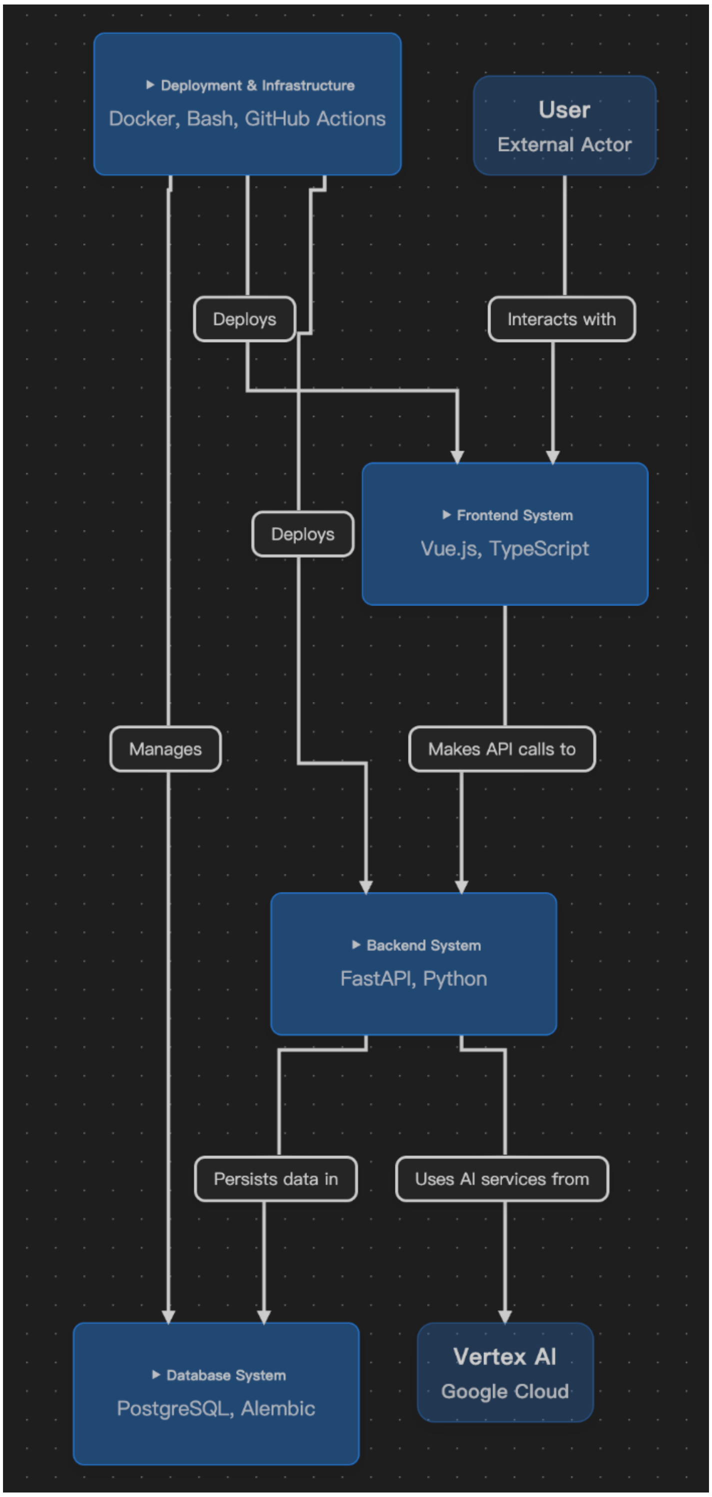
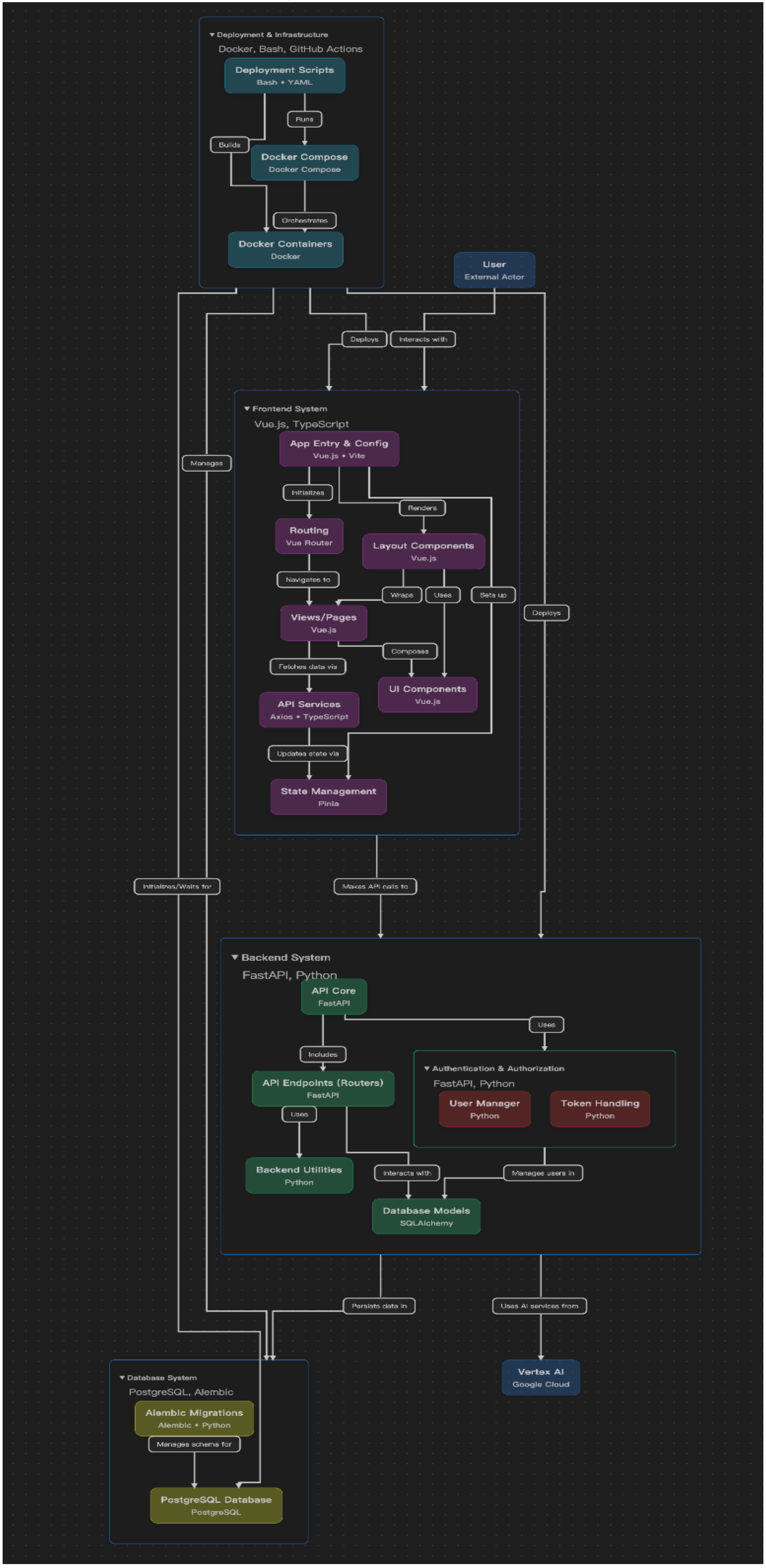
Service dependency diagram.

