



به نام خدا

تمرین سوم درس پایگاه داده پیشرفته

دانشجو:

پریسا مبارک

شماره دانشجویی:

۴۰۲۱۱۴۱۵۰۰۶

استاد مربوطه:

دکتر عبدالرضا رشنو

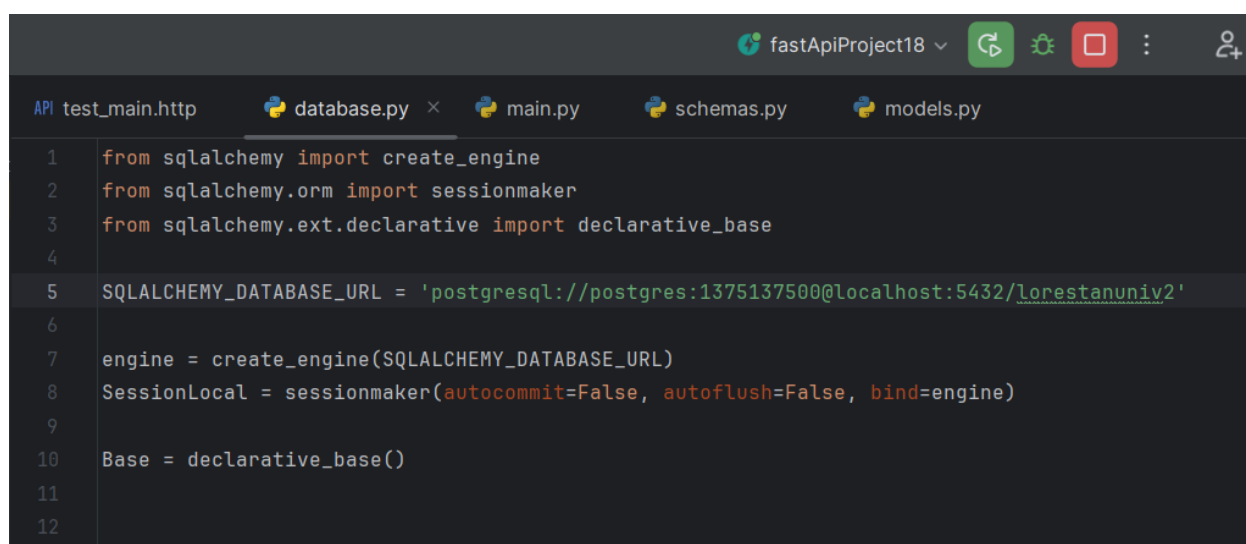
آذر ۱۴۰۲

۱. اتصال پایگاه داده postgresql به fastapi :

ابتدا وارد محیط sqlshell میشویم. بعد از وارد کردن اطلاعات مورد نیاز مانند نام کاربری و پسورد و... به دیتابیس postgres متصل میشویم.

سپس یک دیتابیس با نام lorestanuniv2 ایجاد میکنیم. حال میخواهیم از طریق fastapi به این دیتابیس وصل شویم. در برنامه pycharm چهار فایل main و models و schemas و database همانند تمرین اول، ایجاد میکنیم.

بالین تفاوت که در فایل database به پایگاه داده postgres وصل میشویم. به این صورت:



```
fastApiProject18
API test_main.http
database.py ×
main.py
schemas.py
models.py

1 from sqlalchemy import create_engine
2 from sqlalchemy.orm import sessionmaker
3 from sqlalchemy.ext.declarative import declarative_base
4
5 SQLALCHEMY_DATABASE_URL = 'postgresql://postgres:1375137500@localhost:5432/lorestanuniv2'
6
7 engine = create_engine(SQLALCHEMY_DATABASE_URL)
8 SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)
9
10 Base = declarative_base()
11
12
```

در مرحله بعد عملگرها را تست میکنیم.

تست عملگر درج:

طبق تصویر زیر یک رکورد از طریق برنامه insomnia در جدول درس درج می کنیم.

POST http://127.0.0.1:8000/RegLes/ Send 200 OK TIME 15 ms

JSON Auth Query Header 1 Docs Preview Header 4

```

1 {
2   "cname": "شیفی",
3   "credit": "1",
4   "cid": "76145",
5   "part": "منر"
6 }

```

```

1 {
2   "cname": "شیفی",
3   "part": "منر",
4   "credit": 1,
5   "cid": 76145
6 }

```

حال به محیط sqlshell میرویم و مشاهده میکنیم که رکورد جدید با cid=76145 در جدول درس درج شده است. به این صورت:

```

lorestanuniv2=# SET CLIENT_ENCODING TO UTF8;
SET
lorestanuniv2=# SELECT * FROM lesson;
cid | part | cname | credit
-----+-----+-----+-----
13735 | ج | شیفی | 3
11234 | ج | شیفی | 3
15234 | ج | شیفی | 3
18234 | ج | شیفی | 3
18134 | ج | شیفی | 3
55556 | ج | شیفی | 3
33333 | ج | شیفی | 2
13558 | ج | شیفی | 2
76165 | ج | شیفی | 1
76145 | ج | شیفی | 1
(10 rows)

lorestanuniv2=#

```

تست عملگر آپدیت:

همانند مثال بالا ابتدا در insomnia رکورد با cid=76145 را آپدیت کرده (فیلد credit را از ۱ به ۳ تغییر دادم.) و سپس در محیط sqlshell نتیجه آپدیت را مشاهده میکنیم:

POST <http://127.0.0.1:8000/UpLes/76145/> Send **200 OK** TIME 32 ms

JSON Auth Query Header 1 Docs Preview Header 4 Cookies

```

1 {
2   "cname": "فیمی",
3   "credit": "3",
4   "cid": "76145",
5   "part": "منر"
6 }

```

```

1 {
2   "cname": "فیمی",
3   "part": "منر",
4   "credit": 3,
5   "cid": 76145
6 }

```

```
lorestanuniv2=# SELECT * FROM lesson;
```

cid	part	cname	credit
13735	ج۱	فیمی	3
11234	ج۱	فیمی	3
15234	ج۱	فیمی	3
18234	ج۱	فیمی	3
18134	ج۱	فیمی	3
55556	ج۱	فیمی	3
33333	ج۱	فیمی	2
13558	ج۱	فیمی	2
76165	ج۱	فیمی	1
76147	ج۱	فیمی	3
76145	ج۱	فیمی	3

(11 rows)

تست عملگر حذف:

همانند مثال بالا ابتدا در insomnia حذف میکنیم :

DELETE <http://127.0.0.1:8000/DelLes/76145/> Send **200 OK** TIME 31 ms

JSON Auth Query Header 1 Docs Preview Header 4 Cookies

```

1 {
2   "cname": "فیمی",
3   "credit": "3",
4   "cid": "76145",
5   "part": "منر"
6 }

```

```

1 {
2   "message": "lesson with stdid 76145 has been deleted."
3 }

```

سپس در محیط sqlshell مشاهده میکنیم که رکورد با cid=76145 حذف شده است:

```
lorestanuniv2=# SELECT * FROM lesson;
```

cid	part	cname	credit
13735	۱	۱	3
11234	۱	۱	3
15234	۱	۱	3
18234	۱	۱	3
18134	۱	۱	3
55556	۱	۱	3
33333	۱	۱	2
13558	۱	۱	2
76165	۱	۱	1
76147	۱	۱	3

(10 rows)

برای جدول دیگر یعنی استاد و درس نیز به همین صورت انجام میدهیم و در insomnia اجرا کرده و بعد در محیط sqlshell نتیجه را مشاهده میکنیم. در فایل های ضمیمه شده(فایل شماره1) تمام کدها قرار دارد.

