به نام خدا

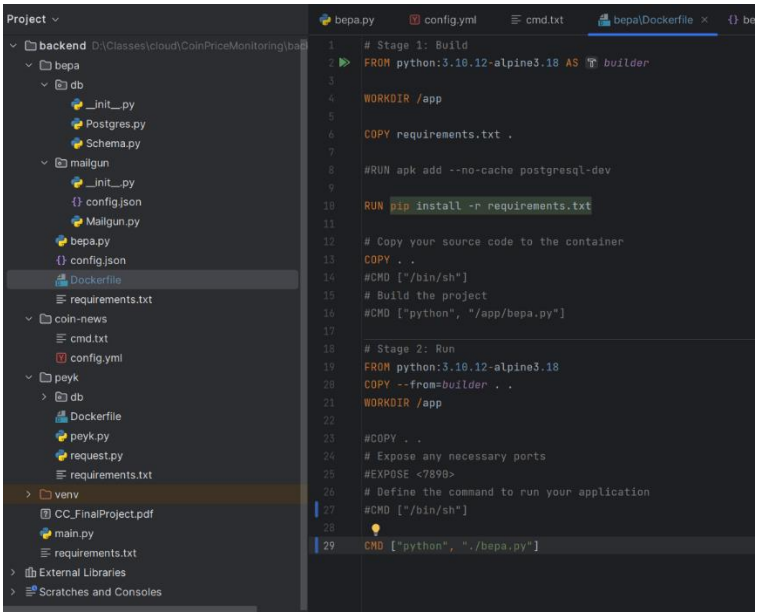
گزارش پروژه نهایی رایانش ابری

مریم کرمانشاهانی – ۹۷۲۳۰۷۳

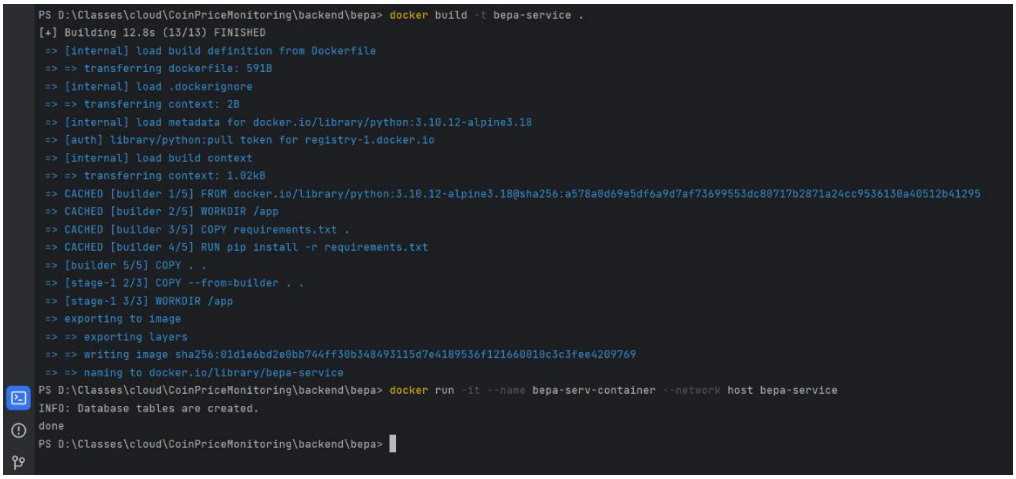
ثمین مهدی پور – ۹۸۳۹۰۳۹

بخش بپا)

