



موضوع: داکرایز کردن پروژه

درس: رایانش ابری

استاد: دکتر هدی طاهری

دانشجوی ناظر: مهندس امیر آبرون

فهرست

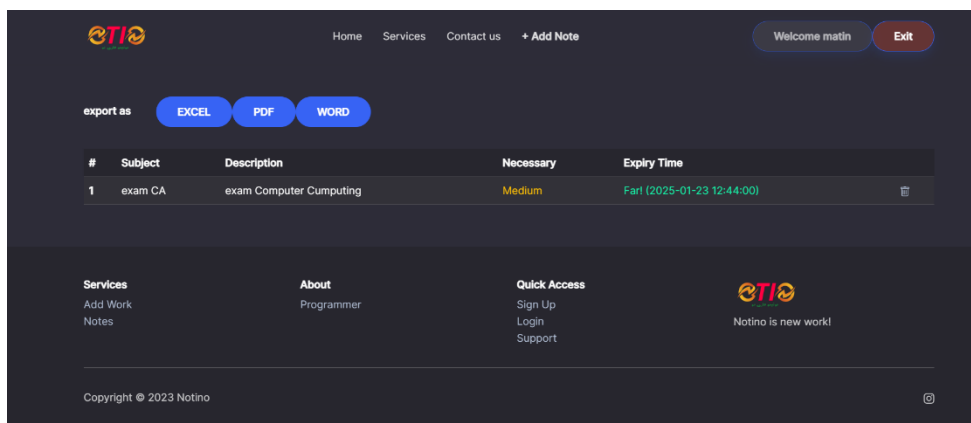
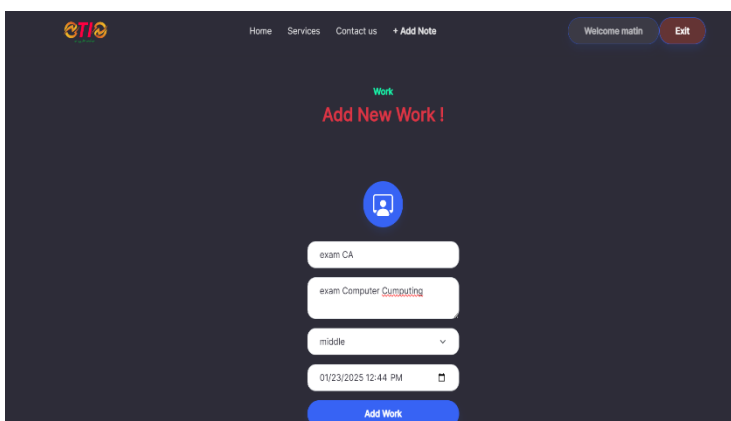
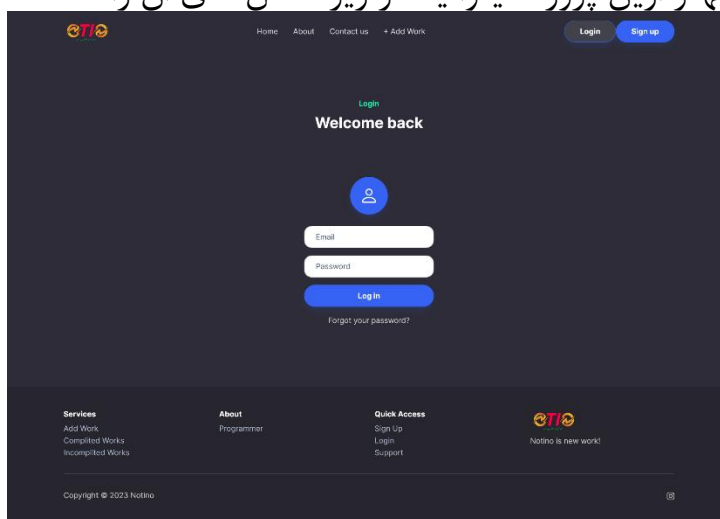
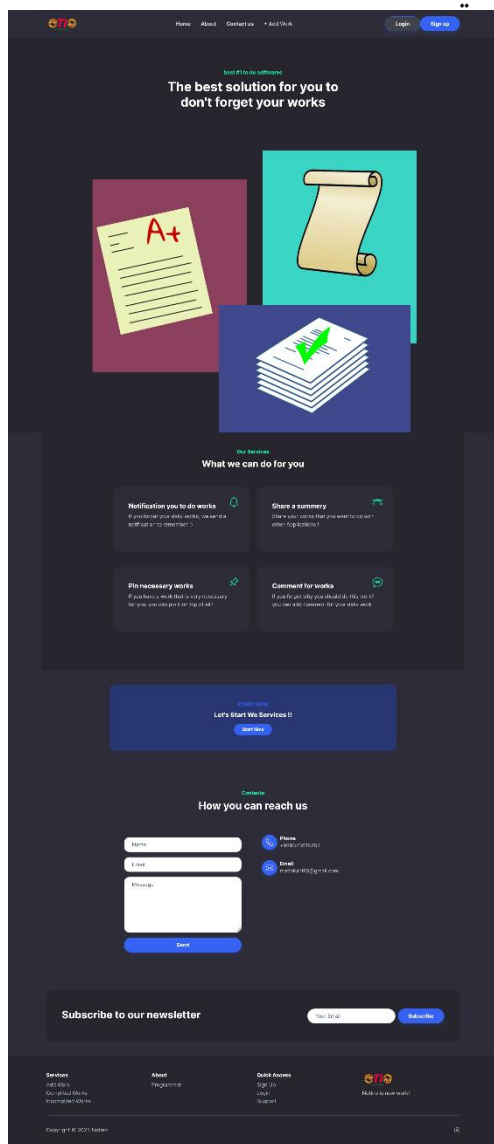
- موضوع پروژه 3
- توضیحات بیشتر راجب پروژه 4
- توضیحاتی راجب داکر 5
- نصب داکر در اوبونتو 6
- داکرایز کردن پروژه.. 7
- استفاده از postman 16

