

دانشگاه صنعتی شریف



دانشکده مهندسی کامپیوتر

## پروژه درس مبانی برنامه‌سازی پایتون

گروه ۷ - پاییز ۱۴۰۳

عنوان پروژه: سامانه آموزشی (LMS)

استاد: علی ابریشمی

طراحان پروژه:

مانی ابراهیمی  
نیما پشبادی  
محمدامین حیدری  
سید امیرمحمد جزایری

تاریخ تحویل: ۱۱ بهمن ۱۴۰۳

## فهرست مطالب

۳	۱ معرفی پروژه
۳	۱.۱ شرح پروژه
۳	۱.۲ انتظارات و اهداف
۳	۱.۳ قوانین مربوط به پروژه
۴	۲ بخش اجباری (۲۰۰۰ نمره)
۴	۲.۱ پیاده‌سازی ساختارهای داده‌ای برای مدل‌ها با استفاده از لغت‌نامه (۲۰۰ نمره)
۴	۲.۱.۱ مدل دانشجو (Student)
۴	۲.۱.۲ مدل استاد (Professor)
۴	۲.۱.۳ مدل درس (Course)
۴	۲.۱.۴ شکل نهایی کد شما
۵	۲.۲ ایجاد فرآیند ثبت‌نام و ورود (۲۰۰ نمره)
۵	۲.۳ پیاده‌سازی فرآیند ایجاد کلاس و ثبت‌نام دانشجو (۲۰۰ نمره)
۵	۲.۴ پیاده‌سازی امکان تعیین بارم‌بندی درس توسط استاد (۱۵۰ نمره)
۵	۲.۵ پیاده‌سازی امکان ثبت نمره دانشجو (۱۵۰ نمره)
۵	۲.۶ ذخیره‌ی اطلاعات کاربران (دانشجویان و اساتید) و کلاس‌ها روی فایل (۳۰۰ نمره)
۵	۲.۷ ساخت شیت لیست کلاسی با Pandas (۲۵۰ نمره)
۶	۲.۸ استخراج شیت کلاسی با فرمت CSV و XLSX (۲۰۰ نمره)
۶	۲.۹ محاسبه نمره و اعمال نمودار روی نمره (۱۵۰ نمره)
۶	۲.۱۰ رسم نمودار پیشرفت دانشجو (۲۰۰ نمره)
۷	۳ بخش اختیاری (۱۰۰۰ نمره)
۷	۳.۱ استفاده از رویکرد شی‌گرا (۱۵۰ نمره)
۷	۳.۱.۱ ساختار کلاس Student
۷	۳.۱.۲ ساختار کلاس Professor
۷	۳.۱.۳ ساختار کلاس Course
۸	۳.۲ پیاده‌سازی امکان عدم نیاز به ورود مجدد پس از ورود دوباره به برنامه (۱۰۰ نمره)
۸	۳.۳ ایجاد تست (Unit Test) (۵۰ نمره)
۸	۳.۴ استفاده از Git (۲۰۰ نمره)
۹	۳.۵ ایجاد رابط کاربری گرافیکی (۴۰۰ نمره)
۹	۳.۵.۱ ویژگی‌ها پنجره‌های اصلی
۹	۳.۶ انتشار روی GitHub (۱۰۰ نمره)
۱۱	۴ ضمیمه: پیشنهادهای ما

## معرفی پروژه

### ۱.۱ شرح پروژه

در این پروژه قرار است شما یک LMS یا سامانه مدیریت یادگیری بسازید، یعنی سامانه‌ای مشابه با CW شریف یا Courses امیرکبیر که خوب است بدانید برپایه Moodle<sup>۱</sup> ساخته شده‌اند. در این راستا، شما با رویکرد شی‌گرا مدل‌های سامانه را تعریف خواهید کرد و با ایجاد ارتباط منطقی میان آن‌ها یک سامانه‌ی قابل اجرا خواهید داشت. همچنین در بخش‌هایی از این پروژه از شما خواسته خواهد شد تا با استفاده از Pandas و Numpy داده‌های کلاسی را پردازش کرده و یک خروجی خوانا برای کاربران سامانه ایجاد کنید.

