



مدرس: رامتین خسروی

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

مهلت تحویل: چهارشنبه ۲ 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 پروژه شما باید یک سامانه دانشگاهی را پیاده‌سازی کنید؛ هدف اصلی این سامانه ایجاد درس‌های دانشگاه و نحوه ارائه آنها است. قابلیت اشتراک گذاری پست، اتصال دانشجویان و اساتید به یکدیگر، ارسال اعلان و ثبت‌نام برای دروس ارائه شده و ... از محدود امکاناتی است که در طول این فاز آن را پیاده‌سازی خواهید 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.

## شرح تمرین

سامانه‌ای که باید در این تمرین پیاده‌سازی کنید، ترکیب ساده‌شده‌ای از سامانه جامع آموزش و یک رسانه اجتماعی<sup>۱</sup> است. در این فاز، ۳ نوع کاربر دانشجوی، استاد و کاربر پیش‌فرض سیستم (مدیر سیستم) پیاده‌سازی می‌شود. اساتید می‌توانند دروسی را ارائه کنند و دانشجویان در صورت دارا بودن شرایط، می‌توانند

<sup>۱</sup> Social Media

این درس‌ها را اخذ کنند. مواردی نظیر ثبت یک ارائه درس توسط مدیر سیستم به سامانه اضافه می‌شود. از طرفی در بخش رسانه اجتماعی، کاربران می‌توانند در سامانه پست ارسال کنند که این پست توسط سایر کاربران قابل مشاهده است. همچنین، در این سامانه امکان اتصال کاربران به یکدیگر نیز وجود دارد و با این روش کاربران می‌توانند همدیگر را دنبال کنند. در صورتی که یک کاربر، کاربر دیگری را دنبال کند، بابت برخی رخدادهایی که توسط کاربر دنبال‌شونده انجام می‌شود، سیستم به صورت خودکار به کاربر دنبال‌کننده اعلان‌هایی در خصوص این رخدادهای ارسال می‌کند.

در ابتدای اجرای برنامه، شما باید یک سری اطلاعات را از روی مجموعه داده<sup>2</sup>هایی که در قالب CSV<sup>3</sup> 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 &lt;/path/to/majors/file&gt; &lt;/path/to/students/file&gt; &lt;/path/to/courses/file&gt; &lt;/path/to/professors/file&gt;</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>

---

<sup>2</sup> Dataset

<sup>3</sup> Comma-Separated Values

## 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.

نمونه فایل دانشجویان
<code>sid,name,major_id,semester,password</code> <code>810102612,Elahe,1,2,ImtheproblemItsme</code> <code>810102342,Nesa,2,4,meoow</code>

## 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
<code>cid,name,credit,prerequisite,majors_id</code> <code>1,Advanced Programming,3,2,1;2</code> <code>2,Operating System,3,5,1</code>

## 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
<code>pid,name,major_id,position,password</code>

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 فرمان<sup>4</sup>، 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، شناسه<sup>5</sup> 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.

<sup>4</sup> command line

<sup>5</sup> ID

## 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).

### 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 وارد شده، 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 دستوری 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 ورود

- هر کاربری که قبلا در سامانه ایجاد شده است، پیش از استفاده از امکانات سامانه باید وارد سیستم شود.
- اگر شماره دانشجویی و یا شناسه استاد در سامانه وجود نداشته باشد، با خطای Not Found مواجه می‌شویم.
- اگر کاربر رمز خود را اشتباه وارد کند در حالی که شماره دانشجویی و یا شناسه استاد وجود داشته باشد، با خطای Permission Denied مواجه می‌شویم.
- در صورتی که کاربر وارد سامانه نشده باشد و هر یک از دستورهای بخش‌های بعد را وارد کند، با خطای Permission Denied مواجه می‌شویم.
- همچنین اگر کاربری از قبل وارد سیستم شده بود با وارد کردن این دستور خطای Permission Denied دریافت می‌کند (برای استفاده از این دستور حتما باید از قبل logout کرده باشیم).

#### ورودی

```
POST login ? id <id> password <password>
```

#### خروجی

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

#### نمونه ورودی اول

```
POST login ? id 810102612 password meoow
```

#### نمونه خروجی اول

```
Permission Denied
```

#### نمونه ورودی دوم

```
POST login ? password ImtheproblemItsme id 810102612
```

#### نمونه خروجی دوم

```
OK
```

## دستور خروج

شخصی که قبلاً در سیستم وارد شده بود با وارد کردن این دستور از سیستم خارج می‌شود. پس از آن می‌تواند دوباره با دستور login به همین حساب کاربری یا یک حساب کاربری دیگر وارد شود.

