
Архитектура распределённых приложений

Лабораторная работа №1

Выполнил: студент группы ИП-312
Дорогин Н. С.
Проверил: преподаватель Кафедры вычислительных систем
Челканова Т. В.

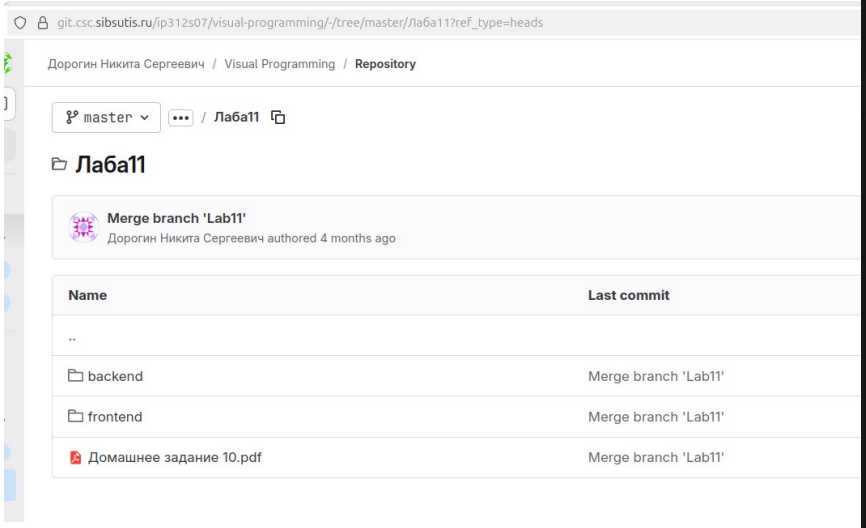
Задание:

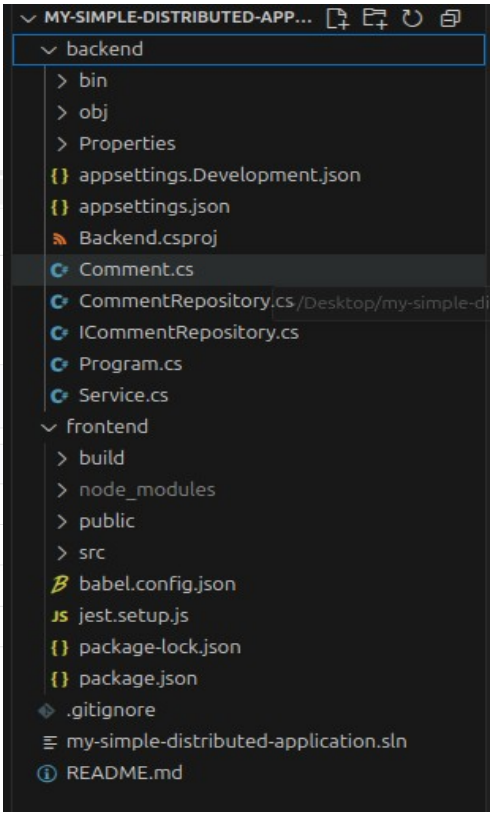
1. Получить доступ к Gitlab кафедры вычислительных систем.
2. Разработать приложение, реализующее простой API по протоколу http. Допускается взять уже существующее.
3. Приложение должно корректно обрабатывать сигналы остановки.
4. Написать для приложения Dockerfile для сборки образа контейнера.
5. Продемонстрировать работу приложения в контейнере. Для этого необходимо запустить контейнер и обеспечить доступ по сети.

Ход работы:

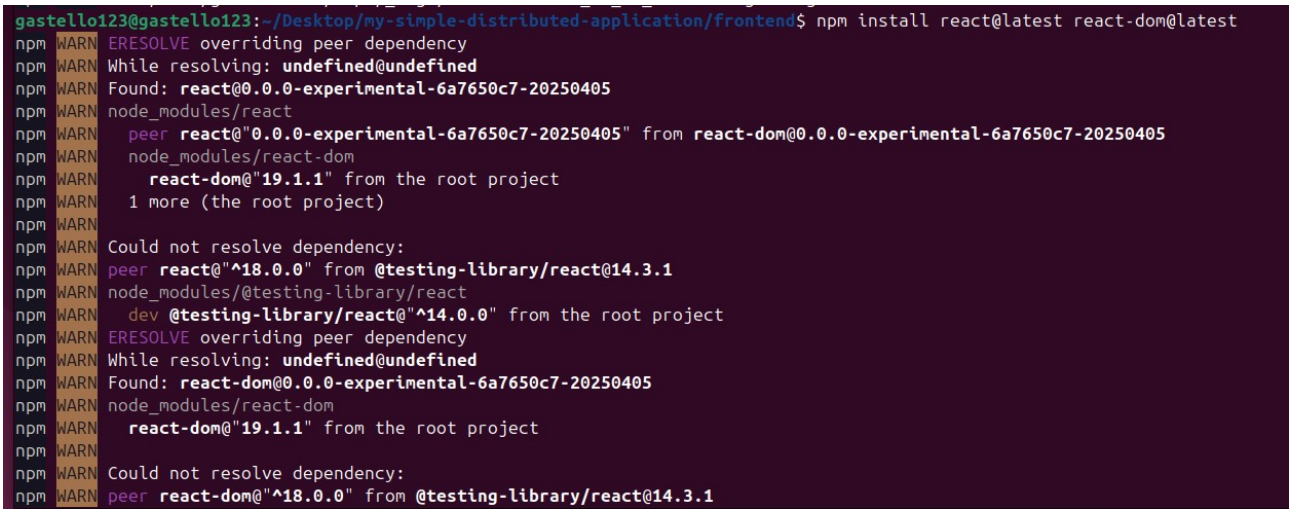
1, 2, 3) Уже имея доступ к Gitlab кафедры вычислительных систем с прошлых курсов, создаю там репозитой my-simple-distributed-application в котором и будет вестись работа в рамках этого курса.

В качестве реализации API по протоколу http возьму приложение с прошлого курса, реализующее таблицу комментариев.





Устанавливаем node.js для запуска React-приложения, реализующего фронтенд и ASP.NET Core для запуска бэкэнда (сервера).