### ۱.۲ انتظارات و اهداف

انتظار می‌رود در انتهای این پروژه:

- رویکرد شی‌گرا را به خوبی آموخته و بتوانید با آن یک پروژه‌ی قابل اجرا بسازید.
  - بتوانید با استفاده از Pandas و Numpy داده‌ها را پردازش کرده و یک خروجی خوانا برای کاربران سامانه ایجاد کنید.
  - تا حد بسیار مبتدی با مفاهیم ذخیره‌سازی داده‌ها و پردازش آن‌ها آشنا شوید.
- و همچنین در صورت انجام بخش اختیاری هم از شما انتظار می‌رود:
- بتوانید با Git تغییرات یک پروژه را مدیریت کنید.
  - با گیت‌هاب آشنا شوید و بتوانید با آن یک پروژه را مدیریت کنید.
  - با مفهوم GUI آشنا شوید و بتوانید با استفاده از Tkinter یا PyQt یک پنجره‌ی GUI بسازید.
  - با اهمیت آزمایش آشنا شده و بتوانید برای یک پروژه Unit Test بنویسید.

### ۱.۳ قوانین مربوط به پروژه

- در این پروژه شما مجاز به استفاده از مدل‌های بزرگ زبانی<sup>۲</sup> مانند ChatGPT یا Claude هستید؛ به شرط اینکه هر دو عضو گروه تسلط کامل بر کد پروژه داشته باشند و بتوانند حین تحویل آن را به خوبی توضیح دهند.
- مستند پروژه را یک بار تا انتها بخوانید، تعدادی از موارد امتیازی را (در صورتی که تمایل به انجام آن‌ها دارید) می‌بایست از همان ابتدای شروع پروژه انجام دهید و در نظر داشته باشید.
- تمام فایل‌های پروژه‌تان را در یک فایل فشرده با فرمت zip، روی کونترا بارگذاری کنید.
- نام فایلی که بارگذاری می‌شود باید به فرمت {STDID2}\_{STDID1}\_FOP\_PROJ باشد که STDID1 شماره دانشجویی عضو اول گروه و STDID2 شماره دانشجویی عضو دوم گروه است. برای مثال اگر یک گروه داشته باشیم که شماره دانشجویی اعضایش 403108123 و 403108987 باشد، باید فایلی به نام FOP\_PROJ\_403108123\_403108987.zip آپلود کنند.
- آپلود هرگونه فایل با نامگذاری خارج از این چارچوب موجب کسر نمره خواهد شد.
- هر دو عضو گروه می‌بایست فایل مربوطه را بارگذاری نمایند. مسئولیت عدم بارگذاری یا بارگذاری فایل‌های متفاوت توسط اعضای یک گروه متوجه خود ایشان است.
- از آنجا که تحویل پروژه اجباری است، مطمئن شوید هر دو عضو گروه کاملاً بر پروژه مسلط باشند. همچنین هر دو عضو موظف به فعالیت هستند و در صورتی که یکی از اعضا فعالیت نکند، تمامی اعضا به عنوان یک گروه دچار کسر نمره خواهد شد.
- پروژه تاخیر نخواهد داشت و زمان پایان اعلام شده، نهایی (هارد ددلاین) خواهد بود.
- اطمینان حاصل شده است که پروژه‌ی شما با تمامی مطالبی که در کلاس آموخته‌اید قابل انجام باشد. پس پیش از انجام پروژه، همه‌ی مطالبی که در کلاس آموخته‌اید را به خوبی مرور بفرمایید.

