Part1

Command: docker compose ps

```
PS C:\Users\javva\tinytasks> docker compose up -d
>>
time="2025-11-16T20:28:32-06:00" level=warning msg="C:\\Users\\javva\\tinytasks\\docker-compose.yml: the attribute `version` is obsolete, it
ill be ignored, please remove it to avoid potential confusion"
[+] Running 4/4
 ✓ Network tinytasks_default  Created                                                                                    0.1
 ✓ Volume tinytasks_pgdata    Created                                                                                    0.0
 ✓ Container tinytasks-db-1   Healthy                                                                                    6.4
 ✓ Container tinytasks-api-1  Started                                                                                    6.5
```

Command: before post: curl http://localhost:8083/tasks

```
PS C:\Users\javva\tinytasks> curl http://localhost:8083/tasks
>>


StatusCode        : 200
StatusDescription : OK
Content           : []

RawContent        : HTTP/1.1 200 OK
                    Connection: close
                    Content-Length: 3
                    Content-Type: application/json
                    Date: Mon, 17 Nov 2025 02:28:51 GMT
                    Server: gunicorn

                    []

Forms             : {}
Headers           : {[Connection, close], [Content-Length, 3], [Content-Type, application/json], [Date, Mon, 17 Nov 2025 02:28:51 GMT]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 3
```

Command: Invoke-WebRequest -Uri "http://localhost:8083/tasks" `
>> -Method POST `
>> -Headers @{ "Content-Type" = "application/json" } `
>> -Body '{ "title": "first task" }'

```
PS C:\Users\javva\tinytasks> Invoke-WebRequest -Uri "http://localhost:8083/tasks" `
>>     -Method POST `
>>     -Headers @{ "Content-Type" = "application/json" } `
>>     -Body '{ "title": "first task" }'
>>


StatusCode        : 201
StatusDescription : CREATED
Content           : {"done":false,"id":1,"title":"first task"}

RawContent        : HTTP/1.1 201 CREATED
                    Connection: close
                    Content-Length: 43
                    Content-Type: application/json
                    Date: Mon, 17 Nov 2025 02:30:25 GMT
                    Server: gunicorn

                    {"done":false,"id":1,"title":"first task"}

Forms             : {}
Headers           : {[Connection, close], [Content-Length, 43], [Content-Type, application/json], [Date, Mon, 17 Nov 2025 02:30:25 GMT]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 43
```

Command: after post: curl http://localhost:8083/tasks

```
PS C:\Users\javva\tinytasks> curl http://localhost:8083/tasks
>>


StatusCode        : 200
StatusDescription : OK
Content           : [{"created_at":"2025-11-17T02:30:25.819685","done":false,"id":1,"title":"first task"}]

RawContent        : HTTP/1.1 200 OK
                    Connection: close
                    Content-Length: 87
                    Content-Type: application/json
                    Date: Mon, 17 Nov 2025 02:30:44 GMT
                    Server: gunicorn

                    [{"created_at":"2025-11-17T02:30:25.819685","done":fals...
Forms             : {}
Headers           : {[Connection, close], [Content-Length, 87], [Content-Type, application/json], [Date, Mon, 17 Nov 2025 02:30:44 GMT]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 87
```

Part 2

1. Non-root container user (user: "1000:1000")

Running the app as a non-root user helps protect the container because even if an attacker exploits a vulnerability in the app, they won't automatically gain root-level privileges inside the container. With reduced permissions, the attacker's actions are limited — they cannot modify critical system files, install system-level tools, or break out of the container as easily. This significantly lowers the overall impact of a potential compromise and strengthens container security.

2. Read-only filesystem (read_only: true)

Setting the container's filesystem to read-only means the app can run, but nothing inside the container can be changed. This protects the app because even if malicious code gets in, it cannot edit files, replace the app, or make harmful changes.
Only special folders you allow (like /tmp) can be written to. This keeps the container safe and stable.

3. Drop all Linux capabilities (cap_drop: ALL), only add back needed ones

Removing default capabilities limits what the container is allowed to do at the operating-system level. This reduces the attack surface, because even if an attacker gets inside the container, they can't perform actions that require higher privileges, making privilege escalation much harder.

