

پروژه نهایی برنامه نویسی پیشرفته

مدیریت اطلاعات دانشجویان، اساتید و دروس

در این پروژه یک API کامل جهت مدیریت دانشجو، استاد و درس پیاده سازی شده، عملیات CRUD برای هر موجودیت پیاده سازی و اعتبارسنجیهای لازم اعمال شده‌اند. پروژه به صورت کامل داکرایز شده و بر روی سرور لینوکسی مستقر شده است.

مراحل انجام پروژه :

- ساخت مدل‌های course, teacher, student و انجام اعتبار سنجی ورودیها با استفاده از SQLAlchemy و Pydantic و متد @validator انجام شده که هر ولیدیتور نیز شامل محدودیت خاصی میباشد.
- اینجا برای ایجاد و مدیریت اطلاعات مربوط به دانشجو، استاد و درس از پایگاه داده‌ی SQLite استفاده شده است. برای کار با آن باید ارتباطی میان برنامه و دیتابیس ایجاد شود که با استفاده از session انجام میشود.
- ساخت مسیرهای دسترسی با کمک Fast API برای انجام عملیات CURD (ایجاد، خواندن، ویرایش، حذف). برای ساختاردهی API از دکوراتورها استفاده شده است.
- با اجرای فایل اصلی، سرور FastAPI راه اندازی شده و API ها فعال میشوند. عملیات CURD برای مدل‌ها از طریق مسیرهای مشخص API انجام میشود.
- پروژه به راحتی از طریق یک سرور لینوکسی با ابزارهایی مثل docker و nginx از طریق اینترنت قابل دسترسی میباشد.
- برای داکرایز کردن ابتدا با استفاده از ابزار PuTTY و وارد کردن دستورات لازم، Docker و Docker Compose را نصب میکنیم تا امکان اجرای پروژه در قالب کانینر فراهم شود.

```
root@id-447205: ~/hannah
root@id-447205:~/hannah# sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

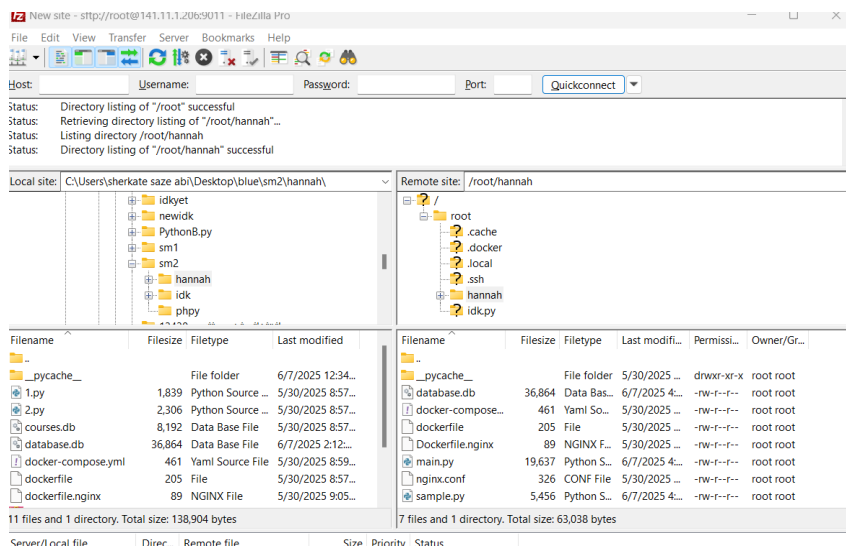
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

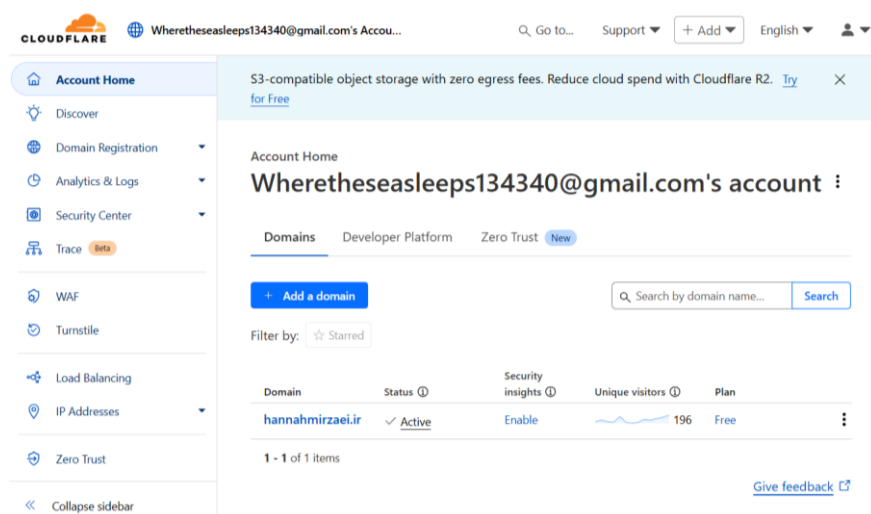
For more examples and ideas, visit:
https://docs.docker.com/get-started/

root@id-447205:~/hannah#
```

- یک فایل dockerfile ایجاد میکنیم که مرحله به مرحله مشخص می کند برنامه چطور نصب و اجرا شود. درواقع با استفاده از فایل Dockerfile، یک Image از برنامه ساخته می شود. سپس فایل docker-compose.yml را ایجاد میکنیم که برای تعریف و اجرای هماهنگ سرویس های پروژه مانند FastAPI و Nginx مورد استفاده قرار می گیرد.
- برای انتقال فایل ها به سرور از نرم افزار FileZilla استفاده میکنیم.



- بعد از وارد کردن دستور `sudo docker-compose up -build -d` نیز کانتینرها روی سرور اجرا میشوند و داکرایز بطور کامل انجام میشود.
- برای اجرای پروژه در یک محیط واقعی و قابل دسترسی از طریق اینترنت، یک سرور مجازی لینوکسی تهیه شد تا پروژه روی آن اجرا شود. بعد با استفاده از PuTTY به سرور متصل شده و ابزارهای مورد نیاز (docker و docker compose) را نصب میکنیم.
- باید یک دامنه نیز تهیه کنیم. پس از تهیه ی دامنه، وارد cloudflare شده و دامنه را در آن ثبت و به IP سرور متصل میکنیم.



- برای عدم نیاز به وارد کردن پورت و بهتر کردن عملکرد از Nginx استفاده میکنیم. برای راهاندازی فایل‌های dockerfile.nginx و nginx.conf را ایجاد کرده و سپس با استفاده از FileZilla آنها را روی هاست نیز آپلود میکنیم.
- بعد از اتمام همه‌ی مراحل، فایل‌ها را در GitHub، push میکنیم.