<sup>۱</sup><https://moodle.org/>  
LLM<sup>۲</sup>

## بخش اجباری (۲۰۰۰ نمره)

### توجه!

جمع نمرات این بخش ۲۰۰۰ نمره است که معادل ۲ نمره از کل درس می‌باشد. انجام موارد ذکر شده در این بخش اجباری است.

### ۲.۱ پیاده‌سازی ساختارهای داده‌ای برای مدل‌ها با استفاده از لغت‌نامه (۲۰۰ نمره)

در این بخش شما موظفید تا ساختارهای داده‌ای مورد نیاز برای مدل‌ها را با استفاده از لغت‌نامه<sup>۱</sup> پیاده‌سازی کنید. این بخش را دقیقاً می‌توانید مانند سوال چهارم تمرین ۳ یا سوال آخر تمرین ۴ پیاده کنید؛ به این صورت که مشخصات هر مدل را به صورت یک کلید<sup>۲</sup> قرار داده و مقداردهی آن را از طریق مقدار<sup>۳</sup> انجام دهید. همچنین می‌توانید به منظور ذخیره‌سازی تمام این لغت‌نامه‌ها از یک لیست<sup>۴</sup> استفاده کنید.

#### ۲.۱.۱ مدل دانشجو (Student)

مشخصات یک مدل دانشجو به شکل زیر است:

کلید	نوع داده
id	int
name	str
email	str
password	str
phone	str

#### ۲.۱.۲ مدل استاد (Professor)

مشخصات یک مدل استاد به شکل زیر است:

کلید	نوع داده
id	int
name	str
email	str
password	str
phone	str

#### ۲.۱.۳ مدل درس (Course)

مدل هر درس به صورت زیر خواهد بود:

#### ۲.۱.۴ شکل نهایی کد شما

بخشی از کدتان که در آن داده‌ساختارها را ذخیره کرده‌اید، به شکل زیر خواهد بود:

dictionary<sup>۱</sup>  
key<sup>۲</sup>  
value<sup>۳</sup>  
list<sup>۴</sup>

کلید	نوع داده
id	int
name	str
description	str
professor	int
students	list[int]

Code

```
STUDENTS = []
TEACHERS = []
COURSES = []
```

## ۲.۲ ایجاد فرآیند ثبت نام و ورود (۲۰۰ نمره)

## ۲.۳ پیاده سازی فرآیند ایجاد کلاس و ثبت نام دانشجو (۲۰۰ نمره)

پیاده سازی فرآیند ایجاد کلاس و ثبت نام دانشجو در این بخش از سیستم، شما وظیفه دارید که کلاس ها را ایجاد و مدیریت کنید. هر کلاس درس اطلاعاتی مانند نام کلاس، مدرس، ظرفیت، و زمان شروع را ذخیره می کند و به سایر ماژول های سیستم مانند ماژول ثبت نام دانشجویان و مدیریت محتوا متصل می شود. همچنین هر دانشجو می تواند با اضافه کردن آیدی کلاس، به آن اضافه شود.

ویژگی ها

۱. ایجاد کلاس روم جدید:

- ایجاد آیدی منحصر به فرد برای کلاس
- تعریف نام کلاس.
- اختصاص مدرس.
- تعیین ظرفیت کلاس.
- تنظیم زمان های کلاس.

۲. ویرایش اطلاعات کلاس درس:

- امکان تغییر نام، مدرس، ظرفیت، یا زمان های کلاس.

۳. حذف کلاس درس:

- حذف کلاس درس از سیستم، با بررسی وابستگی ها (مانند دانشجویان ثبت نام شده).

۴. نمایش لیست کلاس درس ها:

- بازبازی و نمایش همه کلاس درس ها برای مدیریت بهتر.