داکر فایل به صورت زیر تعریف شد



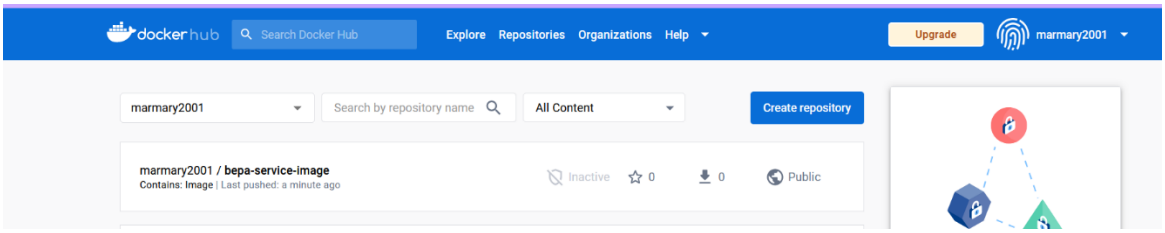
با دستورات زیر ایمیج را ساختیم و کانتینر را ران کردیم



دستورات برای قرار دادن در داکرهاب:

```
PS D:\Classes\cloud\CoinPriceMonitoring\backend\bepa> docker tag bepa-service marmary2001/bepa-service-image:1
PS D:\Classes\cloud\CoinPriceMonitoring\backend\bepa> docker push marmary2001/bepa-service-image:1
The push refers to repository [docker.io/marmary2001/bepa-service-image]
5f70bf18a086: Layer already exists
0359d6413ccf: Pushed
a580b860403a: Layer already exists
be6b7a740d80: Layer already exists
18e21c7b4116: Layer already exists
cd18acbc1cce: Mounted from library/python
78a822fe2a2d: Mounted from library/python
1: digest: sha256:96ed0aa21d19fa859e9dc1ceaa2f986ad75a74364de88539dee1a82355159ff6 size: 1786
PS D:\Classes\cloud\CoinPriceMonitoring\backend\bepa>
```

داکرهاب:



بخش پیک)

داکرفایل بصورت زیر تعریف شد:

```
# Stage 1: Build
FROM python:3.10.12-alpine3.18 AS builder

WORKDIR /app

COPY requirements.txt .

RUN apk add --no-cache gcc musl-dev libffi-dev openssl-dev

RUN pip install --no-cache-dir -r requirements.txt

COPY . .

# Build the project (replace "peyk:app" with the filename if needed)
RUN python -m compileall -b . && rm -rf __pycache__

# Stage 2: Run
FROM python:3.10.12-alpine3.18

WORKDIR /app

COPY --from=builder /app .

RUN apk --no-cache add libpq

# Install UVicorn and other dependencies
RUN pip install --no-cache-dir -r requirements.txt

# Expose the required port
EXPOSE 8080

# Set the entrypoint command
CMD ["uvicorn", "peyk:app", "--host", "0.0.0.0", "--port", "8080"]
```

ایمیج بصورت زیر ساخته شد:

```
C:\Users\Samin\Desktop\University\Term 7\Cloud Computing\PF\CoinPriceMonitoring\backend\peyk>docker build -t peyk-server
 .
[+] Building 99.6s (16/16) FINISHED
 => [internal] load build definition from Dockerfile                                                0.1s
 => => transferring dockerfile: 32B                                                                 0.0s
 => [internal] load .dockerignore                                                                   0.0s
 => => transferring context: 2B                                                                     0.0s
 => [internal] load metadata for docker.io/library/python:3.10.12-alpine3.18                        3.4s
 => [auth] library/python:pull token for registry-1.docker.io                                       0.0s
 => [internal] load build context                                                                   0.1s
 => => transferring context: 565B                                                                   0.0s
 => [builder 1/7] FROM docker.io/library/python:3.10.12-alpine3.18@sha256:a578a0d69e5df6a9d7af73699  0.0s
 => CACHED [builder 2/7] WORKDIR /app                                                               0.0s
 => CACHED [builder 3/7] COPY requirements.txt .                                                    0.0s
 => CACHED [builder 4/7] RUN apk add --no-cache gcc musl-dev libffi-dev openssl-dev                 0.0s
 => CACHED [builder 5/7] RUN pip install --no-cache-dir -r requirements.txt                         0.0s
 => [builder 6/7] COPY . .                                                                          0.1s
 => [builder 7/7] RUN python -m compileall -b . && rm -rf __pycache__                               2.2s
 => [stage-1 3/5] COPY --from=builder /app .                                                        0.1s
 => [stage-1 4/5] RUN apk --no-cache add libpq                                                      7.3s
 => [stage-1 5/5] RUN pip install --no-cache-dir -r requirements.txt                               81.8s
 => exporting to image                                                                              1.8s
 => => exporting layers                                                                             1.7s
 => => writing image sha256:1aaa1575cd25b05787a48cd45c9bf91c6db3d7f4aa5f2a6848edf72de784baa0        0.0s
 => => naming to docker.io/library/peyk-server                                                      0.0s
```

