به نام خدا

نام و نام خانوادگی دانشجو:

هومن علی پور

کد دانشجویی:

40211415049

عنوان گزارش:

پروژه برنامه نویسی پیشرفته

نام استاد:

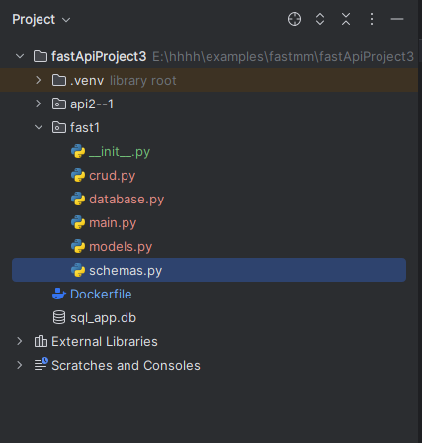
دکتر رشنو

بهار و تابستان 1403

در این پروژه ما فصد داریم که با کمک ابزار ¸ ( fast api) 3 جدول دانشجو و استاد و دروس را بصورت ردیف و جدول در قالب فست ای پی آی ارائه دهیم

برای ساخت این فایل باید در برنامه ایتور خودمان یک دایرکتوری جهت آدرس دهی برای

برنامه گیت هاب ایجاد کنیم .



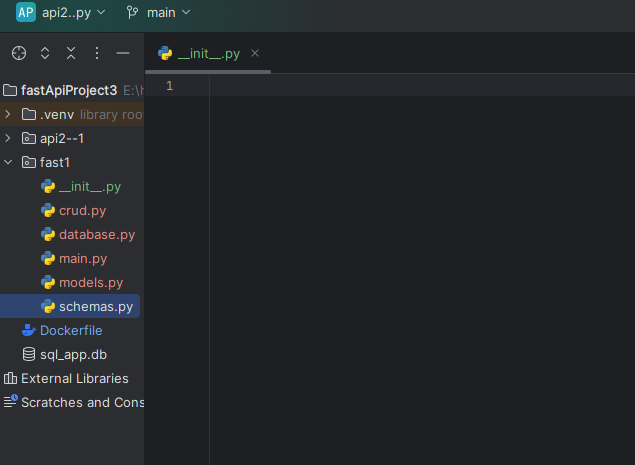
بعد 6 فایل پایتونی به ترتیب زیر ایجاد می منیم که باید فایل اولی خالی باشد تا ادیتور این مجموعه را یک پکیج که یک کار خاصی انجام میدهد تشخیص دهد :

1. 1.\_\_init\_\_.py
2. 2.crud.py
3. 3.database.py
4. 4.main.py
5. 5.models.py
6. 6.schemas.py

وحالا کار هر یک را توضیح میدهیم ..