## ۲.۴ پیاده سازی امکان تعیین بارم بندی درس توسط استاد (۱۵۰ نمره)

شما باید بتوانید شرایطی را مهیا کنید که در آن استاد بتواند بارم بندی درس که شامل ریز نمرات درس به صورت یک نمره مشخص از یک تا بیست و بخشی از درس که نمره به آن تعلق می گیرد را تعیین کند. همچنین این اطلاعات باید توسط دانشجو قابل مشاهده باشد. شایان ذکر است که استاد در زمان های دیگر به غیر از زمانی که بارم بندی اولیه را تنظیم می کند نیز می تواند بارم بندی را تغییر دهد.

نکته: جمع نمرات باید ۲۰ باشد.

## ۲.۵ پیاده سازی امکان ثبت نمره دانشجو (۱۵۰ نمره)

## ۲.۶ ذخیره ی اطلاعات کاربران (دانشجویان و اساتید) و کلاس ها روی فایل (۳۰۰ نمره)

## ۲.۷ ساخت شیت لیست کلاسی با Pandas (۲۵۰ نمره)

۲.۸ ♦ استخراج شیت کلاسی با فرمت CSV و XLSX (۲۰۰ نمره)

---

۲.۹ ♦ محاسبه نمره و اعمال نمودار روی نمره (۱۵۰ نمره)

---

۲.۱۰ ♦ رسم نمودار پیشرفت دانشجو (۲۰۰ نمره)

---

## بخش اختیاری (۱۰۰۰ نمره)

### توجه!

جمع نمرات این بخش ۱۰۰۰ نمره است که معادل ۱ نمره از کل درس می‌باشد. انجام موارد ذکر شده در این بخش اجباری نیست اما نمره‌ی امتیازی بر روی کل درس دارد.

### ۳.۱ استفاده از رویکرد شی‌گرا (۱۵۰ نمره)

در این بخش شما باید مدل‌هایتان را به جای استفاده از لغت‌نامه، با استفاده از رویکرد شی‌گرا ایجاد کنید. سه کلاس `Professor`، `Student` و `Course` را ایجاد کرده و توابع مورد نیاز برای این کلاس‌ها را پیاده‌سازی کنید.

#### ۳.۱.۱ ساختار کلاس `Student`

ساختار کد شما برای کلاس `Student` می‌تواند به شکل زیر باشد:

```
class Student:
    def __init__(self, id, name, email, password, phone):
        # TODO: Implement the Student class
        pass

    # TODO: Implement the other methods
```

#### ۳.۱.۲ ساختار کلاس `Professor`

ساختار کد شما برای کلاس `Professor` می‌تواند به شکل زیر باشد:

```
class Professor:
    def __init__(self, id, name, email, password, phone):
        # TODO: Implement the Professor class
        pass

    # TODO: Implement the other methods
```

#### ۳.۱.۳ ساختار کلاس `Course`

ساختار کد شما برای کلاس `Course` می‌تواند به شکل زیر باشد:

```
Code

class Course:
    def __init__(self, id, name, description, professor):
        # TODO: Implement the Course class
        pass

    # TODO: Implement the other methods
```

توجه شود که در صورت پیاده‌سازی صحیح کلاس‌ها با ساختار شی‌گرا، نیازی به پیاده‌سازی آن‌ها با لغت‌نامه نیست و نمره‌ی آن بخش نیز به صورت کامل به شما تعلق خواهد گرفت.

## ۳.۲ پیاده‌سازی امکان عدم نیاز به ورود مجدد پس از ورود دوباره به برنامه (۱۰۰ نمره)

هرکاربر (اعم از استاد و دانشجو) باید بتواند پس از ورود در صورتی که بخواهد، گزینه‌ای را فعال کند تا دفعه‌ی بعدی که وارد برنامه شد نیاز به ورود نداشته باشد. همچنین توجه شود که این گزینه با خروج کاربر از حساب خودش، غیرفعال خواهد شد و کاربر باید دفعه‌ی بعد حتماً اطلاعات کاربری خود را مجدداً وارد کند.