۲. اتصال پایگاه داده mongodb به fastapi :

ابتدا در محیط mongosh دیتابیس lorestanuniv3 را ایجاد میکنیم:

```
mongosh mongodb://localhost/?directConnection=true&serverSelectionTimeoutMS=2000
Please enter a MongoDB connection string (Default: mongodb://localhost/): mongodb://localhost/
mongodb://localhost/
Current Mongosh Log ID: 65aa38cedfaae5449ac4ea43
Connecting to:      mongodb://localhost/?directConnection=true&serverSelectionTimeoutMS=2000
Using MongoDB:      6.0.11
Using Mongosh:      2.0.2
mongosh 2.1.1 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
  The server generated these startup warnings when booting
  2024-01-18T17:29:00.224+03:30: Access control is not enabled for the database. Read and write operations can change unauthorized data and may impact performance.
  -----

test> use lorestanuniv3
switched to db lorestanuniv3
lorestanuniv3>
```

سپس پایگاه داده mongodb را به fastapi متصل میکنیم. مطابق تصویر زیر:

```
FA fastApiProject22  Version control  ×
API test_main.http  main.py  database.py  schemas.py
1  from motor.motor_asyncio import AsyncIOMotorClient
2
3  mongo_uri = "mongodb://localhost:27017"
4  client = AsyncIOMotorClient(mongo_uri)
5  database = client["lorestanuniv3"]
6
```

سپس در فایل main.py برای جدول student، api های درج و حذف و آپدیت و خواندن را مینویسیم و بعد هرکدام را ابتدا در insomnia انجام داده و تست کرده و سپس در محیط mongosh نیز تست میکنیم.

درج:

مطابق تصویر زیر یک api برای درج برای جدول دانشجو مینویسیم:

```
t_main.http  main.py × database.py schemas.py

from fastapi import FastAPI
from fastapi import HTTPException
import schemas
from database import *

app = FastAPI()

students_collection = database["students"]
teachers_collection = database["teacher"]
lessons_collection = database["lesson"]

@app.post(path: "/RegStu", response_model=schemas.Student)
async def create_student(student: schemas.StudentCreate):
    # بررسی تکراری بودن رکورد دانشجو
    existing_student = await students_collection.find_one({"stdid": student.stdid})
    if existing_student:
        raise HTTPException(status_code=400, detail="Student with the same stdid already exists")

    # ایجاد رکورد دانشجو با استفاده از اطلاعات ارسال شده
    created_student = await students_collection.insert_one(student.dict())
    # بازگرداندن رکورد ایجاد شده
    return student
```

سپس در insomnia تست میکنیم و یک رکورد درج میکنیم:

POST

http://127.0.0.1:8000/RegStu/

Send

200 OK

TIME 16 ms

SIZE 361 B

JSON

Auth

Query

Header 1

Docs

Preview

Header 4

Cookie

```
1 {
2   "fname": "پریماسا",
3   "lname": "مبارک",
4   "father": "کریم",
5   "birth": "1375/08/30",
6   "ids": "43000067",
7   "borncity": "مشهد",
8   "addres": "123 Main Sttttt",
9   "postalcode": "1234567899",
10  "cphone": "09300957712",
11  "hphone": "0615232866",
12  "department": "هنر",
13  "major": "مهندسی برق",
14  "married": "مجرد",
15  "scourseids": "8989",
16  "lids": "21118",
17  "id": "2801037432",
18  "stid": "40211415033"
19 }
20 }
21
```

```
1 {
2   "fname": "پریماسا",
3   "lname": "مبارک",
4   "father": "کریم",
5   "birth": "1375/08/30",
6   "ids": "43000067",
7   "borncity": "مشهد",
8   "addres": "123 Main Sttttt",
9   "postalcode": "1234567899",
10  "cphone": "9300957712",
11  "hphone": "615232866",
12  "department": "هنر",
13  "major": "مهندسی برق",
14  "married": "مجرد",
15  "id": "2801037432",
16  "scourseids": "8989",
17  "lids": "21118",
18  "stid": "40211415033"
19 }
```

در مرحله بعد در محیط mongosh مشاهده میکنیم که رکورد جدید درج شده است.


```
lorestanuniv3> db.students.find()
[
  {
    _id: ObjectId("65aa6f9717ebe73460034dd1"),
    fname: '????????',
    lname: '????',
    father: '????',
    birth: '1375/08/30',
    ids: '430000667',
    borncity: '????',
    addres: '123 Main Stttttt',
    postalcode: 1234567899,
    cphone: Long("9300957712"),
    hphone: 615232866,
    department: '???',
    major: '?????? ???',
    married: '????',
    id: '2801037432',
    scourseids: 8989,
    lids: 21118,
    stid: '40211415033'
  }
]
lorestanuniv3>
```

حذف:

مراحل قبل را تکرار میکنیم. پس در ابتدا یک api برای حذف مینویسیم:

```
@app.delete("/DelStu/{student_stid}")
async def delete_student(student_stid: str):
    # بررسی وجود رکورد با student_stid
    existing_student = await students_collection.find_one({"stid": student_stid})
    if existing_student is None:
        raise HTTPException(status_code=404, detail='STUDENT NOT FOUND')

    # حذف رکورد دانشجو
    await students_collection.delete_one({"stid": student_stid})

    return {"message": "Student deleted successfully"}
```

سپس از طریق insomnia حذف را انجام میدهیم:

DELETE ▼ http://127.0.0.1:8000/DelStu/40211415033/ Send		200 OK	TIME 31 ms	SIZE 42 B
JSON ▼	Auth ▼	Query	Header ¹	Docs
<pre>1 { 2 "fname": "پریسا", 3 "lname": "مبارک", 4 "father": "کریم", 5 "birth": "1375/08/30", 6 "ids": "43000067", 7 "borncity": "مشهد", 8 "addres": "123 Main Sttttt", 9 "postalcode": "1234567899", 10 "cphone": "09300957712", 11 "hphone": "0615232866", 12 "department": "منر", 13 "major": "مهندسی برق", 14 "married": "مجرد", 15 "scourseids": "8989", 16 "lids": "21118", 17 "id": "2801037432", 18 "stid": "40211415033" 19 } 20 }</pre>				
Preview ▼		Header ⁴	Cookie	Time
<pre>1 { 2 "message": "Student deleted successfully" 3 }</pre>				

در مرحله بعد در محیط mongosh مشاهده میکنیم که رکورد حذف شده است:

```
lorestanuniv3> db.students.find()

lorestanuniv3>
```

خواندن:

مراحل قبل را برای عملگر خواندن انجام میدهیم:

```
@app.get(path: "/RegStu/{student_std}", response_model=schemas.StudentRead)
async def read_student(student_std: str):
    # جستجوی دانشجو بر اساس
    student = await students_collection.find_one({"std": student_std})
    # بررسی وجود دانشجو
    if student is None:
        raise HTTPException(status_code=404, detail='STUDENT NOT FOUND')
    return student
```

GET Send **200 OK** TIME 47 ms

JSON Auth Query Header 1 Docs Preview Header 4 Cookies

```
1 {
2   "fname": "پریسا",
3   "lname": "مبارک",
4   "father": "کریم",
5   "birth": "1375/08/30",
6   "ids": "53000067",
7   "borncity": "مشهد",
8   "addres": "123 Main Sttttt",
9   "postalcode": "1234567899",
10  "cphone": "09300957712",
11  "hphone": "0615232866",
12  "department": "هنر",
13  "majon": "مهندسی برق",
14  "married": "مجرد",
15  "scourseids": "8989",
16  "lids": "21118",
17  "id": "2801037435",
18  "std": "40211415034"
19 }
20
```

```
1 {
2   "std": "40211415034",
3   "fname": "پریسا",
4   "lname": "مبارک",
5   "father": "کریم"
6 }
```

آپدیت:

مراحل قبل را برای عملگر آپدیت انجام میدهم.