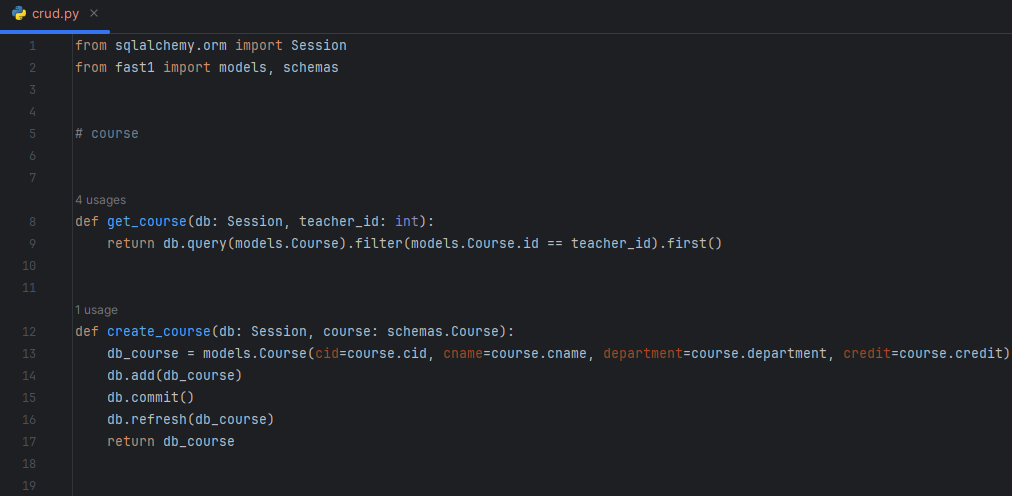
* **فایل in it:**

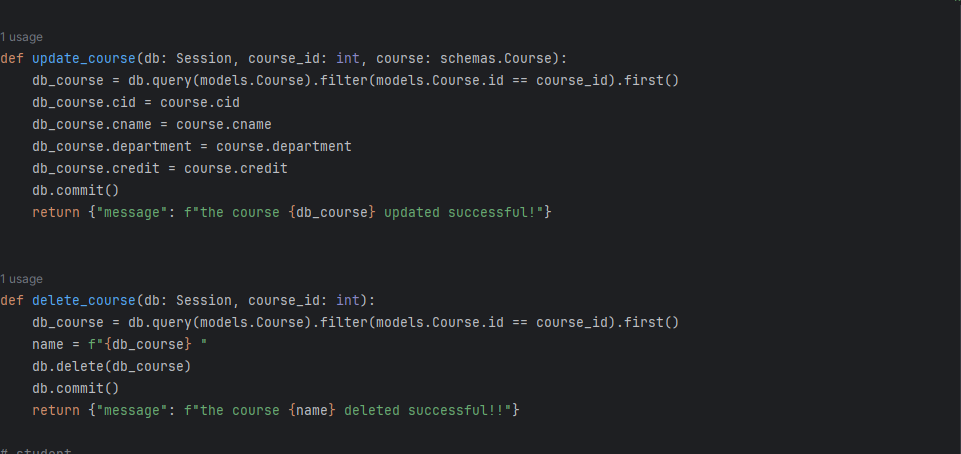
ابتدا فایل پایتونی ( \_\_in it\_\_) که باید خالی از محتوا باشد تا ادیتور پایچارم آنان را یک پکیج کامل در نظر بگیرد!!....



* **فایل crud :**

نام این فایل که مخفف cut read update delete می باشد را باید بسازیم و در آن باید تابع های لازم برای هر یک تز این چهار عمل را بنویسیم :

