

web Application

DEVELOPEMANT & DEPLOYMENT

Prepared For :

Network course at Isfahan university of
technology

Dr . Ali Fanian

Zahra Sarami - Ali Dakik

Deadline:

Thursday 21th Tir



مقدمه :

در این پروژه قرار است یک وب سایت نمایش دهنده پروفایل کاربران را توسعه داده و پیاده سازی کنیم و بعد از آن با استفاده از یک proxy server که خودمان نوشتیم به آن دسترسی پیدا کنیم.

زبان پیشنهادی و کد نیمه آماده ای که برای backend در اختیار شما قرار میگیرد به زبان flask (از فریم ورک های پایتون برای توسعه ی وب) و برای frontend هم از html, css, js استفاده میشود.

فایل کد های آماده را از اینجا دانلود کنید.

پروژه در سه فاز اصلی برنامه نویسی وب، دیپلوی آن و پیاده سازی proxy server انجام میشود.

جهت آشنایی بیشتر با معماری و خدمات سرویس وب این-ویدیو را مشاهده کنید.

جهت آشنایی بیشتر با متدهای http و عملکرد آنان این [داکیومن](#)ت را بخوانید.

معماری :

در این پروژه حداقل از سه host که در یک شبکه ی Nat قرار دارند استفاده میشود .

دقت کنید که تنظیمات شبکه vm ها را به گونه ای تنظیم کنید که هر یک از ماشین ها آی پی مجزا و منحصر به فرد دریافت کنند و بتوانند یک دیگر را پینگ بگیرند.

• هاست اول :

میزبانی سایت، پایگاه داده و web server در اینجا انجام میشود.

• هاست دوم :

proxy server در اینجا پیاده میشود و واسطی میان ارتباط هاست اول و سوم است.

• هاست سوم :

همان کلاینت است که آدرس آی پی هاست اول را در browser خود جست و جو میکند و به سایت وصل میشود.

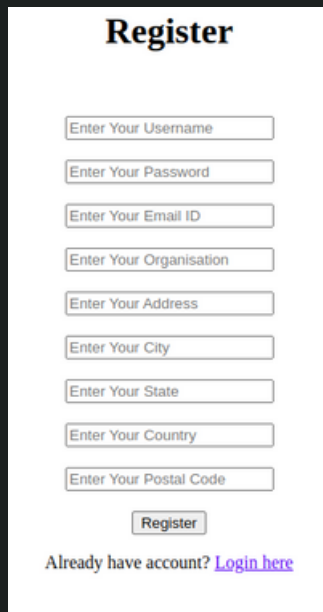


برنامه نویسی وب :

-فاز یک-

ویژگی هایی که سایت به شرح زیر است:

- کاربران باید بتوانند در سایت با وارد کردن اطلاعات زیر ثبت نام شوند :
- بعد از ثبت نام به صفحه لاگین هدایت شده و با استفاده از نام کاربری و رمزعبور وارد شوند.



The screenshot shows a web form titled "Register". It contains several input fields for user registration: "Enter Your Username", "Enter Your Password", "Enter Your Email ID", "Enter Your Organisation", "Enter Your Address", "Enter Your City", "Enter Your State", "Enter Your Country", and "Enter Your Postal Code". Below these fields is a "Register" button. At the bottom, there is a link that says "Already have account? [Login here](#)".

- در این صفحه کاربر اطلاعات خود را دیده و میتواند این اطلاعات به جز نام کاربری را آپدیت کند و یا از صفحه کاربری خود خارج شود .

شما باید علاوه بر ویژگی های موجود دو ویژگی زیر را پیاده سازی کنید :

- پیاده سازی صفحه Home که امکان دیدن لیست همه کاربران موجود در دیتابیس به همراه ایمیلشان در آن وجود داشته باشد
- ویژگی دیلیت اکانت برای کاربران

بهبود بخش فرانت اند سایت،(به طور مثال با افزودن کد جاوا اسکریپت) نمره ی امتیازی دارد.