ایجاد یک api برای آپدیت مطابق تصویر زیر:

```
@app.put(path: "/UpStu/{student_stid}", response_model=schemas.Student)
async def update_student(student_stid: str, updated_student: schemas.StudentCreate):
    # جستجوی رکورد دانشجو با استفاده از شناسه
    existing_student = await students_collection.find_one({"stid": student_stid})

    # بررسی یافت شدن رکورد
    if not existing_student:
        raise HTTPException(status_code=404, detail="Student not found")

    # آپدیت رکورد دانشجو با استفاده از اطلاعات ارسال شده
    updated_student_data = updated_student.dict(exclude_unset=True)
    updated_student = {**existing_student, **updated_student_data}
    await students_collection.update_one({"stid": student_stid}, {"$set": updated_student})

    # بازگرداندن رکورد دانشجو آپدیت شده
    return updated_student
```

آپدیت فیلد birth برای یک رکورد از طریق insomnia مطابق تصویر زیر:

PUT ▼ http://127.0.0.1:8000/UpStu/40211415034/ Send		200 OK	TIME 15 ms	SIZE 353 B
JSON ▼		Preview ▼		
Auth ▼		Header 4		
Query		Cookie		
Header 1		Docs		
<pre>1 { 2 "fname": "سارا", 3 "lname": "مبارک", 4 "father": "کریم", 5 "birth": "1376/04/24", 6 "ids": "53000067", 7 "borncity": "مشهد", 8 "addres": "123 Main Stttttt", 9 "postalcode": "1234567899", 10 "cphone": "09300957712", 11 "hphone": "0615232866", 12 "department": "هنر", 13 "major": "مهندسی برق", 14 "married": "مجرد", 15 "scourseids": "8989", 16 "lids": "21118", 17 "id": "2801037435", 18 "stid": "40211415034" 19 } 20 }</pre>		<pre>1 { 2 "fname": "سارا", 3 "lname": "مبارک", 4 "father": "کریم", 5 "birth": "1376/04/24", 6 "ids": "53000067", 7 "borncity": "مشهد", 8 "addres": "123 Main Stttttt", 9 "postalcode": 1234567899, 10 "cphone": 9300957712, 11 "hphone": 615232866, 12 "department": "هنر", 13 "major": "مهندسی برق", 14 "married": "مجرد", 15 "id": 2801037435, 16 "scourseids": 8989, 17 "lids": 21118, 18 "stid": 40211415034 19 }</pre>		

حال در mongosh مشاهده میکنیم که فیلد birth رکورد موردنظر آپدیت شده است:

```

lorestanuniv3> db.students.find()
[
  {
    _id: ObjectId("65aa879f93ef6871e638de41"),
    fname: '????',
    lname: '?????',
    father: '????',
    birth: '1375/08/30',
    ids: '530000267',
    borncity: '????',
    addres: '123 Main Stttttt',
    postalcode: 1234567899,
    cphone: Long("9300957712"),
    hphone: 615232866,
    department: '???',
    major: '?????? ???',
    married: '????',
    id: '2801037435',
    scourseids: 8989,
    lids: 21118,
    stid: '40211415034'
  }
]
lorestanuniv3> db.students.find()
[
  {
    _id: ObjectId("65aa879f93ef6871e638de41"),
    fname: '????',
    lname: '?????',
    father: '????',
    birth: '1376/04/24',
    ids: '530000267',
    borncity: '????',
    addres: '123 Main Stttttt',
    postalcode: 1234567899,
    cphone: Long("9300957712"),
    hphone: 615232866,
    department: '???',
    major: '?????? ???',
    married: '????',
    id: '2801037435',
    scourseids: 8989,
    lids: 21118,
    stid: '40211415034'
  }
]
lorestanuniv3>

```

برای جدوال دیگر یعنی استاد و درس نیز به همین صورت انجام میدهیم و برای هر کدام api های چهار عمل درج و حذف و خواندن و آپدیت را مینویسیم و سپس در insomnia اجرا کرده و بعد در محیط mongosh نتیجه را مشاهده میکنیم. در فایل های ضمیمه شده (فایل شماره ۲) تمام کدها قرار دارد.

۳. Dockerize کردن فعالیت یک :

ما برای داکرایز کردن پروژه خود باید ۳ فایل در کنار فایل های پروژه ایجاد کنیم:

Dockerfile:

فایل Dockerfile برای ساخت یک تصویر داکر سفارشی استفاده می شود که شامل برنامه ی ما و محیط اجرایی آن است. تصویر ساخته شده می تواند بر روی سیستم های مختلف تکرار شود و برنامه ما را در یک محیط مستقل و قابل حمل اجرا کند. این فایل را مطابق تصویر زیر ایجاد کردم:

```
stApiProject25  Version control  ✓
st_main.http  main.py  database.py  schemas.py  Dockerfile  doc
FROM python:3.10-bullseyepip install --upgrade pip

WORKDIR /source

COPY requirements.txt .

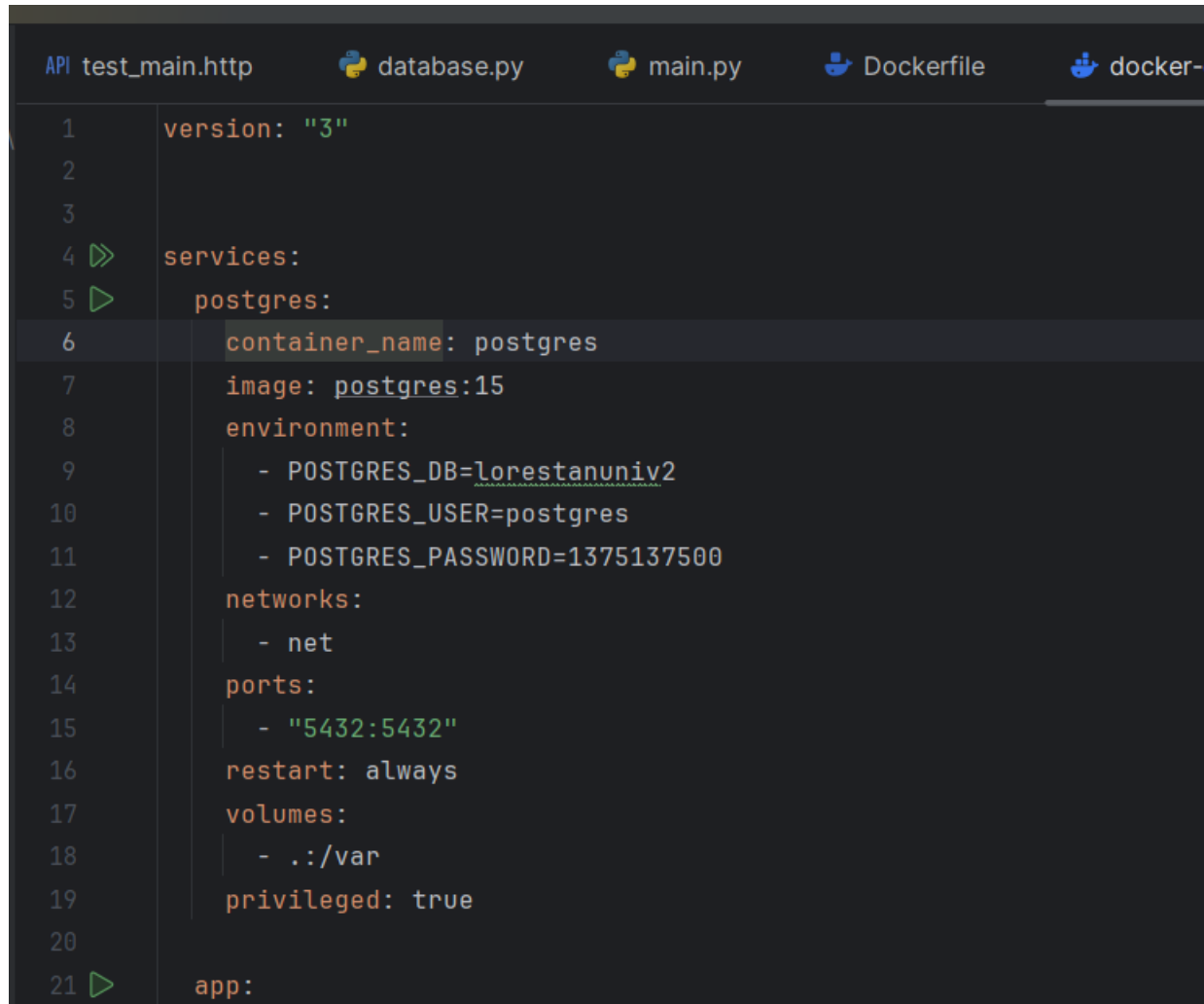
RUN pip install -r requirements.txt

COPY . .

# CMD ["uvicorn", "--host", "0.0.0.0", "--port", "8000", "main:app", "--reload"]
```

فایل docker-compose.yml:

فایل `docker-compose.yml` برای تعریف و مدیریت چندین کانتینر و ایجاد یک برنامه چند بخشی استفاده می‌شود. با استفاده از این فایل، می‌توانیم تصاویر داکر مورد نیاز برنامه‌ی خود را تعیین کنیم. به طور خلاصه در این فایل تمام کانتینرها و تصاویری که پروژه ما نیاز دارد مشخص می‌شود. بنابراین دو کانتینر `postgres` و `app` که پروژه اصلی هست را ایجاد می‌کنیم. این فلایل را به صورت زیر تشکیل دادم:



```
1 version: "3"
2
3
4 services:
5   postgres:
6     container_name: postgres
7     image: postgres:15
8     environment:
9       - POSTGRES_DB=lorestanuniv2
10      - POSTGRES_USER=postgres
11      - POSTGRES_PASSWORD=1375137500
12     networks:
13       - net
14     ports:
15       - "5432:5432"
16     restart: always
17     volumes:
18       - ./var
19     privileged: true
20
21 app:
```

```

app:
  build: .
  container_name: fastapi
  command: uvicorn --host 0.0.0.0 --port 8000 main:app --reload
  volumes:
    - ./source
  privileged: true
  ports:
    - "8080:8000"
  depends_on:
    - postgres
  networks:
    - net
  restart: always

networks:
  net:
    driver: bridge

```

فایل requirements.txt:

در این فایل تمام پکیج های مورد نیاز برای انجام پروژه را مشخص میکنیم:

	API test_main.http	database.py
1	fastapi==0.109.0	
2	uvicorn==0.26.0	
3	starlette==0.35.1	
4	sqlalchemy==2.0.25	
5	psycopg2==2.9.9	

ابتدا با وارد کردن دستورات زیر به ترتیب Container های فعال و درحال اجرا ، تمام Container ها ، و image ها را مشاهده میکنیم.

```
C:\Users\SAM\PycharmProjects\fastApiProject18>Docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                NAMES
478289a7cdd3   docker/welcome-to-docker:latest     "/docker-entrypoint..." 21 hours ago   Up 21 hours   0.0.0.0:8088->80/tcp   welcome-to-docker

C:\Users\SAM\PycharmProjects\fastApiProject18>docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                NAMES
478289a7cdd3   docker/welcome-to-docker:latest     "/docker-entrypoint..." 21 hours ago   Up 21 hours   0.0.0.0:8088->80/tcp   welcome-to-docker

C:\Users\SAM\PycharmProjects\fastApiProject18>docker images
REPOSITORY      TAG    IMAGE ID    CREATED    SIZE
docker/welcome-to-docker   latest   c1f619b6477e   2 months ago   18.6MB
```

حال image های لازم را دانلود و نصب میکنیم:

دانلود image پایتون:

```
C:\Users\SAM\PycharmProjects\fastApiProject18>docker pull python:3.10-bullseye
3.10-bullseye: Pulling from library/python
e455cf41eadb: Pull complete
3e4531da2f06: Pull complete
24098c8d74fa: Pull complete
c6374b4ceb7a: Pull complete
f0057575cfb9: Pull complete
bfbbf33acd5f: Pull complete
75388966077a: Pull complete
77c44461d1a1: Pull complete
Digest: sha256:4ce711926e79e68e0b42d0330c8c246ef81d31a744016626393e7d584a27bcd2
Status: Downloaded newer image for python:3.10-bullseye
docker.io/library/python:3.10-bullseye

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview python:3.10-bullseye
```

دانلود image های mongo و postgres :

```
C:\Users\SAM\PycharmProjects\fastApiProject18>docker pull mongo:6.0.13
6.0.13: Pulling from library/mongo
29202e855b20: Pull complete
e4a0a4cc0db4: Pull complete
81de79cc200c: Pull complete
40805d0ba6f9: Pull complete
e49cd075d869: Pull complete
323e6cdfb337: Pull complete
65ea20a31d41: Pull complete
f0f069fd88ba: Pull complete
cbeb5f85d916: Pull complete
Digest: sha256:656781dda791673991236157e1e25ebb52aa38ba5fde9adb64dc05d67b8ce886
Status: Downloaded newer image for mongo:6.0.13
docker.io/library/mongo:6.0.13

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview mongo:6.0.13

C:\Users\SAM\PycharmProjects\fastApiProject18>docker pull postgres:15
15: Pulling from library/postgres
2f44b7a888fa: Pull complete
6b9394701ee1: Pull complete
0868a116a0eb: Pull complete
87c0e5609f25: Pull complete
02023b353bc1: Pull complete
281b584da88e: Pull complete
acaa844956af: Pull complete
22738de2e568: Pull complete
4c9c0f61a7f9: Pull complete
df75f806b950: Pull complete
6f2bac5542ad: Pull complete
02ca619dca92: Pull complete
f364ed45b816: Pull complete
bb08d32722fa: Pull complete
Digest: sha256:89d430f5f7e680e67f81de534eb459a3b86afd818e3330d9b18d29a6b9ec98cd
Status: Downloaded newer image for postgres:15
docker.io/library/postgres:15

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview postgres:15
```