Overview:

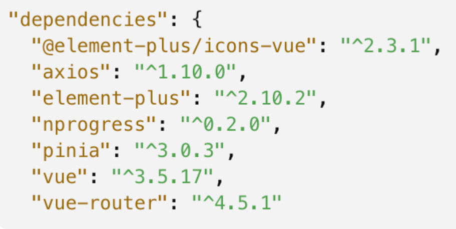


Service dependency diagram



Front-end

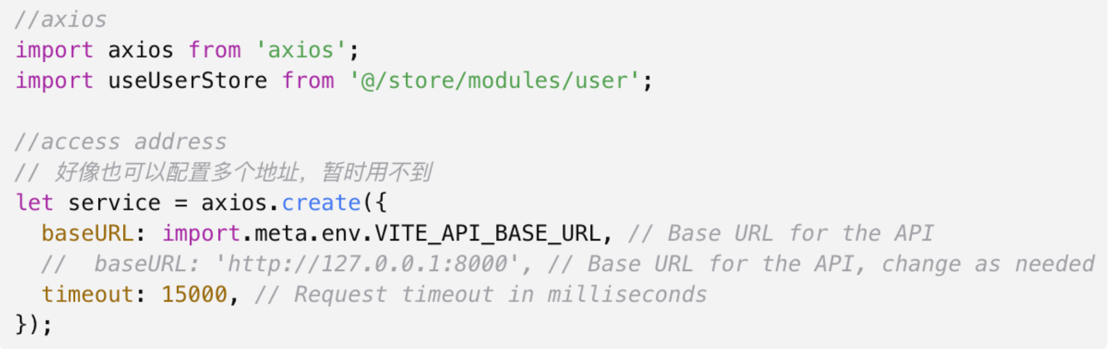
- Vue 3 application using Vite and Axios for HTTP calls



```
"dependencies": {
  "@element-plus/icons-vue": "^2.3.1",
  "axios": "^1.10.0",
  "element-plus": "^2.10.2",
  "nprogress": "^0.2.0",
  "pinia": "^3.0.3",
  "vue": "^3.5.17",
  "vue-router": "^4.5.1"
```

frontend/package.json

- Axios instance configured with a runtime base URL so the app can target the FastAPI service



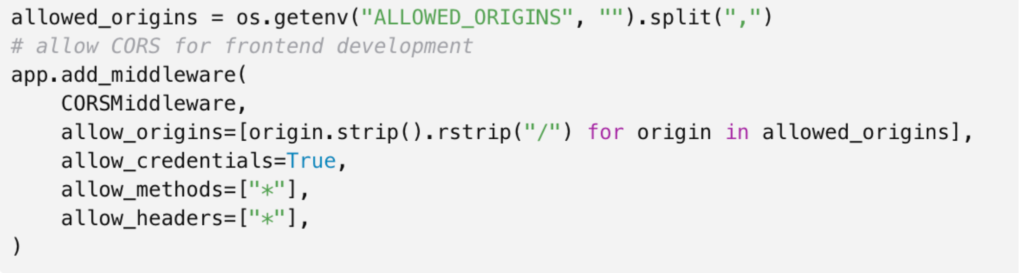
```
//axios
import axios from 'axios';
import useUserStore from '@/store/modules/user';

//access address
// 好像也可以配置多个地址，暂时用不到
let service = axios.create({
  baseURL: import.meta.env.VITE_API_BASE_URL, // Base URL for the API
  //  baseURL: 'http://127.0.0.1:8000', // Base URL for the API, change as needed
  timeout: 15000, // Request timeout in milliseconds
});
```

- Example API usage: /auth/jwt/login, /users/me, and /auth/register

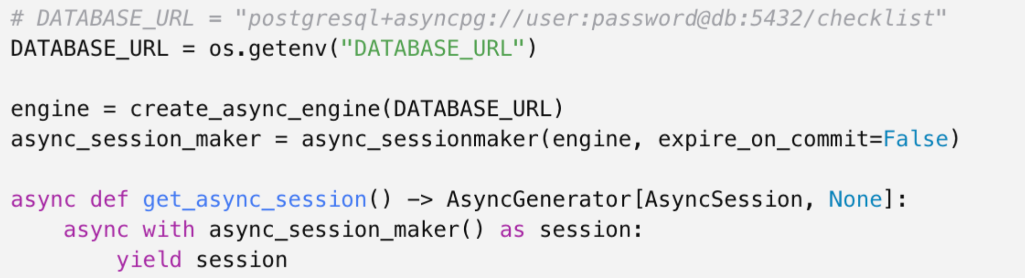
Back-end

- FastAPI application with routers for authentication, users, roles, checklists, and avatars
- CORS middleware allows the browser-based front-end to access the API



```
allowed_origins = os.getenv("ALLOWED_ORIGINS", "").split(",")
# allow CORS for frontend development
app.add_middleware(
    CORSMiddleware,
    allow_origins=[origin.strip().rstrip("/") for origin in allowed_origins],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
```