پایگاه داده :

پایگاه داده ی مورد استفاده در این پروژه mysql است . اگر بخواهید میتوانید از sqlite یا دیگر روش ها استفاده کنید و شیوه کار با آن را در ارائه توضیح دهید. ابتدا نیاز است آن را نصب کرده و با اجرای کد config.sql که در فایل پروژه در اختیار شما قرار داده شده است، دیتابیس و table مربوط به پروژه را ایجاد کنید . اگر از محیط ترمینال استفاده میکنید دستورات زیر برای اجرای کد sql داده شده به کار میرود :

`sudo apt-get install mysql-server // نصب`

`mysql -u yourusername -p yourpassword // باید این کاربر از قبل ایجاد شده باشد`

`mysql> source \path\to\config.sql;`



سپس این قسمت کد app.py را با توجه به ویژگی های یوزری که به دیتابیس دسترسی دارد تغییر دهید.

```
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = 'password'
app.config['MYSQL_DB'] = 'profileApp'
```

محیط تست :

ابتدا در دایرکتوری پروژه یک محیط مجازی پایتونی ایجاد کرده و نیازمندی های پروژه را در آن نصب کنید:

```
virtualenv venv // ساخت محیط مجازی
source venv/bin/activate // فعالسازی محیط مجازی
```

```
pip install -r requirements.txt // نصب نیازمندی ها در آن محیط
flask run // برای تست صحت عملکرد پروژه
```

در صورتی که از صحت عملکرد کد و پایگاه داده تا به اینجا مطمئن شدید میتوانید وارد فاز دیپلوی شوید.

پیاده سازی :

-فاز دو-

اجرای برنامه Flask با استفاده از سرور داخلی (flask run) Flask برای توسعه خوب است زیرا راه اندازی سریع و ساده ای دارد و به شما امکان می دهد به سرعت تغییرات کد خود را تست کنید. با این حال، در محیط production مشکلات زیر را خواهید داشت:

- عدم پایداری و عملکرد: سرور داخلی Flask برای بارهای سنگین و تعداد بالای درخواست ها بهینه سازی نشده است.
- مدیریت درخواست ها: Flask به تنهایی نمیتواند درخواست های همزمان زیادی را مدیریت کند، که ممکن است منجر به مشکلات عملکردی و کاهش سرعت برنامه شود.
- امنیت: سرور داخلی Flask فاقد امکانات امنیتی پیشرفته ای است که برای محیط تولید ضروری است.



برای همین در این پروژه از یک وب سرور NGINX همراه با یک سرور WSGI مانند GUNICORN برای دیپلوی برنامه FLASK خود استفاده میکنیم .

مرحله ۱: نصب NGINX روی ماشین مجازی (VM)
ابتدا باید NGINX را روی ماشین مجازی خود نصب کنید. می‌توانید این کار را با استفاده از مدیر بسته‌ها انجام دهید. به عنوان مثال، در سیستم‌های مبتنی بر دبیان مانند اوبونتو:

```
sudo apt update  
sudo apt install nginx
```

مرحله ۲: نصب و پیکربندی GUNICORN
Gunicorn یک سرور WSGI برای برنامه‌های وب پایتون است. می‌توانید از Gunicorn برای سرویس‌دهی به برنامه Flask خود استفاده کنید و سپس از Nginx برای پراکسی معکوس به Gunicorn بهره ببرید.

نصب Gunicorn:

```
pip install gunicorn
```

اجرای Gunicorn:
به دایرکتوری برنامه Flask خود بروید و Gunicorn را اجرا کنید:

```
gunicorn --bind 0.0.0.0:8000 app:app
```

مرحله ۳: پیکربندی NGINX
ایجاد یک فایل پیکربندی جدید برای NGINX:

```
sudo nano /etc/nginx/sites-available/flask_app
```



اضافه کردن پیکربندی زیر:

```
server {  
    listen 80;  
    server_name your_domain_or_IP;  
  
    location / {  
        proxy_pass http://127.0.0.1:8000;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
        proxy_set_header X-Forwarded-Proto $scheme;  
    }  
  
    location /static {  
        alias /path/to/your/app/static;  
    }  
}
```

فعال کردن پیکربندی: your_domain_or_IP را با آدرس IP یا نام دامنه ماشین مجازی خود جایگزین کنید و مسیر path/to/your/app/static/ را با مسیر فایل‌های استاتیک برنامه Flask خود جایگزین کنید.

```
sudo ln -s /etc/nginx/sites-available/flask_app /etc/nginx/sites-enabled
```

تست پیکربندی NGINX:

```
sudo nginx -t
```

راه‌اندازی مجدد NGINX:

```
sudo systemctl restart nginx
```

مرحله ۴: دسترسی به برنامه Flask از ماشین دیگر

پیدا کردن آدرس IP ماشین میزبان:

روی ماشین میزبان، آدرس IP آن را پیدا کنید و مرورگر وب خود را روی ماشین کلاینت (هاست سه) باز کرده و به آدرس `http://<VM_IP>` بروید. این باید برنامه Flask شما را نمایش دهد.