حال با وارد کردن دستور زیر تمام image ها را مشاهده میکنیم:

```
C:\Users\SAM\PycharmProjects\fastApiProject18>docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mongo	6.0.13	23e54c868737	2 days ago	690MB
postgres	15	d366292ba654	2 weeks ago	419MB
python	3.10-bullseye	2efb40c32a8e	4 weeks ago	911MB
docker/welcome-to-docker	latest	c1f619b6477e	2 months ago	18.6MB

دانلود image با نام hello-world:

```
C:\Users\SAM\PycharmProjects\fastApiProject18>docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:4bd78111b6914a99dbc560e6a20eab57ff6655aea4a80c50b0c5491968cbc2e6
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview hello-world
```

دانلود image با نام busybox:

```
C:\Users\SAM\PycharmProjects\fastApiProject18>docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
9ad63333ebc9: Pull complete
Digest: sha256:6d9ac9237a84afe1516540f40a0fafdc86859b2141954b4d643af7066d598b74
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview busybox
```

حال که تمام image های لازم را دانلود کردیم، با وارد کردن دستور `docker stop welcome-to-docker` کانتینر `welcome-to-docker` که در ابتدا به صورت خودکار تشکیل شده بود را حذف میکنیم:

```
C:\Users\SAM\PycharmProjects\fastApiProject18>docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                NAMES
478289a7cdd3   docker/welcome-to-docker:latest     "/docker-entrypoint.    22 hours ago   Up 22 hours   0.0.0.0:8088->80/tcp   welcome-to-docker

C:\Users\SAM\PycharmProjects\fastApiProject18>docker container
^C
C:\Users\SAM\PycharmProjects\fastApiProject18>docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B

C:\Users\SAM\PycharmProjects\fastApiProject18>docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                NAMES
478289a7cdd3   docker/welcome-to-docker:latest     "/docker-entrypoint.    22 hours ago   Up 22 hours   0.0.0.0:8088->80/tcp   welcome-to-docker

C:\Users\SAM\PycharmProjects\fastApiProject18>docker stop welcome-to-docker
welcome-to-docker

C:\Users\SAM\PycharmProjects\fastApiProject18>docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                NAMES
```

در مرحله بعد با استفاده از دستور زیر تمام container ها را همزمان باهم اجرا میکنیم:

```

C:\Users\SAM\PycharmProjects\fastApiProject18>docker-compose up -d
[+] Running 1/1
  📦 postgres Error
canceled

C:\Users\SAM\PycharmProjects\fastApiProject18>docker-compose up -d
[+] Building 111.6s (10/10) FINISHED
=> [app internal] load .dockerignore
=> => transferring context: 2B
=> [app internal] load build definition from Dockerfile
=> => transferring dockerfile: 249B
=> [app internal] load metadata for docker.io/library/python:3.10-bullseye
=> [app 1/5] FROM docker.io/library/python:3.10-bullseye
=> [app internal] load build context
=> => transferring context: 54.21MB
=> [app 2/5] WORKDIR /source
=> [app 3/5] COPY requirements.txt .
=> [app 4/5] RUN pip install -r requirements.txt
=> [app 5/5] COPY . .
=> [app] exporting to image
=> => exporting layers
=> => writing image sha256:cef659265e1101057380fc3f884cdb425ed5bb625b8eb05e545688a65d96e19e
=> => naming to docker.io/library/fastapiproject18-app
[+] Running 3/3
  📦 Network fastapiproject18_net Created
  📦 Container postgres Started
  📦 Container fastapi Started

```

حال میتوانیم با استفاده از دستورات زیر، container های در حال اجرا و همچنین image ها را مشاهده کنیم:

```

C:\Users\SAM\PycharmProjects\fastApiProject18>docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS                    NAMES
f99fe2ba39a3   fastapiproject18-app "uvicorn --host 0.0.0.0" About a minute ago Up 45 seconds  0.0.0.0:8080->8080/tcp   fastapi
be4d3a207b65   postgres:15         "docker-entrypoint.s..." About a minute ago Up 51 seconds  0.0.0.0:5432->5432/tcp   postgres

C:\Users\SAM\PycharmProjects\fastApiProject18>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED           SIZE
fastapiproject18-app latest             cef659265e11      2 minutes ago   989MB
mongo               6.0.13            23e54c868737      2 days ago     690MB
busybox             latest            3f57d9401f8d      3 days ago     4.26MB
postgres            15               d366292ba654      2 weeks ago    419MB
python              3.10-bullseye     2efb40c32a8e      4 weeks ago    911MB
docker/welcome-to-docker latest            c1f619b6477e      2 months ago   18.6MB
hello-world         latest            d2c94e258dcb      8 months ago   13.3kB

```

سپس برای تست از طریق برنامه insomnia یک رکورد در جدول student درج میکنیم:

POST http://127.0.0.1:8080/RegStu/ Send

Error TIME 0 ms SIZE 0 B

JSON Auth Query Header 1 Docs

Preview Header Cookie Timeline

```
1 {
2   "fname": "پریسا",
3   "lname": "مبارک",
4   "father": "کریم",
5   "birth": "1375/08/30",
6   "ids": "99000067",
7   "borncity": "مشهد",
8   "addres": "123 Main Stttttt",
9   "postalcode": "1234567899",
10  "cphone": "09300957712",
11  "hphone": "0615232866",
12  "department": "منر",
13  "major": "مهندسی برق",
14  "married": "مجرد",
15  "scourseids": "8989",
16  "lids": "21118",
17  "id": "2801037799",
18  "stid": "40211415099"
19
20 }
21
```

Error: Server returned nothing (no headers, no data)

Here are some additional things that may help.

Documentation Contact Support

مشاهده میکنیم که خطا میدهد. برای بررسی خطا برای هر container کد زیر را اجرا میکنیم. این دستور به ما میگوید که کدام container خطا دارد. مطابق تصویر زیر ابتدا برای container با نام postgres دستور docker logs را اجرا میکنیم و همانطور که مشاهده میکنیم هیچ اروری ندارد:

Select Command Prompt

```
C:\Users\SAM\PycharmProjects\fastApiProject18>docker logs postgres
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.utf8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /var/lib/postgresql/data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Etc/UTC
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
initdb: warning: enabling "trust" authentication for local connections
initdb: hint: You can change this by editing pg_hba.conf or using the option -A
syncing data to disk ... ok


Success. You can now start the database server using:

    pg_ctl -D /var/lib/postgresql/data -l logfile start


waiting for server to start....2024-01-21 14:51:15.484 UTC [48] LOG: starting
2024-01-21 14:51:15.768 UTC [48] LOG: listening on Unix socket "/var/run/postg
.2024-01-21 14:51:16.438 UTC [51] LOG: database system was shut down at 2024-0
2024-01-21 14:51:16.605 UTC [48] LOG: database system is ready to accept conne
done
server started
CREATE DATABASE

/usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*

2024-01-21 14:51:17.400 UTC [48] LOG: received fast shutdown request
waiting for server to shut down....2024-01-21 14:51:17.555 UTC [48] LOG: abort
2024-01-21 14:51:17.557 UTC [48] LOG: background worker "logical replication l
2024-01-21 14:51:17.558 UTC [49] LOG: shutting down
2024-01-21 14:51:17.757 UTC [49] LOG: checkpoint starting: shutdown immediate
.....2024-01-21 14:51:22.840 UTC [49] LOG: checkpoint complete: wrote 918 buff
average=0.012 s; distance=4223 kB, estimate=4223 kB
2024-01-21 14:51:22.854 UTC [48] LOG: database system is shut down
done
server stopped
```

```
server stopped

PostgreSQL init process complete; ready for start up.