- Asynchronous SQLAlchemy connection to a database specified by DATABASE_URL



```
# DATABASE_URL = "postgresql+asyncpg://user:password@db:5432/checklist"
DATABASE_URL = os.getenv("DATABASE_URL")

engine = create_async_engine(DATABASE_URL)
async_session_maker = async_sessionmaker(engine, expire_on_commit=False)

async def get_async_session() -> AsyncGenerator[AsyncSession, None]:
    async with async_session_maker() as session:
        yield session
```

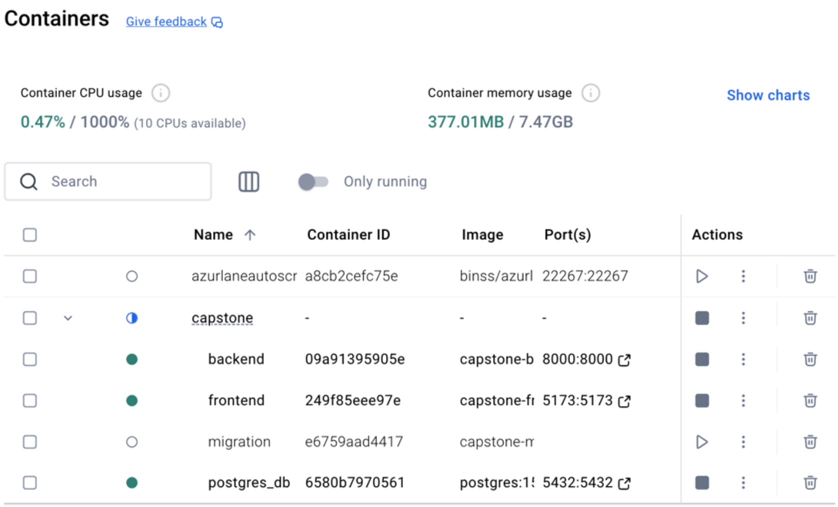
Database

- PostgreSQL service defined in docker-compose.yml and mounted into the backend via DATABASE_URL

External APIs

- Google Generative AI (Gemini) used to generate checklist content via google.generativeai
- Azure OpenAI plus Azure Blob Storage for avatar generation and persistence

Monitoring (logs location, component health checks etc).
For monitoring, all logs, health check can be easily controlled by the Docker containers. For local deployment, for example:



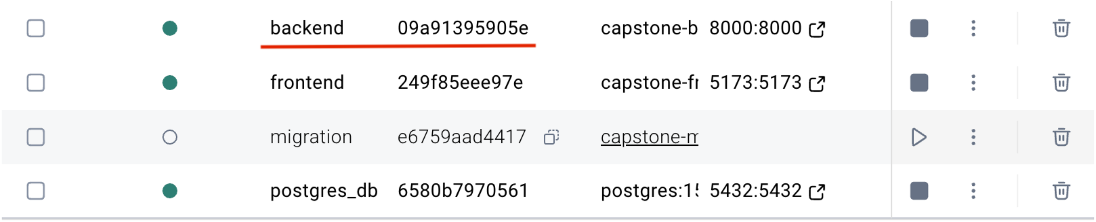
Containers Give feedback

Container CPU usage
0.47% / 1000% (10 CPUs available)

Container memory usage
377.01MB / 7.47GB

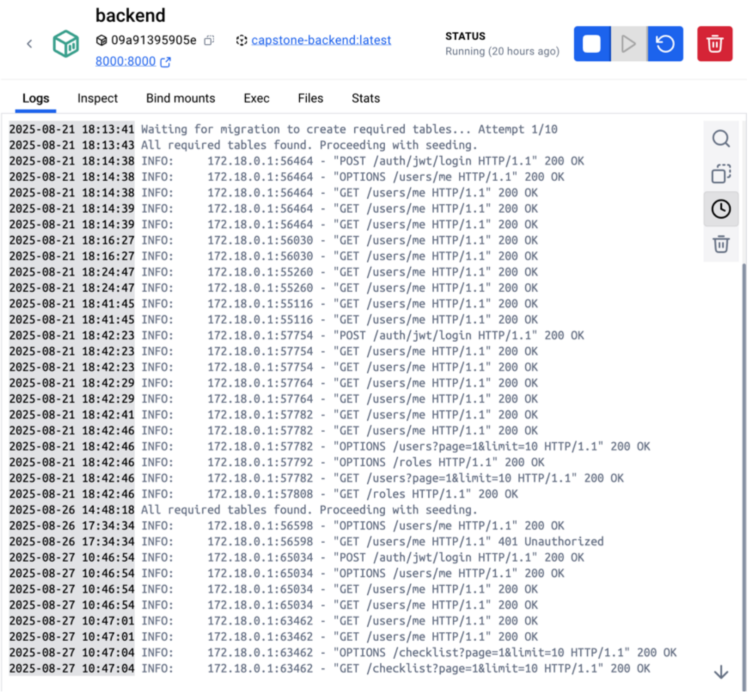
□			Name ↑	Container ID	Image	Port(s)	Actions
□		○	azurlaneautoscr	a8cb2cefc75e	binss/azurl	22267:22267	▷ ⋮ 🗑
□	⌄	◑	capstone	-	-	-	■ ⋮ 🗑
□		●	backend	09a91395905e	capstone-b	8000:8000	■ ⋮ 🗑
□		●	frontend	249f85eee97e	capstone-fı	5173:5173	■ ⋮ 🗑
□		○	migration	e6759aad4417	capstone-m		▷ ⋮ 🗑
□		●	postgres_db	6580b7970561	postgres:1!	5432:5432	■ ⋮ 🗑

We can see the backend, frontend and database works well right now. Similarly, all logs is store and managed in Docker. For instance: if click each part, for example the backend:
Click here:



□						
□	●	backend	09a91395905e	capstone-b	8000:8000	■ ⋮ 🗑
□	●	frontend	249f85eee97e	capstone-fı	5173:5173	■ ⋮ 🗑
□	○	migration	e6759aad4417	capstone-m		▷ ⋮ 🗑
□	●	postgres_db	6580b7970561	postgres:1!	5432:5432	■ ⋮ 🗑

Client can see all the logs for all service status.



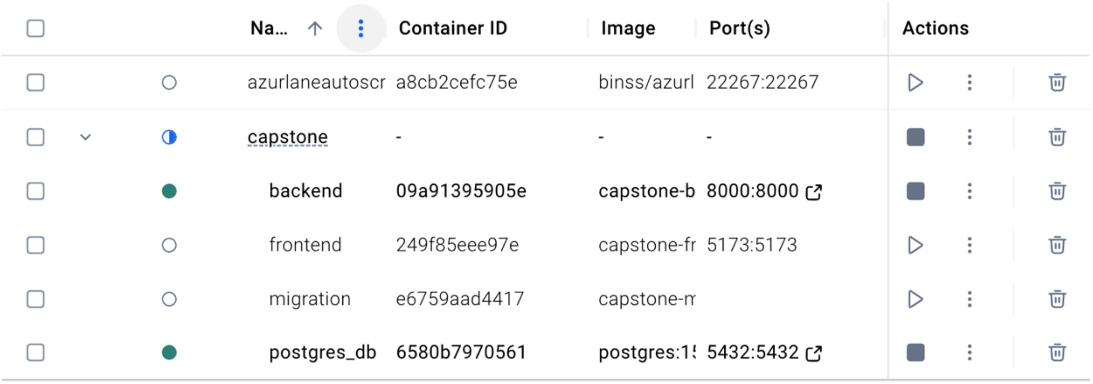
backend
09a91395905e capstone-backend:latest
8000:8000

STATUS
Running (20 hours ago)

```
2025-08-21 18:13:41 Waiting for migration to create required tables... Attempt 1/10
2025-08-21 18:13:43 All required tables found. Proceeding with seeding.
2025-08-21 18:14:38 INFO:     172.18.0.1:56464 - "POST /auth/jwt/login HTTP/1.1" 200 OK
2025-08-21 18:14:38 INFO:     172.18.0.1:56464 - "OPTIONS /users/me HTTP/1.1" 200 OK
2025-08-21 18:14:38 INFO:     172.18.0.1:56464 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:14:39 INFO:     172.18.0.1:56464 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:14:39 INFO:     172.18.0.1:56464 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:16:27 INFO:     172.18.0.1:56030 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:16:27 INFO:     172.18.0.1:56030 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:24:47 INFO:     172.18.0.1:55260 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:24:47 INFO:     172.18.0.1:55260 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:41:45 INFO:     172.18.0.1:55116 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:41:45 INFO:     172.18.0.1:55116 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:42:23 INFO:     172.18.0.1:57754 - "POST /auth/jwt/login HTTP/1.1" 200 OK
2025-08-21 18:42:23 INFO:     172.18.0.1:57754 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:42:23 INFO:     172.18.0.1:57754 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:42:29 INFO:     172.18.0.1:57764 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:42:29 INFO:     172.18.0.1:57764 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:42:41 INFO:     172.18.0.1:57782 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:42:46 INFO:     172.18.0.1:57782 - "GET /users/me HTTP/1.1" 200 OK
2025-08-21 18:42:46 INFO:     172.18.0.1:57782 - "OPTIONS /users?page=1&limit=10 HTTP/1.1" 200 OK
2025-08-21 18:42:46 INFO:     172.18.0.1:57792 - "OPTIONS /roles HTTP/1.1" 200 OK
2025-08-21 18:42:46 INFO:     172.18.0.1:57782 - "GET /users?page=1&limit=10 HTTP/1.1" 200 OK
2025-08-21 18:42:46 INFO:     172.18.0.1:57808 - "GET /roles HTTP/1.1" 200 OK
2025-08-26 14:48:18 All required tables found. Proceeding with seeding.
2025-08-26 17:34:34 INFO:     172.18.0.1:56598 - "OPTIONS /users/me HTTP/1.1" 200 OK
2025-08-26 17:34:34 INFO:     172.18.0.1:56598 - "GET /users/me HTTP/1.1" 401 Unauthorized
2025-08-27 10:46:54 INFO:     172.18.0.1:65034 - "POST /auth/jwt/login HTTP/1.1" 200 OK
2025-08-27 10:46:54 INFO:     172.18.0.1:65034 - "OPTIONS /users/me HTTP/1.1" 200 OK
2025-08-27 10:46:54 INFO:     172.18.0.1:65034 - "GET /users/me HTTP/1.1" 200 OK
2025-08-27 10:46:54 INFO:     172.18.0.1:65034 - "GET /users/me HTTP/1.1" 200 OK
2025-08-27 10:47:01 INFO:     172.18.0.1:63462 - "GET /users/me HTTP/1.1" 200 OK
2025-08-27 10:47:01 INFO:     172.18.0.1:63462 - "GET /users/me HTTP/1.1" 200 OK
2025-08-27 10:47:04 INFO:     172.18.0.1:63462 - "OPTIONS /checklist?page=1&limit=10 HTTP/1.1" 200 OK
2025-08-27 10:47:04 INFO:     172.18.0.1:63462 - "GET /checklist?page=1&limit=10 HTTP/1.1" 200 OK
```

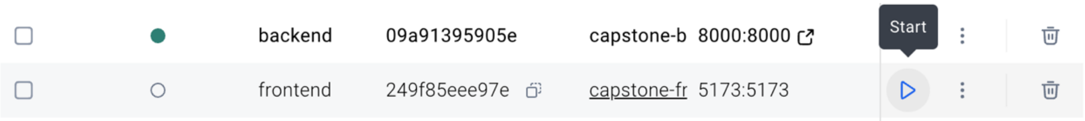
Common incidents & recovery steps (examples: database connection loss, service crash).