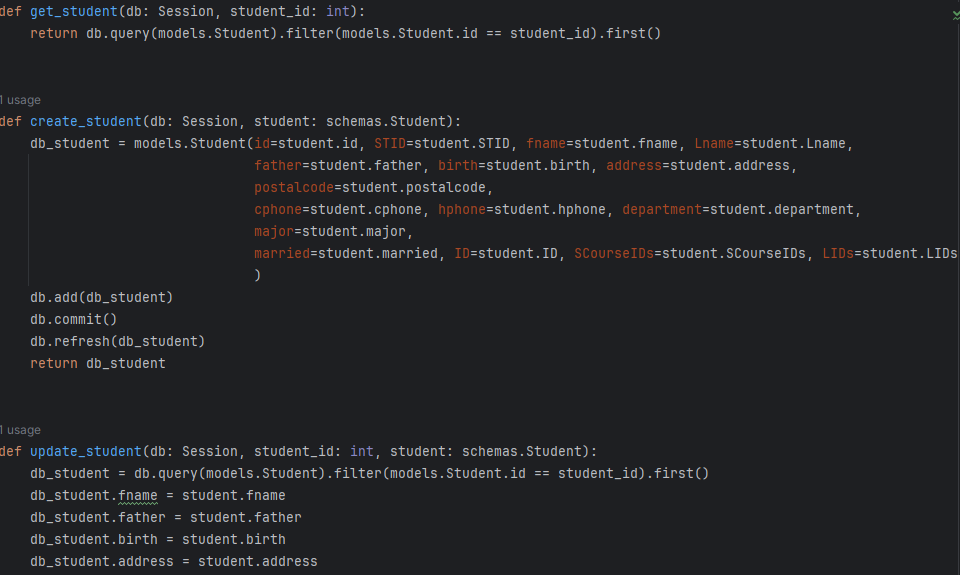


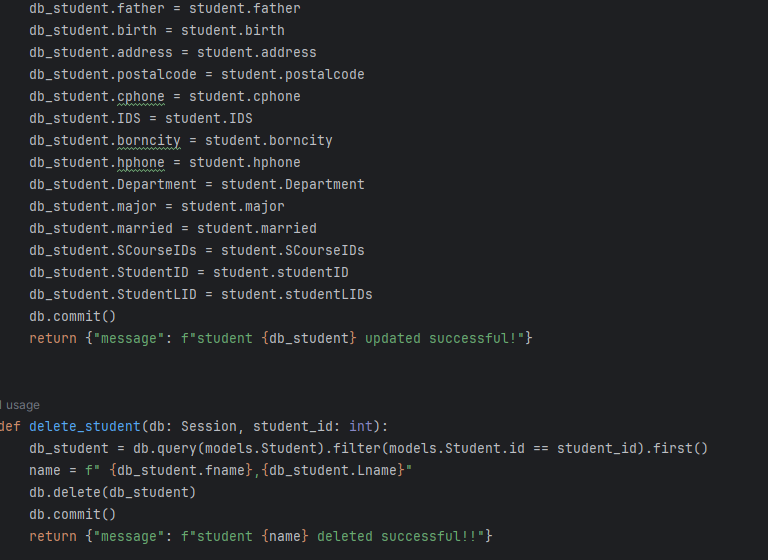


برای مثال برای جدول درس که با نام course نامگذاری اش کردیم برایش 4 تابع در این فایل می نویسیم که به ترتیب اولین برای خواندن دومی برای ایجاد کردن یک فضای جدولی برای آن از طریق تابع Session که یک پل ارتباطی میان پایگاه داده و ادیتور ما میباشد.

تابع سوم برای آپدیت کردن و اعمال تغییرات روی ورودی هاست. آخرین تابع یعنی delete نیز برای پاک کردن و حذف یک ردیف از ورودی ها برای همین جدول دروس می باشد .

و به همین ترتیب برای دو جدول دانشجو و استاد هم همین 4 تابع مهم را مینویسیم بدین صورت :





در تابع اول با متغیری که آنرا وارد میکنیم آنرا ابتدا فیلتر و سپس اولین آنرا از طریق query parameters به پایگاه داده ارسال می کند .

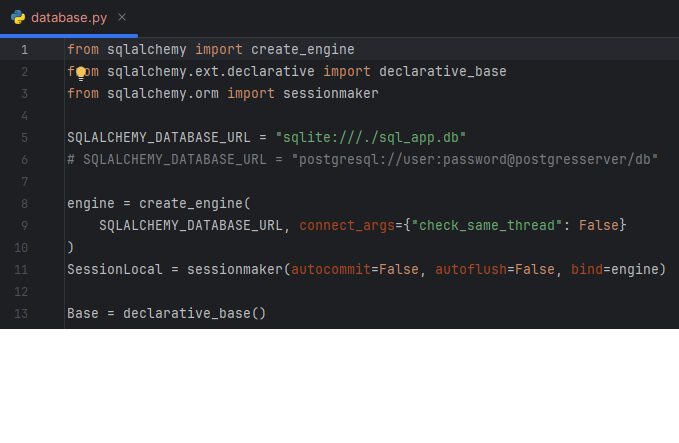
تابع دوم وظیفه ایجاد یک فضا در پایگاه داده با توجه به همه مقادیر لازم برایش ایجاد کرده و در آخر آنان را ثبت و ذخیره میکند. سیستم را Refreshکرده وهمان را برایمان برمیگرداند تا نمایش دهد.

تابع سوم هم تمامی مقادیر را آپدیت کرده و ثبت کرده و دوباره برمیگرداند.

تابع چهام هم از نامش میتوانم فهمید که دقیقا چه وظیفه ای دارد و مثلا همان مقادیر نام و منام خانوادگی را حذف میکند . وبه همین ترتیب برای 2 جدول استاد و دانشجو این کار ها را تکرار میکنیم.

* **فایل دیتابیس :**

در این فایل باید آن پایگاه داده ای که با آن قرار است ستون و جدول هایی که باید بسازیم را به کمک آن وارد کنیم برای مثال ساختار s q l alchemy :



در این قسمت تمامی ماژول های پایگاه داده و ساختار s q l alchemy را وارد کرده که برای کل پروژه لازم داریم .