- اگر پیش از دستور login، دستور logout را وارد کنیم، با خطای Permission Denied مواجه می‌شویم.

ورودی
POST logout ?
خروجی
OK   Permission Denied
نمونه ورودی
POST logout ?
نمونه خروجی
OK

## دستور نمایش دروس ارائه شده

با استفاده از این دستور کاربر (دانشجو یا استاد) می‌تواند لیست تمام دروس ارائه شده در دانشگاه را مشاهده کند. همچنین کاربر می‌تواند با وارد کردن شناسه ارائه درس دلخواه اطلاعات کامل ارائه را مشاهده کند.

پس از وارد کردن این دستور بدون وارد کردن آرگومان، لیست همه دروس ارائه شده به همراه اطلاعات ارائه درس شامل شناسه ارائه درس، نام درس، ظرفیت و استاد به ترتیب شناسه از کوچک به بزرگ نمایش داده می‌شود. برای دریافت اطلاعات کامل درس ارائه شده کاربر بایستی پس از وارد کردن آرگومان id، شناسه ارائه درس را وارد کند.

- در صورتی که شناسه وارد شده عدد طبیعی نبود با خطای Bad Request مواجه می‌شویم.
- در صورتی که شناسه وارد شده در بین دروس ارائه شده وجود نداشت با خطای Not Found مواجه می‌شویم.
- اگر دستور را بدون آرگومان اجرا کنیم و هیچ درسی ارائه نشده باشد Empty دریافت می‌کنیم.

ورودی
GET courses ? id <id>



### خروجی

```
<id> <name> <capacity> <professor> |  
<id> <name> <capacity> <professor> <time> <exam_date>  
<class_number> |  
Bad Request | Not Found | Empty
```

### نمونه ورودی اول

```
GET courses ? id 7
```

### نمونه خروجی اول

```
7 Advanced Programming 110 Ramtin Khosravi Tuesday:10-12 1403/4/2  
1
```

### نمونه ورودی دوم

```
GET courses ?
```

### نمونه خروجی دوم

```
2 Advanced Engineering Mathematics 50 Kambiz Afrooz  
5 Operating System 90 Mohammad Mohammadi  
7 Advanced Programming 110 Ramtin Khosravi
```

## 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 (اعلان‌ها و کانکت شدن با افراد دیگر در ادامه توضیح داده شده است).

- شناسه هر پست برای پست‌های هر کاربر محاسبه می‌شود و از یک شروع می‌شود. با اضافه شدن هر پست مقدار شناسه یکی زیاد می‌شود.
- در این دستور متن‌ها داخل double quotation (") قرار می‌گیرند.

### ورودی

```
POST post ? title <title> message <message>
```

### خروجی

```
OK | Bad Request
```

### نمونه ورودی

```
POST post ? title "Good Day" message "Ye rooz joonane dige"
```

## نمونه خروجی

OK

## دستور حذف پست

- با این دستور هر کاربر می‌تواند با وارد کردن شناسه پست، آن پست را از صفحه شخصی خود حذف کند. با حذف پست، هیچ اعلانی برای کسانی که با کاربر کانکت هستند ارسال نمی‌شود.
- با حذف پست، شناسه آن پست از بین می‌رود و دیگر در صفحه کاربر نمایش داده نمی‌شود.
  - حذف پست تغییری در روند تولید کردن شناسه برای پست‌های بعدی ایجاد نمی‌کند و بدون توجه به آن شناسه‌های بعدی را تولید کنید.
  - در صورتی که شناسه وارد شده در بین پست‌های کاربر موجود نبود، با خطای Not Found مواجه می‌شویم.

## ورودی

```
DELETE post ? id <id>
```

## خروجی

OK | Bad Request | Not Found

## نمونه ورودی

```
DELETE post ? id 3
```

## نمونه خروجی

OK

## دستور مشاهده صفحه شخصی کاربران

- با این دستور هر کاربر می‌تواند با وارد کردن شناسه کاربران صفحه شخصی آن‌ها از جمله خود را ببیند. برای مشاهده صفحه شخصی افراد نیازی به کانکت شدن با آن‌ها نیست. با این دستور اطلاعات مالک صفحه و شناسه و عنوان پست‌های او نمایش داده می‌شود (براساس زمان مرتب می‌شوند و پست جدیدتر بالاتر نمایش داده می‌شود).
- اگر صفحه برای یک دانشجو بود:
  - نام، رشته تحصیلی، ترم فعلی و دروس شرکت‌شده توسط دانشجو نمایش داده می‌شود.
  - اگر صفحه برای یک استاد بود:
  - نام، درجه، درس‌های ارائه شده توسط استاد نمایش داده می‌شود.
  - اگر صفحه برای کاربر پیش‌فرض سیستم بود:
  - فقط عبارت UT\_account نمایش داده می‌شود (کاربر پیش‌فرض جلوتر توضیح داده شده است).