All service running status can be monitored by Docker. After deployment, if service crash happened like below: (frontend is down)



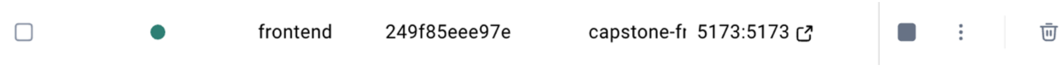
□			Na... ↑ ⋮	Container ID	Image	Port(s)	Actions
□		○	azurlaneautoscr	a8cb2cefc75e	binss/azurl	22267:22267	▷ ⋮ 🗑
□	⌄	◑	capstone	-	-	-	■ ⋮ 🗑
□		●	backend	09a91395905e	capstone-b	8000:8000 ↗	■ ⋮ 🗑
□		○	frontend	249f85eee97e	capstone-fr	5173:5173	▷ ⋮ 🗑
□		○	migration	e6759aad4417	capstone-m		▷ ⋮ 🗑
□		●	postgres_db	6580b7970561	postgres:1!	5432:5432 ↗	■ ⋮ 🗑

Click the start button to restart the service.



□	●	backend	09a91395905e	capstone-b	8000:8000 ↗	⋮ 🗑
□	○	frontend	249f85eee97e	capstone-fr	5173:5173	▷ ⋮ 🗑

Frontend recovered.



□	●	frontend	249f85eee97e	capstone-fr	5173:5173 ↗	■ ⋮ 🗑

Or try redeploying the entire system:
In bash, enter

```
docker compose down -v
```

Then use:

```
./start.sh
```

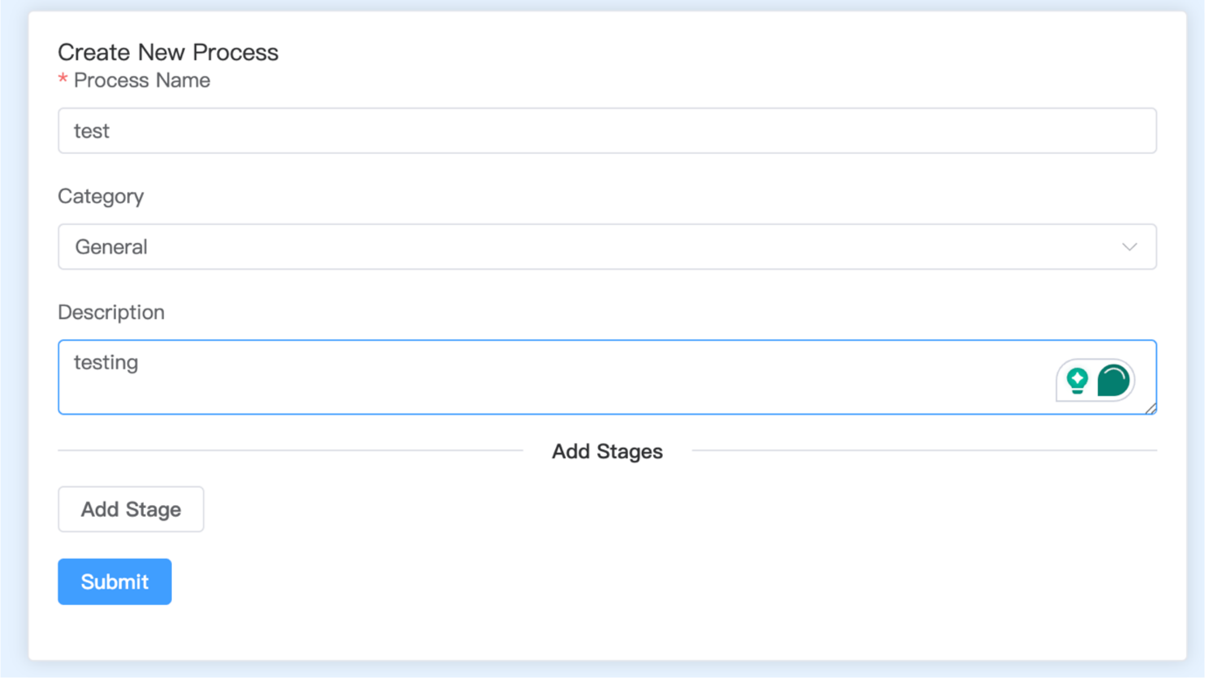
To complete full automatic redeployment.

Testing scenarios & results

Considering over all testing cases includes several hundreds pages, testing can be found in the “Testing Scenarios” folders.