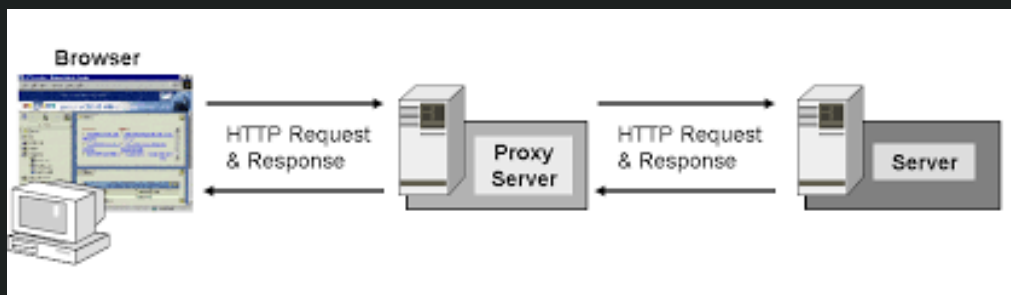
راه اندازی یک HTTP PROXY :

- فاز سه -

تعریف http proxy

همانطور که در فصل 2 توضیح داده شد عملیات proxy پروتکل http در لایه کاربرد انجام می گیرد.

به عبارتی درخواست کاربران برای سرور پراکسی ارسال می شود سرور پراکسی داده ها را دریافت و از جانب خود به سرور اصلی درخواست را ارسال می کند و پاسخ دریافتی را برای کاربر ارسال می کند. در نتیجه می توان گفت سرور پراکسی دارای دو نقش سروری و کلاینتی است. سروری است چون روی یک پورت گوش می کند و درخواست ارسالی کاربر را دریافت و کلاینتی است چون با سرور اصلی ارتباط برقرار می کند .



در شکل فوق ملاحظه می شود در مرورگر کلاینت با توجه به این که پراکسی برای ارتباط http ان ست شده است درخواست خود را برای سرور اصلی که نام آن server است ارسال می کند. اما چون در مرورگر proxy ست شده است درخواست بجای این که برای server به صورت مستقیم برود برای پراکسی ارسال می شود . پراکسی داده ها را دریافت و از جانب خود به server درخواست می فرستد. به عبارتی server، کلاینت را مرورگر کاربر نمی بیند بلکه ماشین پراکسی می بیند.

آشنایی با scrapy :

کتابخانه Scrapy یک ابزار قدرتمند و چند منظوره در پایتون است که برای دستکاری بسته های شبکه استفاده می شود.

با استفاده از Scrapy، می توان بسته های شبکه را ایجاد، ارسال، دریافت و تحلیل کرد. این کتابخانه برای اهداف مختلفی از جمله آزمایش نفوذ، تحلیل شبکه، ایجاد بسته های سفارشی و شبیه سازی پروتکل های شبکه مورد استفاده قرار می گیرد.



برخی از ویژگی‌های مهم Scapy عبارتند از:

- **ایجاد و ارسال بسته‌ها:** با Scapy می‌توان بسته‌های شبکه با انواع مختلف پروتکل‌ها را ایجاد و به شبکه ارسال کرد.
- **دریافت و تحلیل بسته‌ها:** می‌توان بسته‌های دریافتی از شبکه را تحلیل و بررسی کرد.
- **پشتیبانی از پروتکل‌های مختلف:** Scapy از بسیاری از پروتکل‌های شبکه مانند IP, TCP, UDP, ICMP, ARP و بسیاری دیگر پشتیبانی می‌کند.
- **سفارشی‌سازی بسته‌ها:** امکان ایجاد بسته‌های سفارشی با مقادیر دلخواه برای فیلدهای مختلف پروتکل‌ها وجود دارد.
- **تست و شبیه‌سازی حملات:** Scapy می‌تواند برای شبیه‌سازی حملات شبکه‌ای و تست امنیت شبکه مورد استفاده قرار گیرد.

برای اطلاعات بیشتر می‌توانید به مستندات رسمی Scapy مراجعه کنید. چند نمونه کارکرد های این کتابخانه در پیوست آورده شده است.

قطعه کدی که در اختیار شما قرار گرفته است نقش یک TCP bridge را ایجاد می‌کند. یعنی سرور روی یک آدرس و پورت گوش می‌کند و هر بسته‌ای که دریافت می‌کند را برای یک سرور دیگر می‌فرستد و نتیجه را باز می‌گرداند. کد را اجرا کنید و عملکرد آن را از ماشین مجازی سوم تست کنید.

