**ASSIGNMENT 4.4**

**on**

**Microservices Best Practice**

**Submitted by:**

**Haseebullah Shaikh (2303.KHI.DEG.015)**

**and**

**Faiza Gulzar Ahmed (2303.khi.deg.001)**

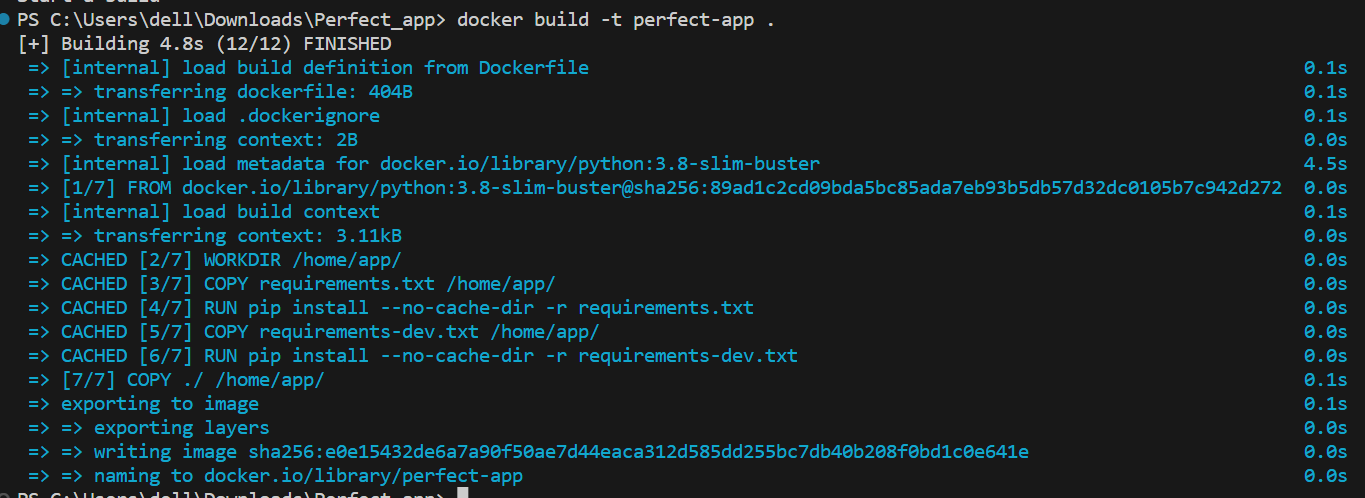
**Dated:** 14th May 2023

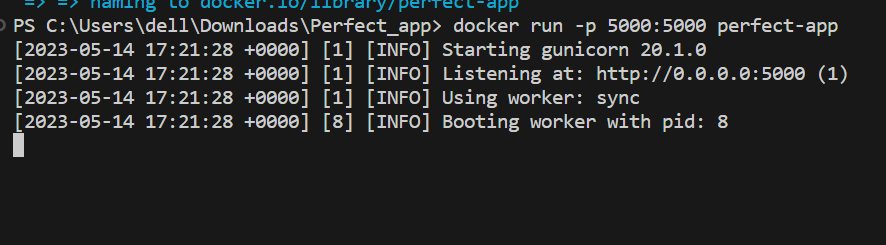
**Task:**  **Browse to: tasks/4\_microservices\_development/day\_4\_best\_practices/**  
**app\_that\_doesnt\_follow\_best\_practices/**  
**analyze the application - which microservice best practices it doesn’t follow?**  
**Think about what needs to be improved first. Have a look at the areas\_for\_improvement.txt file for hints. Improve the application**