Test for integration:

Try adding a checklist:



Create New Process

* Process Name

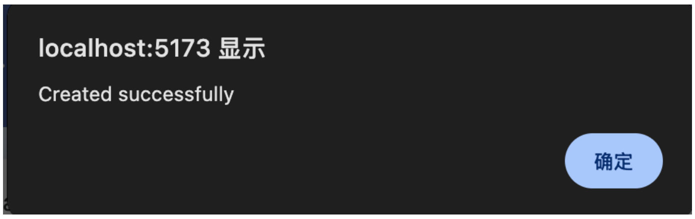
test

Category

General

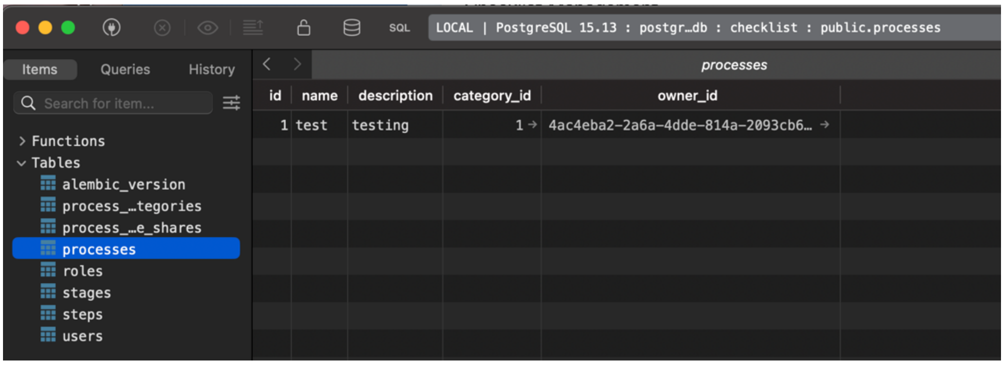
Description

testing



localhost:5173 显示

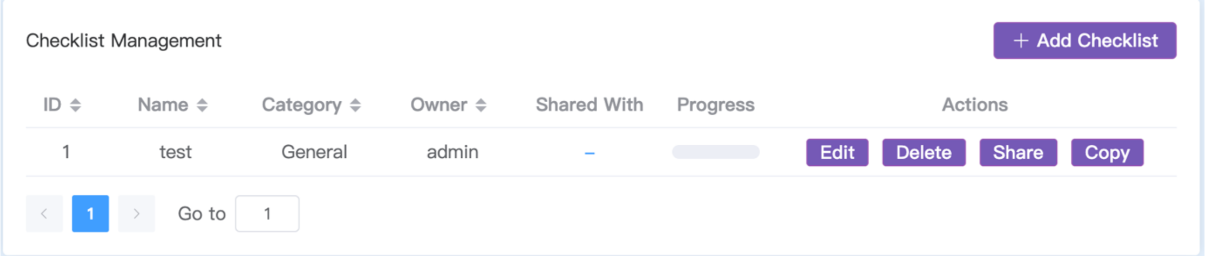
Created successfully



LOCAL | PostgreSQL 15.13 : postgr…db : checklist : public.processes

```
> Functions
v Tables
    alembic_version
    process_…tegories
    process_…e_shares
    processes
    roles
    stages
    steps
    users
```

id	name	description	category_id	owner_id
1	test	testing	1 →	4ac4eba2-2a6a-4dde-814a-2093cb6… →



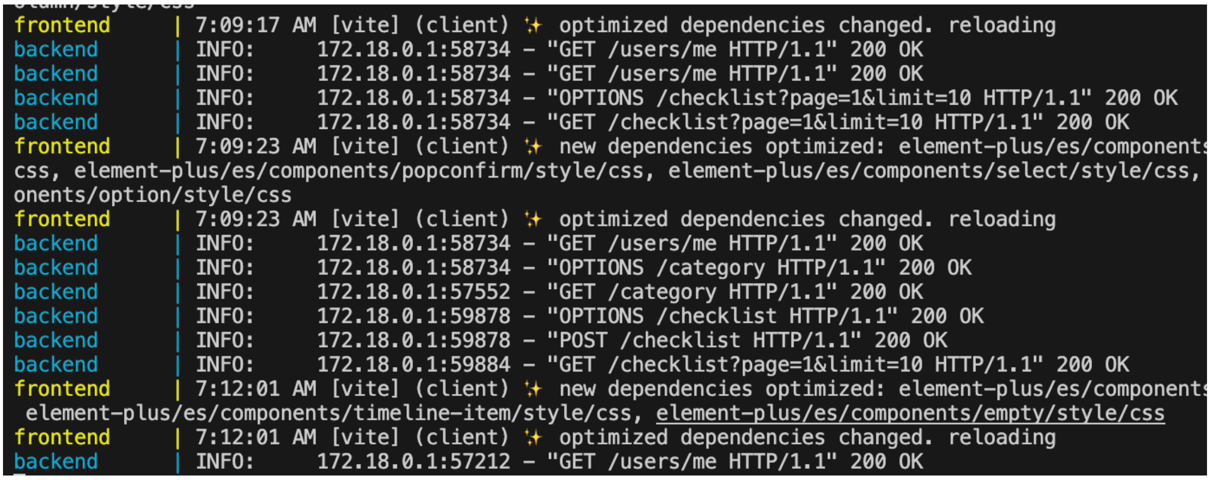
Checklist Management

ID	Name	Category	Owner	Shared With	Progress	Actions
1	test	General	admin	–		Edit Delete Share Copy

Integration is great. Checklist successfully added and changes reflected on both frontend page and database record.

Related logs as evidence:

API correctly called and processed



```
olumn/style/css
frontend     | 7:09:17 AM [vite] (client) ✨ optimized dependencies changed. reloading
backend      | INFO:     172.18.0.1:58734 - "GET /users/me HTTP/1.1" 200 OK
backend      | INFO:     172.18.0.1:58734 - "GET /users/me HTTP/1.1" 200 OK
backend      | INFO:     172.18.0.1:58734 - "OPTIONS /checklist?page=1&limit=10 HTTP/1.1" 200 OK
backend      | INFO:     172.18.0.1:58734 - "GET /checklist?page=1&limit=10 HTTP/1.1" 200 OK
frontend     | 7:09:23 AM [vite] (client) ✨ new dependencies optimized: element-plus/es/components
css, element-plus/es/components/popconfirm/style/css, element-plus/es/components/select/style/css,
onents/option/style/css
frontend     | 7:09:23 AM [vite] (client) ✨ optimized dependencies changed. reloading
backend      | INFO:     172.18.0.1:58734 - "GET /users/me HTTP/1.1" 200 OK
backend      | INFO:     172.18.0.1:58734 - "OPTIONS /category HTTP/1.1" 200 OK
backend      | INFO:     172.18.0.1:57552 - "GET /category HTTP/1.1" 200 OK
backend      | INFO:     172.18.0.1:59878 - "OPTIONS /checklist HTTP/1.1" 200 OK
backend      | INFO:     172.18.0.1:59878 - "POST /checklist HTTP/1.1" 200 OK
backend      | INFO:     172.18.0.1:59884 - "GET /checklist?page=1&limit=10 HTTP/1.1" 200 OK
frontend     | 7:12:01 AM [vite] (client) ✨ new dependencies optimized: element-plus/es/components
 element-plus/es/components/timeline-item/style/css, element-plus/es/components/empty/style/css
frontend     | 7:12:01 AM [vite] (client) ✨ optimized dependencies changed. reloading
backend      | INFO:     172.18.0.1:57212 - "GET /users/me HTTP/1.1" 200 OK
```