موضوع پروژه

من یک پروژه ساده با استفاده از php بدون استفاده از هیچ فریم ورکی نوشتم که در واقع اسم آن را notino گذاشتم که به معنای یادداشتی جدید است.

کاربران میتوانند با ثبتنام در سایت وارد پنل خود شوند و یادداشتی جدید اضافه کنند همچنین تاریخ پایان یادداشت و اولویت آن را اضافه میتوانند کنند.

برای درک بهتر از این پروژه میتوانید در زیر عکس های آن را مشاهده کنید.



توضیحات بیشتر راجب پروژه

لازم به ذکر است لوگوی پروژه توسط خودم با استفاده از فوتوشاپ طراحی شده که همان notino است.



این پروژه رو دو سال قبل که نوشتم داکرایز نشده بود ولی برای درس رایانش ابری این پروژه را داکرایز کردم و در گیتهاب قرار دادم

فرانت پروژه با استفاده از bootstrap studio ساخته شده چون من دانش کافی برای ساخت یک فرانت خوب را ندارم پس سعی کردم از نرم افزار هایی که با drag & drop میتوان فرانت خوب ساخت استفاده کنم چرا که فرانت و gui خوب اعتماد بیشتری را جلب میکند.

همچنین سعی کردم بک اند را به صورت کد نویسی های جدا بنویسم که بعدا برای تغییر یا اضافه کردن چیزی راحت باشم و نیاز به گشتن کل کد ها نباشد.

در واقع من این پروژه را در درس رایانش ابری به شکل گفته شده در درس داکرایز کردم

توضیحاتی راجب داکر

داکر (Docker) یک پلتفرم منبع باز است که به توسعه دهندگان و مدیران سیستم ها اجازه می دهد تا برنامه ها را در قالب کانتینرها (Containers) اجرا و مدیریت کنند. کانتینرها محیطی ایزوله و سبک وزن هستند که همه نیازمندی های یک برنامه، مانند کتابخانه ها، فایل ها و وابستگی ها، را در خود جای می دهند.

مزایای داکر:

1. پرتابل بودن: کانتینرها در هر محیطی (لپ تاپ، سرور یا فضای ابری) اجرا می شوند.
2. ایزولاسیون: هر کانتینر مستقل از دیگر کانتینرها عمل می کند و از منابع سیستم جداست.
3. سبک وزنی: در مقایسه با ماشین های مجازی (VMS)، کانتینرها سریع تر و کم حجم تر هستند.
4. مقیاس پذیری: اجرای همزمان چندین کانتینر به سادگی امکان پذیر است.
5. مدیریت وابستگی ها: هر کانتینر دارای تمام نیازمندی های برنامه است و به نصب دستی کتابخانه ها نیازی نیست.

مفاهیم اصلی:

1. ایمیج (Image): قالبی آماده که شامل برنامه و وابستگی های آن است. ایمیج به عنوان پایه برای ایجاد کانتینر استفاده می شود.
2. کانتینر (Container): نمونه اجرایی از یک ایمیج که برنامه را اجرا می کند.
3. داکر فایل (Dockerfile): یک فایل متنی که مراحل ساخت یک ایمیج را تعریف می کند.

کاربردها:

- توسعه و تست نرم افزار در محیط های یکسان.
- مدیریت زیرساخت ها در مقیاس بالا.
- اجرای برنامه های وب، پایگاه های داده و میکروسرویس ها.



نصب داکر در اوبونتو

چون در کلاس گفته شد که مجاز نیستیم با WSL در ویندوز از داکر استفاده کنیم برای همین یک سیستم لینوکسی را نصب کردم که من توزیعی که اختیار کردم اوبونتو هست شما میتوانید از توزیع های دیگر استفاده کنید.