در صورت وجود چند درس برای کاربران آن‌ها را با کاراکتر کاما (,) از هم جدا می‌کنیم. پس از نمایش اطلاعات کاربر، پست‌های مربوطه نمایش داده می‌شوند.

- در صورتی که شناسه وارد شده عدد حسابی نبود با خطای Bad Request مواجه می‌شویم.
- در صورتی که شناسه وارد شده در بین کاربران نبود با خطای Not Found مواجه می‌شویم.

#### ورودی

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

#### خروجی

```
<name> <major> <semester> <courses>
<post_id> "<title>"
<post_id> "<title>" ... |
<name> <major> <position> <courses>
<post_id> "<title>"
<post_id> "<title>" ... |
UT_account
<post_id> "<title>"
<post_id> "<title>" ... |
Bad Request | Not Found
```

#### نمونه ورودی 1

```
GET personal_page ? id 810102612
```

#### نمونه خروجی 1

```
Elahe Computer Engineering 2 Advanced Programming, Operating System
5 "title"
2 "title"
1 "title"
```

#### نمونه ورودی 2

```
GET personal_page ? id 810420432
```

#### نمونه خروجی 2

```
Kambiz Afrooz Computer Engineering Professor
```

در نمونه خروجی 2 استاد مورد نظر درسی ارائه نمی‌دهد و هنوز پستی منتشر نکرده است. خروجی در شرایط مشابه برای دانشجویان نیز به همین صورت خواهد بود.

## دستور مشاهده پست‌ها

با این دستور هر کاربر می‌تواند محتوای پستی که شناسه آن عنوان شده را به طور کامل مشاهده کند. ابتدا مشخصات صفحه شخصی کاربر ذکر شده و سپس جزئیات پست نمایش داده می‌شود.

- اگر شخصی با آیدی ذکر شده در سامانه وجود نداشت با خطای Not Found مواجه می‌شویم.
- اگر شخص پستی با شناسه عنوان شده نداشت با خطای Not Found مواجه می‌شویم.
- اگر شناسه‌های وارد شده عدد طبیعی نبودند با خطای Bad Request مواجه می‌شویم.

### ورودی

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

### خروجی

```
<name> <major> <semester> <courses>
<post_id> "<title>" "<message>" |
<name> <major> <position> <courses>
<post_id> "<title>" "<message>" |
UT_account
<post_id> "<title>" "<message>" |
Bad Request | Not Found
```

### نمونه ورودی

```
GET post ? id 810102612 post_id 5
```

### نمونه خروجی

```
Elahe Computer Engineering 2 Advanced Programming,Operating System
5 "title" "message"
```

## دستور کانکت شدن

بعد از وارد کردن این دستور دو کاربر (کاربر فعلی و کاربری که شناسه آن وارد شده) به هم متصل می‌شوند و اعلان‌های مربوط به فعالیت‌های جدید یکدیگر را دریافت خواهند کرد.

- در صورتی که شناسه وارد شده عدد طبیعی نبود با خطای Bad Request مواجه می‌شویم.
- اگر دو کاربر از قبل با یکدیگر کانکت بودند با خطای Bad Request مواجه می‌شویم.
- در صورتی که شناسه وارد شده در بین کاربران نبود با خطای Not Found مواجه می‌شویم.

### ورودی

POST connect ? id <id>

خروجی

OK | Bad Request | Not Found

نمونه ورودی

POST connect ? id 810102612

نمونه خروجی

OK

## دستور مشاهده اعلان‌ها

بعد از وارد کردن این دستور لیست اعلان‌های ارسال شده در صورت وجود برای کاربر نشان داده می‌شود. برای هر اعلان شناسه و نام ارسال‌کننده به همراه متن اعلان نمایش داده خواهد شد. پس از وارد کردن این دستور اعلان‌های نمایش داده شده از تاریخچه کاربر حذف خواهد شد و بار دیگر که این دستور را وارد کند اعلان‌های تکراری نمایش داده نخواهند شد.