آن را بصورت زیر ران کردیم:

```
C:\Users\Samin\Desktop\University\Term 7\Cloud Computing\PF\CoinPriceMonitoring\backend\peyk>docker run -p 8080:8080 --n
ame peyk-server peyk-server
INFO:     Started server process [1]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8080 (Press CTRL+C to quit)
```

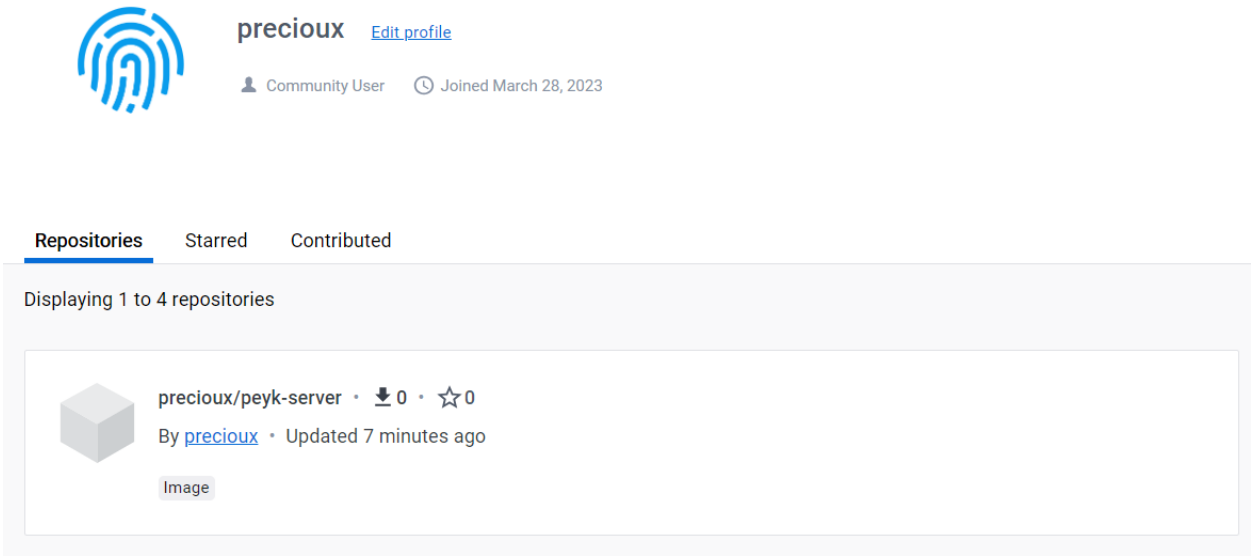
همانطور که مشاهده میشود سرور آماده پاسخگویی است.

آن را بصورت زیر در داکرهاب قرار دادیم:

```
C:\Users\Samin\Desktop\University\Term 7\Cloud Computing\PF\CoinPriceMonitoring\backend\peyk>docker tag
peyk-server precioux/peyk-server:latest

C:\Users\Samin\Desktop\University\Term 7\Cloud Computing\PF\CoinPriceMonitoring\backend\peyk>docker push
 precioux/peyk-server:latest
The push refers to repository [docker.io/precioux/peyk-server]
2a43d4f32fae: Pushing  55.39MB/59.48MB
23b36fde1770: Pushed
04048931be8f: Pushed
f17e85f413e7: Pushed
a580b860403a: Mounted from library/python
be6b7a740d80: Mounted from library/python
18e21c7b4116: Mounted from library/python
cd18acbc1cce: Mounted from library/python
78a822fe2a2d: Mounted from library/python
^C
C:\Users\Samin\Desktop\University\Term 7\Cloud Computing\PF\CoinPriceMonitoring\backend\peyk>docker push
 precioux/peyk-server:latest
The push refers to repository [docker.io/precioux/peyk-server]
2a43d4f32fae: Pushed
23b36fde1770: Layer already exists
04048931be8f: Layer already exists
f17e85f413e7: Layer already exists
a580b860403a: Layer already exists
be6b7a740d80: Layer already exists
18e21c7b4116: Layer already exists
cd18acbc1cce: Layer already exists
78a822fe2a2d: Layer already exists
latest: digest: sha256:057de9b6f66b50bca16036fd6ec23f94aca749eca821b95e47e19f8ba23503b8 size: 2205
```

صحت کار را روی داکرهاب بررسی میکنیم:



همانطور که مشاهده میشود فایل با موفقیت روی داکرهاب قرار داده شده است.

گام سوم-

موارد زیر را در گزارش خود بیاورید.

- با استفاده از دستور kubectl get صحت ایجاد منابع بر روی کلاستر را نمایش دهید.
- آدرس آی‌پی[۲۱] پادها و نحوه برقراری ارتباط میان آنها و سرویس ساخته شده.
- برای دیپلویمنت مربوط به پایگاه داده چه تعداد پاد ایجاد کردید؟ دلیل کار خود را توضیح دهید.
- به کمک یک پاد با ایمیج کرل[۲۲] و یا قابلیت port-forwarding کوبرنتیز، سیستم نهایی خود را بیازمایید. تصاویر صحت عملکرد درخواست‌های **subscribe_coin** و **get_price_history** و **ارسال ایمیل به کاربر** را بیاورید.

منابع به صورت زیر خواهند بود:

```
C:\Users\Samin\Desktop\University\Term 7\Cloud Computing\PF\CoinPriceMonitoring\backend>kubectl get all
NAME                                              READY   STATUS      RESTARTS        AGE
pod/bepa-cronjob-28134573-6nw5l                   0/1     Completed   0               7m48s
pod/bepa-cronjob-28134576-2586z                   0/1     Completed   0               4m48s
pod/bepa-cronjob-28134579-7gsck                   0/1     Completed   0               108s
pod/coinnews-79b6fb6c49-sqmxd                     1/1     Running     0               3h17m
pod/db-deployment-6f487688f6-5m8m7                1/1     Running     0               29m
pod/fastapi-server-deployment-768857fcdb-2hdww    1/1     Running     6 (3h21m ago)   40d
pod/fastapi-server-deployment-768857fcdb-9qmhh    1/1     Running     6 (3h21m ago)   40d
pod/peyk-server-deployment-6bd7776bbf-wxrrb       1/1     Running     0               7m13s
pod/redis-cache-deployment-6f4bc5558d-nmng9       1/1     Running     7 (3h21m ago)   40d
pod/ubuntu-deployment-546bf655c4-bpwhr            1/1     Running     7 (3h21m ago)   43d

NAME                              TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
service/coinnews                  ClusterIP   10.107.175.133   <none>        8000/TCP   3h17m
service/db-service                ClusterIP   10.111.182.196   <none>        5432/TCP   8h
service/fastapi-server-service    ClusterIP   10.103.236.244   <none>        8000/TCP   40d
service/kubernetes                ClusterIP   10.96.0.1        <none>        443/TCP    46d
service/peyk-server-service       ClusterIP   10.107.126.126   <none>        8080/TCP   8h
service/redis-service             ClusterIP   10.100.56.166    <none>        6379/TCP   40d

NAME                                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/coinnews                     1/1     1            1           3h17m
deployment.apps/db-deployment                1/1     1            1           29m
deployment.apps/fastapi-server-deployment    2/2     2            2           40d
deployment.apps/peyk-server-deployment       1/1     1            1           7m13s
deployment.apps/redis-cache-deployment       1/1     1            1           40d
deployment.apps/ubuntu-deployment            1/1     1            1           43d

NAME                                                    DESIRED   CURRENT   READY   AGE
replicaset.apps/coinnews-79b6fb6c49                     1         1         1       3h17m
replicaset.apps/db-deployment-6f487688f6                1         1         1       29m
replicaset.apps/fastapi-server-deployment-768857fcdb    2         2         2       40d
replicaset.apps/peyk-server-deployment-6bd7776bbf       1         1         1       7m13s
replicaset.apps/redis-cache-deployment-6f4bc5558d       1         1         1       40d
replicaset.apps/ubuntu-deployment-546bf655c4            1         1         1       43d

NAME                          SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
cronjob.batch/bepa-cronjob    */3 * * * *   False     0        108s            19m

NAME                                COMPLETIONS   DURATION   AGE
job.batch/bepa-cronjob-28134573     1/1           14s        7m48s
job.batch/bepa-cronjob-28134576     1/1           15s        4m48s
job.batch/bepa-cronjob-28134579     1/1           22s        108s
```