Post-deployment tests:

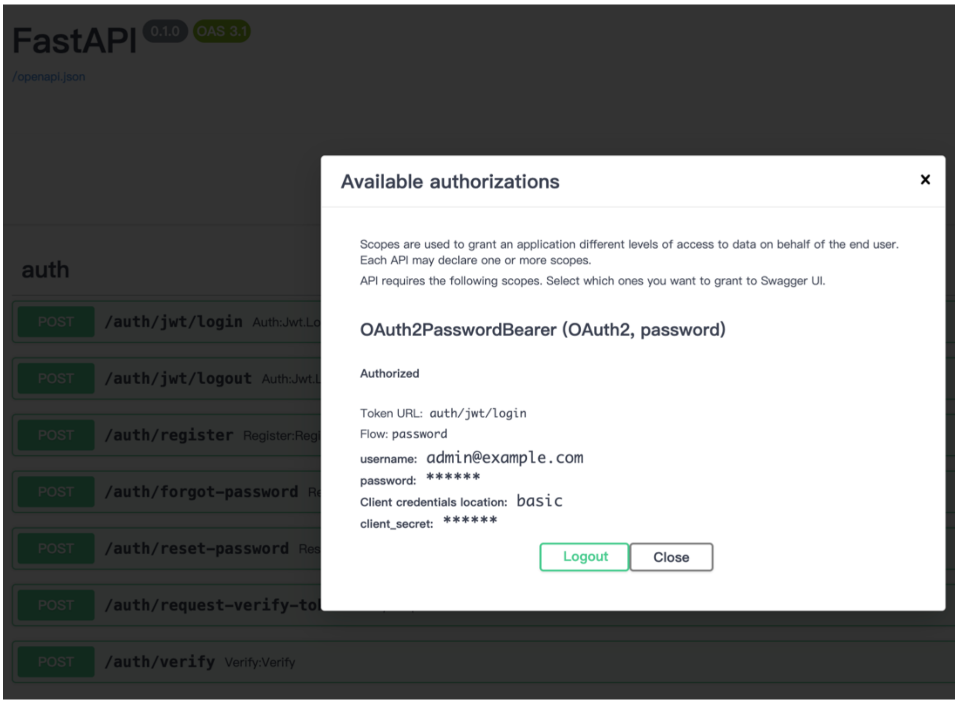
Access:

https://capstoneapp-frontend.agreeablepebble-d64b0d17.westus.azurecontainerapps.io

https://capstoneapp-backend.agreeablepebble-d64b0d17.westus.azurecontainerapps.io/docs

Result:

Backend able to process our login requests



FastAPI
0.1.0
OAS 3.1
/openapi.json
auth
POST /auth/jwt/login Auth:Jwt.Lo
POST /auth/jwt/logout Auth:Jwt.L
POST /auth/register Register:Regi
POST /auth/forgot-password R
POST /auth/reset-password Res
POST /auth/request-verify-to
POST /auth/verify Verify:Verify
Available authorizations
×

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.

API requires the following scopes. Select which ones you want to grant to Swagger UI.

OAuth2PasswordBearer (OAuth2, password)