4. Enable no-new-privileges: true

This ensures that the process and any subprocesses cannot gain extra privileges after they start, which helps block many common privilege-escalation attacks.

5. Healthchecks for both services

Automated container health checks ensure that only healthy application and database containers stay active, helping quickly detect issues and recover from faults or potential attacks.

Part 3

Issue 1: DB password mismatch

```
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ curl http://localhost:8083/tasks
<!doctype html>
<html lang=en>
<title>500 Internal Server Error</title>
<h1>Internal Server Error</h1>
<p>The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.</p>
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ docker compose logs -f api
WARN[0000] /home/hunterb978/tinytask/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
api-1  | [2025-11-24 04:49:09 +0000] [1] [INFO] Starting gunicorn 22.0.0
api-1  | [2025-11-24 04:49:09 +0000] [1] [INFO] Listening at: http://0.0.0.0:8080 (1)
api-1  | [2025-11-24 04:49:09 +0000] [1] [INFO] Using worker: sync
api-1  | [2025-11-24 04:49:09 +0000] [7] [INFO] Booting worker with pid: 7
api-1  | [2025-11-24 04:51:04,219] ERROR in app: Exception on /tasks [GET]
api-1  | Traceback (most recent call last):
api-1  |   File "/usr/local/lib/python3.12/site-packages/flask/app.py", line 1473, in wsgi_app
api-1  |     response = self.full_dispatch_request()
api-1  |                ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
api-1  |   File "/usr/local/lib/python3.12/site-packages/flask/app.py", line 882, in full_dispatch_request
api-1  |     rv = self.handle_user_exception(e)
api-1  |          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
api-1  |   File "/usr/local/lib/python3.12/site-packages/flask/app.py", line 880, in full_dispatch_request
api-1  |     rv = self.dispatch_request()
api-1  |          ^^^^^^^^^^^^^^^^^^^^^^^
api-1  |   File "/usr/local/lib/python3.12/site-packages/flask/app.py", line 865, in dispatch_request
api-1  |     return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)  # type: ignore[no-any-return]
api-1  |            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
api-1  |   File "/app/app.py", line 54, in list_tasks
api-1  |     with get_conn() as conn:
api-1  |          ^^^^^^^^^
api-1  |   File "/app/app.py", line 17, in get_conn
api-1  |     return psycopg2.connect(
api-1  |            ^^^^^^^^^^^^^^^^^
api-1  |   File "/usr/local/lib/python3.12/site-packages/psycopg2/__init__.py", line 122, in connect
api-1  |     conn = _connect(dsn, connection_factory=connection_factory, **kwasync)
api-1  |            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
api-1  | psycopg2.OperationalError: connection to server at "db" (172.19.0.2), port 5432 failed: FATAL:  password authentication failed for user "appuser"
api-1  |
^C
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ docker compose down -v
WARN[0000] /home/hunterb978/tinytask/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 4/4
 ✔ Container tinytask-api-1  Removed                 1.1s
 ✔ Container tinytask-db-1   Removed                 0.5s
 ✔ Network tinytask_default  Removed                 0.2s
 ✔ Volume tinytask_pgdata    Removed                 0.4s
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ docker compose up -d
WARN[0000] /home/hunterb978/tinytask/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 4/4
 ✔ Network tinytask_default  Created                 0.1s
 ✔ Volume "tinytask_pgdata"  Created                 0.0s
 ✔ Container tinytask-db-1   Healthy                12.1s
 ✔ Container tinytask-api-1  Started                12.7s
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$
```