ایجاد یک HTTP Proxy (بخش امتیازی)

این قطعه کد را تغییر دهید تا بتواند به عنوان یک HTTP Proxy کار کند. برای این کار باید از کتابخانه scapy کمک بگیرید.

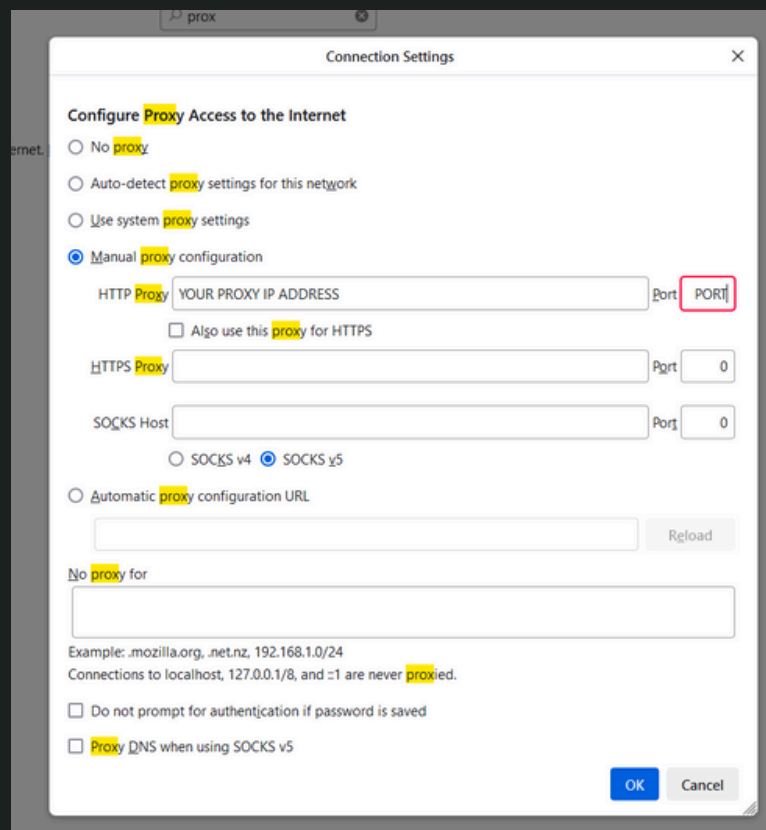
برای تست پروکسی باید آدرس آن را در مرورگر تنظیم کنید و از آن به بعد تمام درخواست‌های HTTP به سمت سرور شما می‌آید. سرور شما باید درخواست‌ها را دریافت کند و به سمت سرور مورد نظر بفرستد و پاسخ را برای مرورگر ارسال کند.

نکته: در هنگام استفاده از http proxy همیشه با CTRL+F5 صفحه مرورگر را بروز کنید زیرا با توجه به این که تمام مکانیزم‌های http proxy پیاده سازی نمی‌شوند، ممکن است محتوای صفحه‌ای که برای بار اول نمایش داده می‌شود cache شود و همه آدرس‌ها یک صفحه ثابت را نمایش دهند.



نکته مهم: وقتی روی مرورگر proxy تنظیم می‌شود، header پروتکل http اندکی تغییر می‌کند، باید این تغییر را پیدا کنید و در کد http proxy مدیریت کنید تا سیستم درست کار کند. برای پیدا کردن تغییر مورد نظر از نرم افزار wireshark استفاده کنید و هدر http را در دو حالت با هم مقایسه کنید. اگر تغییر مورد نظر را پیدا نکنید در این صورت برنامه شما درست کار نخواهد کرد.

تنظیم پروکسی در فایرفاکس:



The screenshot shows the 'Connection Settings' dialog box in Firefox. The 'Configure Proxy Access to the Internet' section has four radio buttons: 'No proxy', 'Auto-detect proxy settings for this network', 'Use system proxy settings', and 'Manual proxy configuration'. The 'Manual proxy configuration' option is selected. Below it, there are input fields for 'HTTP Proxy' (containing 'YOUR PROXY IP ADDRESS') and 'Port' (containing 'PORT'). There is a checkbox 'Also use this proxy for HTTPS' which is unchecked. Below that are input fields for 'HTTPS Proxy' and 'Port' (both containing '0'). There are also input fields for 'SOCKS Host' and 'Port' (both containing '0'). Below these are two radio buttons: 'SOCKS v4' and 'SOCKS v5', with 'SOCKS v5' selected. There is an option 'Automatic proxy configuration URL' with an input field and a 'Reload' button. Below that is a section 'No proxy for' with an input field. At the bottom, there are two checkboxes: 'Do not prompt for authentication if password is saved' and 'Proxy DNS when using SOCKS v5'. The 'OK' and 'Cancel' buttons are at the bottom right.



