Found | Permission Denied

نمونه ورودی

```
POST ta_request? professor_id 810420432 form_id 3
```

نمونه خروجی

OK

نکات و نحوه تحویل

- تمام فایل‌های خود را در قالب یک پرونده‌ی زیپ با نام `A6-<SID>.zip` در صفحه‌ی Elearn دربارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۰۲۰۰۰ است، نام پرونده‌ی شما باید `A7-810102000.zip` باشد.
 - برای مثال، نمونه فایل مورد قبول در زیر آمده است:

`A7-810102000.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 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.
- 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 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 `utms.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 شماسست. 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 انسجام⁵
 - عدم وجود کد تکراری
 - رعایت دندانه‌گذاری⁶
 - عدم استفاده از متغیرهای گلوبال
 - استفاده صحیح از متغیرهای ثابت⁷ 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 و چندفایلی
- دقت کنید که موارد ذکر شده لزوماً کل نمره شما را تشکیل نمی‌دهند و ممکن است با تغییراتی همراه باشند.

⁵ Consistency

⁶ Indentation

⁷ Constant