Fix was to update password back to changeme123 within .env file

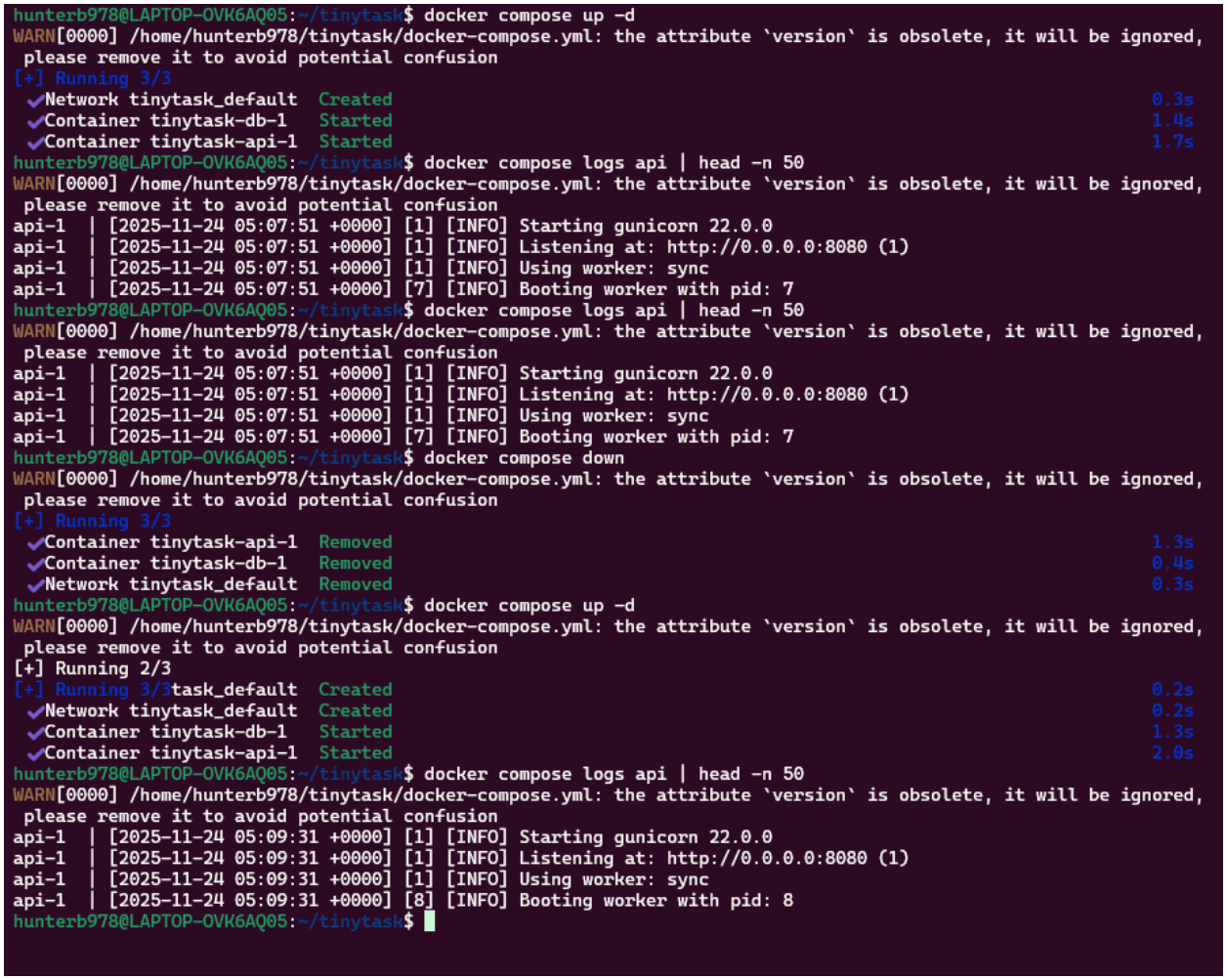
Issue 2: Port already in use

```
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ python3 -m http.server 8083 &
[1] 183779
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ Serving HTTP on 0.0.0.0 port 8083 (http://0.0.0.0:8083/) ...

hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ docker compose up -d
WARN[0000] /home/hunterb978/tinytask/docker-compose.yml: the attribute `version` is obsolete, it will be ignored,
 please remove it to avoid potential confusion
[+] Running 2/3
 ✔ Network tinytask_default  Created                                                                         0.2s
 ✔ Container tinytask-db-1   Healthy                                                                         7.4s
 ⠦ Container tinytask-api-1  Starting                                                                        8.2s
Error response from daemon: failed to set up container networking: driver failed programming external connectivit
y on endpoint tinytask-api-1 (07600e5022b7c1e748ebe4ea5310a1b35fca577ac5354da6c3dcc5cdfcfd60ae): failed to bind h
ost port for 0.0.0.0:8083:172.19.0.3:8080/tcp: address already in use
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ sudo lsof -i :8083 -sTCP:LISTEN
COMMAND    PID       USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
python3 183779 hunterb978    3u  IPv4 763625      0t0  TCP *:8083 (LISTEN)
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ kill <182779>
-bash: syntax error near unexpected token `182779'
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ kill <183779>
-bash: syntax error near unexpected token `183779'
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ kill 183779
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ docker compose up -d
WARN[0000] /home/hunterb978/tinytask/docker-compose.yml: the attribute `version` is obsolete, it will be ignored,
 please remove it to avoid potential confusion
[+] Running 2/2
 ✔ Container tinytask-db-1   Healthy                                                                         0.5s
 ✔ Container tinytask-api-1  Started                                                                         0.9s
[1]+  Terminated              python3 -m http.server 8083
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$ curl http://localhost:8083/health
{"ok":true}
hunterb978@LAPTOP-OVK6AQ05:~/tinytask$
```