میتوانید از سایت خود داکر راهنمای نصب ان را مشاهده کنید

<https://docs.docker.com/engine/install/ubuntu>

برای نصب داکر در اوبونتو

1) # Add Docker's official GPG key:

```
sudo apt-get update
```

```
sudo apt-get install ca-certificates curl
```

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o  
/etc/apt/keyrings/docker.asc
```

```
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Add the repository to Apt sources:

```
echo \
```

```
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]  
https://download.docker.com/linux/ubuntu \
```

```
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
```

```
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

2) sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

3) sudo docker run hello-world

داکرایز کردن پروژه

من برای داکرایز کردن پروژه سه تا فایل

Dockerfile

docker-compose.db.yml

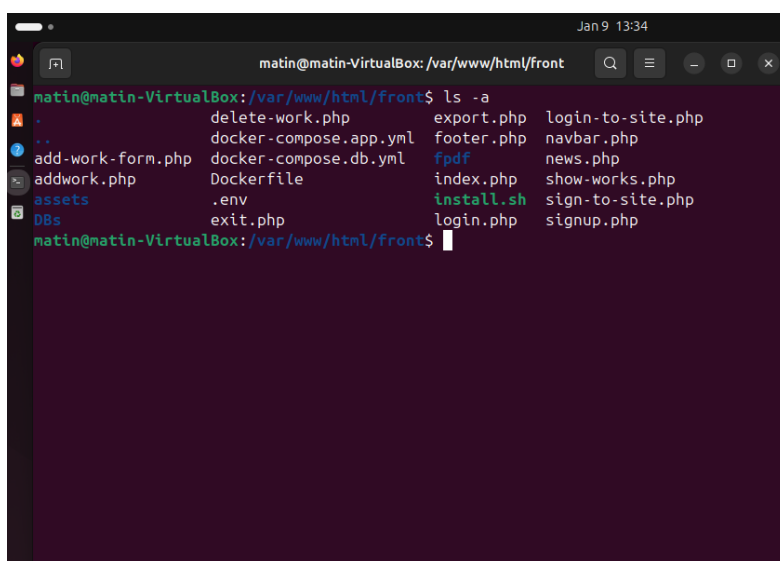
docker-compose.app.yml

در پروژه ام ایجاد کردم که فایل های داکر هست.

همچنین یک فایل **env** هم ایجاد کردم که یک سری متغیر های مربوط به فایل های داکر را در ان تعریف کرده ام.

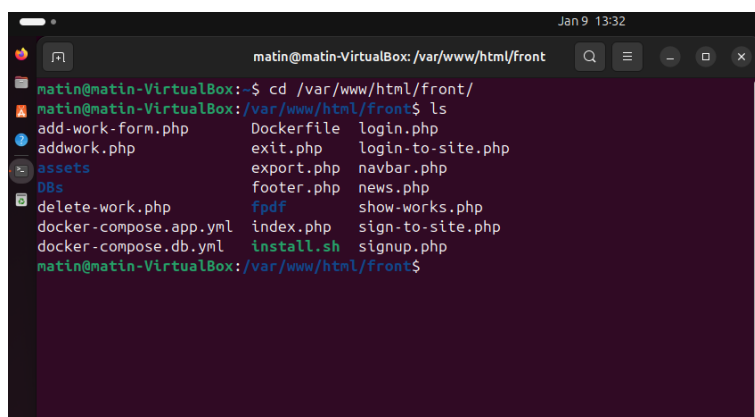
.env

حال یک **ls** از فولدری که پروژه در آن است میگیرم که لیست فایل های موجود در پروژه را نشان میدهد.



```
matin@matin-VirtualBox: /var/www/html/front$ ls -a
.          delete-work.php  export.php  login-to-site.php
..         docker-compose.app.yml  footer.php  navbar.php
add-work-form.php  docker-compose.db.yml  fpdf       news.php
addwork.php      Dockerfile         index.php  show-works.php
assets           .env               install.sh  sign-to-site.php
DBs             exit.php           login.php  signup.php
matin@matin-VirtualBox: /var/www/html/front$
```

از آنجایی که **env** هست این فایل وقتی **ls** میگیریم نمیداد و برای مشاهده این فایل میتوانیم از دستور **ls -a** استفاده کنیم



```
matin@matin-VirtualBox: /var/www/html/front$ cd /var/www/html/front/
matin@matin-VirtualBox: /var/www/html/front$ ls
add-work-form.php  Dockerfile  login.php
addwork.php        exit.php   login-to-site.php
assets             export.php  navbar.php
DBs               footer.php  news.php
delete-work.php   fpdf       show-works.php
docker-compose.app.yml  index.php  sign-to-site.php
docker-compose.db.yml  install.sh  signup.php
matin@matin-VirtualBox: /var/www/html/front$
```

حال داخل تک تک فایل های موجود را با هم بررسی میکنیم.

فایل [.env](#):

DB_HOST=db

DB_USER=root

DB_PASSWORD=root

DB_NAME=notino

MYSQL_ROOT_PASSWORD=root

MYSQL_DATABASE=notino

MYSQL_USER=root

MYSQL_PASSWORD=root