**Solution:**

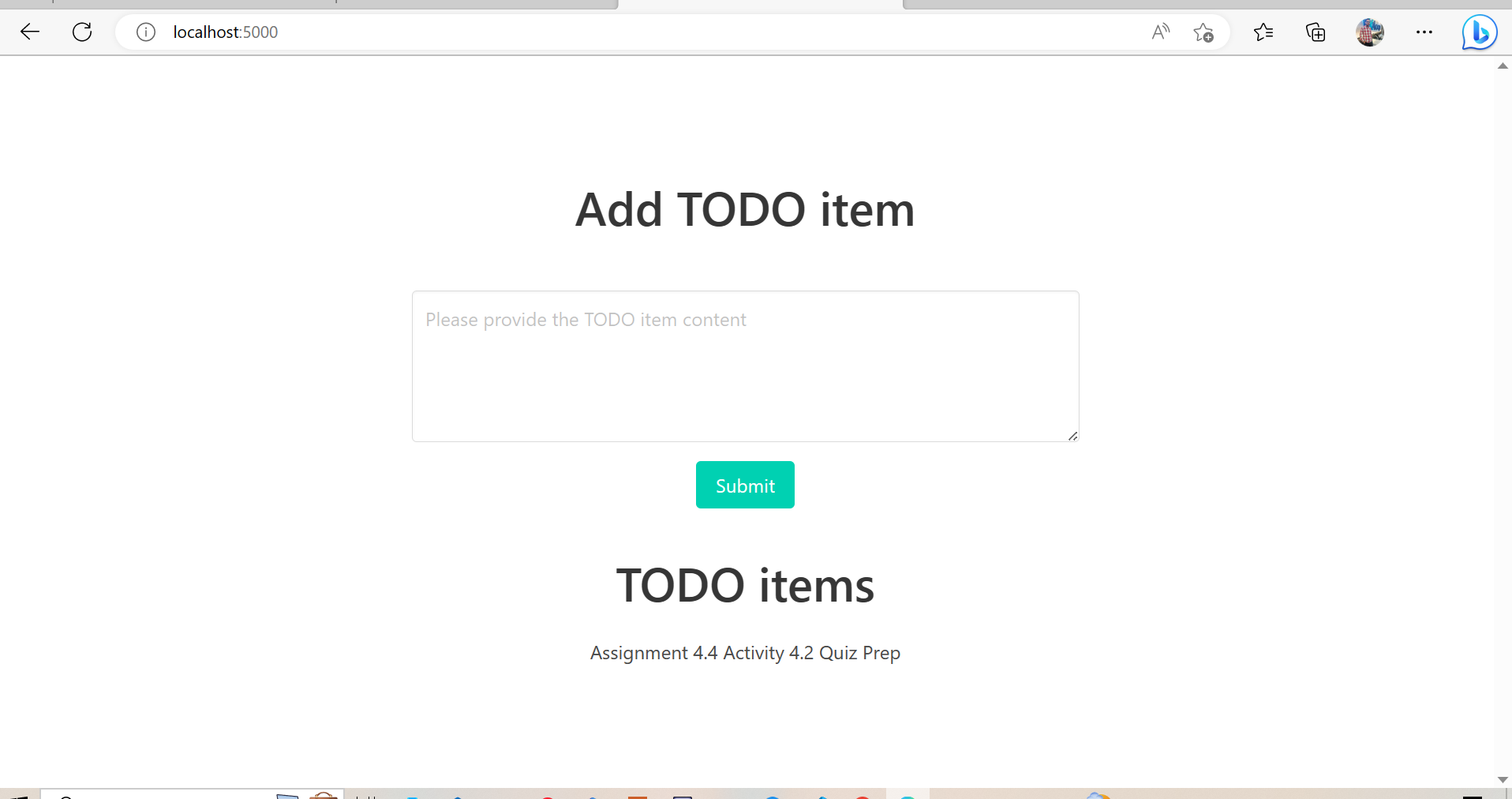
The app has been updated by considering the area of improvements, app and all related files are uploaded on you git repository.

