# List concerts

curl http://localhost:5000/concerts

# Buy tickets for Imagine Dragons

curl -X POST http://localhost:5000/buy -H "Content-Type: application/json" -d "{\"concert\_id\": 1, \"quantity\": 2}"

# Buy tickets for Coldplay

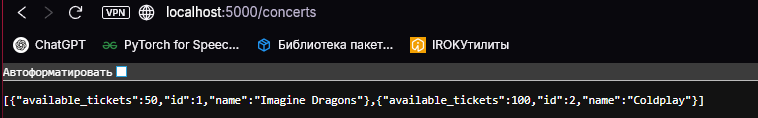
curl -X POST http://localhost:5000/buy -H "Content-Type: application/json" -d "{\"concert\_id\": 2, \"quantity\": 3}"

SRENURIFINAL:

TASK 1

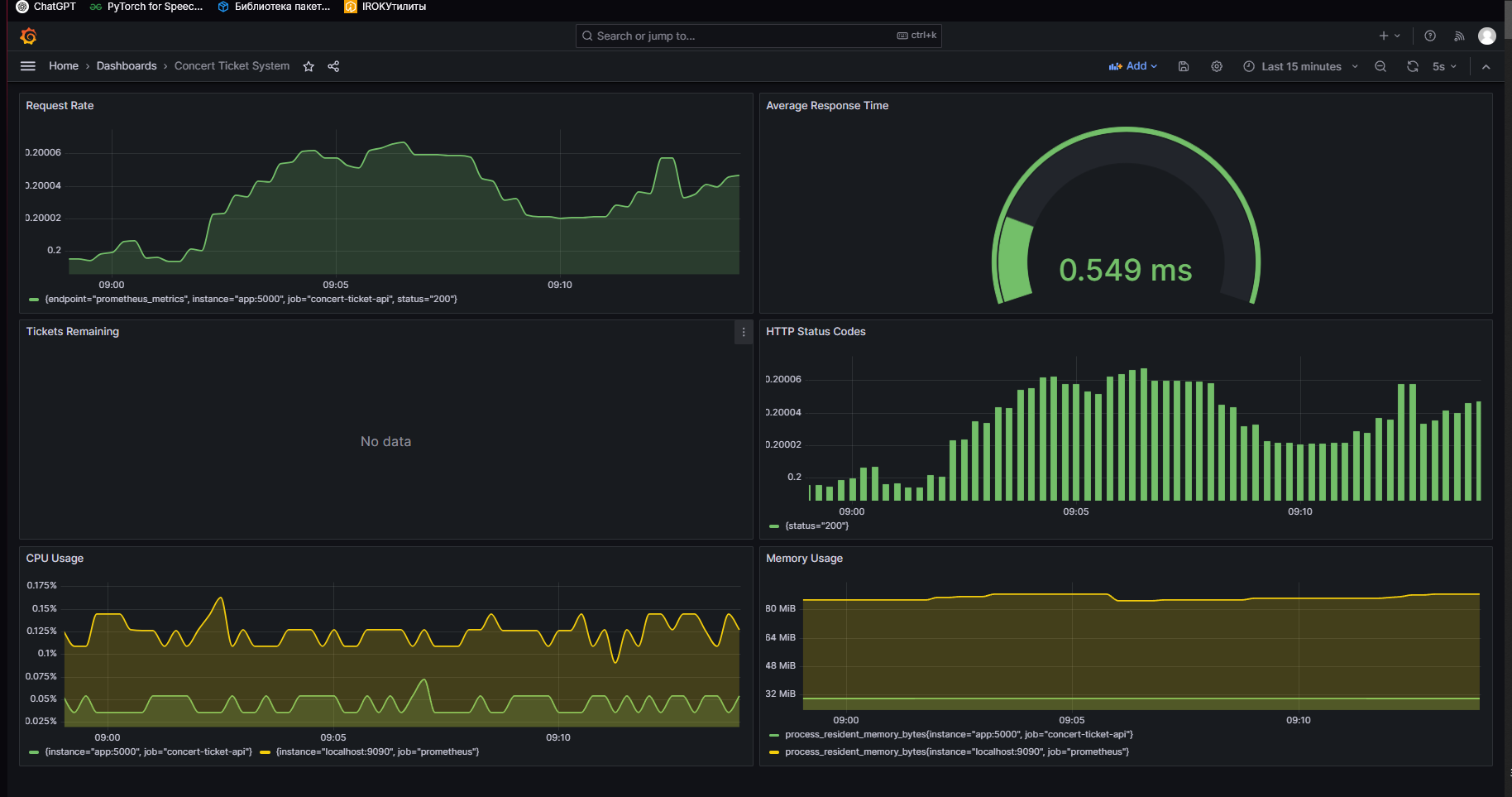
1) created a concert ticketing web application

Backend at http://localhost:5000



2) Monitoring & Alerting:

Set up Prometheus and Grafana



Defined key metrics:

HTTP Request Rate

Average Response Time

CPU Usage

Memory Usage

**Task 2 - Infrastructure as Code Collaboration**

Application & Infrastructure Design:

Multi-container architecture using Docker Compose

Services: Frontend, Backend, Database, Prometheus, Grafana

High availability considerations through container orchestration

IaC Implementation:

Used Docker Compose for infrastructure definition

Services are containerized and easily deployable

Used terraform

Integration & Deployment:

All services are integrated and work together

Environment variables and networking properly configured

Volumes for data persistence

Documentation:

Used best practices

Problems and challenges:  
had very big problems with terraform. Added permissions, even managed my own policy for bucket S3 and added it to my user, only after that I got from error 403 to successful terraform init