2024-01-21 14:51:23.548 UTC [1] LOG:  starting PostgreSQL 15.5 (Debian 15.5-1.pgdg120+1) on x86_64-pc-linux
2024-01-21 14:51:23.549 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
2024-01-21 14:51:23.549 UTC [1] LOG:  listening on IPv6 address "::", port 5432
2024-01-21 14:51:24.139 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2024-01-21 14:51:25.468 UTC [64] LOG:  database system was shut down at 2024-01-21 14:51:22 UTC
2024-01-21 14:51:25.945 UTC [1] LOG:  database system is ready to accept connections
```

حال دستور `docker logs` را برای `container` با نام `fastapi` اجرا میکنیم و همانطور که در تصویر زیر مشاهده میکنیم با یک ارور برخورد میکنیم:

```
C:\Users\SAM\PycharmProjects\fastApiProject18>docker logs fastapi
INFO: Will watch for changes in these directories: ['/source']
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO: Started reloader process [1] using StatReload
/usr/local/lib/python3.10/site-packages/pydantic/_internal/_config.py:321: UserWarning: Valid config keys have changed in V2:
* 'orm_mode' has been renamed to 'from_attributes'
  warnings.warn(message, UserWarning)
Process SpawnProcess-1:
Traceback (most recent call last):
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/base.py", line 145, in __init__
    self._dbapi_connection = engine.raw_connection()
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/base.py", line 3293, in raw_connection
    return self.pool.connect()
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 452, in connect
    return _ConnectionFairy._checkout(self)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 1269, in _checkout
    fairy = _ConnectionRecord.checkout(pool)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 716, in checkout
    rec = pool._do_get()
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/impl.py", line 169, in _do_get
    with util.safe_reraise():
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/util/langhelpers.py", line 146, in __exit__
    raise exc_value.with_traceback(exc_tb)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/impl.py", line 167, in _do_get
    return self._create_connection()
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 393, in _create_connection
    return _ConnectionRecord(self)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 678, in __init__
    self._connect()
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 902, in __connect
    with util.safe_reraise():
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/util/langhelpers.py", line 146, in __exit__
    raise exc_value.with_traceback(exc_tb)
  File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 898, in __connect
    self.dbapi_connection = connection = pool._invoke_creator(self)
```



```

Select Command Prompt
with self.connect() as conn:
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/base.py", line 3269, in connect
    return self._connection_cls(self)
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/base.py", line 147, in __init__
    Connection._handle_dbapi_exception_noconnection(
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/base.py", line 2431, in _handle_db
    raise sqlalchemy_exception.with_traceback(exc_info[2]) from e
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/base.py", line 145, in __init__
    self._dbapi_connection = engine.raw_connection()
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/base.py", line 3293, in raw_conne
    return self.pool.connect()
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 452, in connect
    return _ConnectionFairy._checkout(self)
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 1269, in _checkout
    fairy = _ConnectionRecord.checkout(pool)
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 716, in checkout
    rec = pool._do_get()
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/impl.py", line 169, in _do_get
    with util.safe_reraise():
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/util/langhelpers.py", line 146, in __exit
    raise exc_value.with_traceback(exc_tb)
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/impl.py", line 167, in _do_get
    return self._create_connection()
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 393, in _create_conne
    return _ConnectionRecord(self)
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 678, in __init__
    self.__connect()
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 902, in __connect
    with util.safe_reraise():
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/util/langhelpers.py", line 146, in __exit
    raise exc_value.with_traceback(exc_tb)
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/pool/base.py", line 898, in __connect
    self.dbapi_connection = connection = pool._invoke_creator(self)
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/create.py", line 645, in connect
    return dialect.connect(*cargs, **cparams)
File "/usr/local/lib/python3.10/site-packages/sqlalchemy/engine/default.py", line 616, in connect
    return self.loaded_dbapi.connect(*cargs, **cparams)
File "/usr/local/lib/python3.10/site-packages/psycopg2/_init_.py", line 122, in connect
    conn = _connect(dsn, connection_factory=connection_factory, **kwasync)
sqlalchemy.exc.OperationalError: (psycopg2.OperationalError) could not connect to server: Connection
    Is the server running on host "localhost" (127.0.0.1) and accepting
    TCP/IP connections on port 5432?
could not connect to server: Cannot assign requested address
    Is the server running on host "localhost" (:::1) and accepting
    TCP/IP connections on port 5432?

(Background on this error at: https://sqlalche.me/e/20/e3q8)

```

پیام خطای ارائه شده نشان می‌دهد که در SQLAlchemy یک خطای عملیاتی (OperationalError) رخ داده است و به ما اعلام می‌کند که اتصال به سرور PostgreSQL در حال اجرا در میزبان "localhost" (۱۲۷.۰.۰.۱) به پورت ۵۴۳۲ انجام نشده است. این پیام مشکل احتمالی در اجرای سرور یا عدم پذیرش اتصالات TCP/IP را مطرح می‌کند.

۴. Dockerize کردن فعالیت دوم :

ابتدا ۳ فایل را همانند بخش قبل ایجاد میکنیم:

:Dockerfile

```
API test_main.http  main.py  database.py  schemas.py  Dockerfile  docker-c
1  FROM python:3.10-bullseyepip install --upgrade pip
2
3  WORKDIR /source
4
5  COPY requirements.txt .
6
7  RUN pip install -r requirements.txt .
8
9  COPY . .
10
11
12  # CMD ["uvicorn", "--host", "0.0.0.0", "--port", "8000", "main:app", "--reload"]
13
```

:Docker-compose.yml

دو کانتینر mongo و app که پروژه اصلی ما می باشد را ایجاد میکنیم:

```
API test_main.http  main.py  database.py  schemas.py  Docker

1  version: "3"
2
3  >> services:
4  >    mongo:
5      container_name: mongo
6      image: mongo:6.0.13
7      environment:|
8          - MONGO_INITDB_DATABASE=lorestanuniv3
9
10     networks:
11         - net
12     ports:
13         - "27017:27017"
14     restart: always
15     volumes:
16         - ./var
17     privileged: true
18
19  > app:
20     build: .
```

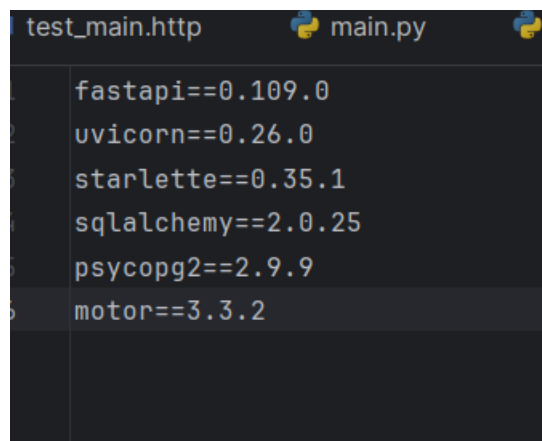
```

app:
  build: .
  container_name: fastapi2
  command: uvicorn --host 0.0.0.0 --port 8000 main:app --reload
  volumes:
    - ./source
  privileged: true
  ports:
    - "8081:8000"
  depends_on:
    - mongo
  networks:
    - net
  restart: always

networks:
  net:
    driver: bridge

```

فایل requirements.txt:



```

test_main.http  main.py
fastapi==0.109.0
uvicorn==0.26.0
starlette==0.35.1
sqlalchemy==2.0.25
psycopg2==2.9.9
motor==3.3.2

```

ابتدا image ها و container ها را مشاهده میکنیم:

```
C:\Users\SAM\PycharmProjects\fastApiProject25>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
fastapiproject18-app latest             0c7f91b7f0f1       6 days ago         1.02GB
mongo                6.0.13            23e54c868737       8 days ago         690MB
busybox              latest            3f57d9401f8d       9 days ago         4.26MB
postgres             15               d366292ba654       3 weeks ago        419MB
python               3.10-bullseye     2efb40c32a8e       5 weeks ago        911MB
docker/welcome-to-docker latest            c1f619b6477e       2 months ago       18.6MB
hello-world          latest            d2c94e258dcb       9 months ago       13.3kB

C:\Users\SAM\PycharmProjects\fastApiProject25>docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED             STATUS              PORTS                               NAMES
c434b995a986   fastapiproject18-app   "uvicorn --host 0.0.0..." 52 minutes ago     Up 52 minutes      0.0.0.0:8080->8000/tcp             fastapi
7e9d419c4e4f   postgres:15           "docker-entrypoint.s..." 3 days ago         Up 4 hours         0.0.0.0:5432->5432/tcp             postgres
```

مطابق تصویر بالا image های mongo و python از بخش قبل وجود دارند پس لازم نیست دانلود و نصب کنیم.

در مرحله بعد پروژه بخش دوم که اتصال mongodb به fastapi بود را داکرایز میکنیم :

```
[+] Running 2/2charmProjects\fastApiProject25>docker-compose up -d
 Container mongo      Running
 Container fastapi2   Started
```

و مشاهده میکنیم که container و image مربوطه ایجاد و run میشوند:

```
C:\Users\SAM\PycharmProjects\fastApiProject25>docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED             STATUS              PORTS                               NAMES
0026b26072a3   fastapiproject25-app   "uvicorn --host 0.0.0..." About a minute ago   Up About a minute   0.0.0.0:8081->8000/tcp             fastapi2
9f78aaa7e6e0   mongo:6.0.13          "docker-entrypoint.s..." 15 minutes ago      Up 4 minutes        0.0.0.0:27017->27017/tcp           mongo
c434b995a986   fastapiproject18-app   "uvicorn --host 0.0.0..." About an hour ago    Up About an hour    0.0.0.0:8080->8000/tcp             fastapi
7e9d419c4e4f   postgres:15           "docker-entrypoint.s..." 3 days ago          Up 4 hours         0.0.0.0:5432->5432/tcp             postgres

C:\Users\SAM\PycharmProjects\fastApiProject25>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
fastapiproject25-app latest             5373477dd10b       15 minutes ago     1.03GB
fastapiproject18-app latest            0c7f91b7f0f1       6 days ago         1.02GB
mongo                6.0.13            23e54c868737       9 days ago         690MB
busybox              latest            3f57d9401f8d       9 days ago         4.26MB
postgres             15               d366292ba654       3 weeks ago        419MB
python               3.10-bullseye     2efb40c32a8e       5 weeks ago        911MB
docker/welcome-to-docker latest            c1f619b6477e       2 months ago       18.6MB
hello-world          latest            d2c94e258dcb       9 months ago       13.3kB
```

مطابق تصویر بالا دو container با نام mongo و fastapi2 ایجاد شده اند. همچنین image با نام fastapiproject25-app ایجاد شده است.

حال برنامه باید به درستی اجرا شود و خروجی نمایش داده شود. بنابراین در insomnia تست میکنیم:

POST

http://127.0.0.1:8081/RegStu/

Send

500 Internal Server Error

TIME 30.3

JSON

Auth

Query

Header 1

Docs

Preview

Header 4

Cookie

1 {
2 "fname": "پریم",
3 "lname": "مبارک",
4 "father": "کریم",
5 "birth": "1375/08/30",
6 "ids": "99000067",
7 "borncity": "مشهد",
8 "addres": "123 Main Sttttt",
9 "postalcode": "1234567899",
10 "cphone": "09300957712",
11 "hphone": "0615232866",
12 "department": "هنر",
13 "major": "مهندسی برق",
14 "married": "مجرد",
15 "scourseids": "8989",
16 "lids": "21118",
17 "id": "2801037799",
18 "stid": "40211415099"
19 }
20
21

1 Internal Server Error

مطابق تصویر بالا مشاهده میکنیم که اجرای برنامه با خطا مواجه شده است. من بعد از این مرحله با استفاده از url زیر بررسی کردم که آیا میتوانم داکيومنت را مشاهده کنم یا خیر:

FastAPI 0.1.0 OAS 3.1

/openapi.json

default

POST /RegStu Create Student

GET /RegStu/{student_stid} Read Student

DELETE /DealStu/{student_stid} Delete Student

PUT /UpStu/{student_stid} Update Student

POST /RegTeach Create Teacher

GET /RegTeach/{lid} Read Teacher

DELETE /DealTeach/{teacher_lid} Delete Teacher

PUT /UpTeach/{lid} Update Teacher

POST /RegLes Create Lesson

GET /RegLes/{lesson_cid} Read Lesson

پس مطابق تصویر بالا داکيومنت نمايش داده ميشود.

بنابراین باید بررسی کنیم که کدام کانترینر خطا دارد. ابتدا برای container با نام mongo بررسی میکنیم:

```
C:\Users\SAM\PycharmProjects\fastApiProject25>docker logs mongo
about to fork child process, waiting until server is ready for connections.
forked process: 27

