

فایل main.py هسته‌ی اصلی پروژه است و شامل پیاده‌سازی کامل API با استفاده از FastAPI می‌باشد. این فایل وظایف زیر را بر عهده دارد:

- تعریف مدل‌های داده‌ای برای سه موجودیت اصلی پروژه Student ، Teacher و Course با استفاده از SQLAlchemy
- اعتبارسنجی کامل مقادیر ورودی با استفاده از validator@ برای اطمینان از صحت و سازگاری داده‌ها با نیازمندی‌های پروژه (مثلاً فرمت کد ملی، شماره دانشجویی، تاریخ تولد شمسی و ...).
- پیاده‌سازی عملیات CRUD (ایجاد، خواندن، ویرایش و حذف) برای هر سه موجودیت از طریق endpoint های FastAPI
- تعریف session و اتصال به پایگاه داده SQLite برای مدیریت داده‌ها.
- ساختاردهی API با استفاده از @app.get, @app.post, @app.put, @app.delete برای مسیریهای RESTful
- این فایل با اجرای مستقیم سرور FastAPI ، سرویس‌دهی API را فعال می‌کند و آماده است تا روی سرور لینوکسی با Docker و Nginx مستقر شود.
- این فایل بدون نیاز به فایل‌های اضافی قابل اجراست و پایگاه داده را در اولین اجرا به صورت خودکار ایجاد می‌کند. (on\_startup) تمام خطاهای احتمالی ناشی از ورود داده‌های نامعتبر از طریق FastAPI با پیام مناسب به کاربر برگشت داده می‌شود.

## در واقع فعالیت های ۱ تا ۴ پروژه ما را در بر میگیرد

## ۱. به‌روزرسانی سرور

با استفاده از ابزار PuTTY به سرور متصل شدم و دستورات زیر را اجرا کردم تا سیستم به‌روز شود:

`sudo apt update`

`sudo apt upgrade`

```
root@id-345247: ~  
Get:21 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [157 kB]  
Get:22 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]  
Get:23 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1,131 kB]  
Get:24 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [235 kB]  
Get:25 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]  
Get:26 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [842 kB]  
Get:27 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [184 kB]  
Get:28 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]  
Get:29 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]  
Fetched 8,552 kB in 4s (2,052 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
2 packages can be upgraded. Run 'apt list --upgradable' to see them.  
root@id-345247:~# apt upgrade
```

## ۲. نصب Docker و Docker Compose

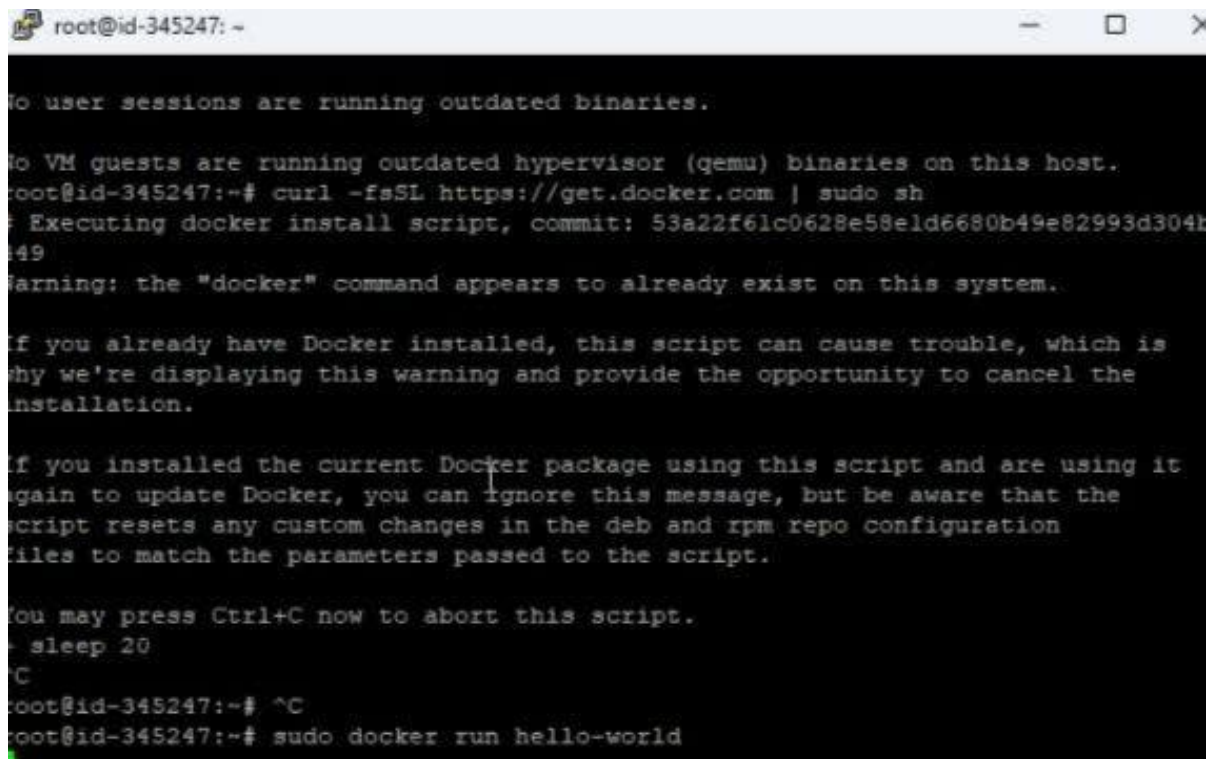
برای نصب داکر و داکر کامپوز از دستورات زیر استفاده کرد

```
sudo curl -fsSL https://get.docker.com | sh
```

