

ایجاد دامنه در ایرنیک

فهرست تمامی دامنه‌ها

نام دامنه:	نام دامنه:
asrarahimi.ir	asrarahimi.ir

در حساب خود یک دامنه داریم.

لطفاً این دامنه را برای ثبت آماده کنید.

asrarahimi.ir

اطلاعات تماس با ایرنیک

آدرس بسته الکترونیک: info@iran-nic.ir

تلفن: +(۰۲۶) ۳۴۵۸ ۷۰۰۰

(فقط در روزهای کاری از شنبه تا چهارشنبه بین ساعت ۱۵:۴۵ تا ۸:۰۰ پاسخگو می‌باشد.)

تماس: +(۰۲۶) ۳۴۹۹ ۵۷۰۰

سپس داده شده در ایرنیک را در cloudflare کپی میکنید

Account Home

Asra.rahami1384@gmail.com's account :

Domains Developer Platform Zero Trust New

+ Add a domain Search

Filter by: Starred

Domain	Status	Security insights	Unique visitors	Plan
asrarahimi.ir	✓ Active	0	174	Free

1 - 1 of 1 items Give feedback

Support | System Status | Careers | Terms of Use | Report Security Issues | Privacy Policy | Cookie Preferences © 2025 Cloudflare, Inc.

The screenshot shows the Cloudflare dashboard for the domain `asrarahimi.ir`. The left sidebar is open, showing various monitoring and configuration options. The main area displays performance metrics and a timeline from June 4 to June 5. A red box highlights the 'Overview' tab in the sidebar.

Overview

asrarahimi.ir

Monitor security and performance for asrarahimi.ir. Configure products and services from the menu.

Unique Visitors: 30

Total Requests: 91

Percent Cached: 6.83%

Total Data Served: 146 kB

DNS

DNS Records

Quick Actions

- Under Attack Mode: Show visitors a JavaScript challenge when they visit your site. (Toggle off)
- About Under Attack Mode
- Development Mode: Temporarily bypass our cache. See changes to your origin server in real-time. (Toggle off)
- About Development Mode
- Run speed test
- Configure caching

Domain Registration

Registrar: Unknown

The screenshot shows the 'DNS Records' page in the Cloudflare dashboard. The left sidebar is open, showing various monitoring and configuration options. The main area displays a form for adding a new DNS record. A red box highlights the 'Add record' button in the top right corner of the form area.

Ipv4 address **NS IRNIC** **Name** **@**

[name] points to [Ipv4 address] and has [ns] traffic proxied through Cloudflare.

Type: A Name (required): asrarahimi.ir Ipv4 address (required): 51.89.52.213 Proxy status: Proxied TTL: Auto

Record Attributes

Comment

Type	Name	Content	Proxy status	TTL	Actions
A	asrarahimi.ir	51.89.52.213	Proxied	Auto	Edit

Cloudflare Nameservers

Every DNS zone on Cloudflare is assigned a set of Cloudflare-branded nameservers.

The screenshot shows the Cloudflare DNS Records page for the domain asrarahimi.ir. The left sidebar has 'Records' selected. The main area displays a table of records with one entry:

Type	Name	Content	Proxy status	TTL	Actions
A	asrarahimi.ir	51.89.52.213	Proxied	Auto	Edit

Below the table, under 'Cloudflare Nameservers', there is a table showing two NS records:

Type	Value
NS	grace.ns.cloudflare.com
NS	nick.ns.cloudflare.com

خرید هاست از سرو هاست

The screenshot shows the 'HostAsrAhimi' website. The user is selecting a VPS package. A red box highlights the 'Toman ۳۵۰,۰۰۰' price field. Another red box highlights the 'نامهای متن‌وارد' (hostnames) section where 'ns1' and 'ns2' are listed. A red arrow points from the Cloudflare screenshot to the 'ns1' field.