ورودی

GET notification ?

خروجی

<id> <name>: New Post |  
<id> <name>: New Course Offering |  
<id> <name>: Get Course |  
<id> <name>: Delete Course |  
Empty

نمونه ورودی

GET notification ?

نمونه خروجی

810232612 Erfan: Get Course  
810102612 Elahe: New Post  
810100992 Shahnam: Delete Course  
810102342 Nesa: New Post  
810420432 Kambiz Afrooz: New Course Offering

## دستورات کاربر پیش فرض دانشگاه

در این سیستم یک اکانت پیش فرض با شناسه صفر و رمز ورود "UT\_account" وجود دارد که دستورات خاصی را می تواند اجرا کند.

### دستور ورود و خروج

همانند ورود و خروج دیگر کاربران است، صرفا شناسه و رمز ورود مختص به این حساب وارد می شود.

ورودی
<code>POST login ? id 0 password UT_account</code>
خروجی
OK

### دستور ارسال پست

کاملا مشابه با ارسال و حذف پست برای کاربران معمولی است، با این تفاوت که اعلان آن برای تمامی اکانت های سیستم ارسال می شود.

ورودی
<code>POST post ? title &lt;title&gt; message &lt;message&gt;</code>
خروجی
OK   Bad Request
نمونه ورودی
<code>POST post ? title &lt;title&gt; message &lt;message&gt;</code>
نمونه خروجی
OK

### دستور ارائه درس

با استفاده از این دستور، کاربر پیش فرض می تواند یک ارائه درس جدید را ایجاد کند و اعلان ایجاد ارائه درس برای تمامی کاربران سیستم ارسال می شود.

- هر ارائه درس با نام درس ارائه شده، نام استاد ارائه‌دهنده، ظرفیت شرکت‌کنندگان، زمان برگزاری، تاریخ امتحان و کلاس برگزاری تعریف می‌شود.
- هر ارائه درس با استفاده از یک شماره یکتا شناسایی می‌شود.
- در هنگام ایجاد هر ارائه درس یک شناسه یکتا به این ارائه درس اختصاص داده می‌شود. این شناسه‌ها از یک شروع شده و به ازای هر ارائه درس جدیدی که در سامانه ایجاد می‌شود، یک واحد افزایش می‌یابد. یعنی اولین ارائه درس ایجاد شده دارای شناسه یک، دومین ارائه درس ایجاد شده دارای شناسه دو و ... است.
- در صورتی که ایجاد ارائه درس موفقیت‌آمیز باشد، اعلان ارائه درس ایجاد شده برای تمامی کاربران سیستم ارسال می‌شود، شناسه و نام در اعلان ارسال شده شناسه و نام استاد ارائه دهنده خواهد بود.
- اگر شناسه‌ها، شماره کلاس و ظرفیت شرکت‌کنندگان عدد طبیعی نباشد، با خطای Bad Request مواجه می‌شویم.
- اگر درس یا کاربر مورد نظر در سیستم وجود نداشتند، با خطای Not Found مواجه می‌شویم.
- اگر کاربری که شناسه آن را وارد کردیم استاد نبود، با خطای Permission Denied مواجه می‌شویم.
- اگر رشته استاد وارد شده در لیست رشته‌هایی که مجاز به اخذ و ارائه درس هستند نبود، با خطای Permission Denied مواجه می‌شویم.
- اگر زمان ارائه درس جدید با یکی از دروسی که استاد در حال حاضر ارائه می‌دهد تداخل داشت، با خطای Permission Denied مواجه می‌شویم.
- تضمین می‌شود که فرمت زمان برگزاری و همچنین تاریخ امتحان درست باشد و نیاز به چک کردن این دو مورد نخواهید داشت.

#### ورودی

```
POST course_offer ? course_id <course_id> professor_id
<professor_id> capacity <capacity> time <time> exam_date
<exam_date> class_number <class_number>
```

#### خروجی

OK | Bad Request | Permission Denied | Not Found

#### نمونه ورودی

```
POST course_offer ? course_id 1 professor_id 810420432 capacity 70
time Sunday:13-15 exam_date 1403/4/4 class_number 2
```

#### نمونه خروجی

OK

## فرمت ورودی زمان برگزاری

- می‌تواند از شنبه تا چهارشنبه و از ساعت 1 تا 24 باشد.
- در ابتدای ساعات تکریمی عدد 0 نمی‌آید.
- ساعت شروع همواره از ساعت اتمام کمتر است. برای مثال کلاسی که ساعت 22 شروع شود و 3 تمام شود یا 15 شروع شود و 15 تمام شود نخواهیم داشت.
- دو کلاسی که به صورت متوالی برگزار می‌شوند تداخل ندارند. برای مثال 12 تا 13 و 13 تا 14 تداخل ندارند.