```

1  # Use an official PHP image with Apache
2  FROM ubuntu:20.04
3
4  ENV DEBIAN_FRONTEND=noninteractive \
5      TZ=Asia/UTC
6
7  RUN apt-get update && apt-get install -y \
8      curl \
9      git \
10     unzip \
11     apache2 \
12     libapache2-mod-php \
13     php-cli \
14     php-mysql \
15     php-zip \
16     php-xml \
17     php-strings \
18     php-curl \
19     && apt-get clean \
20     && rm -rf /var/lib/apt/lists/*
21
22 # Enable necessary Apache modules
23 RUN a2enmod rewrite
24
25 FROM php:8.2-apache
26 RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
27
28 # Install Composer
29 RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer
30
31 # Copy project files to the container
32 COPY . /var/www/html/
33
34 # Set working directory
35 WORKDIR /var/www/html/
36
37 # Set file permissions
38 RUN chown -R www-data:www-data /var/www/html/ && chmod -R 755 /var/www/html/
39
40 # Install PHP libraries using Composer
41 #RUN composer require phpooffice/phpspreadsheet phpooffice/phpword
42 # Install required PHP extensions
43
44 RUN apt-get update && apt-get install -y \
45     libzip-dev \
46     libgd-dev \
47     && docker-php-ext-install zip gd
48
49 # Install PHP libraries using Composer
50 RUN composer require phpooffice/phpspreadsheet phpooffice/phpword
51
52 # Restart Apache to apply changes
53 CMD ["apache2-foreground"]

```

از انجایی که این فایل طولانی بود من ترجیح دادم اسکرین شاتی از آن را بیاورم جهت مشاهده میتوانید از گیت‌هابم نگاه کنید.

در این فایل دستورات مورد نیاز برای نصب آپاچی php8 و کامپوسر و ... آورده شده است.

در صورتی که سیستم عامل شما فرق دارد میتوانید آن را تغییر دهید.

فایل [:docker-compose.app.yml](#)

از انجایی که در پروژه گفته شد دیتابیس و برنامه خود را در دو کانتینر جدا قرار دهید من این فایل را ساختم که به این صورت است

```
1  version: '3.8'
2
3  services:
4    app:
5      build:
6        context: .
7        dockerfile: Dockerfile
8      container_name: notino_app
9      volumes:
10       - ./var/www/html
11      ports:
12       - "8080:80"
13      networks:
14       - shared_network
15      environment:
16       - DB_HOST=${DB_HOST}
17       - DB_USER=${DB_USER}
18       - DB_PASSWORD=${DB_PASSWORD}
19       - DB_NAME=${DB_NAME}
20
21     networks:
22       shared_network:
23         external: true
```

در این فایل تنظیمات و پیکربندی برای اجرای برنامه پروژه قرار گرفته است که میگوید تحت پورت 80 یا 8080 و کانتینر ان notion_app و داکر فایل آن همان داکر فایلی است که در بالا گفته شد

. همچنین گفته با share_link شبکه میشه

یعنی با کانتینر دیتابیس ارتباط برقرار میکنه

و دیگر تنظیمات

فایل [:docker-compose.db.yml](#)

این فایل پیکربندی برای اجرای یک سرویس پایگاه داده MySQL با استفاده از Docker Compose است.