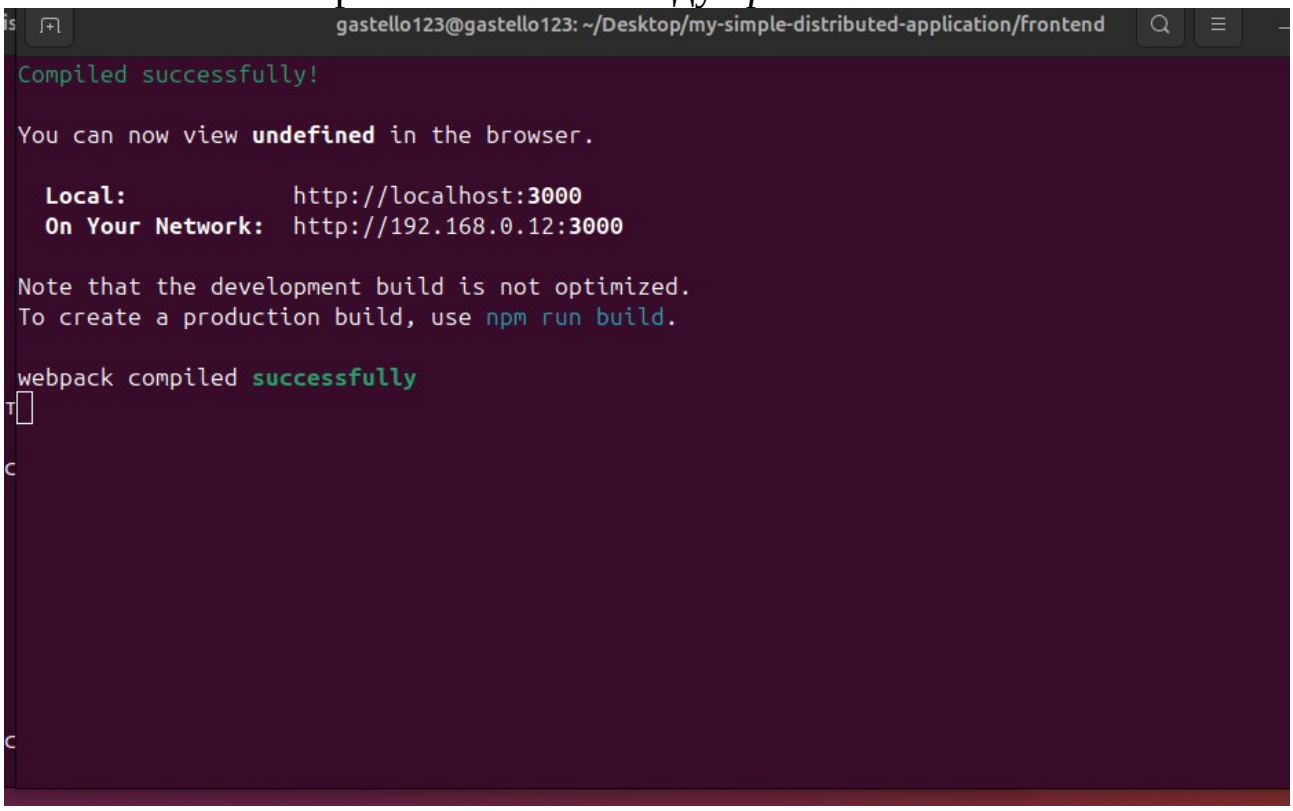
```
gastello123@gastello123:~/Desktop$ wget https://dot.net/v1/dotnet-install.sh -O dotnet-install.sh && chmod +x dotnet-install.sh && ./dotnet-install.sh --channel 9.0 && echo 'exp
ort DOTNET_ROOT=$HOME/.dotnet' >> ~/.bashrc && echo 'export PATH=$PATH:$HOME/.dotnet:$HOME/.dotnet/tools' >> ~/.bashrc && source ~/.bashrc

--2025-09-20 21:57:46--  https://dot.net/v1/dotnet-install.sh
Распознаётся dot.net (dot.net)… 20.236.44.162, 20.231.239.246, 20.70.246.20, ...
Подключение к dot.net (dot.net)|20.236.44.162|:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа… 301 Moved Permanently
Адрес: https://builds.dotnet.microsoft.com/dotnet/scripts/v1/dotnet-install.sh [переход]
--2025-09-20 21:57:47--  https://builds.dotnet.microsoft.com/dotnet/scripts/v1/dotnet-install.sh
Распознаётся builds.dotnet.microsoft.com (builds.dotnet.microsoft.com)… 23.196.236.91, 23.196.236.82, 2001:2030:0:17::50ef:8a92, ...
Подключение к builds.dotnet.microsoft.com (builds.dotnet.microsoft.com)|23.196.236.91|:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа… 200 OK
Длина: нет данных [application/octet-stream]
Сохранение в: ‘dotnet-install.sh’

dotnet-install.sh                           [ <=>                                                                                ]  62.15K  --.-KB/s    за
0.03s

2025-09-20 21:57:48 (2.14 MB/s) - ‘dotnet-install.sh’ сохранён [63644]

dotnet-install: Attempting to download using aka.ms link https://builds.dotnet.microsoft.com/dotnet/Sdk/9.0.305/dotnet-sdk-9.0.305-linux-x64.tar.gz
dotnet-install: Remote file https://builds.dotnet.microsoft.com/dotnet/Sdk/9.0.305/dotnet-sdk-9.0.305-linux-x64.tar.gz size is 218127088 bytes.
gastello123@gastello123:~/Desktop/my-simple-distributed-application/backend$ export DOTNET_ROOT=$HOME/.dotnet && export PATH=$HOME/.dotnet:$HOME/.dotnet/tools:$PATH

gastello123@gastello123:~/Desktop/my-simple-distributed-application/backend$ dotnet --version
9.0.305
gastello123@gastello123:~/Desktop/my-simple-distributed-application/backend$ echo 'export DOTNET_ROOT=$HOME/.dotnet' >> ~/.bashrc && echo 'export PATH=$HOME/.dotnet:$HOME/.dot
t/tools:$PATH' >> ~/.bashrc && source ~/.bashrc

gastello123@gastello123:~/Desktop/my-simple-distributed-application/backend$ dotnet run

Вас приветствует .NET 9.0!
```

Запускаем приложение:

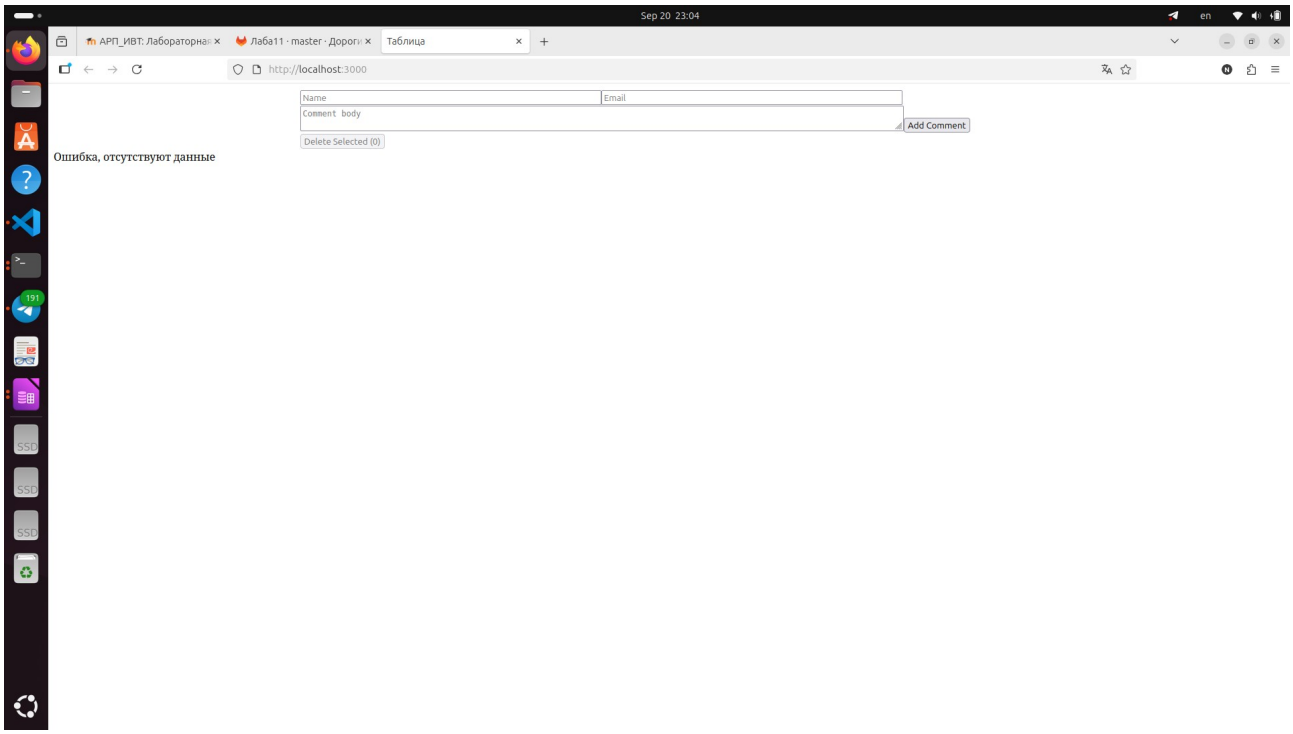
В папке backend прописываем команду *dotnet run*

```
gastello123@gastello123: ~/Desktop/my-simple-dis

---------------
Установлен сертификат разработки ASP.NET Core HTTPS.
Чтобы доверять сертификату, выполните команду "dotnet dev-certs https --trust"
Дополнительные сведения об HTTPS: https://aka.ms/dotnet-https

---------------
Как написать свое первое приложение: https://aka.ms/dotnet-hello-world
Узнать о новых возможностях: https://aka.ms/dotnet-whats-new
Просмотреть документацию: https://aka.ms/dotnet-docs
Сообщить о проблемах и найти исходный код на GitHub: https://github.com/dotnet/core
Для просмотра доступных команд введите команду "dotnet --help" или посетите следующую ст
--------------------------------------------------------------------------------
Используются параметры запуска из /home/gastello123/Desktop/my-simple-distributed-applic
Сборка…
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5139
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /home/gastello123/Desktop/my-simple-distributed-application/bac
```