## ۳. تست موفق نصب Docker

برای اطمینان از نصب موفق داکر، از دستور زیر استفاده کردم:

```
sudo docker run hello-world
```



```
root@id-345247: ~  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
root@id-345247:~# curl -fsSL https://get.docker.com | sudo sh  
+ Executing docker install script, commit: 53a22f61c0628e58eld6680b49e82993d304b  
49  
Warning: the "docker" command appears to already exist on this system.  
  
If you already have Docker installed, this script can cause trouble, which is  
why we're displaying this warning and provide the opportunity to cancel the  
installation.  
  
If you installed the current Docker package using this script and are using it  
again to update Docker, you can ignore this message, but be aware that the  
script resets any custom changes in the deb and rpm repo configuration  
files to match the parameters passed to the script.  
  
You may press Ctrl+C now to abort this script.  
+ sleep 20  
^C  
root@id-345247:~# ^C  
root@id-345247:~# sudo docker run hello-world
```

نتیجه اجرای دستور پیام موفقیت آمیز بود.

## ۴. آماده سازی فایل های Docker

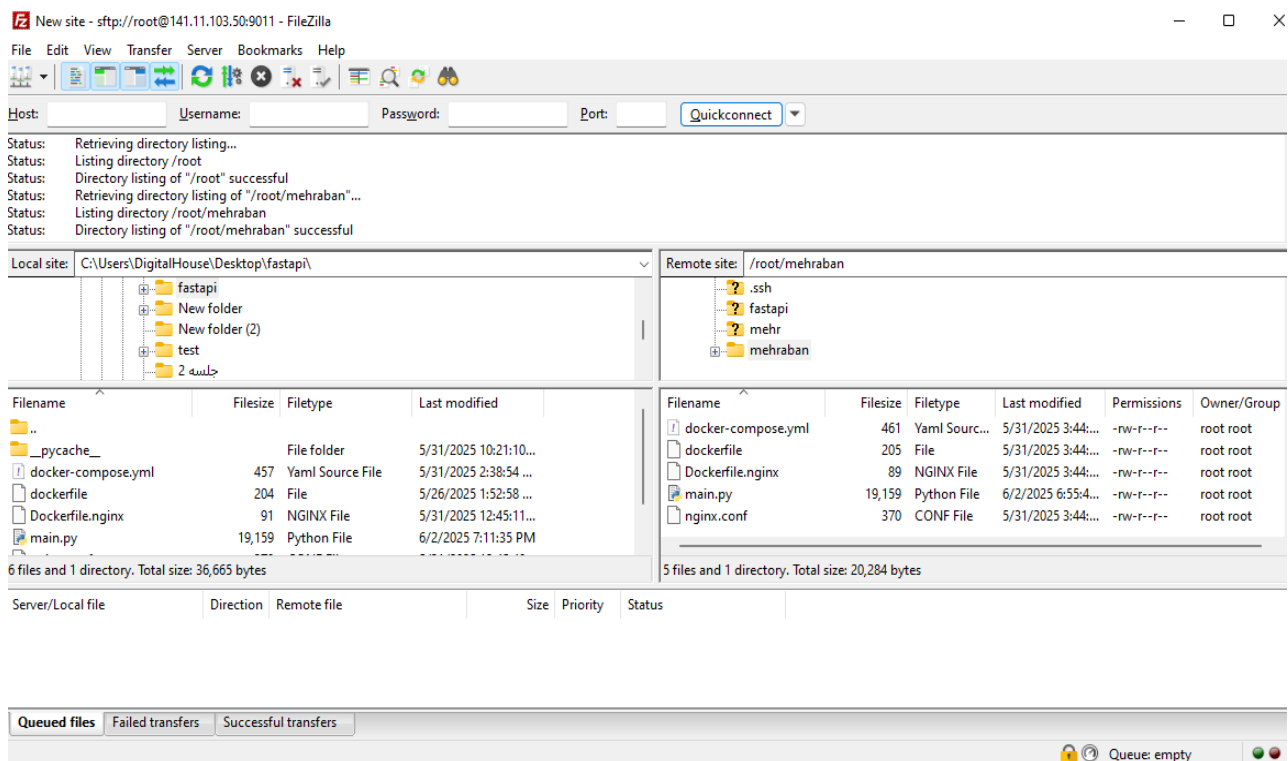
در محیط توسعه، دو فایل اصلی زیر را ایجاد کردم:

Dockerfile برای ساخت image پروژه FastAPI

docker-compose.yml برای اجرای پروژه و پایگاه داده به صورت همزمان

## ۵. انتقال فایل ها به سرور

با استفاده از نرم افزار FileZilla، فایل های پروژه و فایل های داکر به سرور منتقل شدند.



۶. اجرای کانتینرها روی سرور

در نهایت در مسیر پروژه روی سرور، کانتینرها را با دستور زیر اجرا کردم:

```
sudo docker-compose up --build -d
```

و بعد از اتمام این کارها داکرایز کردن انجام شد

و در

<http://mehrabanmalek.ir:۸۰۸۰/docs>

در مرورگر به درستی بالا می آید.

۸. استفاده از Nginx

برای اجرای nginx باید کارهای زیر را انجام دهیم و این تغییرات برای این است که بدون وارد کردن پورت به لینک خود در سووگر وصل بشیم یعنی به صورت:

<http://mehrabanmalek.ir/docs>

ابتدا با ایجاد کردن دو فایل با نام های

Dockerfile.nginx و nginx.conf

این فایل ها را با فایل زیلا بر روی سرور مجازی (هاست) اپلود میکنیم و اجرای دوباره داکر

کامپاس

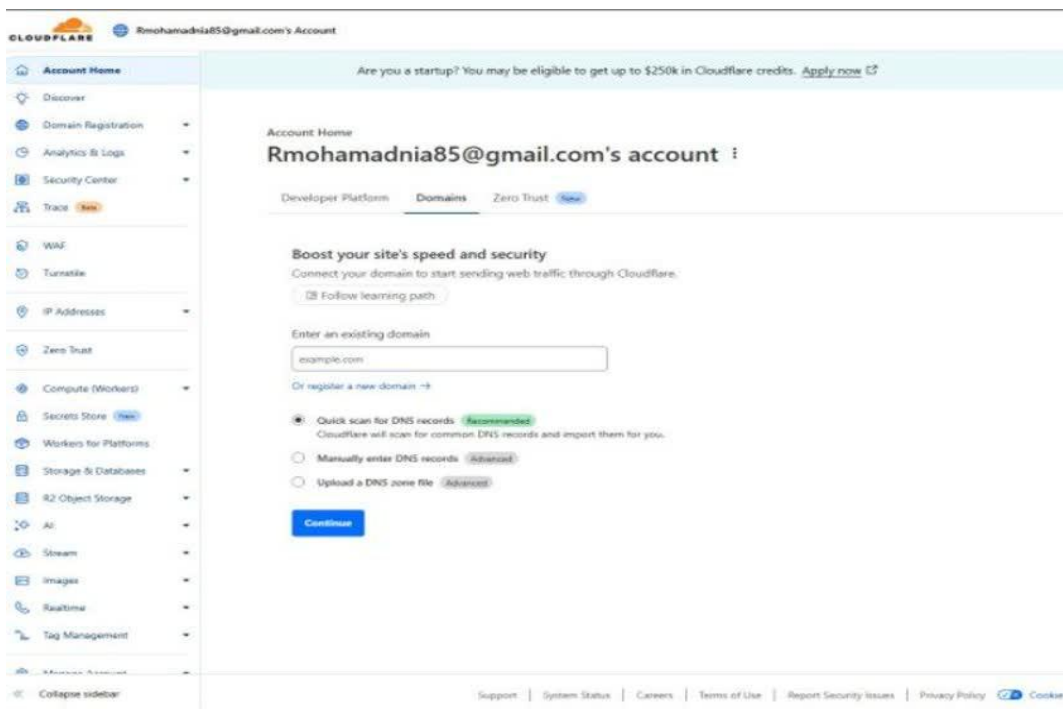
حال لینک بدون پورت بالا می آید.

## ۹. خرید دامین از سات ایرنیک و اتصال به cloudflare

### یک دامنه از سایت ایرنیک تهیه کردیم



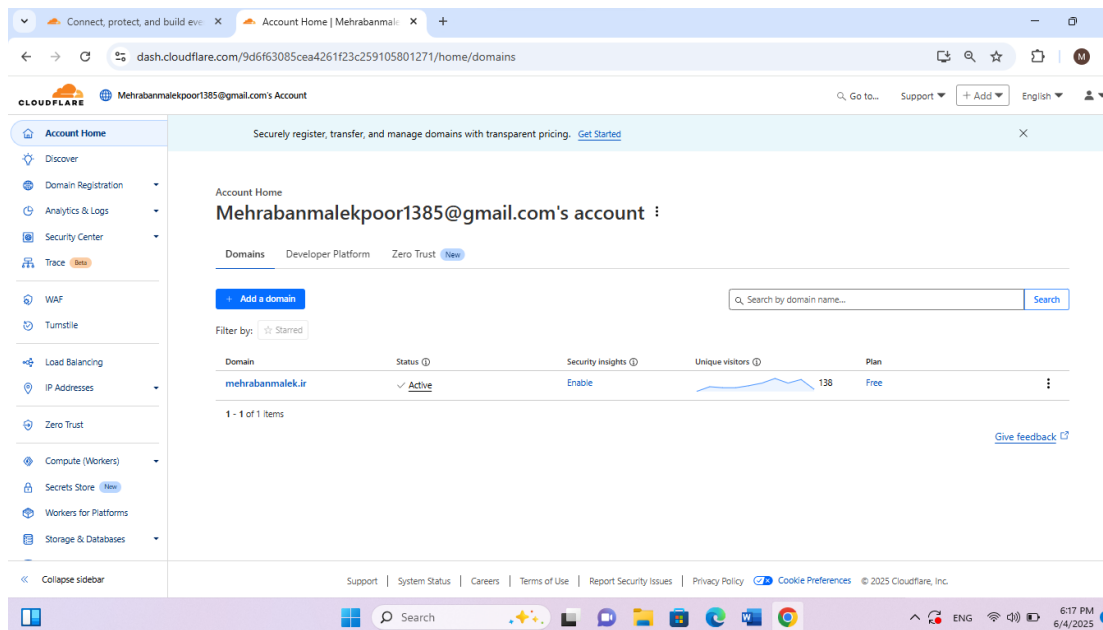
پس از خرید دامین خود باید ان را به cloudflare اتصال دهیم برای اسن کر وارد کلود فلر شده در این قسمت نام دامین خود را وارد میکنیم



و پس از فرستادن دامین خود رمز هایی که کلود فلر برای ورود میدهد را در ایرنیک وارد میکنیم

پس وارد کردن نام و رمز ها اتصال را میزنیم و اتصال برقرار میشود

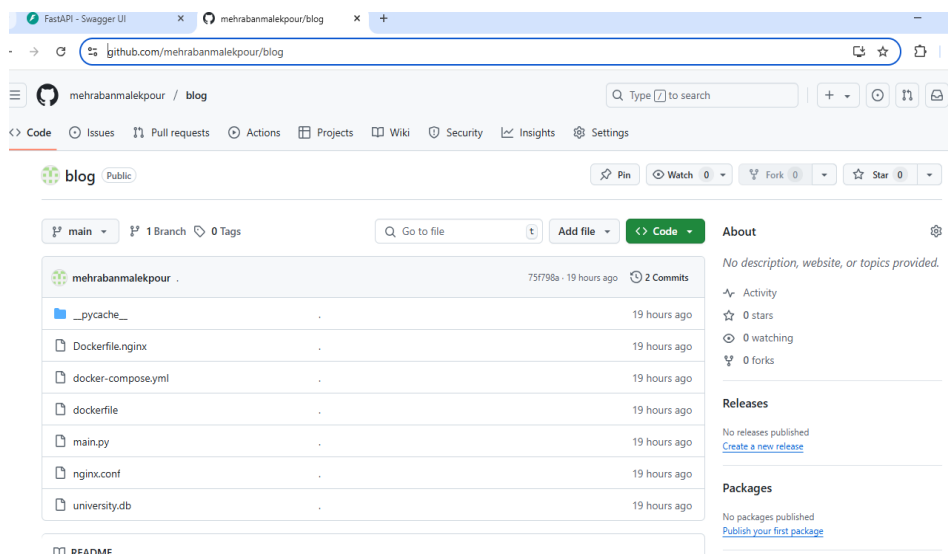
پس دو روز سرور در خواست را پذیرفته و ان را اکتیو میکند



پس از تمام این کارها در فعالیت ششم باید تمام فایل ها را در بستر GitHub

از طریق IDE و push کنیم

<https://github.com/mehrabanmalekpour/blog>



۱۲. در آخر همه API ها بر روی دامنه قابل دسترس است.

<http://mehrabanmalrk.ir/docs>