پیوست :

در این قسمت چند نمونه از کاربرد های scapy آورده شده است.

نصب کتابخانه // `sudo apt-get install python3-scapy`

شنود بسته های TCP:

این قسمت نشان می دهد که چگونه می توان بسته ها را بر اساس پورت های مبدأ و مقصد فیلتر و نمایش داد.

```
1 # Import necessary modules
2 from scapy.all import *
3
4 # Define a packet handler function to filter TCP packets based on ports
5 def packet_handler(packet):
6     if packet.haslayer(TCP):
7         tcp_layer = packet.getlayer(TCP)
8         ip_layer = packet.getlayer(IP)
9         print(f"TCP Packet: {ip_layer.src}:{tcp_layer.sport} -> {ip_layer.dst}:{tcp_layer.dport}")
10
11 # Start sniffing packets with a filter for TCP packets only
12 print("Starting to sniff TCP packets...")
13 sniff(filter="tcp", prn=packet_handler, count=5)
14 |
```

ایجاد و ارسال یک بسته:

```
# Import necessary modules
from scapy.all import *

# Define source and destination ports
src_port = 12345
dst_port = 80

# Create a TCP packet with specific source and destination ports
tcp_packet = IP(dst="8.8.8.8")/TCP(sport=src_port, dport=dst_port)

# Display the packet summary
print("TCP Packet Summary:")
print(tcp_packet.summary())

# Send the packet
send(tcp_packet)
```



استفاده از SCAPY برای کار با لایه HTTP :

packet.haslayer(HTTPRequest) بررسی می‌کند که آیا بسته دارای لایه درخواست HTTP است. اگر یک درخواست HTTP یافت شود، جزئیاتی مانند روش HTTP، میزبان، مسیر و عامل کاربر استخراج و چاپ می‌شوند.

```
1 # Import necessary modules from Scapy
2 from scapy.all import *
3 from scapy.layers.http import HTTP, HTTPRequest
4
5 def http_packet_callback(packet):
6     # Check if the packet has an HTTP layer
7     if packet.haslayer(HTTPRequest):
8         # Extract HTTP request details
9         http_layer = packet[HTTPRequest]
10        print(f"HTTP Method: {http_layer.Method.decode('utf-8')}")
11        print(f"Host: {http_layer.Host.decode('utf-8')}")
12        print(f"Path: {http_layer.Path.decode('utf-8')}")
13        print(f>User-Agent: {http_layer.User_Agent.decode('utf-8')}")
14
15 # Start sniffing packets with a filter for HTTP requests
16 print("Starting packet sniffing...")
17 sniff(filter="tcp port 80", prn=http_packet_callback, store=False)
18
```



نکات پایانی:

اختیارات :

در صورتی که علاقه مند باشید هریک از بخش ها را با فریموک یا تکنولوژی دیگری پیاده سازی کنید، میتوانید با هماهنگی با دستیاران آموزشی از آنها استفاده کنید که لازمه ی آن تسلط کافی شما برآن ابزار است .

مهلت تحویل :

شما تا پایان مهلت ددلاین فرصت دارید فایل کد پروژه به همراه فایل های پیوست آن را به صورت یک فایل فشرده (Zip) در سامانه ی یکتا در ماژول مربوطه بارگزاری کنید .

کد هر vm را در یک پوشه ی مجزا قرارداداده و بعد فشرده کنید.

ارائه :

ارائه ی پروژه به صورت فردی و مجازی بوده و باید بتوانید پروژه را در سیستم خود اجرا کنید همچنین باید بتوانید کد ، نحوه ی اجرای برنامه و سیاست های پیاده سازی شده در برنامه خود را توضیح دهید .
تاریخ ارائه متاقبا اعلام میشود .

نمره دهی :

مجموع نمره ی بخش های امتیازی از ۱۰۰ محاسبه شده و توزیع آن به این صورت است :

- بهبود فرانت اند : 20
- پیاده سازی پروکسی سرور : 80

راه های ارتباطی :

Telegram : @zhra_sarami

Telegram : @X_AFDK_X