{"t":{"$date":"2024-01-27T16:46:22.265+00:00"},"s":"I", "c":"CONTROL", "id":20698, "ctx":
{"t":{"$date":"2024-01-27T16:46:22.281+00:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":
n":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWir
{"t":{"$date":"2024-01-27T16:46:22.289+00:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":
{"t":{"$date":"2024-01-27T16:46:22.303+00:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":
, and tcpFastOpenQueueSize."}
{"t":{"$date":"2024-01-27T16:46:22.364+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":
"config.tenantMigrationDonors"}}
{"t":{"$date":"2024-01-27T16:46:22.364+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":
ce":"config.tenantMigrationRecipients"}}
{"t":{"$date":"2024-01-27T16:46:22.364+00:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":
ig.tenantSplitDonors"}}
{"t":{"$date":"2024-01-27T16:46:22.365+00:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":
{"t":{"$date":"2024-01-27T16:46:22.367+00:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":
st":"9f78aaa7e6e0"}}
{"t":{"$date":"2024-01-27T16:46:22.367+00:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":
15d51bbd20c","opensslVersion":"OpenSSL 3.0.2 15 Mar 2022","modules":[],"allocator":"tcmalloc
{"t":{"$date":"2024-01-27T16:46:22.367+00:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":
["+":{"$date":"2024-01-27T16:46:22.367+00:00"},"s":"I", "c":"CONTROL", "id":21951, "ctx":

history_store:0,recovery:0,rts:0,salvage:0,tiered:0,timestamp:0,transaction:0,verify:0,log:0,}}
{"t":{"$date":"2024-01-27T16:46:39.627+00:00"},"s":"I", "c":"STORAGE", "id":4795906, "ctx":"initandlisten","msg"
{"t":{"$date":"2024-01-27T16:46:39.627+00:00"},"s":"I", "c":"RECOVERY", "id":23987, "ctx":"initandlisten","msg"
{"t":{"$date":"2024-01-27T16:46:39.836+00:00"},"s":"W", "c":"CONTROL", "id":22178, "ctx":"initandlisten","msg"
s":["startupWarnings"]}
{"t":{"$date":"2024-01-27T16:46:39.836+00:00"},"s":"W", "c":"CONTROL", "id":5123300, "ctx":"initandlisten","msg"
ns":838860,"tags":["startupWarnings"]}
{"t":{"$date":"2024-01-27T16:46:39.841+00:00"},"s":"I", "c":"NETWORK", "id":4915702, "ctx":"initandlisten","msg"
xWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6
on":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":17,"maxWireVersion":17},"outgoing":{"minWire
{"t":{"$date":"2024-01-27T16:46:39.842+00:00"},"s":"I", "c":"REPL", "id":5853300, "ctx":"initandlisten","msg"
":"startup"}}
{"t":{"$date":"2024-01-27T16:46:39.842+00:00"},"s":"I", "c":"STORAGE", "id":5071100, "ctx":"initandlisten","msg"
{"t":{"$date":"2024-01-27T16:46:39.844+00:00"},"s":"I", "c":"CONTROL", "id":20536, "ctx":"initandlisten","msg"
{"t":{"$date":"2024-01-27T16:46:39.845+00:00"},"s":"I", "c":"FTDC", "id":20625, "ctx":"initandlisten","msg"
ta"))}
{"t":{"$date":"2024-01-27T16:46:39.849+00:00"},"s":"I", "c":"REPL", "id":6015317, "ctx":"initandlisten","msg"
figPreStart"}}
{"t":{"$date":"2024-01-27T16:46:39.850+00:00"},"s":"I", "c":"STORAGE", "id":22262, "ctx":"initandlisten","msg"
{"t":{"$date":"2024-01-27T16:46:39.852+00:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Lis
{"t":{"$date":"2024-01-27T16:46:39.852+00:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Lis
{"t":{"$date":"2024-01-27T16:46:39.852+00:00"},"s":"I", "c":"NETWORK", "id":23016, "ctx":"listener","msg":"Wai
{"t":{"$date":"2024-01-27T16:46:39.852+00:00"},"s":"I", "c":"CONTROL", "id":8423403, "ctx":"initandlisten","msg"
,"Statistics":{"Transport layer setup":"0 ms","Run initial syncer crash recovery":"0 ms","Create storage engine lo
ms","Get metadata describing storage engine":"0 ms","Get metadata describing storage engine":"0 ms","Validate opti
rent startup options":"0 ms","Create storage engine":"0 ms","Create storage engine":"2108 ms","Write current PID t
ms","Initialize FCV before rebuilding indexes":"6 ms","Drop abandoned idents and get back indexes that need to be
that need to be rebuilt or builds that need to be restarted":"0 ms","Rebuild indexes for collections":"0 ms","Rebu
.system.users collection":"0 ms","Verify indexes for admin.system.users collection":"0 ms","Set up the background
plication coordinator":"2 ms","Start transport layer":"1 ms","_initAndListen total elapsed time":"2334 ms"}}}}
C:\Users\SAM\PycharmProjects\fastApiProject25>
```


پس container با نام mongo خطایی ندارد و به درستی اجرا میشود.

حال برای container با نام fastapi2 بررسی میکنیم:

```
C:\Users\SAM\PycharmProjects\fastApiProject25>docker logs fastapi2
INFO: Will watch for changes in these directories: ['/source']
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO: Started reload process [1] using StatReload
Process SpawnProcess-1:
Traceback (most recent call last):
  File "/usr/local/lib/python3.10/multiprocessing/process.py", line 314, in _bootstrap
    self.run()
  File "/usr/local/lib/python3.10/multiprocessing/process.py", line 108, in run
    self._target(*self._args, **self._kwargs)
  File "/usr/local/lib/python3.10/site-packages/uvicorn/_subprocess.py", line 78, in subprocess_started
    target(sockets=sockets)
  File "/usr/local/lib/python3.10/site-packages/uvicorn/server.py", line 62, in run
    return asyncio.run(self.serve(sockets=sockets))
  File "/usr/local/lib/python3.10/asyncio/runners.py", line 44, in run
    return loop.run_until_complete(main)
  File "/usr/local/lib/python3.10/asyncio/base_events.py", line 649, in run_until_complete
    return future.result()
  File "/usr/local/lib/python3.10/site-packages/uvicorn/server.py", line 69, in serve
    config.load()
  File "/usr/local/lib/python3.10/site-packages/uvicorn/config.py", line 458, in load
    self.loaded_app = import_from_string(self.app)
  File "/usr/local/lib/python3.10/site-packages/uvicorn/importer.py", line 21, in import_from_string
    module = importlib.import_module(module_str)
  File "/usr/local/lib/python3.10/importlib/_init_.py", line 126, in import_module
    return _bootstrap.gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1050, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1027, in _find_and_load
  File "<frozen importlib._bootstrap>", line 1006, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 688, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 879, in exec_module
```

```
File "/usr/local/lib/python3.10/site-packages/pymongo/topology.py", line 259, in _select_servers_loop
    raise ServerSelectionTimeoutError(
pymongo.errors.ServerSelectionTimeoutError: localhost:27017: [Errno 111] Connection refused (configured timeouts: socketTimeoutMS: 20000.0ms, connectTimeoutMS: 20000.0ms), Timeout: 30s, Topology Description: <TopologyDescription id: 65b535691173b4266e6b7001, topology_type: Unknown, servers: [{ServerDescription('localhost', 27017) server_type: Unknown, rtt: None, error=AutoReconnect('localhost:27017: [Errno 111] Connection refused (configured timeouts: socketTimeoutMS: 20000.0ms, connectTimeoutMS: 20000.0ms))'}]>
INFO: 172.20.0.1:44608 - "GET /docs HTTP/1.1" 200 OK
INFO: 172.20.0.1:44608 - "GET /openapi.json HTTP/1.1" 200 OK
C:\Users\SAM\PycharmProjects\fastApiProject25>
```

مطابق تصویر بالا مشاهده میکنیم که این کانترینر خطا دارد. این پیام خطا از طرف کتابخانه pymongo نشان می‌دهد که ارتباط با سرویس دیتابیس MongoDB با آدرس localhost:27017 ناموفق بوده است.