لطفا دوره پرداخت خود را انتخاب نماید

تام ۳۵۰,۰۰۰

نامهای متن‌وارد

ns1 ns2

افزونه‌های این سرویس

اطلاعات اضافی مورد نیاز

* Operating System: centos-V.9-x86_64

* ISO Disk: none

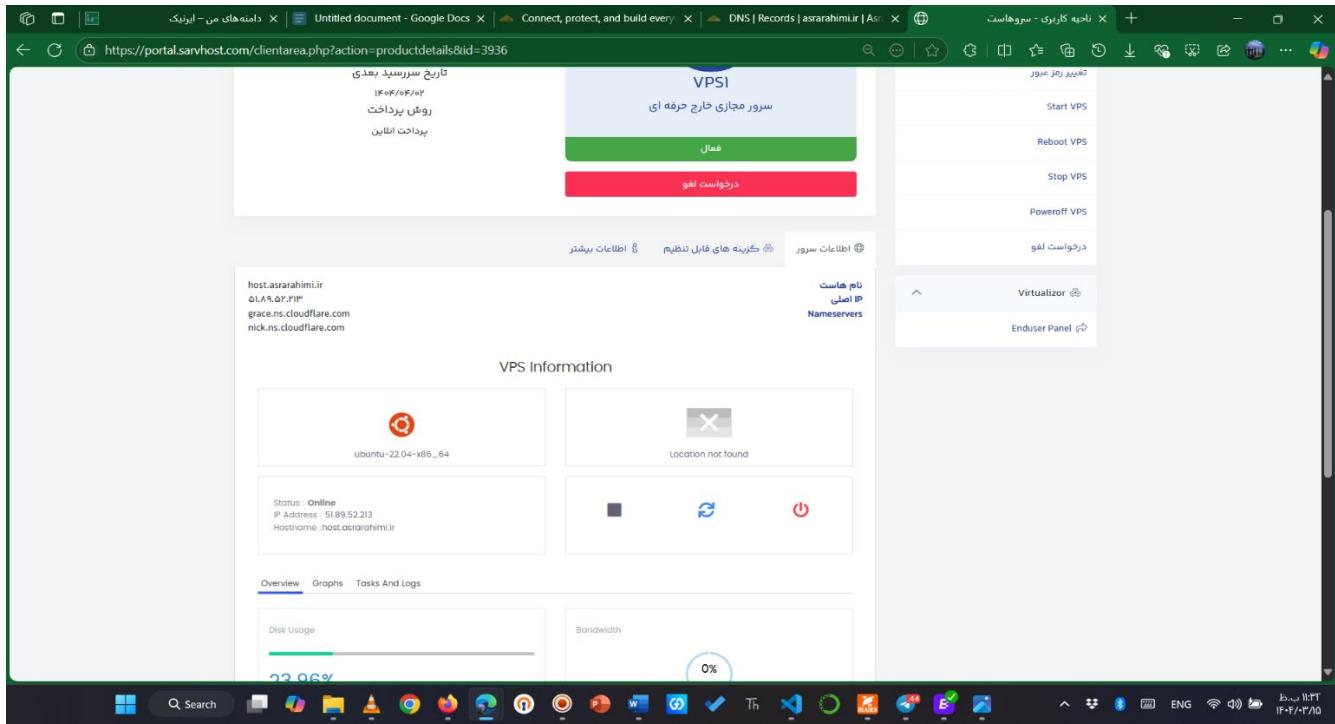
* کشور مورد نظر

آلمان

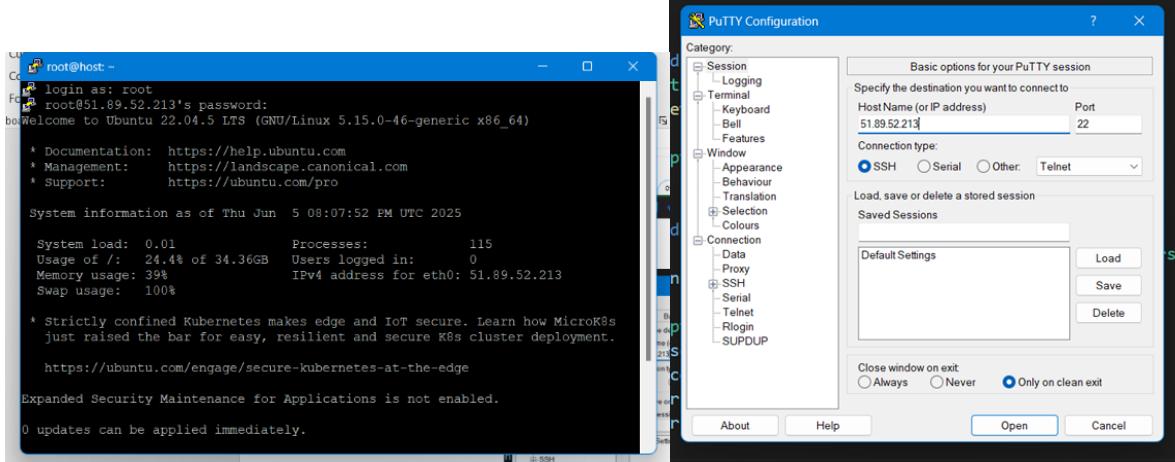
سوالی دارید؟ چهت اطلاعات بینشتر با تیم فروش تماس بگیرید. آینده را مخاطب شوید

خلاصه سفارش

نصب برنامه filezilla و putty



سپس پس از اجرای putty



و سیس دستورات apt update, apt upgrade, apt install python3, curl -fsSL https://get.docker.com | sudo sh