## ۳.۳ ایجاد تست (Unit Test) (۵۰ نمره)

برای این بخش، شما باید تعداد تست واحد<sup>۱</sup> مناسب برای چند تابع پیاده کنید. برای یادگیری بیشتر در مورد تست‌های واحد به ضمیمه ۲ مراجعه کنید. همچنین شما باید از ماژول `unittest` استفاده کنید. برای کسب نمره‌ی کامل این بخش، شما باید برای حداقل ۳ تابع از توابع کدتان تست واحد بنویسید.

## ۳.۴ استفاده از Git (۲۰۰ نمره)

برای این بخش، شما از ابتدا باید برای پروژه‌تان یک مخزن<sup>۲</sup> Git ایجاد نمایید. برای این کار ابتدا وارد پوشه‌ای می‌شوید که در آن بناس‌ت پروژه را ذخیره کنید و سپس دستور زیر را در ترمینال یا CMD یا PowerShell اجرا کنید:

```
Terminal

C:/Users/username/Desktop/project> git init
```

سپس هربار تغییری در کد می‌دهید می‌توانید به صورت زیر از تغییراتتان ذخیره کنید:

```
Terminal

C:/Users/username/Desktop/project> git add .
C:/Users/username/Desktop/project> git commit -m "commit message"
```

که در آن `message commit` یک متن است که شما می‌توانید برای توضیح تغییراتتان استفاده کنید. برای مطالعه‌ی بیشتر در مورد Git به ضمیمه ۲ مراجعه کنید. همچنین برای کسب نمره کامل از این بخش، مخزن Git شما باید دارای حداقل ۵ تغییر<sup>۳</sup> باشد. می‌توانید برای دیدن سوابق تغییراتتان از دستور زیر استفاده کنید:

Unit Test<sup>۱</sup>  
Repository<sup>۲</sup>  
Commit<sup>۳</sup>



```
Terminal
C:/Users/username/Desktop/project> git log
```

## ۳.۵ ♦ ایجاد رابط کاربری گرافیکی (۴۰۰ نمره)

در ابتدا کاربر باید انتخاب کند که می‌خواهد به عنوان دانشجو یا به عنوان استاد وارد پتل شود. سپس با انتخاب هر کدام پنجره متفاوتی برای او باز می‌شود.

### ۳.۵.۱ ■ ویژگی‌ها پنجره‌های اصلی

۱. پنجره دانشجو:

- لیست کلاس‌های ثبت‌نام شده
- امکان اضافه کردن یک کلاس جدید با وارد کردن آیدی آن کلاس
- مشاهده جزئیات کلاس (مثل نام استاد، ظرفیت، زمان برگزاری)

۲. پنجره استاد:

- ایجاد کلاس
- ویرایش اطلاعات کلاس
- حذف کلاس
- نمایش لیست کلاس‌های ایجاد شده توسط استاد

## ۳.۶ ♦ انتشار روی GitHub (۱۰۰ نمره)

در این بخش، شما وظیفه دارید که پروژه خود را بر روی اکانت گیت‌هاب خود منتشر کنید. برای اینکار، ابتدا نیاز است به وبسایت GitHub بروید و در آنجا اکانت خود را بسازید. پس از ساختن اکانت، وارد آن شوید و مراحل زیر را انجام دهید:

۱. ساختن مخزن

- (آ) روی دکمه New Repository کلیک کنید.
- (ب) نامی برای مخزن انتخاب کنید.
- (ج) مخزن را عمومی یا خصوصی تنظیم کنید.
- (د) روی Create Repository کلیک کنید.