* **فایل main :**

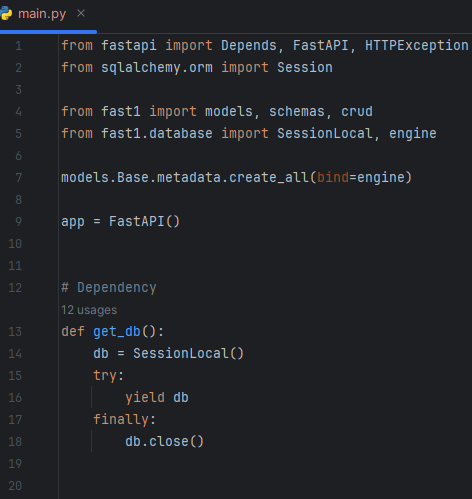
در این فایل متد های فست ای پی آی را برای خواندن و وارد کردن و

تغییر دادن و در نهایت حذف کردن یک داده را (در صورت لزوم) وارد کنیم .

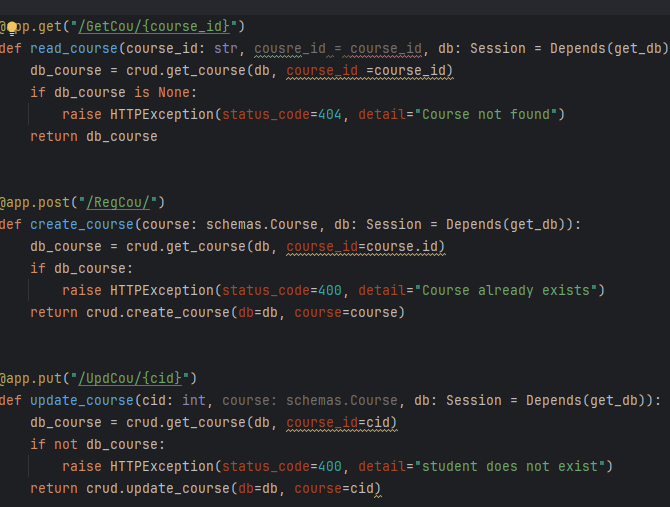
برای هر جدول چهار متد لازم داریم !

ابتدا تمامی ماژول های fast API و پایگاه داده و موارد لازم را وارد کرده که در کنارشان تابع get \_ d b را که مختص به خود پایگاه داده بوده را نوشته و سپس متد های

هر جدول را وارد میکنیم.



متد ها را وارد میکنیم:



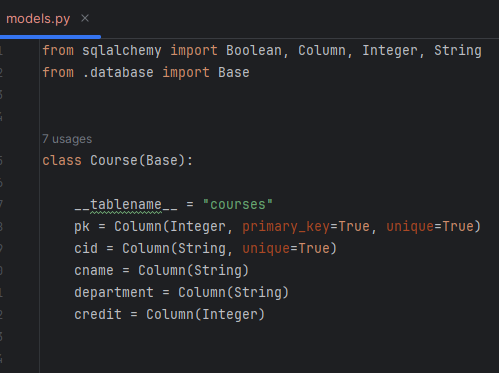


متدها طبق fast API باید نوشته شوند . متد get برای خواندن اطلاعات و post ارسال اطلاعات و متد put برای آپدیدت کردن اطلاعات بوده و delete هم برای حذف اطلاعات میباشد.

در ادامه در هر تابع یک سری دستورات و شروطی را طبق گفته سوال مینویسیم که اگر برقرار نبودند خطا هایی را برای ما raise کند.

* **فایل models:**

در این فایل پایتونی ما نام جدول ها و ستون ها و سطر ها و نوعشان را مینویسیم و به ورود کردن دیتابیس نوع ورودی ارسال شده از این فایل را به آنان معلوم میکنیم . البته که باید برای هر جدول یک class جداگانه ساخت!!..

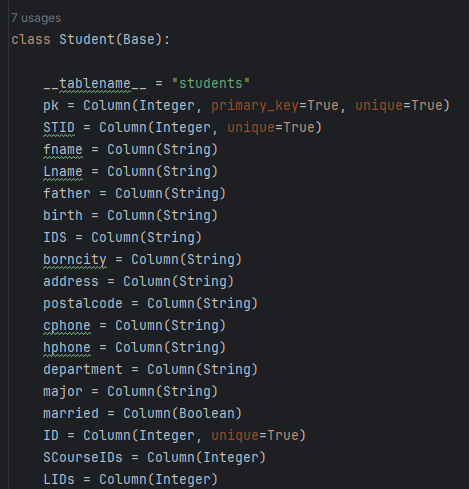
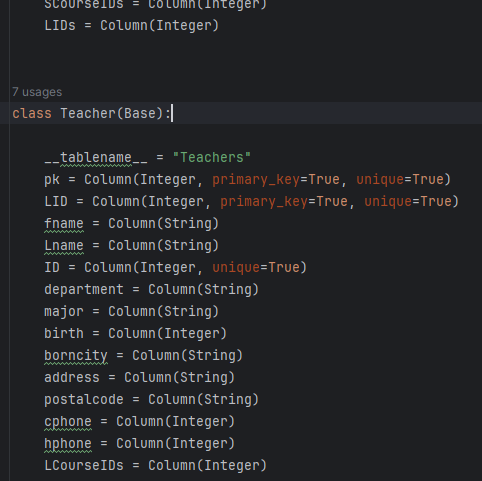


باید در این فایل ابتدا ماژول هایی نظیر String و base و Column و.. را از دایرکتوری های دیگر وارد کرد و برای هر سه جدول که باید بسازیم کلاسی را تعریف میکنیم که شامل اسم جدول بوده و مقادیر خود جدول را می نویسیم .

*عبارت primary key برای زمانی است که یک پیش فرض برای داشته باشیم.*

*عبارت unique هم برای زمانی است که از فقط یکی و برای یک نفر داشته باشیم.*

*برای دو جدول دیگر هم بدین شکل مینویسیم:*



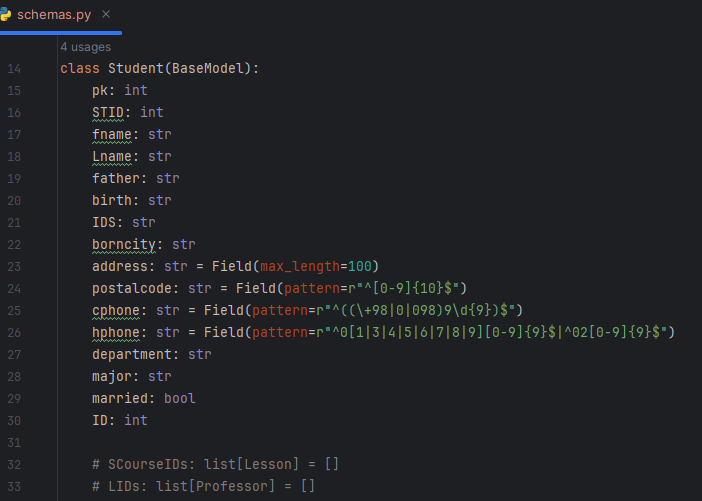
* **فایل schemas :**

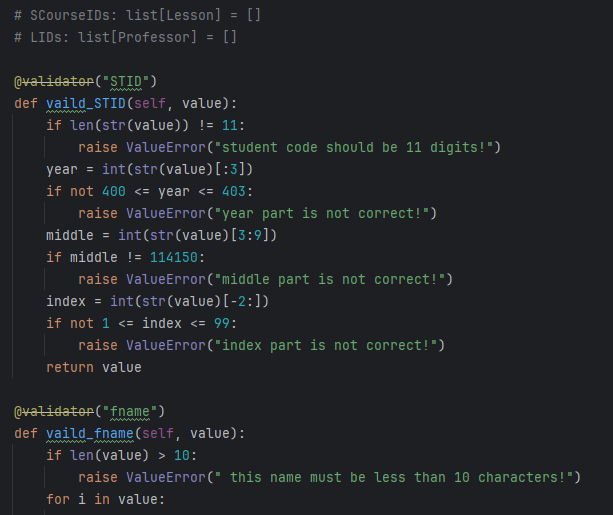
در این فایل فیلدها و مشخصات مهم هر قسمت از متغیر های ورودی هر جدول را مشخص کنیم و شرایط خاص بودن و تعیین نوع متغیر و مقدار پیش فرض برای برخی از متغیر ها را معلوم کرد .