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

ایجاد 2 فایل به نام های dockerfile و docker-compose.yml

```

1 services:
2   backend:
3     build: .
4     image: backend
5     ports:
6       - "8080:8080"
7     volumes:
8       - ./app
9     command: uvicorn main:app --host 0.0.0.0 --port 8080
10

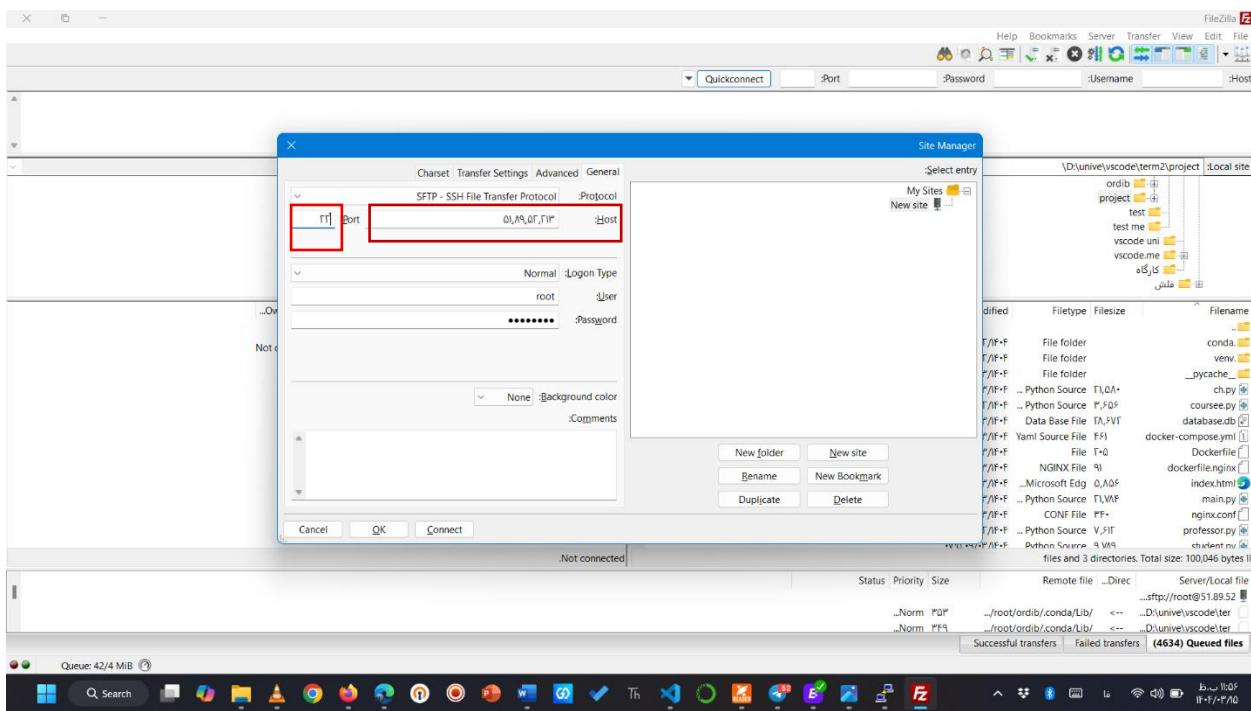
```

```

1 FROM python:3.11-slim
2 WORKDIR /app
3 COPY . /app
4 RUN pip install --no-cache-dir fastapi uvicorn[standard] sqlmodel pydantic
5 EXPOSE 8080
6 CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8080"]