**Commands to run app**





**App is accesible at port: 5000**

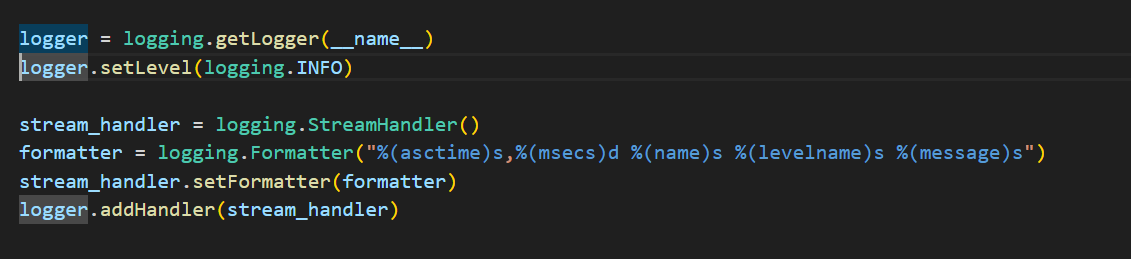


**Below is the detailed description of each and every thing that has been updated.**

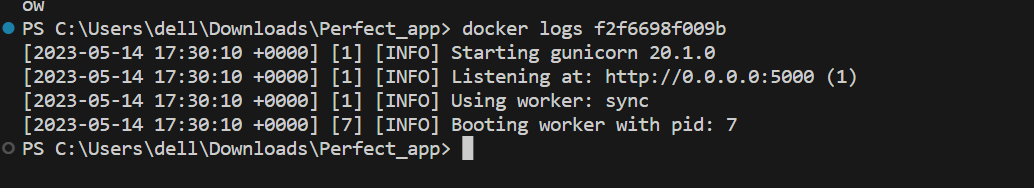
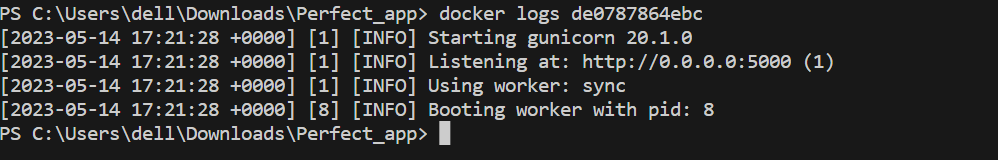
**Given areas of improvements**

1. **The logs shouldn’t written to a file, but to the container output.**

We have used built-in logging module of python to write the logs to the container output. Every log that will be generated will be written to the console standard output.



**Logs of two created container for running app and statless verification.**



1. **It should be stateless, so that:**

**- it can easily be restarted without loss of data,**

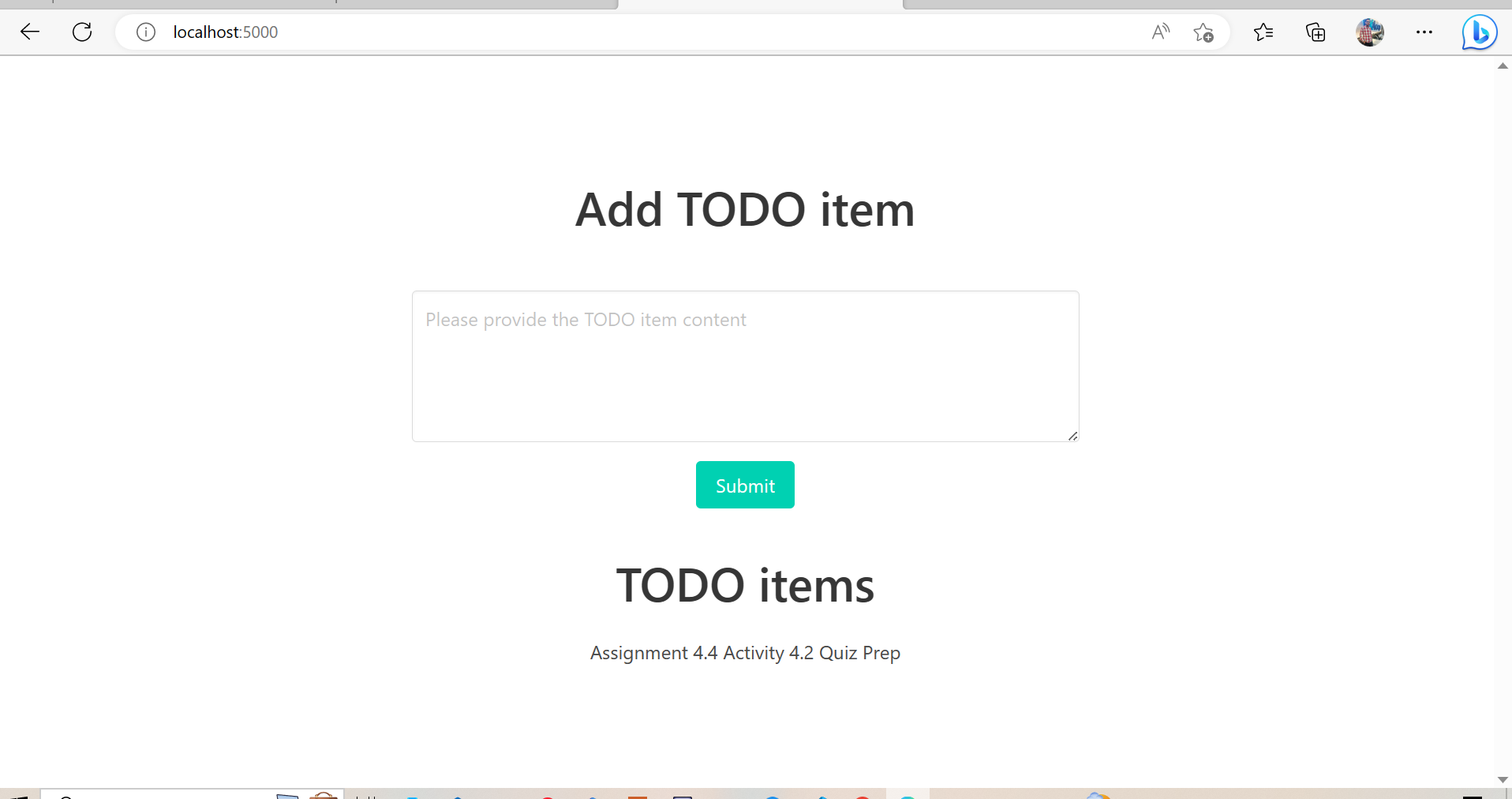
**- it is easy to spawn multiple instances of the application.**