Fix: kill <pid> and docker compose up -d again

Issue 3: Startup Race



Fix: could not cause error as only may cause race error but changing back

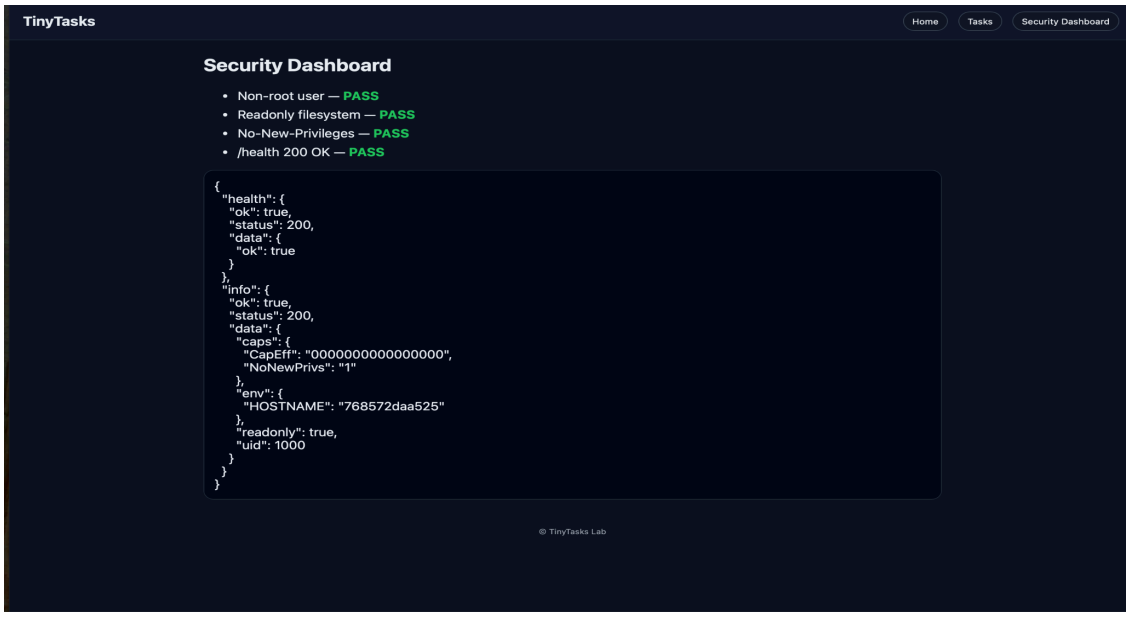
db:

condition: service_healthy

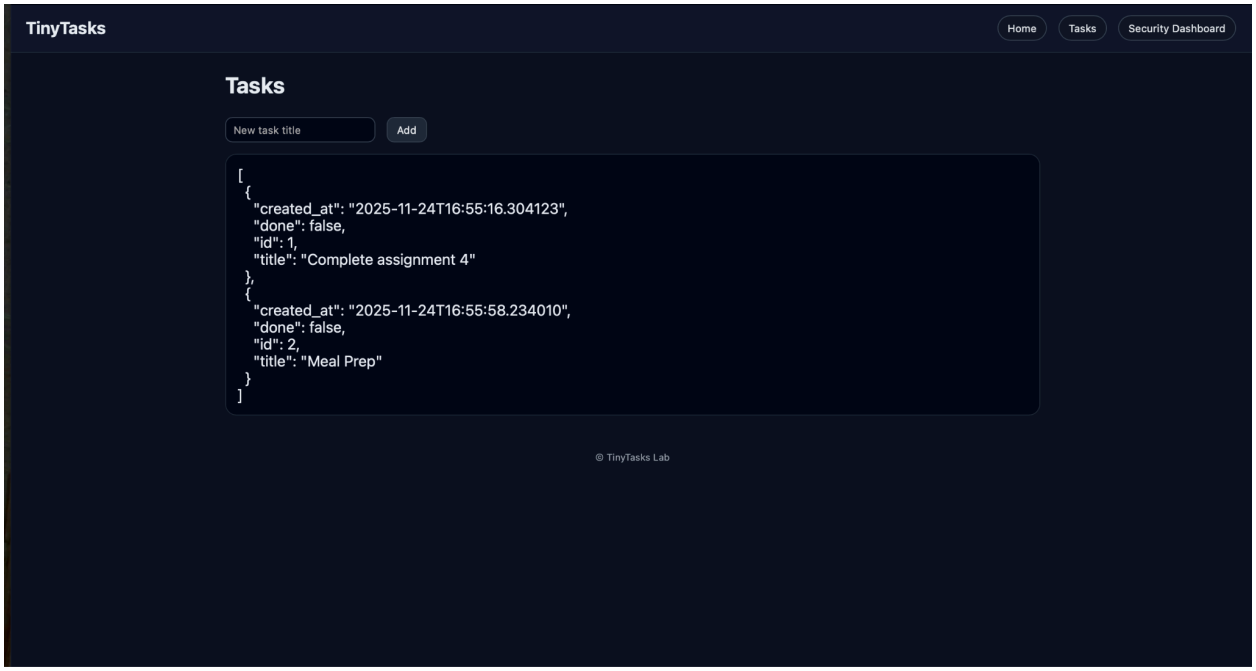
Fixes error race error if caused.

Front end security dashboard:

UI security dashboard-



UI tasks page-



Smoke test-

```
sadiaalam@macbookpro tinytasks % curl http://localhost:8083/health
curl http://localhost:8083/tasks
curl -X POST http://localhost:8083/tasks \
-H "Content-Type: application/json" \
-d '{"title":"first task"}'
curl http://localhost:8083/tasks
{"ok":true}
[{"created_at":"2025-11-24T16:55:16.304123","done":false,"id":1,"title":"Complet
e assignment 4"},{"created_at":"2025-11-24T16:55:58.234010","done":false,"id":2,
"title":"Meal Prep"}]
{"done":false,"id":3,"title":"first task"}
[{"created_at":"2025-11-24T16:55:16.304123","done":false,"id":1,"title":"Complet
e assignment 4"},{"created_at":"2025-11-24T16:55:58.234010","done":false,"id":2,
"title":"Meal Prep"},{"created_at":"2025-11-24T18:01:34.382638","done":false,"id
":3,"title":"first task"}]
sadiaalam@macbookpro tinytasks %
```

How hardening control reduces risk and production lesson we learned:

Hardening makes it difficult for the users to compromise a system. When an app is running as a non-root user, attackers can’t get full system control. It ensures the principle of least privilege (users have bare minimum control to complete their functionalities). This limits damage if any exploitation occurs. The health checks for failures so that the system can restart any unhealthy containers. In short, hardening control protects systems by reducing blast radius. We learnt that we can create a secure system from the start by preventing damage before any attack by limiting permissions and safe design.