```

بعد از آن در `ctrl+s` وارد کردن ای پی داده شده از هاست و پورت مخصوص لینوکس گزینه `connect` را می‌زنیم و بعد از آن فolder پروژه مورد نظر را اپلود می‌کنیم.



تا اینجا کد قابل اجرا است (`http://asrarahimi.ir:8000/docs`) ولی برای ایجاد `nginx` (برای ایجاد `http://asrarahimi.ir/docs`) باید مراحل زیر را انجام دهیم.

در فolder جدید پروژه را مجدد کپی می‌کنیم و در `docker-compose.yml` زیر را مینویسیم

در همان فolder فایل هایی به نام Dockerfile.nginx و nginx.conf

```
1 server {
2     listen 80;
3     server_name _;
4
5     location / {
6         proxy_pass http://backend:8080;
7         proxy_set_header Host $host;
8         proxy_set_header X-Real-IP $remote_addr;
9         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
10        proxy_set_header X-Forwarded-Proto $scheme;
11    }
12 }
13 }
```

1	for i in range(10):
2	print(i)
3	if i == 5:
4	break
5	else:
6	continue
7	print("Loop has ended")
8	print("Program has ended")
9	print("Program has ended")
10	print("Program has ended")
11	print("Program has ended")
12	print("Program has ended")
13	print("Program has ended")
14	print("Program has ended")
15	print("Program has ended")
16	print("Program has ended")
17	print("Program has ended")
18	print("Program has ended")
19	print("Program has ended")
20	print("Program has ended")
21	print("Program has ended")
22	print("Program has ended")
23	print("Program has ended")
24	print("Program has ended")
25	print("Program has ended")
26	print("Program has ended")
27	print("Program has ended")

```
1 FROM nginx:1.26-alpine
2
3 COPY nginx.conf /etc/nginx/conf.d/default.conf
4
5 EXPOSE 80 443
6
```

و بعد از آن فایل جدید را در فایل زیلا اپلود میکنیم و بعد از آن در پوتی دستور های زیر را به ترتیب اجرا میکنیم:

Ls

که فایل هایی که روی سرور اپلود کردیم را نشان میدهد فایل جدید و فایل دارای nginx

فایل قدیمی Cd

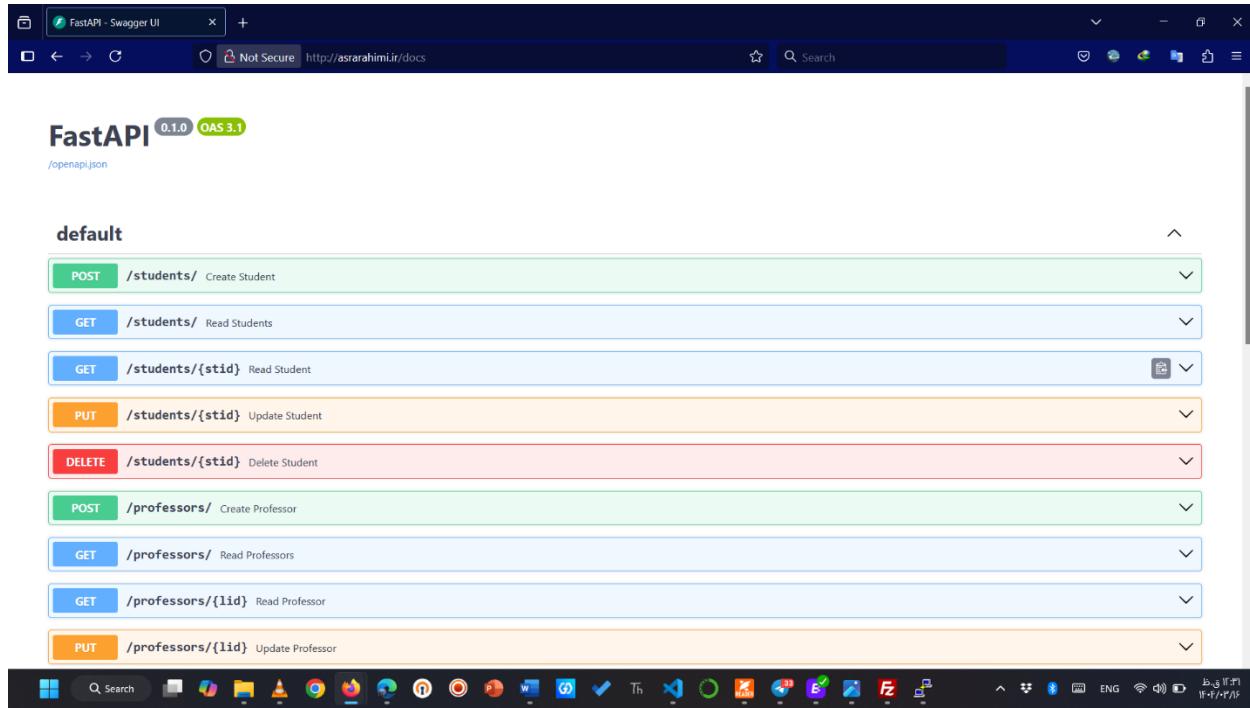
Docker-compose down (توقف و حذف سرویس)

Cd ..

فایل جدید Cd

(راهنمای سرویس) Docker-compose up -d --build

در اینجا کد به صورت کامل با دستور <http://asrarahimi.ir/docs> قابل اجرا است.



توضیح کد:

کاربرد	مطلوب
(مثل) تعریف نوع های جنریک و اختیاری برای تایپدهی List[str], Optional[int])	typing (پایتون)
ساخت اپلیکیشن، مدیریت و استگی ها، تعریف خطاهای پارامتر های کوئری	FastAPI
تعریف مدل ها، فیلدها، جلسه ارتباط با دیتابیس و اجرای کوئری SQL	SQLModel
اعتبار سنجی داده ها، تعریف کلاس پایه، مدیریت خطاهای ولیدیشن	Pydantic
استفاده از الگوهای با قاعده برای بررسی یا استخراج از رشته ها	re (Regular Expressions)
(در) برای اجازه دسترسی بین دامنه ای CORS فعال سازی Front و Backend)	FastAPI.middleware
گرفتن تاریخ و زمان فعلی، استفاده برای فیلدهای زمانی مثل created_at	datetime (پایتون)

```

from typing import Annotated, List, Optional
from fastapi import Depends, FastAPI, HTTPException, Query
from sqlmodel import Field, Session, SQLModel, create_engine, select
from pydantic import validator, ValidationError, BaseModel
import re
from fastapi.middleware.cors import CORSMiddleware
from datetime import datetime

```

نام اصلی (Config Option)	مقدار	توضیح
validate_assignment	True	اعتبارسنجی خودکار هنگام تغییر مقدار فیلد پس از ساخت مدل
extra	"forbid"	"allow", "ignore", یا "forbid": جلوگیری از ورود فیلدهای اضافه (مجاز)
strict	True	فعالسازی بررسی سختگیرانه نوع دادهها در هنگام مقداردهی

```
 1 class Person(SQLModel):
 2     fname: str = Field(max_length=10)
 3     lname: str = Field(max_length=10)
 4     id: str = Field(max_length=10, min_length=10)
 5     born_city: str
 6     birth_date: str
 7     address: str = Field(max_length=100)
 8     postal_code: str = Field(max_length=10, min_length=10)
 9     hphone: str
10     phone: str
11     department: str
12     major:str
13
14     class Config:
15         validate_assignment = True
16         extra = "forbid"
17         strict = True
18
```

