

Analyze the application - which Microservice best practices does it not follow?

The following aspects need to be improved in the application:

- The logs shouldn't be written to a file, but to the container output.
- It should be stateless, so that:
 - it can easily be restarted without loss of data,
 - it is easy to spawn multiple instances of the application.
- Requirements installation should be moved from runtime to build time.
- App should be able to be executed both during development, with debugging enabled, and in production, with debugging disabled.
- The application should be built in such a way that the database can easily be replaced (development with production instance).

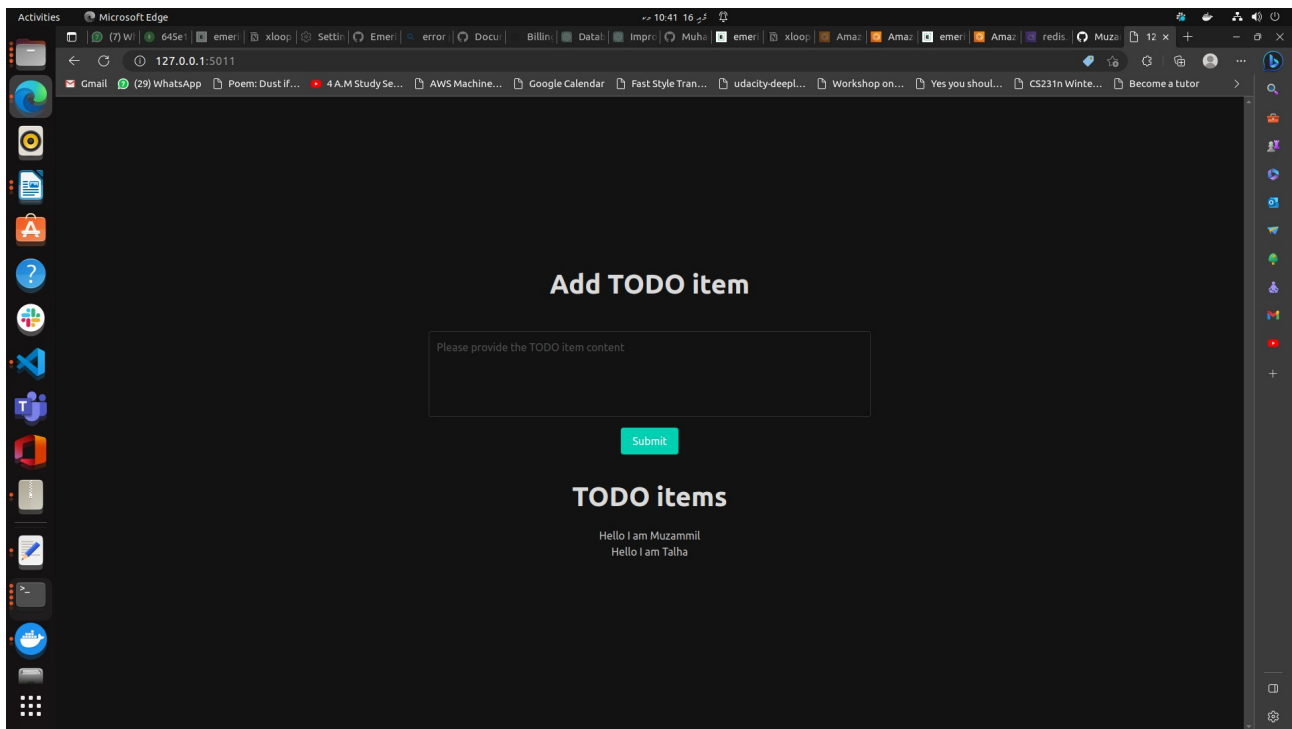
The logs output

```
muzammil@all-MS-7D35: ~/Documents/Muzammil-Mehmood-2303.028... x muzammil@all-MS-7D35: ~/Documents/Muzammil-Mehmood-2303.028... x v
(base) muzammil@all-MS-7D35:~/Documents/Muzammil-Mehmood-2303.028.KHI.DEG-/Week_4/4.4 Best Practices$ docker-compose logs
Attaching to 44bestpractices_app_2, 44bestpractices_app_1
app_1 | [2023-05-16 05:33:00 +0000] [1] [INFO] Starting gunicorn 20.1.0
app_1 | [2023-05-16 05:33:00 +0000] [1] [INFO] Listening at: http://0.0.0.0:5000 (1)
app_1 | [2023-05-16 05:33:00 +0000] [1] [INFO] Using worker: sync
app_1 | [2023-05-16 05:33:00 +0000] [8] [INFO] Booting worker with pid: 8
app_2 | [2023-05-16 05:33:00 +0000] [1] [INFO] Starting gunicorn 20.1.0
app_2 | [2023-05-16 05:33:00 +0000] [1] [INFO] Listening at: http://0.0.0.0:5000 (1)
app_2 | [2023-05-16 05:33:00 +0000] [1] [INFO] Using worker: sync
app_2 | [2023-05-16 05:33:00 +0000] [8] [INFO] Booting worker with pid: 8
app_2 | [2023-05-16 05:33:28,648] INFO in main: Added new item: Hello I am Muzammil
app_2 | 05:33:28,648 main INFO Added new item: Hello I am Muzammil
app_2 | 05:33:35,747 main INFO Added new item: Hello I am Talha
app_2 | [2023-05-16 05:33:35,747] INFO in main: Added new item: Hello I am Talha
(base) muzammil@all-MS-7D35:~/Documents/Muzammil-Mehmood-2303.028.KHI.DEG-/Week_4/4.4 Best Practices$
```