We have made the app stateless in a way:

* The SQLite database is implemented to store the data, instead of saving the data locally.
* The code is defined within the module scope as it’s not relying upon any external files.
* Developed database can accessed by the multiple instances of the application.
* There will be no any data loss, app can be restarted without data loss.

**Stateless Verification**

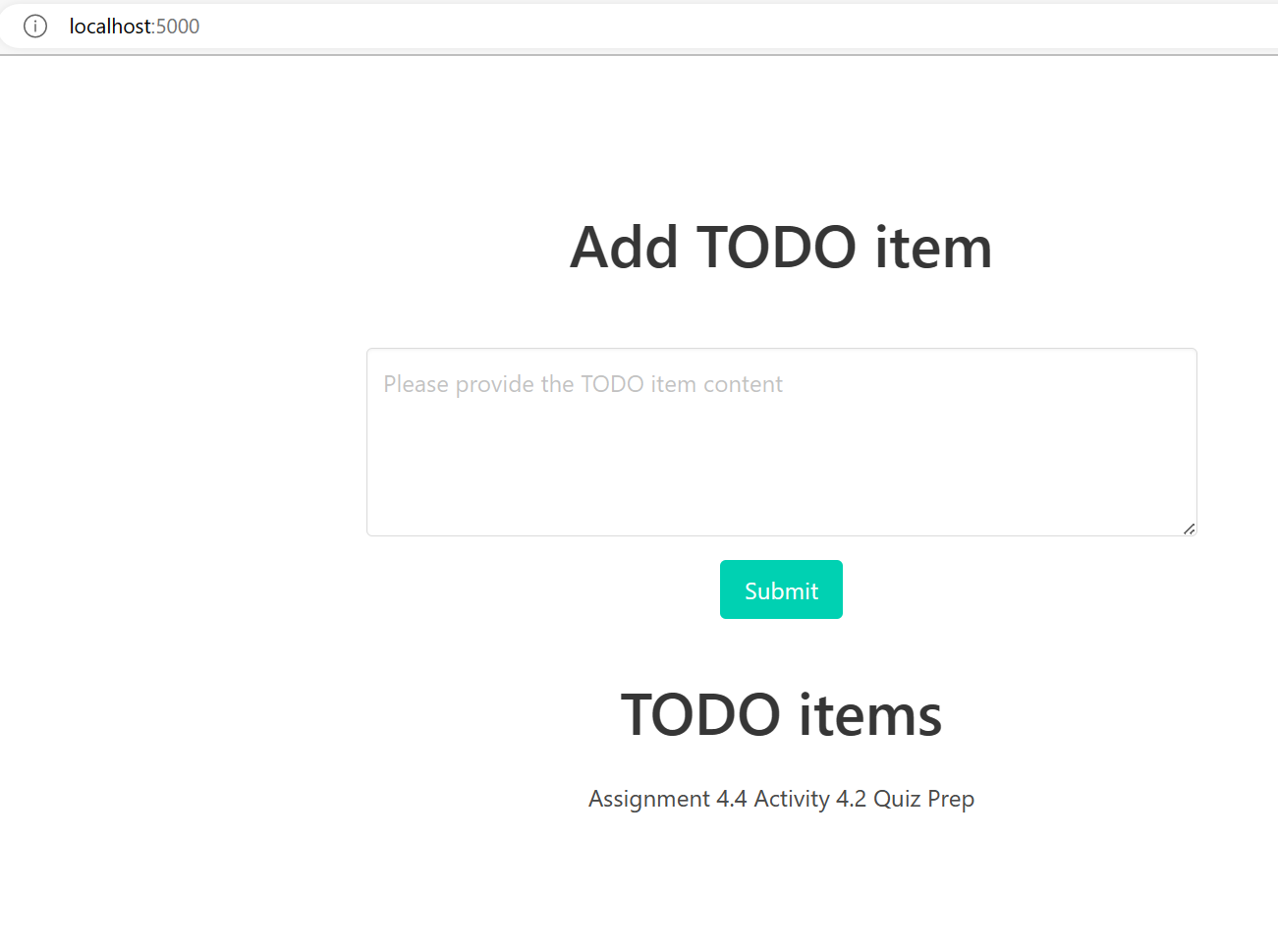
App running on first container, items created and submitted.