- آدرس آی پی پادها و نحوه ارتباطشان:

```
C:\Users\Samin\Desktop\University\Term 7\Cloud Computing\PF\CoinPriceMonitoring\backend\peyk>kubectl get pods -o wide
NAME                                      READY   STATUS      RESTARTS        AGE     IP             NODE       NOMINATED NODE   READINESS GATES
bepa-cronjob-28134816-wdr9g               0/1     Completed   0               7m29s   10.244.1.177   minikube   <none>           <none>
bepa-cronjob-28134819-vqsbw               0/1     Completed   0               4m29s   10.244.1.178   minikube   <none>           <none>
bepa-cronjob-28134822-6hspm               0/1     Completed   0               89s     10.244.1.179   minikube   <none>           <none>
coinnews-79b6fb6c49-sqmxd                 1/1     Running     0               7h20m   10.244.1.51    minikube   <none>           <none>
db-deployment-6f487688f6-g476x            1/1     Running     0               3h15m   10.244.1.109   minikube   <none>           <none>
fastapi-server-deployment-768857fcdb-2hdww   1/1     Running     6 (7h24m ago)   40d     10.244.1.43    minikube   <none>           <none>
fastapi-server-deployment-768857fcdb-9qmhh   1/1     Running     6 (7h24m ago)   40d     10.244.1.2     minikube   <none>           <none>
peyk-server-deployment-6bd7776bbf-mz2rf   1/1     Running     0               20m     10.244.1.172   minikube   <none>           <none>
redis-cache-deployment-6f4bc5558d-nmng9   1/1     Running     7 (7h24m ago)   40d     10.244.1.40    minikube   <none>           <none>
ubuntu-deployment-546bf655c4-bpwhr        1/1     Running     7 (7h24m ago)   43d     10.244.1.21    minikube   <none>           <none>
```

نحوه ارتباط پادها توسط سرویس هایی است که تعریف کردیم که آی پی آنها بصورت زیر است ولی میتوان به صورت اسمی نیز آنهارا به همراه پورتشان استفاده کرد:

```
C:\Users\Samin\Desktop\University\Term 7\Cloud Computing\PF\CoinPriceMonitoring\backend\peyk>kubectl get services -o wide
NAME                   TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE     SELECTOR
coinnews               ClusterIP   10.107.175.133   <none>        8000/TCP   7h24m   app=coinnews
db-service             ClusterIP   10.111.182.196   <none>        5432/TCP   12h     app=postgresdb
fastapi-server-service   ClusterIP   10.103.236.244   <none>        8000/TCP   40d     app=fastapi-server-deployment
kubernetes             ClusterIP   10.96.0.1        <none>        443/TCP    46d     <none>
peyk-server-service    ClusterIP   10.107.126.126   <none>        8080/TCP   12h     app=peyk-server
redis-service          ClusterIP   10.100.56.166    <none>        6379/TCP   40d     app=redis-cache-deployment
```

- یک پاد برای پایگاه داده استفاده شده چون درخواست ها از طریق سرویسی که برای آن تعبیر کرده بودیم بررسی و پاسخ دهی میشوند و نیازی به چندین پاد نداریم.