Authorized

Token URL: auth/jwt/login

Flow: password

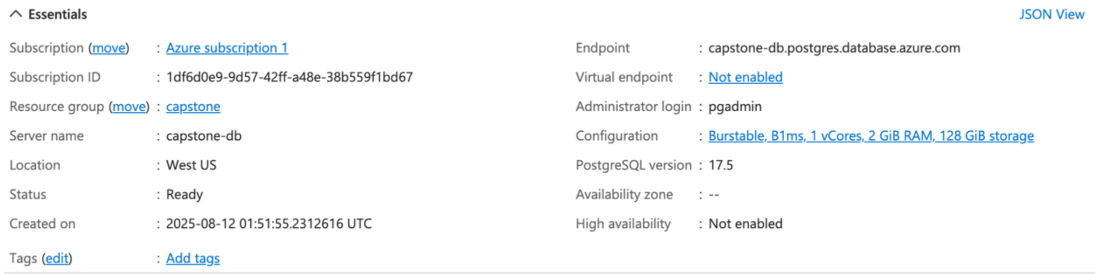
username: admin@example.com

password: ******

Client credentials location: basic

client_secret: ******

Database status ready



^ Essentials

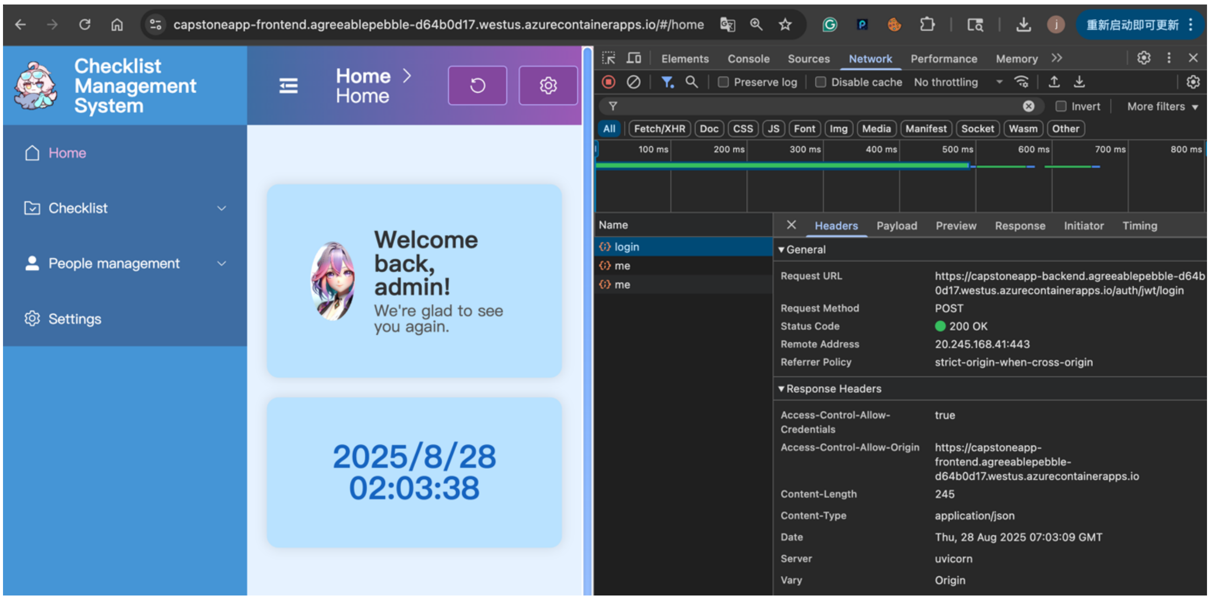
JSON View

Subscription (move) : Azure subscription 1
Subscription ID : 1df6d0e9-9d57-42ff-a48e-38b559f1bd67
Resource group (move) : capstone
Server name : capstone-db
Location : West US
Status : Ready
Created on : 2025-08-12 01:51:55.2312616 UTC
Tags (edit) : Add tags

Endpoint : capstone-db.postgres.database.azure.com
Virtual endpoint : Not enabled
Administrator login : pgadmin
Configuration : Burstable, B1ms, 1 vCores, 2 GiB RAM, 128 GiB storage
PostgreSQL version : 17.5
Availability zone : --
High availability : Not enabled

Able to see the frontend, and able to do the API test.

Server able to process all requests, routing functions well
(Login request for example)



capstoneapp-frontend.agreeablepebble-d64b0d17.westus.azurecontainerapps.io/#/home

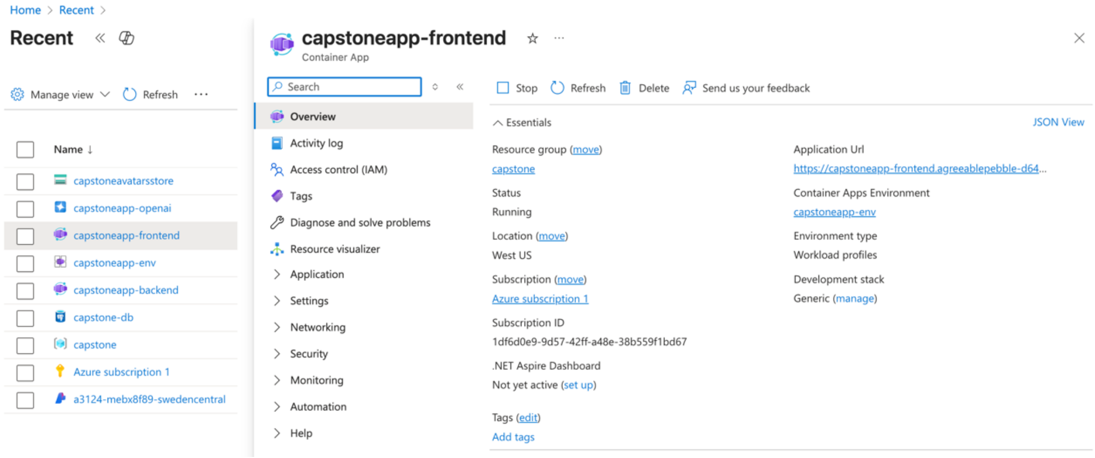
▼General

Request URL: https://capstoneapp-backend.agreeablepebble-d64b0d17.westus.azurecontainerapps.io/auth/jwt/login
Request Method: POST
Status Code: ● 200 OK
Remote Address: 20.245.168.41:443
Referrer Policy: strict-origin-when-cross-origin

▼Response Headers

Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: https://capstoneapp-frontend.agreeablepebble-d64b0d17.westus.azurecontainerapps.io
Content-Length: 245
Content-Type: application/json
Date: Thu, 28 Aug 2025 07:03:09 GMT
Server: uvicorn
Vary: Origin

For current cloud deployment (Microsoft Azure):
All services are hosted by the cloud platform provider **Microsoft Azure**. The author is solely responsible for maintaining, managing, and explaining the services of this specific instance. (As of now, the frontend service is confirmed to be running.)



capstoneapp-frontend
Container App

^ Essentials

JSON View

Resource group (move)
capstone

Status
Running

Location (move)
West US

Subscription (move)
Azure subscription 1

Subscription ID
1df6d0e9-9d57-42ff-a48e-38b559f1bd67

.NET Aspire Dashboard
Not yet active (set up)

Tags (edit)
Add tags

Application Url
https://capstoneapp-frontend.agreeablepebble-d64...

Container Apps Environment
capstoneapp-env

Environment type
Workload profiles

Development stack
Generic (manage)

For self-deployment on other cloud service providers:
All rights are reserved by the respective cloud service provider. The author bears no responsibility for costs, billing, or ongoing management outside of the provided deployment instance. Any fees, infrastructure management, or operational expenses are entirely subject to the chosen cloud provider’s policies and are **not associated with or covered by the author**.