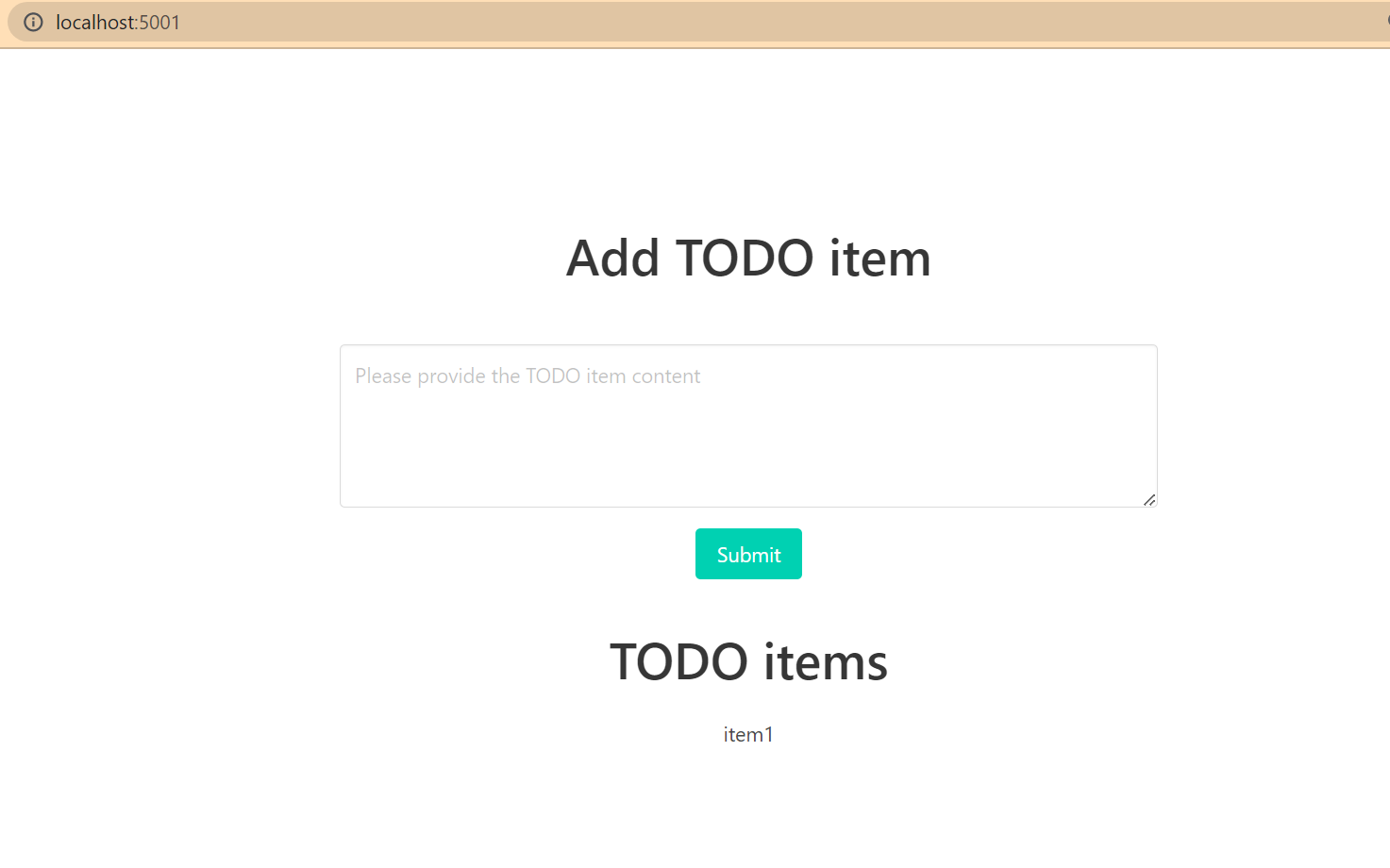
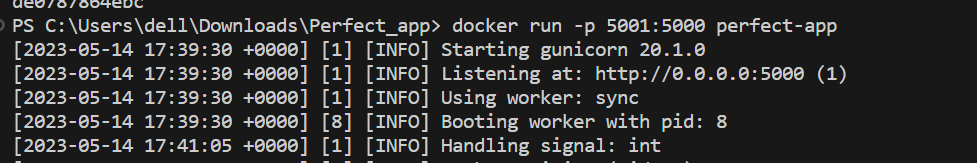
**Container stopped**