- بررسی صحت سیستم:

برای انجام اینکار از minikube service استفاده میکنیم تا به ما آی پی خروجی و قابل دسترس عمومی دهد:

برای سرویس پایگاه داده:

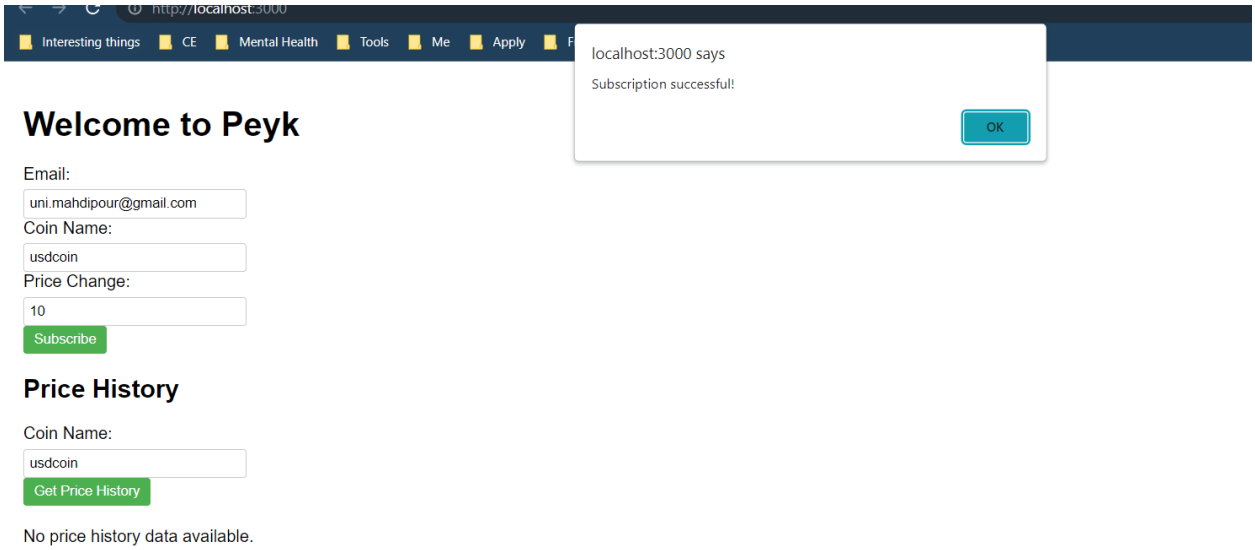
```
C:\Users\Samin\Desktop\University\Term 7\Cloud Computing\PF\CoinPriceMonitoring\backend\peyk>minikube service db-service

|-----------|------------|-------------|--------------|
| NAMESPACE |    NAME    | TARGET PORT |     URL      |
|-----------|------------|-------------|--------------|
| default   | db-service |             | No node port |
|-----------|------------|-------------|--------------|
* service default/db-service has no node port
* Starting tunnel for service db-service.
|-----------|------------|-------------|------------------------|
| NAMESPACE |    NAME    | TARGET PORT |          URL           |
|-----------|------------|-------------|------------------------|
| default   | db-service |             | http://127.0.0.1:28875 |
|-----------|------------|-------------|------------------------|
* Opening service default/db-service in default browser...
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

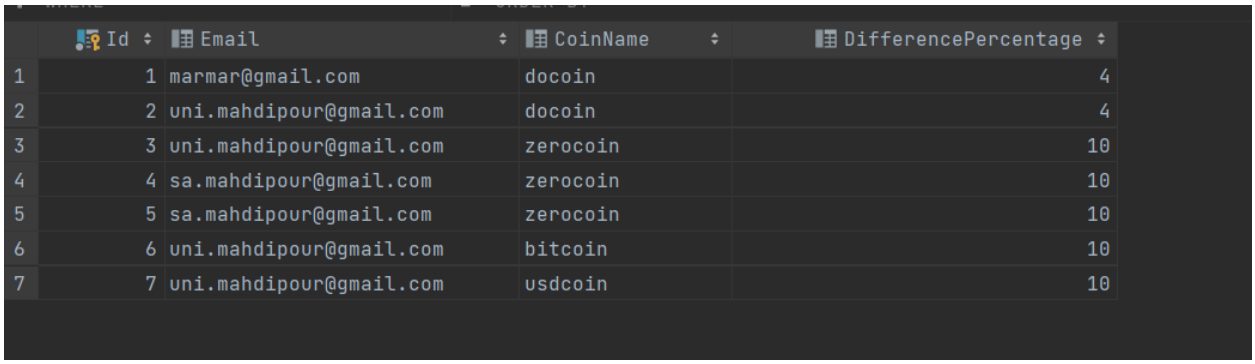
برای سرویس پیک:

```
C:\Users\Samin\Desktop\University\Term 7\Cloud Computing\PF\CoinPriceMonitoring\backend\kuber\peyk>minikube service peyk
-server-service
|-----------|---------------------|-------------|--------------|
| NAMESPACE |        NAME         | TARGET PORT |     URL      |
|-----------|---------------------|-------------|--------------|
| default   | peyk-server-service |             | No node port |
|-----------|---------------------|-------------|--------------|
* service default/peyk-server-service has no node port
* Starting tunnel for service peyk-server-service.
|-----------|---------------------|-------------|------------------------|
| NAMESPACE |        NAME         | TARGET PORT |          URL           |
|-----------|---------------------|-------------|------------------------|
| default   | peyk-server-service |             | http://127.0.0.1:31614 |
|-----------|---------------------|-------------|------------------------|
* Opening service default/peyk-server-service in default browser...
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

حالا فرانت را اجرا میکنیم:



مشاهده میکنیم که درخواست ما به درستی ثبت شده است:



	Id	Email	CoinName	DifferencePercentage
1	1	marmar@gmail.com	docoin	4
2	2	uni.mahdipour@gmail.com	docoin	4
3	3	uni.mahdipour@gmail.com	zerocoin	10
4	4	sa.mahdipour@gmail.com	zerocoin	10
5	5	sa.mahdipour@gmail.com	zerocoin	10
6	6	uni.mahdipour@gmail.com	bitcoin	10
7	7	uni.mahdipour@gmail.com	usdcoin	10

حالا گذشته usdcoin را بررسی میکنیم:

Price History

Coin Name:

usdcoin

Get Price History

Timestamp	Price
2023-06-29T07:38:00.334349586Z	7999
2023-06-29T07:38:10.334567186Z	7999
2023-06-29T09:48:00.913601301Z	7980
2023-06-29T09:51:00.923189618Z	7984
2023-06-29T09:54:00.955764539Z	7977
2023-06-29T09:57:00.967522963Z	7979
2023-06-29T10:06:01.005967902Z	7979
2023-06-29T10:09:01.02191253Z	7982
2023-06-29T10:18:01.059361787Z	7981
2023-06-29T10:21:01.073555347Z	7980
2023-06-29T10:27:01.149861418Z	7979
2023-06-29T10:30:01.15922706Z	7994
2023-06-29T10:36:01.242624907Z	7998
2023-06-29T10:42:01.259954109Z	8000
2023-06-29T10:45:01.338299227Z	8007

مشاهده میشود که گذشته قیمت های این کوین به درستی آورده میشود.

همچنین میبینیم که برای ایمیل های ثبت شده پس از هر اپدیت ایمیل هشداری با توجه به کوین مدنظرشان ارسال شده است: