# بخش اول

فایل‌های این قسمت در پوشه src/app قرار گرفته‌اند. برای پیش‌بینی برچسب از مدل Inception که برروی داده‌های ImageNet آموزش دیده است استفاده شده است. برای تبدیل بایت‌های ورودی api به تنسور از کتابخانه PIL استفاده شده و تعدادی تبدیل شامل تغییر اندازه و میانگین‌گیری برای نرمالیزه کردن تصویر ورودی به مدل نیز استفاده شده است. ابتدا با دستور زیر پکیج‌های مورد نیاز را نصب می‌کنیم:

> cd src/app

> pip install -r requirements.txt

خروجی برنامه‌ی اصلی به صورت زیر است:

> python main.py

INFO: Started server process [16560]

INFO: Waiting for application startup.

INFO: Application startup complete.

INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)

INFO: 127.0.0.1:61652 - "POST /predict HTTP/1.1" 200 OK

INFO: 127.0.0.1:61654 - "POST /predict HTTP/1.1" 200 OK

برای تست برنامه‌ی نوشته شده فایل test\_app.py را اجرا می‌کنیم:

> python test\_app.py

Response: {'label': 1, 'labelName': 'goldfish, Carassius auratus'}

Latency: 2.394197940826416 seconds

# بخش دوم

فایل داکر در مسیر بخش اول نوشته شده است. خروجی دستورات ساخت و اجرای image در زیر آورده شده است.

$ docker build -t hw3-predict-api .

[+] Building 287.7s (11/11) FINISHED

=> [internal] load .dockerignore 0.5s

=> => transferring context: 57B 0.0s

=> [internal] load build definition from Dockerfile 0.5s

=> => transferring dockerfile: 213B 0.0s

=> [internal] load metadata for docker.io/library/python:3.11 1.8s

=> [auth] library/python:pull token for registry-1.docker.io 0.0s

=> [1/5] FROM docker.io/library/python:3.11@sha256:d73088ce13d5a1eec1dd05b47736041ae6921d08d2f240035d99642db98bc8d4 29.0s

=> => resolve docker.io/library/python:3.11@sha256:d73088ce13d5a1eec1dd05b47736041ae6921d08d2f240035d99642db98bc8d4 0.3s

=> => sha256:94c2dca43c9c127e42dfd021039cc83d8399752097612b49bdc7b00716b6d826 2.01kB / 2.01kB 0.0s

=> => sha256:c0e63845ae986c52da5cd6ac4d56eebf293439bb22a3cee198dd818fd12ba555 7.53kB / 7.53kB 0.0s

=> => sha256:d73088ce13d5a1eec1dd05b47736041ae6921d08d2f240035d99642db98bc8d4 2.14kB / 2.14kB 0.0s

=> => sha256:d52e4f012db158bb7c0fe215b98af1facaddcbaee530efd69b1bae07d597b711 49.55MB / 49.55MB 1.4s

=> => sha256:7dd206bea61ff3e3b54be1c20b58d8475ddd6f89df176146ddb7a2fd2c747ea2 24.03MB / 24.03MB 0.8s

=> => sha256:2320f9be4a9c605d1ac847cf67cec42b91484a7cf7c94996417a0c7c316deadc 64.11MB / 64.11MB 1.5s

=> => sha256:6e5565e0ba8dfce32b9049f21ceeb212946e0bb810d94cbd2db94ca61082f657 211.00MB / 211.00MB 4.1s

=> => sha256:70f90dfe001bab12bb117972f14fb9429be3d5af96cc78084e8bd0c53b635ea3 19.76MB / 19.76MB 2.9s

=> => sha256:d3797e13cc41e1387a937908677cc66838a672cdc8ba8de9b966f1801723d07b 6.39MB / 6.39MB 2.2s

=> => extracting sha256:d52e4f012db158bb7c0fe215b98af1facaddcbaee530efd69b1bae07d597b711 3.1s