**Re Running app again**

Items are restored, no any data loss

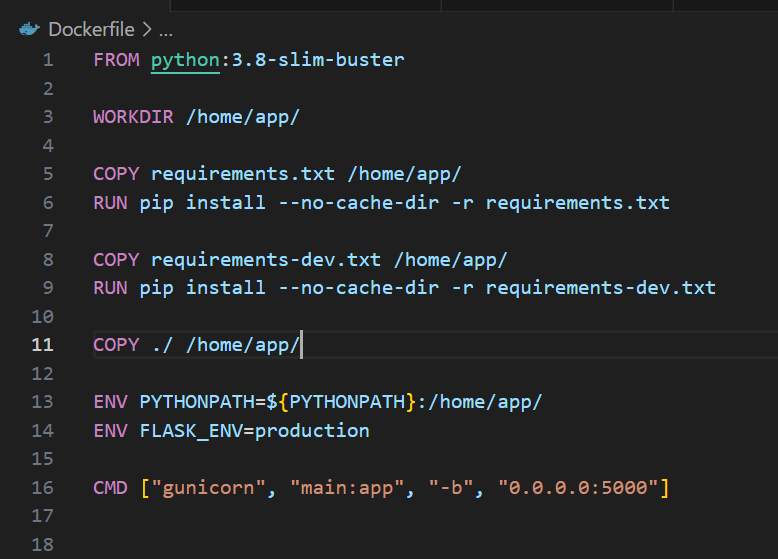


**Verfying app is accesible at multiple instances and can share the same database.**



1. **Requirements installation should be moved from runtime to build time.**

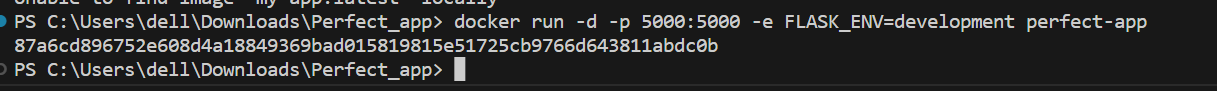
We have updated docker file, moved the requirements installation from run time to the build time.

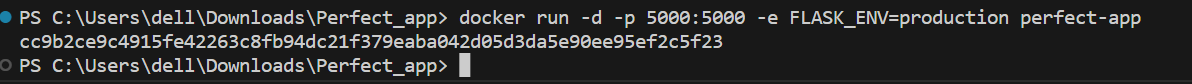


1. **App should be able to be executed both during development, with debugging enabled, and in production, with debugging disabled.**

We have used environment variable “FLASK\_ENV” which made it happen to execute app in development with debugging enabled while in production debigging disabled.

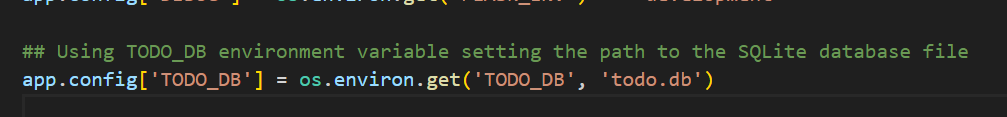
**Verfying the debugging on both**

**App is accessible at port 5000** 



1. **The application should be built in such a way that the database can easily be replaced (development with production instance).**

We have achieved it by using enviroment variable “TODO\_DB”, by which we have set the path to the SQLite database file .



* ***The End* 😊**