که متغیر رشته‌ای به نام `sqlite_file_name` تعریف می‌شود و مقدار "database.db" به آن تخصیص داده می‌شود تعبیین نام فایل دیتابیس SQLite که داده‌ها) مثل جداول مدل‌های Person یا Student در آن ذخیره می‌شوند. این فایل در سیستم فایل محلی مثلاً در کنار فایل‌های پروژه قرار می‌گیرد. متغیر `sqlite_url` با استفاده از یک f-string تعریف می‌شود که URL اتصال به دیتابیس SQLite ایجاد می‌کند. فرمت URL برای SQLite به شکل `sqlite:///path/to/databasse.db` است.

```
 1 sqlite_file_name = "database.db"
 2 sqlite_url = f"sqlite:///({sqlite_file_name})"
 3
 4 connect_args = {"check_same_thread": False}
 5 engine = create_engine(sqlite_url, connect_args=connect_args)
 6
```

نام فایل پایگاهداده SQLite مستحب می‌شود. در اینجا قابل پایگاهداده قرار است نام داشته باشد. اگر این فایل وجود نداشته باشد، SQLite ان را ایجاد می‌کند.	"sqlite_file_name = "database.db"
تعیین مسیر دیتابیس برای SQLAlchemy از چول لسلن برای مسیر مغلق استفاده کرد. و شنید آدرس (URL) برای اتصال با پایگاهداده ساخته می‌شود	"sqlite_url = f"sqlite:///{{sqlite_file_name}}"
SQLite طور پیش فرض فقط اجازه می‌دهد که اتصال به پایگاهداده در همان (thread)*** زد ای استفاده شود که آن را خواسته است. اما بر نرم‌افزارهای مثل FastAPI که چندین ترد (ya async task) دارند، ممکن است خطای بدهد.	connect_args = {"check_same_thread": False}
با تنظیم check_same_thread=False در این محدودیت برداشته می‌شود تا بتوان از اتصال در تردهای مختلف استفاده کرد.	engine = create_engine(sqlite_url, connect_args=connect_args)

بن چند خط کد در مجموع یک اتصال به پایگاهداده SQLite ایجاد می‌کنند که بتوان از آن در برنامه‌هایی مانند SQLModel با FastAP استفاده کرد.

(1) با ایجاد کلاس person شرط ها و لینیتور های مشترک بین 2 کلاس student , professor ایجاد میکنیم تا کلاس student , professor پایگاه داده داشته باشد.

2. پایگاه داده

- اتصال :

- از SQLite با فایل database.db استفاده می‌کند.
- موتور با create_db_and_tables ایجاد شده و create_engine را در زمان راه اندازی می‌سازد.

- سشن :

- تابع Annotated[SessionDep] برای تزریق سشن پایگاه داده با (SessionDep از نوع get_session تعريف شده.)

FastAPI 3. تنظیمات

- اپلیکیشن FastAPI با میان افزار CORS راه اندازی شده تا در خواست های بین منشأ (cross-origin) مجاز باشد (allow_origins=["*"]).
- در زمان شروع اپلیکیشن (@app.on_event("startup")) جداول پایگاه داده ایجاد می‌شوند.

CRUD 4. عملیات

عملیات CRUD (ایجاد، خواندن، بروزرسانی، حذف) را برای سه کلاس انجام می‌دهد.

این کد یک API برای مدیریت اطلاعات دانشجویان، استادی و دروس در یک سیستم دانشگاهی ارائه می‌دهد، با پایگاه داده SQLite، اعتبارسنجی قوی.

course | Table | database.db

sqliteviewer.app/#/database.db/table/course

database.db > course

Records: 3

cid Cname department credit

1	15730	مهندسی	فني مهندسي
2	69895	منابع	اقتصاد
3	17934	رياضي	فني مهندسي

Try Hydejack, a Jekyll theme for hackers, nerds, and academics ↗

professor | Table | database.db

sqliteviewer.app/#/database.db/table/professor

database.db > professor

Records: 1

fname lname id born_city birth_date address

1	مسعود	رمضاني	1234567890	شہزاد	1370/12/10	لرستان

Try Hydejack, a Jekyll theme for hackers, nerds, and academics ↗