لازم است بگوییم که صحت سنجی متغیر ها نیز باید در این فایل پایتونی سنجیده گردد .

یعنی در واقع validation برای همه متغیر ها چون ما برای تعدادی از این مشخصات شرایط خاصی ذکر داریم .

برای هر جدول یا همان کلاس ساخته برایش در فایل models متغیر های همان جدول را صحت سنجی میکنیم و می دانیم که باید همه ی validator های متغیر های مرتبط با یک class باید در امتداد همان کلاس نوشته شوند .

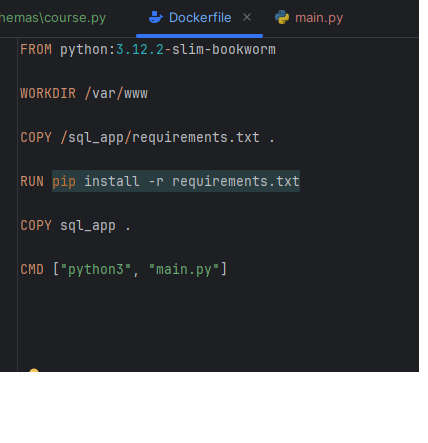




برای هر جدول یک کلاس نوشته و نوع مقادیر را تعیین میکنیم و در صورت لزوم از regex یا شرایط خاص برای فیلد آن میتوان به شکل دیگری آنرا validation کرد یعنی آنها را صحت سنجی کرد .

برای validatin هر مقدار باید کلاس متدی با همین نام برایش ساخت که در راستای همان کلاس اصلی جدول باشند. مثلا برای آدرس دهی طول رشته باید 100 باشد که نیاز به ولیدیشن مجزا نیست ولی برای کددانشجویی لازم است یک لیدیتور جداگانه ساخت. وبه همین ترتیب الی آخر تا آخرین مقدار از جدول سوم همین کار ها را تکرار میکنیم.

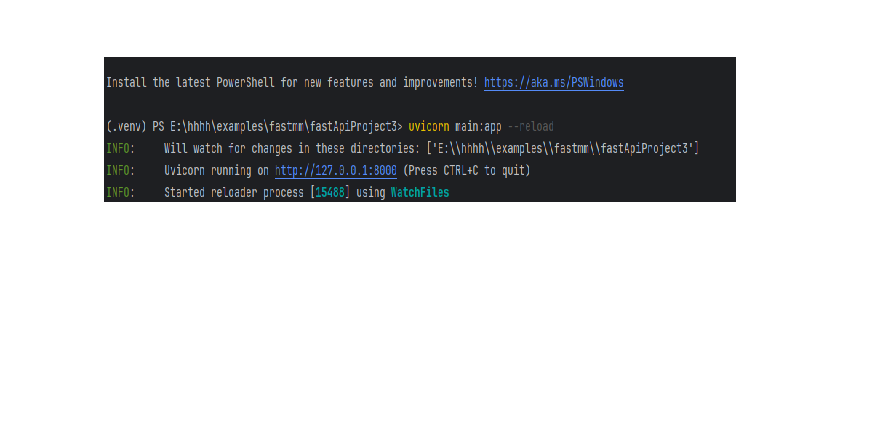
و در آخر فایل docker fileرا برای این پروژه ایجاد می کنیم که یک قالب مهم برای اجرا شدن این نمونه در هر محیطی می باشد .



در نهایت برای اجرا کردن نهایی این پکیج در قسمت terminal در ادیتور پایچارم

دستور زیر را وارد میکنیم ودر حین این کار بهتر است که فایل main در حال اجرا باشد:

Uvicornfast1/main :app -- reload



(البته که در اینجا fast1 نام دایرکتوری است که ما در آن پکیج را ساختیم.)