۲. اتصال مخزن محلی به GitHub:

برای اتصال مخزن Git محلی به مخزن ایجادشده در GitHub، دستور زیر را اجرا کنید:

```
Terminal
git remote add origin https://github.com/username/repository.git
```

۳. ارسال تغییرات به GitHub:

برای ارسال تغییرات ثبت‌شده به GitHub، دستورات زیر را اجرا کنید

```
Terminal
git push -u origin main
```

#### ۴. کلون کردن مخزن از GitHub:

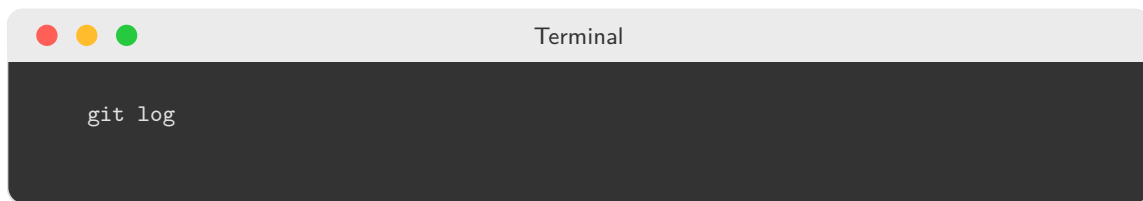
برای دریافت یک مخزن موجود از GitHub روی سیستم خود، دستور زیر را اجرا کنید:

A terminal window with a title bar containing three colored circles (red, yellow, green) and the word "Terminal". The command `git clone https://github.com/username/repository.git` is entered in the terminal.

```
git clone https://github.com/username/repository.git
```

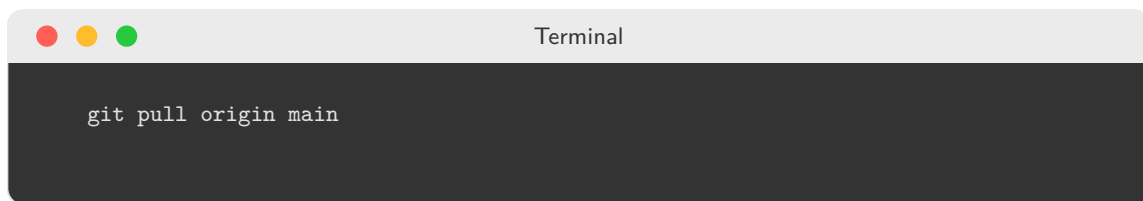
#### ۵. مدیریت تغییرات در GitHub:

برای مشاهده سوابق تغییرات، دستور زیر را اجرا کنید:

A terminal window with a title bar containing three colored circles (red, yellow, green) and the word "Terminal". The command `git log` is entered in the terminal.

```
git log
```

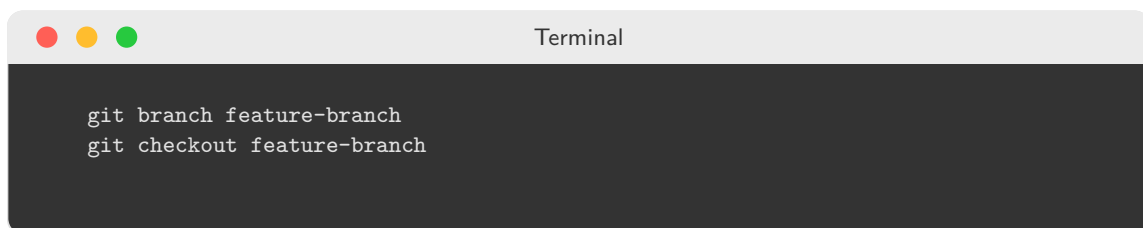
#### ۶. برای همگام‌سازی تغییرات جدید از مخزن GitHub، از دستور زیر استفاده کنید::

A terminal window with a title bar containing three colored circles (red, yellow, green) and the word "Terminal". The command `git pull origin main` is entered in the terminal.

```
git pull origin main
```

#### ۷. ایجاد شاخه جدید برای ویژگی‌ها:

برای توسعه ویژگی‌های جدید بدون تأثیر بر شاخه اصلی:

A terminal window with a title bar containing three colored circles (red, yellow, green) and the word "Terminal". Two commands are entered in the terminal:

```
git branch feature-branch  
git checkout feature-branch
```

## ضمیمه: پیشنهادهای ما