=> => sha256:bd75605de417c150af6942a796e71c4bcd8567194d2c2a1e9ed90d8660550a39 244B / 244B 3.0s

=> => sha256:3d0e1a4b14bc5f3e735fb7730c802b651d49d295a8ff12b8661d6c0e5653485e 3.09MB / 3.09MB 3.2s

=> => extracting sha256:7dd206bea61ff3e3b54be1c20b58d8475ddd6f89df176146ddb7a2fd2c747ea2 0.9s

=> => extracting sha256:2320f9be4a9c605d1ac847cf67cec42b91484a7cf7c94996417a0c7c316deadc 3.3s

=> => extracting sha256:6e5565e0ba8dfce32b9049f21ceeb212946e0bb810d94cbd2db94ca61082f657 10.3s

=> => extracting sha256:d3797e13cc41e1387a937908677cc66838a672cdc8ba8de9b966f1801723d07b 0.7s

=> => extracting sha256:70f90dfe001bab12bb117972f14fb9429be3d5af96cc78084e8bd0c53b635ea3 0.8s

=> => extracting sha256:bd75605de417c150af6942a796e71c4bcd8567194d2c2a1e9ed90d8660550a39 0.0s

=> => extracting sha256:3d0e1a4b14bc5f3e735fb7730c802b651d49d295a8ff12b8661d6c0e5653485e 0.3s

=> [internal] load build context 0.3s

=> => transferring context: 37.05kB 0.0s

=> [2/5] WORKDIR /app 0.2s

=> [3/5] COPY requirements.txt . 0.3s

=> [4/5] RUN pip install -r requirements.txt 150.9s

=> [5/5] COPY . . 0.5s

=> exporting to image 103.2s

=> => exporting layers 102.7s

=> => writing image sha256:8b7fd559d0ddd028469a1882e84cce0e64dae515e3cf425acfee396d4fcb05ad 0.1s

=> => naming to docker.io/library/hw3-predict-api 0.1s

$ docker run -dp 8000:8000 hw3-predict-api

a56618c7c7445843e64bdd313d5fe735b2b9a75de8408fbb773fc70f4ebad72b

برای تست برنامه‌ی نوشته شده فایل test\_app.py را اجرا می‌کنیم:

$ python test\_app.py

Response: {'label': 1, 'labelName': 'goldfish, Carassius auratus'}

Latency: 0.2344062328338623 seconds

# بخش سوم

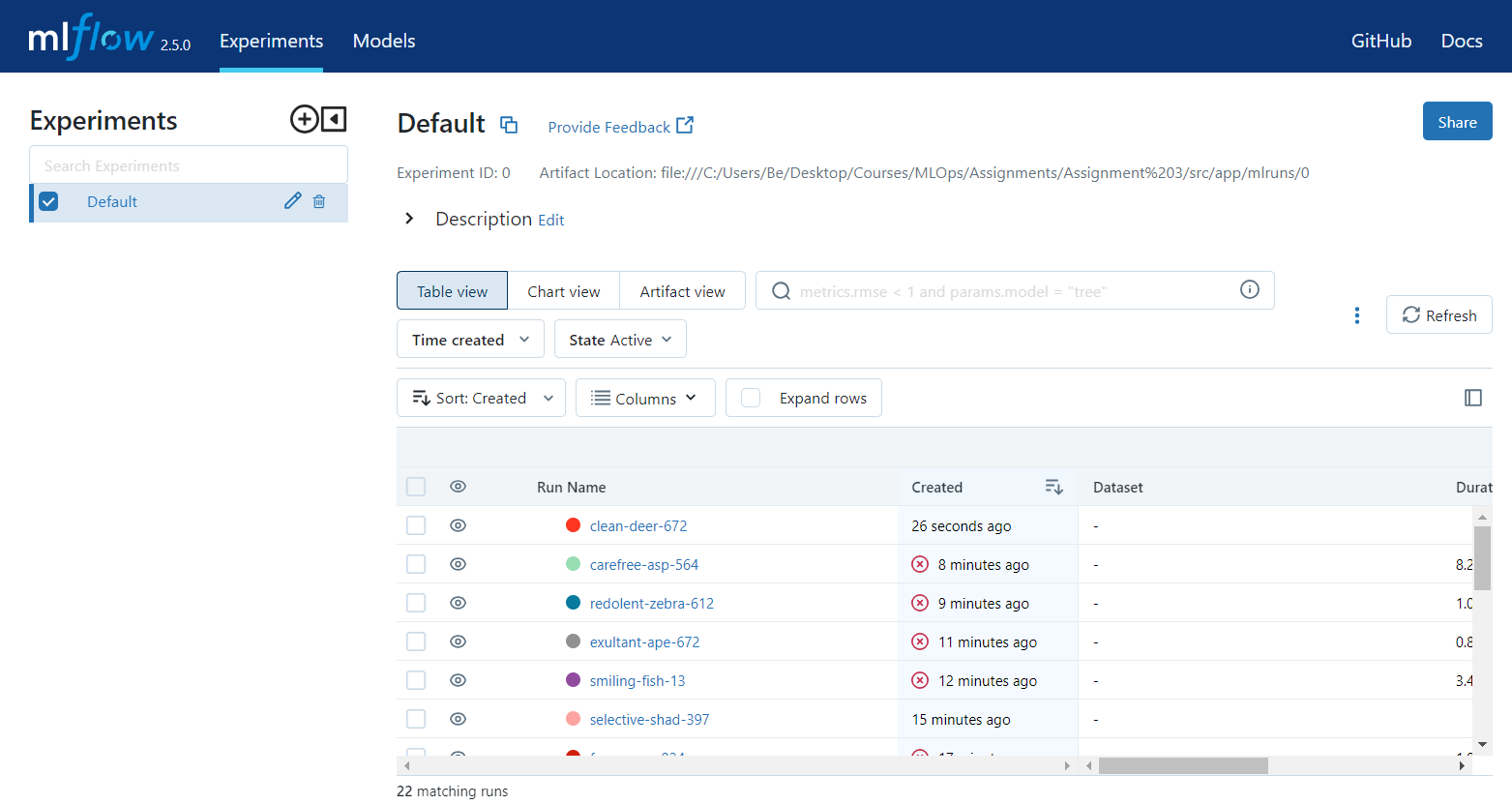
فایل مربوط به monitor کردن به نام monitor.py در مسیر بخش اول نوشته شده است. برای شروع دستورات زیر را اجرا می‌کنیم:

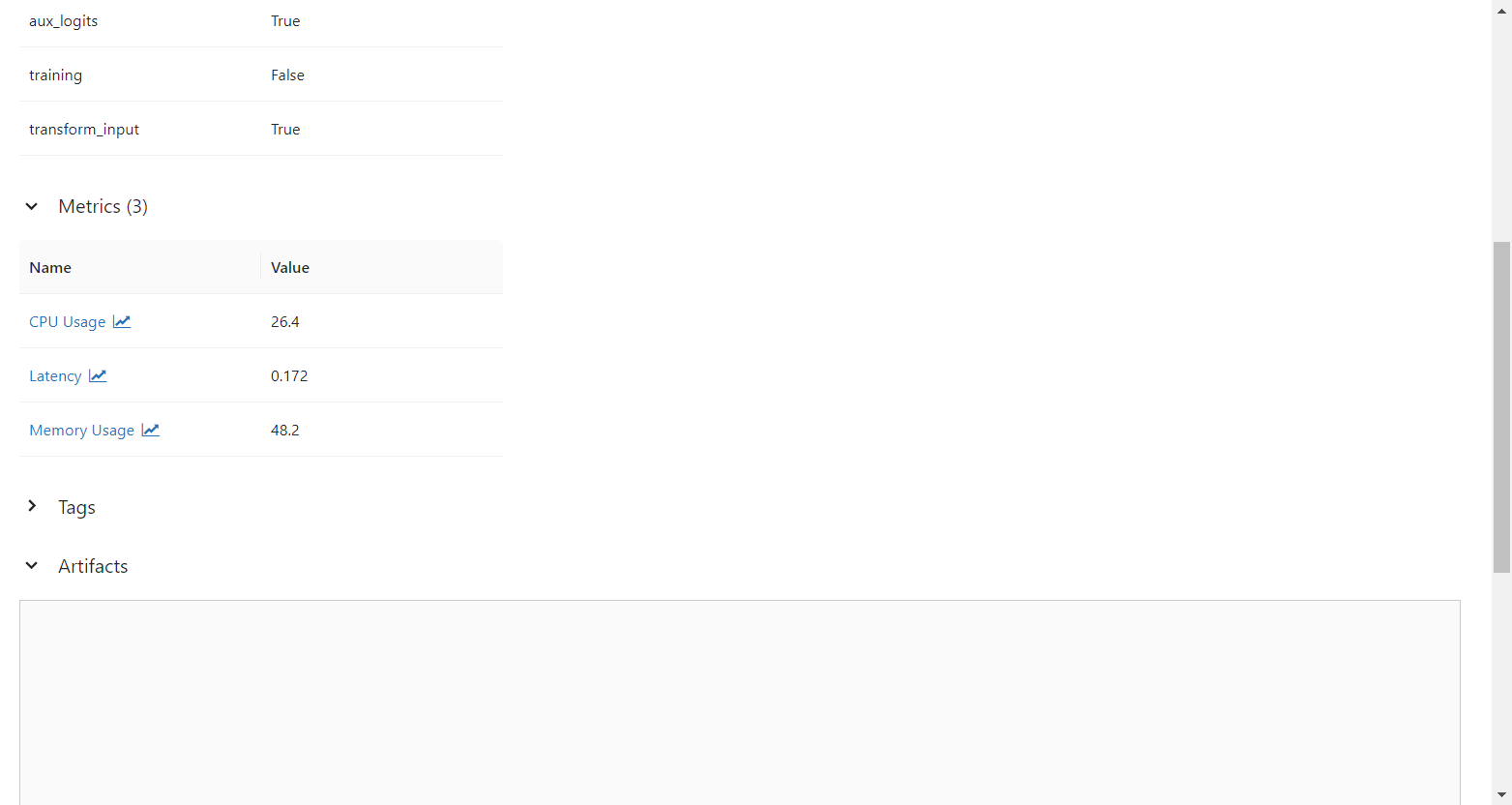
> python monitor.py

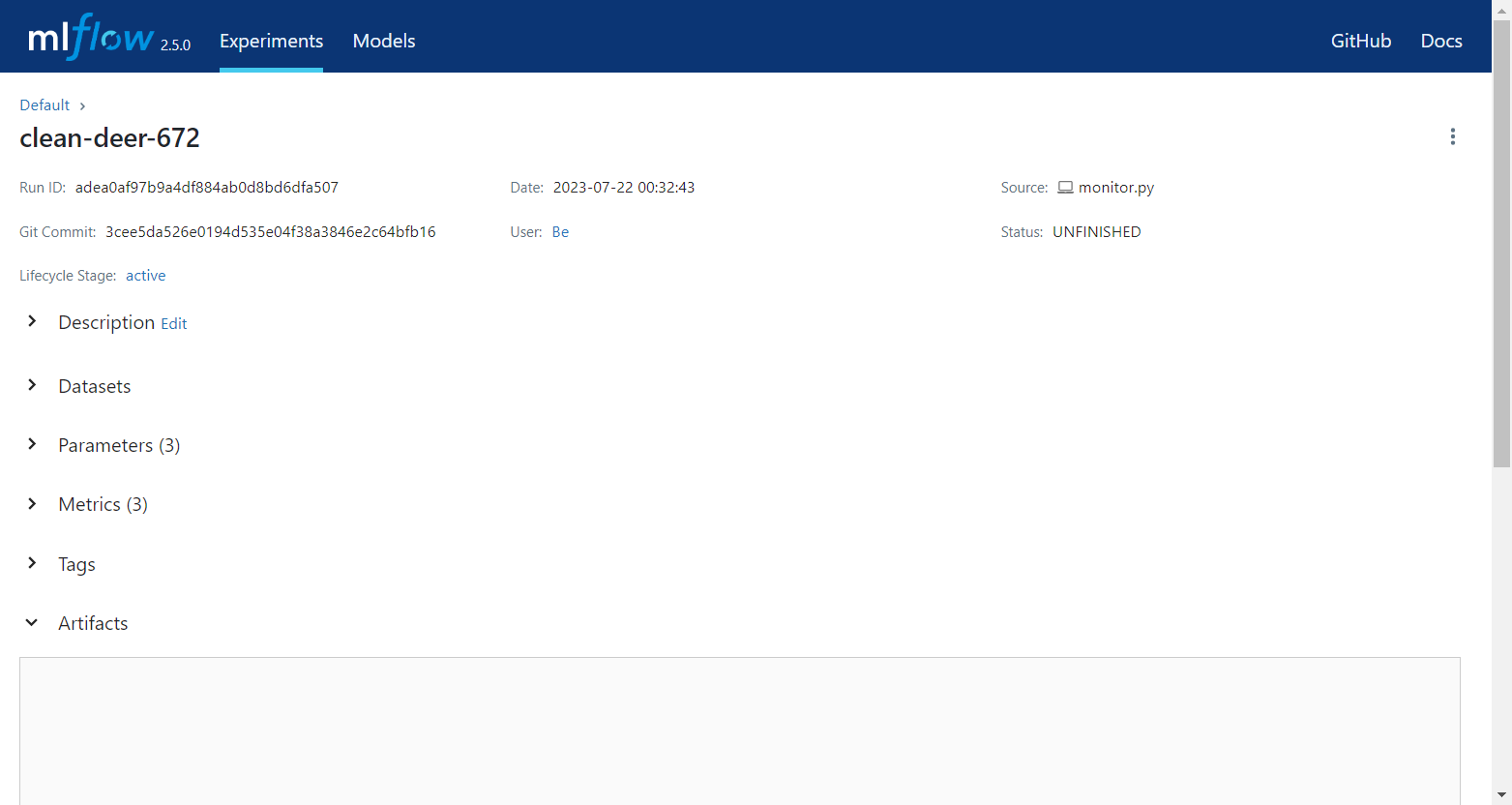
> mlflow ui

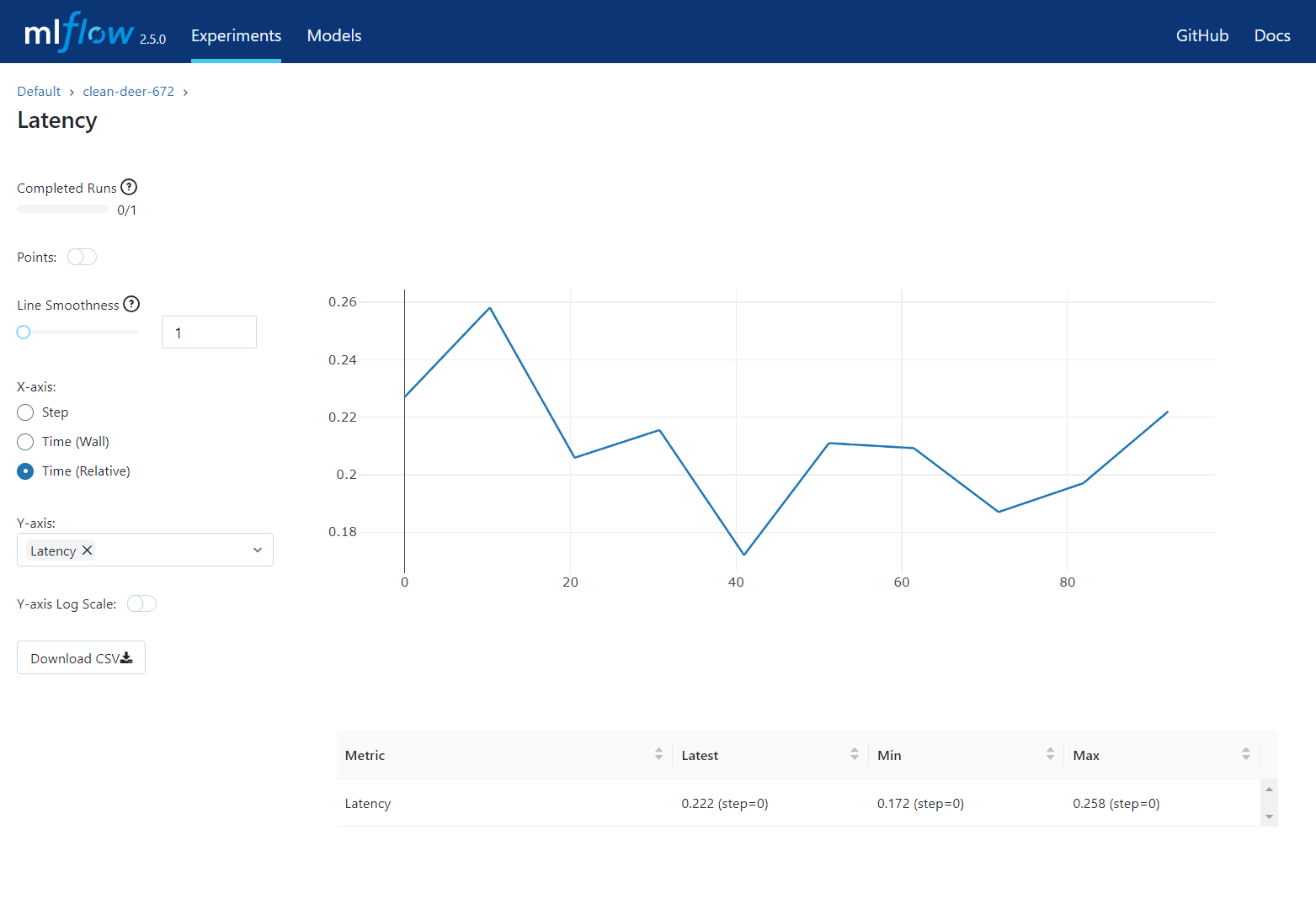
INFO:waitress:Serving on http://127.0.0.1:5000

برای مشاهده لاگ‌ها به صفحه روی پورت 5000 می‌رویم:









# بخش امتیازی

برای استفاده از API، یک صفحه وب با استفاده از فریمورک Angular نوشته شده که در پوشه src/web قرار دارد. برای استفاده از آن لازم است که دستورات زیر اجرا شوند:

> **cd src/web**

> **npm i**

added 960 packages, and audited 961 packages in 1m

105 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

> **npm start**

> web-app@0.0.0 start

> ng serve

✔ Browser application bundle generation complete.

Initial Chunk Files | Names | Raw Size

vendor.js | vendor | 3.38 MB |

styles.css, styles.js | styles | 345.94 kB |

polyfills.js | polyfills | 335.90 kB |

main.js | main | 22.27 kB |

runtime.js | runtime | 6.51 kB |

| Initial Total | 4.08 MB

Build at: 2023-07-21T19:28:02.206Z - Hash: acd8adf1c2803eea - Time: 18914ms

\*\* Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ \*\*

✔ Compiled successfully.

نمونه تصاویر از محیط رابط کاربری:

