

## گزارش پروژه‌ی پایان‌ترم برنامه‌نویسی پیشرفته

درس: برنامه‌نویسی پیشرفته

تهیه‌کننده: ثنا ساکی

استاد درس: دکتر آرمین رشنو

آدرس گیت‌هاب پروژه: <https://github.com/SANA-botSANA-bot/uni-api>

آدرس مستندات پروژه: <http://sanasaki.ir/docs>

### مقدمه

این گزارش به معرفی و تشریح پروژه‌ی پایان‌ترم درس «برنامه‌نویسی پیشرفته» می‌پردازد. پروژه با استفاده از ابزارهای مرتبط توسعه داده شده است. هدف اصلی این پروژه، پیاده‌سازی یک سیستم ساده FastAPI فریمورک مدیریت دانشگاهی است که امکان مدیریت اطلاعات دانشجویان، اساتید و دروس را فراهم می‌کند. طراحی شده و برای بهینه‌سازی و استقرار از ابزارهایی نظیر RESTful API این سامانه با بهره‌گیری از معماری استفاده شده است SQLite و پایگاه داده Nginx و Docker.

### توضیحات کلی پروژه

است که برای مدیریت سه موجودیت اصلی دانشگاهی FastAPI مبتنی بر فریمورک API پروژه شامل یک انجام شده و اعتبارسنجی (SQLModel) دانشجو، استاد، درس (طراحی شده است. مدل‌سازی داده‌ها با استفاده از پیاده‌سازی گردیده است Pydantic دقیق داده‌ها به کمک استفاده شده و Docker Compose و Docker به‌منظور استقرار و اجرای پروژه در محیط کانتینری، از برای بهینه‌سازی و هدایت ترافیک تعریف شده است Reverse Proxy به‌عنوان Nginx سرویس.

### ساختار پروژه

(main.py) فایل اصلی پروژه ۱. (Endpoints) تعریف مدل‌ها، مسیرها، API این فایل نقطه‌ی آغاز برنامه است که شامل پیاده‌سازی منطق اصلی و ارتباط با پایگاه داده می‌باشد.

- استفاده شده است HTTP برای ایجاد اپلیکیشن و مدیریت درخواست‌های FastAPI از -
- تعریف شده‌اند SQLAlchemy مدل‌های داده‌ای با استفاده از -
- انجام شده است Pydantic اعتبارسنجی داده‌ها با کمک -

## مدل‌های داده‌ای

در پروژه، چهار کلاس اصلی تعریف شده است:

- **Person**: کلاس پایه شامل ویژگی‌های مشترک بین دانشجو و استاد مانند نام، نام خانوادگی، کد ملی، شهر محل تولد، تاریخ تولد، آدرس، کد پستی، شماره تلفن، دانشکده و رشته تحصیلی
- **Student**: با ویژگی‌های خاص دانشجویی نظیر شماره دانشجویی، نام پدر، سریال **Person** ارث‌بر کلاس -  
شناسنامه، وضعیت تأهل، لیست دروس و اساتید
- **Professor**: با ویژگی‌های خاص استاد مانند کد استاد و لیست دروس تدریسی **Person** ارث‌بر کلاس -  
کلاس تعریف درس با ویژگی‌هایی مانند کد درس، نام درس، دانشکده و تعداد واحد
- **Course**: کلاس تعریف درس با ویژگی‌هایی مانند کد درس، نام درس، دانشکده و تعداد واحد

اعتبارسنجی‌ها:

- نام و نام خانوادگی باید فقط شامل حروف فارسی و حداکثر ۱۰ کاراکتر باشند -
- کد ملی و کد پستی باید عددی و ۱۰ رقمی باشند -
- شماره دانشجویی باید ۱۱ رقمی و شامل سال ورود (بین ۱۳۸۵ تا ۱۴۰۳) (و بخش ثابت باشد -
- شهر محل تولد باید یکی از مراکز استان‌های ایران باشد -
- رشته تحصیلی باید با دانشکده انتخاب‌شده همخوانی داشته باشد -
- و در بازه‌ی سال‌های ۱۳۰۰ تا ۱۴۰۰ باشد (YYYY/MM/DD) تاریخ تولد باید در فرمت شمسی -

## CRUD عملیات‌های

برای هر یک از موجودیت‌ها (دانشجو، استاد، درس)، عملیات‌های زیر پیاده‌سازی شده‌اند:

- ایجاد (POST)
- خواندن (GET)
- به‌روزرسانی (PUT)
- حذف (DELETE)

در دسترس هستند در /courses/، /professors/، /students/ این عملیات‌ها از طریق مسیرهای استفاده شده و پیام‌های فارسی مناسب مانند خطای ۴۰۴ به کاربر HTTPException صورت وجود خطا، از

نمایش داده می‌شود

## پایگاه داده

پایاده‌سازی شده است SQLite پایگاه داده با استفاده از  
جداول لازم در دیتابیس ایجاد می‌شوند، `create_db_and_tables` در زمان اجرای اپلیکیشن، با اجرای تابع  
انجام می‌گیرد Session و Depends مدیریت جلسات ارتباط با پایگاه داده از طریق

## استقرار و بهینه‌سازی

اجرا می‌شود Docker و Docker Compose پروژه به‌صورت کامل کانتینریزه شده و با استفاده از

اجزای اصلی استقرار

که روی پورت 8080 اجرا می‌شود FastAPI شامل اپلیکیشن: Backend -  
و پشتیبانی از پورت‌های 80 و Backend برای هدایت درخواست‌ها به Reverse Proxy به‌عنوان: Nginx -  
443.

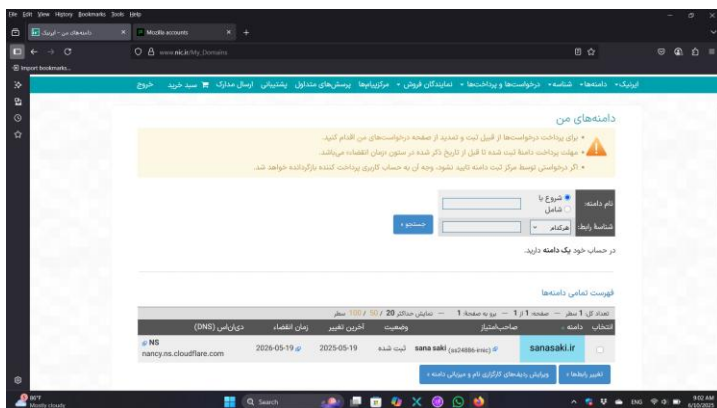
انجام شده است و هر دو سرویس در یک شبکه `docker-compose.yml` در فایل Docker تنظیمات مربوط به  
قرار دارند `app-network` مشترک به نام  
و کتابخانه‌های مورد نیاز مانند Python 3.11 شامل نصب وابستگی‌های Backend برای Dockerfile فایل  
می‌باشد Pydantic و SQLModel، FastAPI  
به کار رفته‌اند Nginx برای تنظیمات `nginx.conf` و `Dockerfile.nginx` فایل  
متصل شده Nginx به کانتینر Volume به‌صورت `/etc/letsencrypt` پوشه‌ی، HTTPS برای پشتیبانی از  
است.

## اعتبارسنجی و مدیریت خطاها

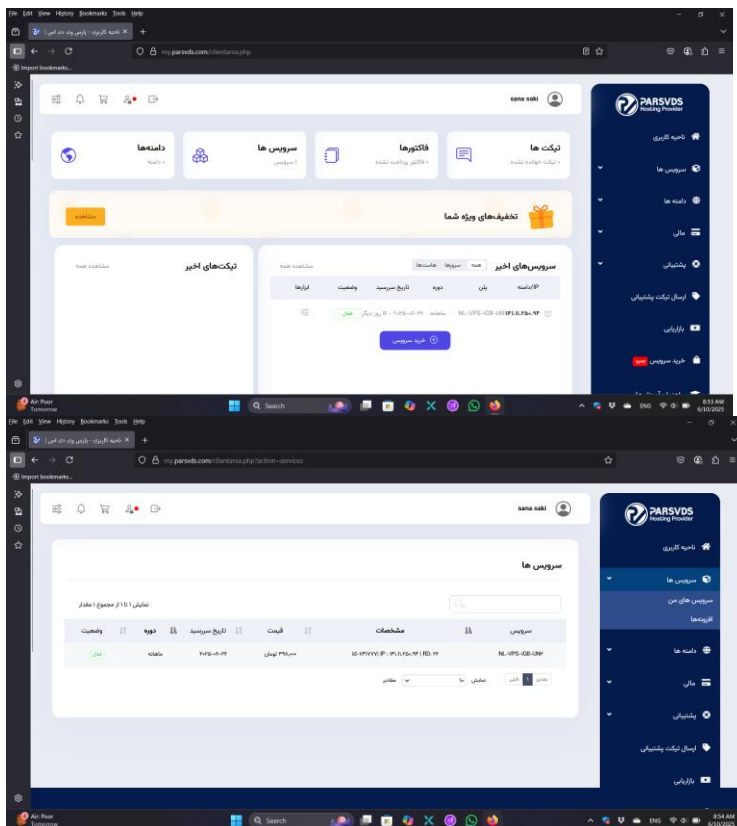
اعتبارسنجی کامل برای همه‌ی فیلدها جهت جلوگیری از ورود اطلاعات نادرست انجام شده است  
HTTP تمامی خطاهای قابل شناسایی (عدم وجود رکورد، ورودی ناصحیح و ...) با پیام‌های فارسی و استانداردهای  
مدیریت می‌شوند.

## تنظیمات هاست و دامین و اتصال آنها با کلود فلیر

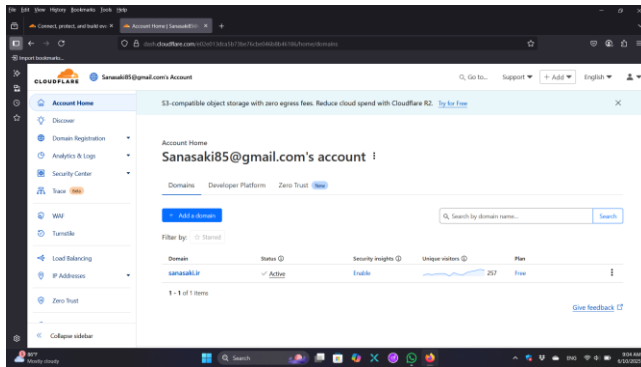
تهیه می‌کنیم nic.ir: یک دامنه از سایت



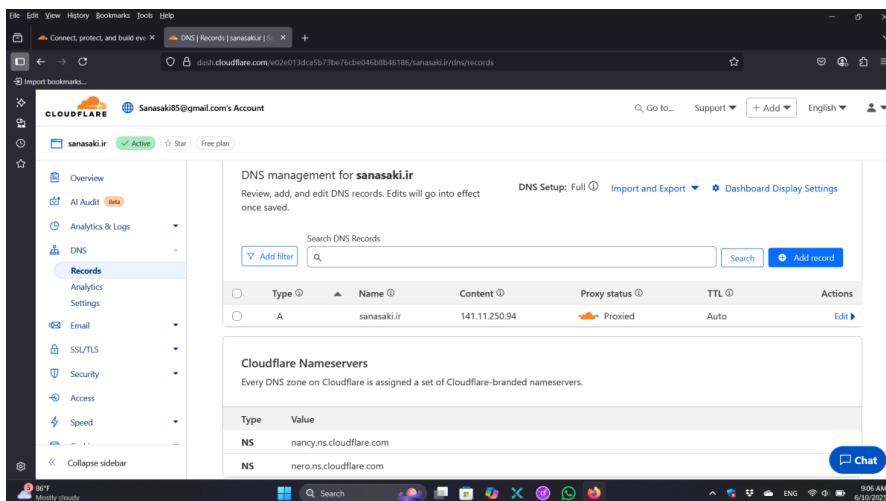
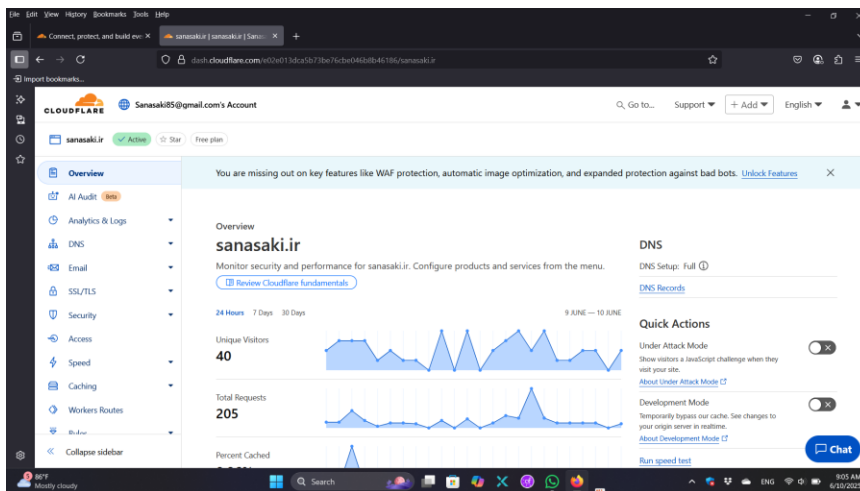
و رمز هاست را IP خریداری می‌کنیم و parsvds.com یا sarvhost.com گام دوم: یک هاست از سایت‌هایی مانند نزد خود نگه می‌داریم



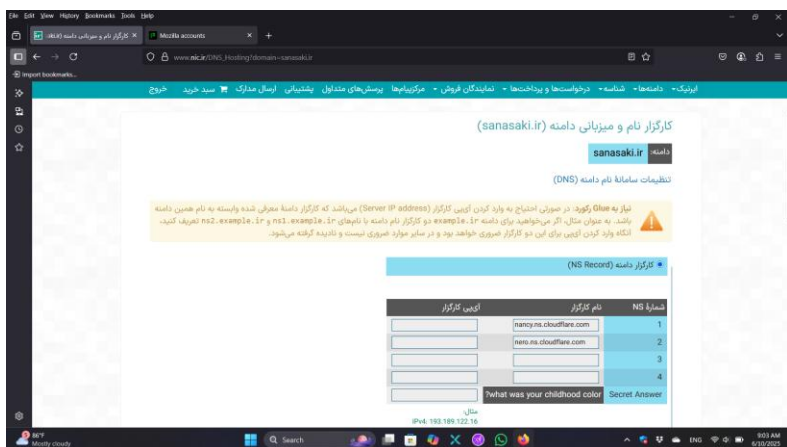
دامنه خود را اضافه می‌کنیم "add a domain" می‌رویم و با استفاده از گزینه Cloudflare.com گام سوم: به سایت



مشاهده می‌شود name server دو DNS Record گام چهارم: با کلیک روی دامنه و رفتن به بخش



دامنه در ایرنیک، اتصال ایرنیک و کلود ns ها و پیست آن‌ها در بخش name server گام پنجم: در این بخش با کپی فلیپر را برقرار می‌کنیم.



و نام دامنه و آدرس هاست ایجاد می‌کنیم a در نهایت یک رکورد جدید با تایپ