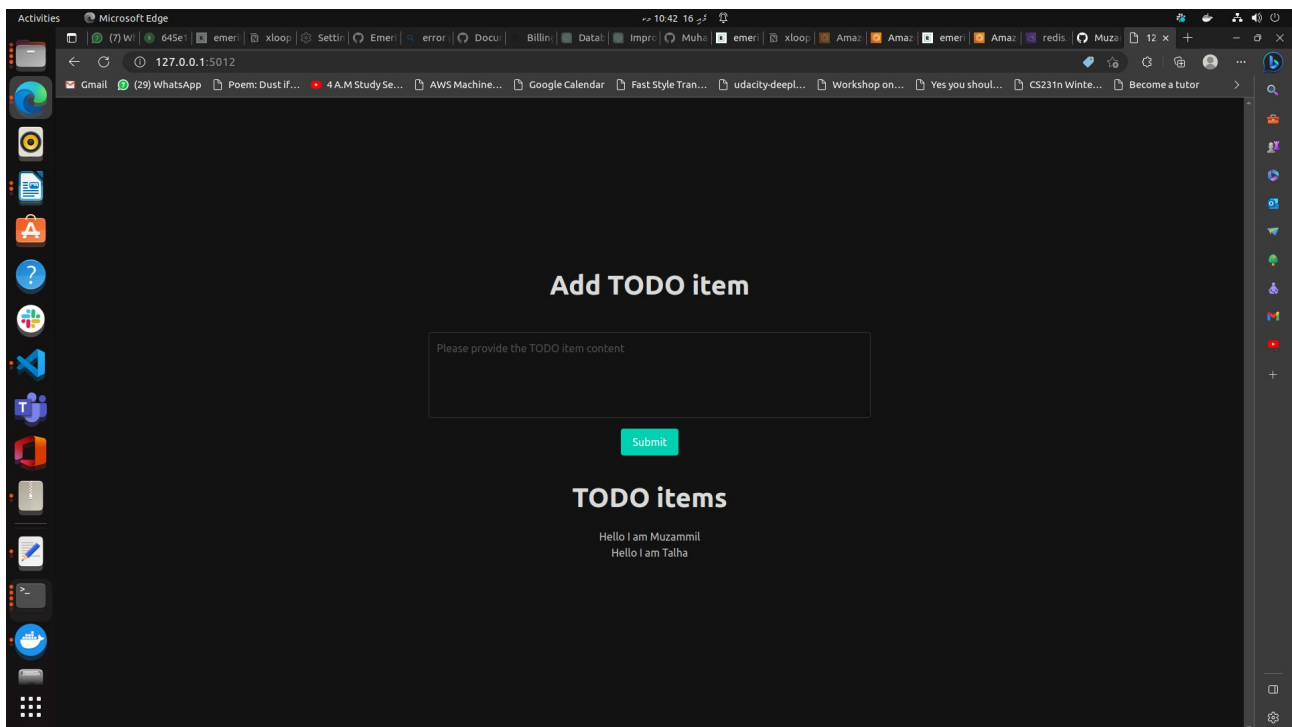
It is stateless as after stopping the docker container by using docker-compose down

```
muzammil@all-MS-7D35: ~/Documents/Muzammil-Mehmood-2303.028... x muzammil@all-MS-7D35: ~/Documents/Muzammil-Mehmood-2303.028... x v
app_2 | [2023-05-16 05:33:00 +0000] [8] [INFO] Booting worker with pid: 8
app_2 | 05:33:28,648 main INFO Added new item: Hello I am Muzammil
app_2 | [2023-05-16 05:33:28,648] INFO in main: Added new item: Hello I am Muzammil
app_2 | 05:33:35,747 main INFO Added new item: Hello I am Talha
app_2 | [2023-05-16 05:33:35,747] INFO in main: Added new item: Hello I am Talha
^CGracefully stopping... (press Ctrl+C again to force)
Stopping 44bestpractices_app_2 ... done
Stopping 44bestpractices_app_1 ... done
(base) muzammil@all-MS-7D35:~/Documents/Muzammil-Mehmood-2303.028.KHI.DEG-/Week_4/4.4 Best Practices$ docker-compose down
Removing 44bestpractices_app_2 ... done
Removing 44bestpractices_app_1 ... done
Removing network 44bestpractices_default
(base) muzammil@all-MS-7D35:~/Documents/Muzammil-Mehmood-2303.028.KHI.DEG-/Week_4/4.4 Best Practices$ docker-compose up
Creating network "44bestpractices_default" with the default driver
WARNING: The "app" service specifies a port on the host. If multiple containers for this service are created on a single host,
the port will clash.
```

It is still retaining the data



It is replicating on multiple ports,



Other things can be verify by seeing the code