```
1  version: '3.8'
2
3  services:
4    db:
5      image: mysql:8.0
6      container_name: mysql_db
7      ports:
8        - "3306:3306"
9      volumes:
10       - db_data:/var/lib/mysql
11      networks:
12        - shared_network
13      environment:
14        - MYSQL_ROOT_PASSWORD=${MYSQL_ROOT_PASSWORD}
15        - MYSQL_DATABASE=${MYSQL_DATABASE}
16        - MYSQL_USER=${MYSQL_USER}
17        - MYSQL_PASSWORD=${MYSQL_PASSWORD}
18
19  volumes:
20    db_data:
21
22  networks:
23    shared_network:
24      external: true
```

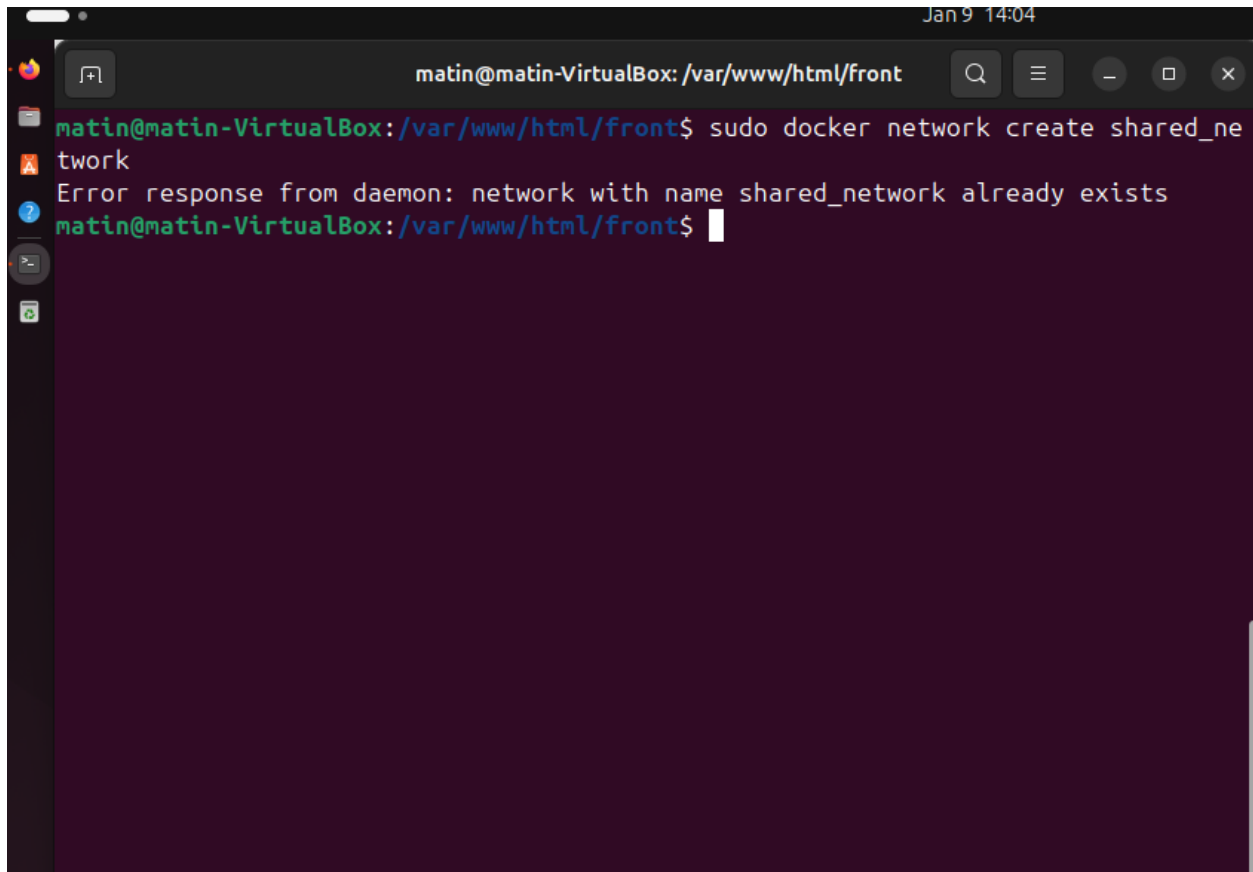
این فایل یک کانتینر با نام `mysql_db` است و از نسخه `mysql 8` استفاده میکند و پورت آن `3306` است و ولوم `db_data` برای ذخیره داده‌های پایگاه داده به مسیر `var/lib/mysql/` در کانتینر متصل می‌شود. این کار باعث پایداری داده‌ها حتی پس از حذف کانتینر می‌شود.

و همچنین با `share_network` شبکه شده تا با کانتینر `app` ارتباط برقرار کند.

حال که فایل های داکر را ساختیم با استفاده از دستور

`docker network create shared_network`

یک شبکه `share_network` برای ارتباط بین دو کانتینر میسازیم

A screenshot of a terminal window with a dark background. The window title is "Jan 9 14:04" and the address bar shows "matin@matin-VirtualBox: /var/www/html/front". The terminal shows a command being executed: "sudo docker network create shared_network". The output is "Error response from daemon: network with name shared_network already exists". The prompt "matin@matin-VirtualBox: /var/www/html/front\$" is visible at the end of the line.

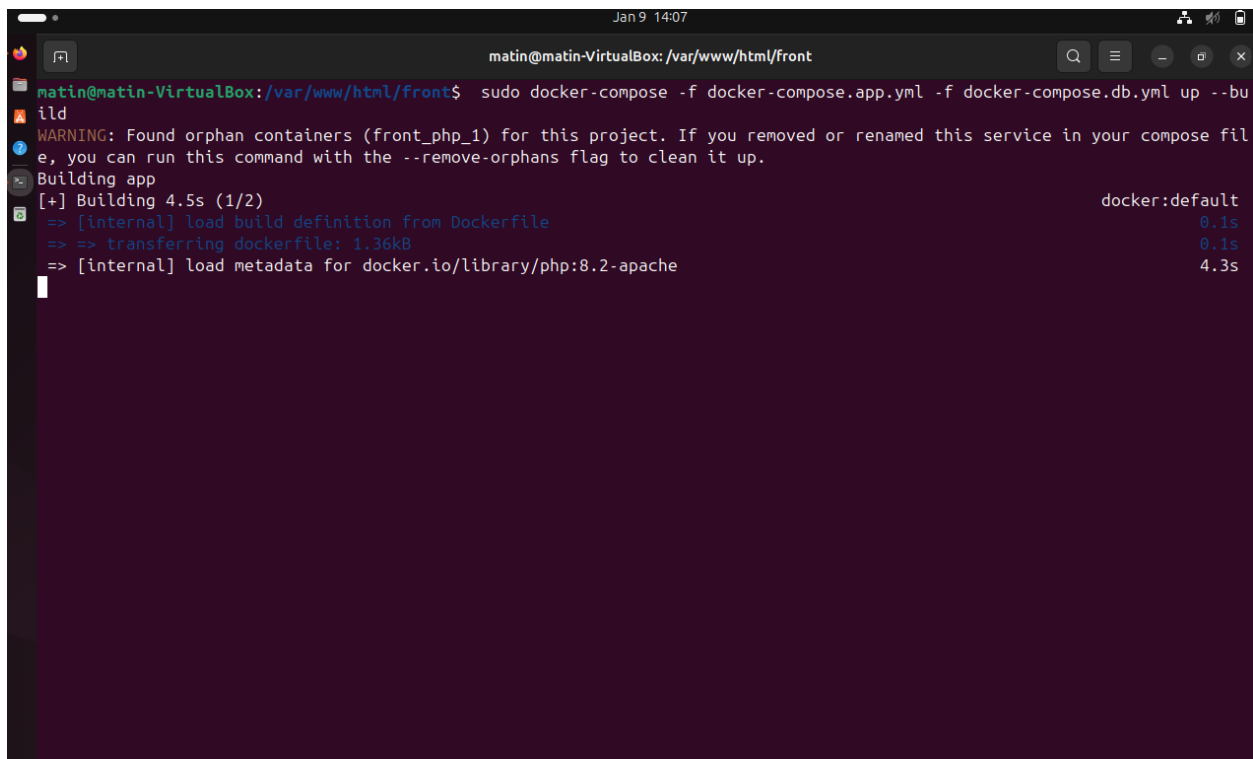
```
matin@matin-VirtualBox: /var/www/html/front$ sudo docker network create shared_network
Error response from daemon: network with name shared_network already exists
matin@matin-VirtualBox: /var/www/html/front$
```

چون من قبلا زدم میگه این نتورک از قبل وجود دارد.

حال تقریبا کار تمامه و فقط کافیه build کنیم و داکر را up کنیم که با دستور زیر هم میتوانیم build کنیم و پس از build خودکار up میشود.

همچنین از انجایی که داکر مارا تحریم کرده است بهتر است با تحریم شکن و یا فیلتر شکن برویم.

