**Summary:**

This rate-limiting HTTP service uses FastAPI in Python. Its primary objective is to restrict the number of requests a user can make to a specific endpoint within a defined time window. The rate limiter uses the client's IP address to track and control the number of requests. If the user exceeds the configured limit, the service responds with a 429 HTTP status code, indicating too many requests. The request log is saved persistently in a JSON file (state.json) so that the limiter can recover after service restarts or crashes.

The application is built with Python’s FastAPI framework and served using Uvicorn. It is fully containerized using Docker and deployed to a local Kubernetes cluster via Minikube. The service exposes a Swagger UI at /docs for API interaction and testing.

**Result:**

Code run in PyCharm

A screenshot of a computer program

AI-generated content may be incorrect.

Rate limit app running in Docker

A screenshot of a computer

AI-generated content may be incorrect.

Deploy service from Minikube

A screenshot of a computer

AI-generated content may be incorrect.

Go to http://localhost:8000 → See JSON message  
A screenshot of a computer

AI-generated content may be incorrect.

Go to <http://localhost:8000/limited> → See JSON message if request Status Code 200, so request allowed

A black rectangular object with a black border

AI-generated content may be incorrect.

Go to http://localhost:8000/docs → See FastAPI UI to test endpoints.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Result show if Status Code 200 “Request allowed and Status Code 429 “Too many requests”

A screenshot of a computer

AI-generated content may be incorrect.