### قالب کلی

<week\_day>:<start\_time>-<end\_time>

### نمونه

Saturday:14-16  
Sunday:9-12  
Monday:10-17  
Tuesday:7-20  
Wednesday:9-24

## فرمت ورودی تاریخ امتحان

با سال، ماه و روز مشخص می‌شود.

### قالب کلی

<year>/<month>/<day>

### نمونه

1403/3/27

## دستورات دانشجو

### دستور ثبت‌نام برای درس ارائه‌شده

- با استفاده از این دستور دانشجو می‌تواند با توجه به لیست دروس ارائه شده، با وارد کردن شناسه یکتا درس ارائه شده مورد نظر، برای آن ثبت‌نام کند. پس از اینکه ثبت‌نام قطعی شد، درس به دروس ثبت‌نام شده دانشجو اضافه می‌شود و برای افرادی که با آن connect هستند اعلان ارسال می‌شود.
- در صورتی که شناسه وارد شده عدد طبیعی نبود با خطای Bad Request مواجه می‌شویم.



- در صورتی که درس وارد شده جزو درس‌های ارائه شده نباشد، با خطای Not Found مواجه می‌شویم.
- اگر ترم دانشجو از حداقل ترم مورد نیاز برای اخذ درس کمتر باشد، با خطای Permission Denied مواجه می‌شویم.
- در صورتی که رشته تحصیل دانشجو جزو رشته‌های مجاز ارائه و اخذ درس نباشد، با خطای Permission Denied مواجه می‌شویم.
- اگر زمان ارائه درس یا تاریخ برگزاری امتحان درس با دیگر درس‌هایی که دانشجو از قبل آنها را اخذ کرده تداخل داشته باشد با خطای Permission Denied مواجه می‌شویم.

ورودی
<code>PUT my_courses ? id &lt;id&gt;</code>
خروجی
<code>OK   Not Found   Bad Request   Permission Denied</code>
نمونه ورودی
<code>PUT my_courses ? id 6</code>
نمونه خروجی
<code>OK</code>

## دستور حذف درس برداشته شده

با اجرای این دستور درس مدنظر از لیست درس‌های اخذ شده توسط دانشجو حذف می‌شود و برای افرادی که با آن connect هستند اعلان ارسال می‌شود.

- در صورتی که شناسه وارد شده عدد طبیعی نبود با خطای Bad Request مواجه می‌شویم.
- در صورتی که درس وارد شده جزو درس‌هایی که دانشجو آن را اخذ کرده نباشد، با خطای Not Found مواجه می‌شویم.

ورودی
<code>DELETE my_courses ? id &lt;id&gt;</code>
خروجی
<code>OK   Not Found   Bad Request</code>
نمونه ورودی

```
DELETE my_courses ? id 6
```

نمونه خروجی

OK

## دستور مشاهده درس‌های برداشته‌شده

این دستور درس‌هایی را که دانشجو آنها را اخذ کرده است به او نشان می‌دهد.

- در صورتی که هیچ درسی برداشته نشده بود خطای Empty چاپ شود.

ورودی

```
GET my_courses ?
```

خروجی

```
<id> <name> <capacity> <professor> <time> <exam_date>  
<class_number> ... |  
Empty
```

نمونه ورودی

```
GET my_courses ?
```

نمونه خروجی

```
2 Advanced Engineering Mathematics 50 Kambiz Afrooz Sunday:9-11  
1403/4/5 5
```

# نکات و نحوه تحویل

- تمام فایل‌های خود را در قالب یک پرونده‌ی زیپ با نام A6-<SID>.zip در صفحه‌ی Elearn درس بارگذاری 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 انسجام<sup>6</sup>
    - عدم وجود کد تکراری
    - رعایت دندانه‌گذاری<sup>7</sup>
    - عدم استفاده از متغیرهای گلوبال
    - استفاده صحیح از متغیرهای ثابت<sup>8</sup> 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.

---

<sup>6</sup> Consistency

<sup>7</sup> Indentation

<sup>8</sup> Constant