```
sudo docker-compose -f docker-compose.app.yml -f docker-compose.db.yml  
up --build
```



```
Jan 9 14:07  
matin@matin-VirtualBox: /var/www/html/front  
matin@matin-VirtualBox:/var/www/html/front$ sudo docker-compose -f docker-compose.app.yml -f docker-compose.db.yml up --bu  
ild  
WARNING: Found orphan containers (front_php_1) for this project. If you removed or renamed this service in your compose fil  
e, you can run this command with the --remove-orphans flag to clean it up.  
Building app  
[+] Building 4.5s (1/2)                                docker:default  
=> [internal] load build definition from Dockerfile                                0.1s  
=> => transferring dockerfile: 1.36kB                                           0.1s  
=> [internal] load metadata for docker.io/library/php:8.2-apache                4.3s
```

این فرایند ممکن است طول بکشد ولی چون من از قبل اینکار را کرده ام در cache داکر است و سریع تر انجام میشود

من در حین کار به ارور زیر برخورد کردم که دلایلش این بود چون سرویس `mysql` ام از قبل مورد استفاده بود باید غیر فعال میشد

```
Jan 9 14:09
matin@matin-VirtualBox: /var/www/html/front

=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.36kB
=> [internal] load metadata for docker.io/library/php:8.2-apache
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [stage-1 1/8] FROM docker.io/library/php:8.2-apache@sha256:56d8b242c3430aa5eb27fc112194d2b22c5c72fe7c3b3db094063
=> [internal] load build context
=> => transferring context: 6.54kB
=> CACHED [stage-1 2/8] RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
=> CACHED [stage-1 3/8] RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --file
=> CACHED [stage-1 4/8] COPY . /var/www/html/
=> CACHED [stage-1 5/8] WORKDIR /var/www/html/
=> CACHED [stage-1 6/8] RUN chown -R www-data:www-data /var/www/html/ && chmod -R 755 /var/www/html/
=> CACHED [stage-1 7/8] RUN apt-get update && apt-get install -y libzip-dev libgd-dev && docker-php-ext
=> CACHED [stage-1 8/8] RUN composer require phpoffice/phpspreadsheet phpoffice/phpword
=> exporting to image
=> => exporting layers
=> => writing image sha256:426fe9dbdae9a333096c7e75d66b89dad1c46d28ee899003fa019ae74a0cc42f
=> => naming to docker.io/library/front_app
notino_app is up-to-date
Starting mysql_db ...
Starting mysql_db ... error

ERROR: for mysql_db Cannot start service db: driver failed programming external connectivity on endpoint mysql_db (fa49d87592f32d00d31fbf7ea7647fdd2a957ca07308b2fad1f575ba8965b288): failed to bind port 0.0.0.0:3306/tcp: Error starting userland proxy: listen tcp4 0.0.0.0:3306: bind: address already in use

ERROR: for db Cannot start service db: driver failed programming external connectivity on endpoint mysql_db (fa49d87592f32d00d31fbf7ea7647fdd2a957ca07308b2fad1f575ba8965b288): failed to bind port 0.0.0.0:3306/tcp: Error starting userland proxy: listen tcp4 0.0.0.0:3306: bind: address already in use
ERROR: Encountered errors while bringing up the project.
matin@matin-VirtualBox: /var/www/html/front$
```

و برای حل این ارور هم دستورات زیر را زدم و سپس دوباره تلاش کردم

```
Jan 9 14:12
matin@matin-VirtualBox: /var/www/html/front

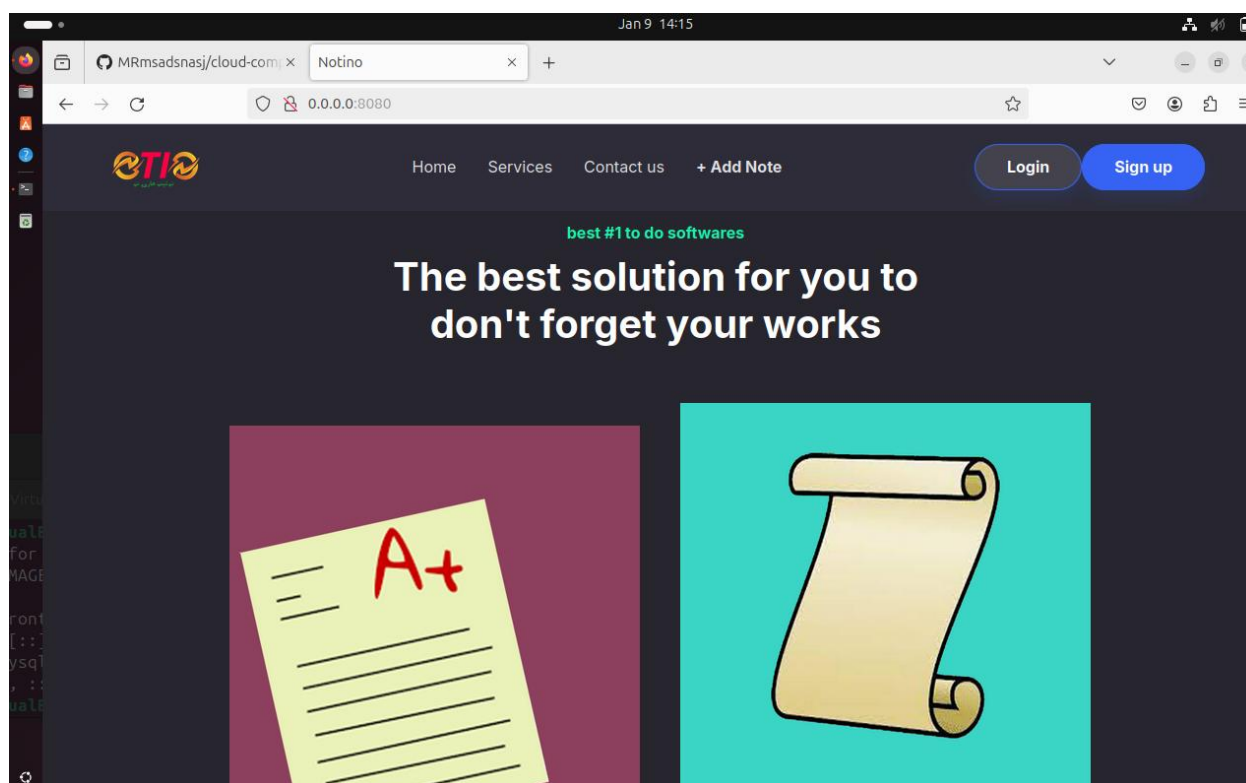
matin@matin-VirtualBox: /var/www/html/front$ sudo service mysql stop
matin@matin-VirtualBox: /var/www/html/front$ sudo docker-compose -f docker-compose.app.yml -f docker-compose.db.yml up --build
WARNING: Found orphan containers (front_php_1) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up.
Building app
[+] Building 2.2s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.36kB
=> [internal] load metadata for docker.io/library/php:8.2-apache
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [stage-1 1/8] FROM docker.io/library/php:8.2-apache@sha256:56d8b242c3430aa5eb27fc112194d2b22c5c72fe7c3b3db094063
=> [internal] load build context
=> => transferring context: 6.54kB
=> CACHED [stage-1 2/8] RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
=> CACHED [stage-1 3/8] RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --file
=> CACHED [stage-1 4/8] COPY . /var/www/html/
=> CACHED [stage-1 5/8] WORKDIR /var/www/html/
=> CACHED [stage-1 6/8] RUN chown -R www-data:www-data /var/www/html/ && chmod -R 755 /var/www/html/
=> CACHED [stage-1 7/8] RUN apt-get update && apt-get install -y libzip-dev libgd-dev && docker-php-ext
=> CACHED [stage-1 8/8] RUN composer require phpoffice/phpspreadsheet phpoffice/phpword
=> exporting to image
=> => exporting layers
=> => writing image sha256:426fe9dbdae9a333096c7e75d66b89dad1c46d28ee899003fa019ae74a0cc42f
=> => naming to docker.io/library/front_app
notino_app is up-to-date
Starting mysql_db ... done
Attaching to notino_app, mysql_db
notino_app | [Thu Jan 09 08:27:16.801076 2025] [mpm_prefork:notice] [pid 1:tid 1] AH00163: Apache/2.4.62 (Debian) PHP/8.2.2
7 configured -- resuming normal operations
notino_app | [Thu Jan 09 08:27:16.801171 2025] [core:notice] [pid 1:tid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
notino_app | [Thu Jan 09 08:31:23.208013 2025] [mpm_prefork:notice] [pid 1:tid 1] AH00170: caught SIGWINCH, shutting down n
```

همانطور که مشاهده میکنید کانتینر ها به خوبی ران شدند

برای مشاهده از `docker ps` استفاده میکنم.

```
Jan 9 14:13
matin@matin-VirtualBox: /var/www/html/front
matin@matin-VirtualBox: /var/www/html/front
matin@matin-VirtualBox: /var/www/html/front$ sudo docker ps
[sudo] password for matin:
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
a3b4a017b407   front_app     "docker-php-entrypoi..." 2 hours ago    Up 5 minutes   0.0.0.0:8080->80/tcp, [::]:8080->80/t
cp             notino_app
b6cb1d515ebe   mysql:8.0     "docker-entrypoint.s..." 2 hours ago    Up About a minute 0.0.0.0:3306->3306/tcp, :::3306->3306
/tcp, 33060/tcp mysql_db
matin@matin-VirtualBox: /var/www/html/front$
```

حال طبق آن پورت و چیز های گفته شده آن را در مرورگر باز میکنیم.



همانطور که میبینید پروژه به خوبی با استفاده از داکر بالا اومد .

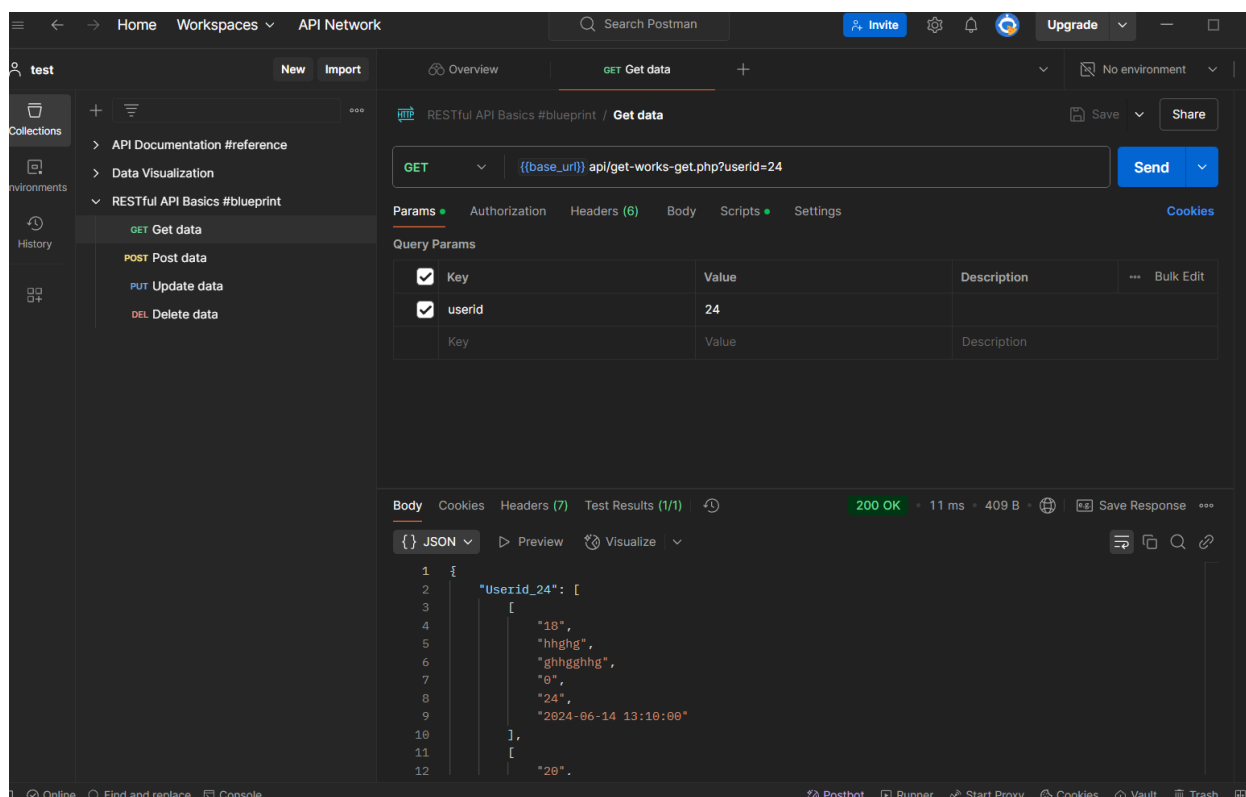
استفاده از postman

من از پستمن برای تست api ها استفاده میکنم که میتوانید از لینک زیر دانلود کنید.

همچنین میتوانید نسخه تحت وب ان را هم استفاده کنید.

من وقتی نصب کردم چون روی لوکال بود نمیتوانستم درخواست ارسال کنم. ولی به من گفت برنامه postman-agent رو دانلود کنم تا بتوانم روی لوکال اینکار را کنم.

در زیر شات هایی از اجرای postman میبینید



این api میاد لیست تمامی کارهای یوزر ایدی 24 را میدهد.