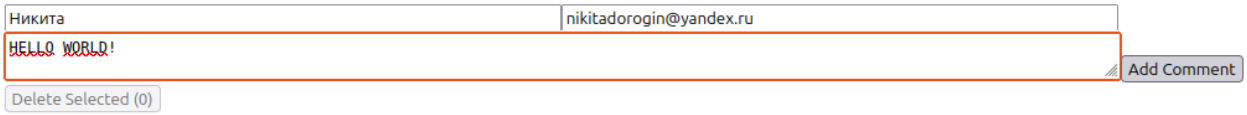
В папке frontend прописываем команду *npm start*



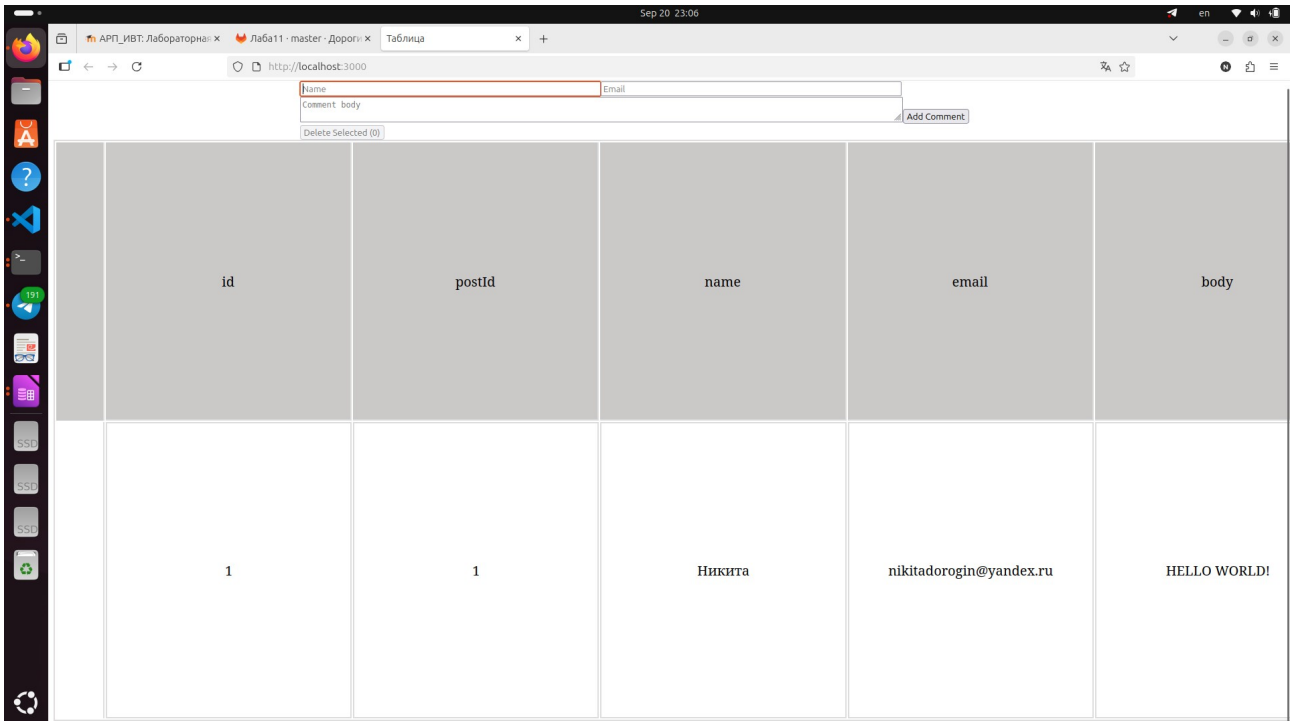
Приложение открывается:



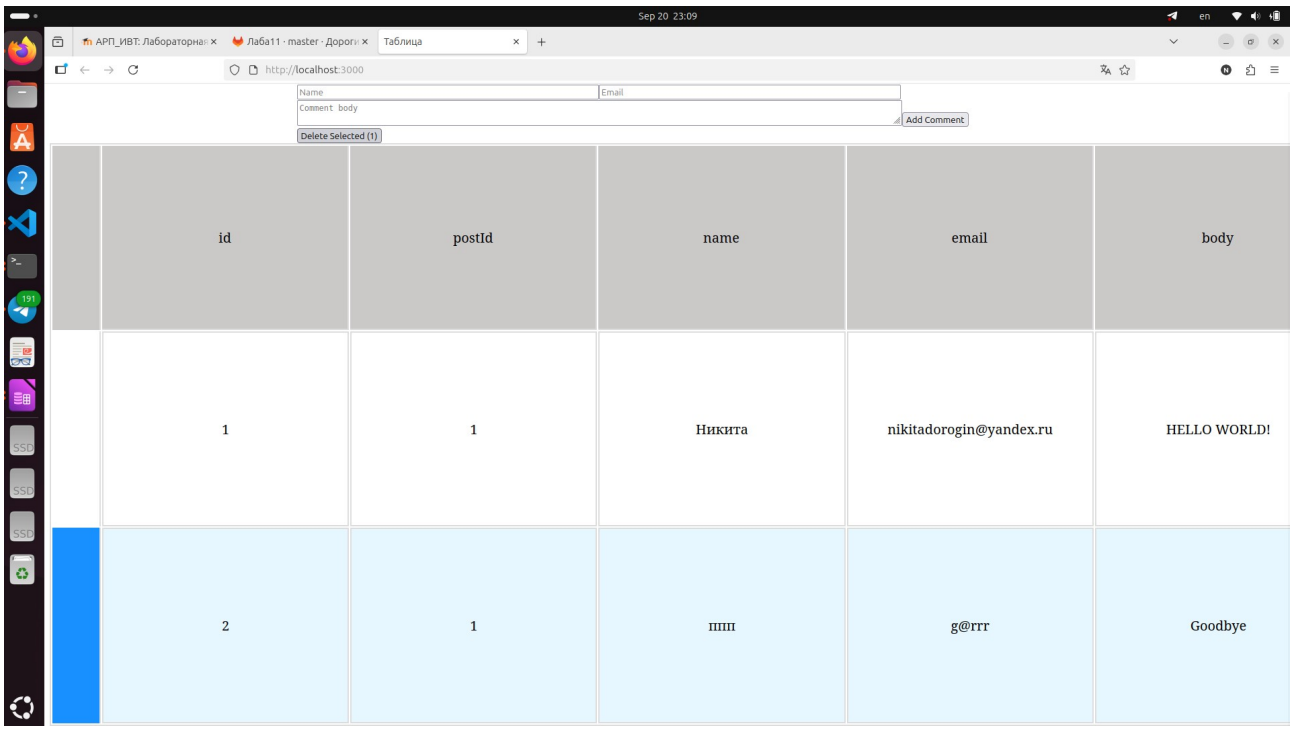
Пока что в таблицу ничего не занесено, поэтому выводится сообщение об отсутствии данных.



Вписав корректные данные и нажав AddComment мы отправим на сервер http запрос POST и таким образом добавим туда наши данные.



Нажав несколько раз на ячейку и изменив в ней значение мы отправим на сервер http запрос PATCH, изменив уже имеющиеся там данные.



Выбрав строку и нажав кнопку Delete Selected мы отправим на сервер http запрос DELETE и таким образом удалим оттуда выбранные данные.

При этом на протяжении всей работы приложения React-приложение посылает серверу http запрос GET, чтобы получать данные, которые оно выводит в таблице.

4) Dockerfile для приложения выглядит следующим образом:

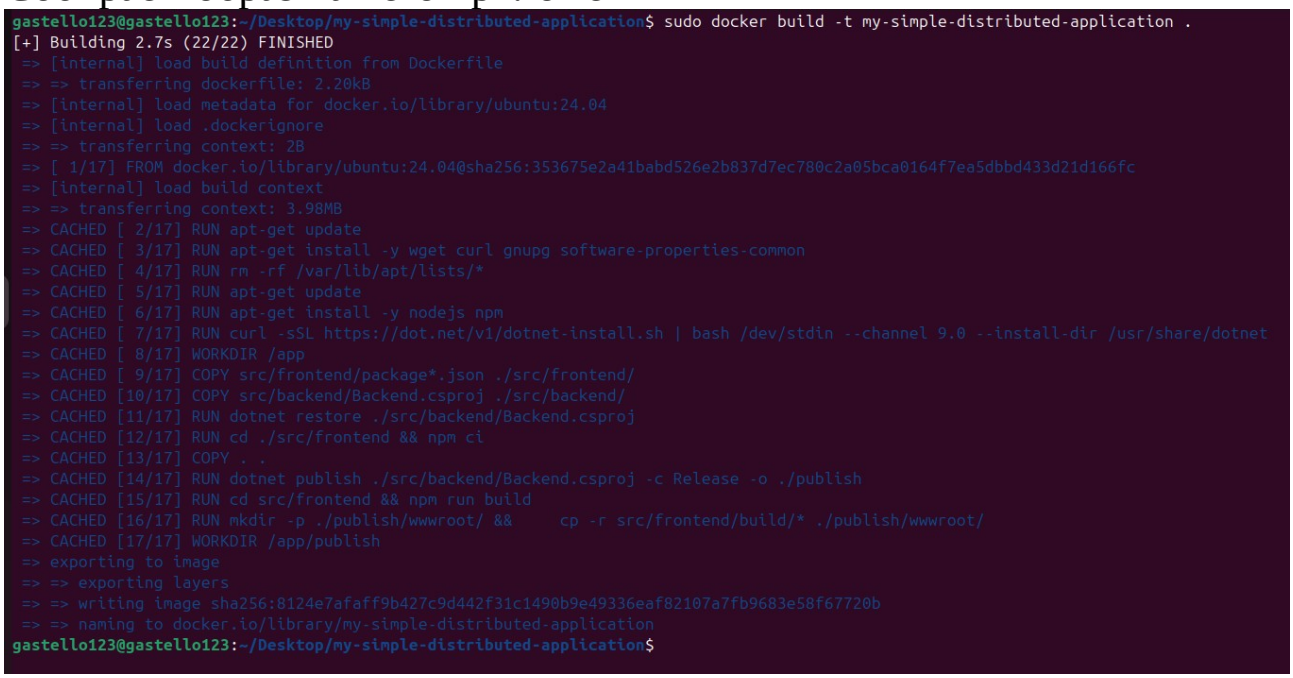
```
#my-dockerfile

# Указываем систему, с которой запускаем
FROM ubuntu:24.04


# Установка системных зависимостей
RUN apt-get update
RUN apt-get install -y wget curl gnupg software-properties-common
RUN rm -rf /var/lib/apt/lists/*
```

```
# Установка React
RUN apt-get update
RUN apt-get install -y nodejs npm


# Установка ASP.NET Core
RUN curl -sSL https://dot.net/v1/dotnet-install.sh
| bash /dev/stdin --channel 9.0 --install-dir
/usr/share/dotnet
# Добавляем .NET в PATH
ENV DOTNET_ROOT=/usr/share/dotnet
ENV PATH=$PATH:/usr/share/dotnet

# Создаём рабочую директорию и переходим в неё
WORKDIR /app

# Копируем из исходников в рабочую директорию
package.json из фронтенда
COPY src/frontend/package*.json ./src/frontend/
# и основной файл C# проекта из бэкенда
COPY src/backend/Backend.csproj ./src/backend/

# Восстановление зависимостей бэкенда
RUN dotnet restore ./src/backend/Backend.csproj

#Устанавиваем зависимости для Реакта.
RUN cd ./src/frontend && npm ci

#Копируем код из исходника в рабочую директорию
COPY . .

# Компилируем и собираем бэкенд (в папку
./publish)
RUN dotnet publish ./src/backend/Backend.csproj -c
Release -o ./publish

# Собираем бэкенд
```

```
RUN cd src/frontend && npm run build

# Создаём в бэкенде если нет папку wwwroot для
хранения статических файлов фронтенда
RUN mkdir -p ./publish/wwwroot/ && \
    cp -r src/frontend/build/* ./publish/wwwroot/

# Перемещаемся в папку с собранным бэкендом
WORKDIR /app/publish

# Ставим порт, через который будем получать доступ
к приложению.
ENV ASPNETCORE_URLS=http://0.0.0.0:5000
EXPOSE 5000

# Устанавливаем точку входа.
ENTRYPOINT ["dotnet", "Backend.dll"]
```

5) Проверяем работу приложения в контейнере.

Собираем образ нашего приложения



(сборка проходит быстро, очевидно ввиду того, что кэш помнит прошлые попытки)

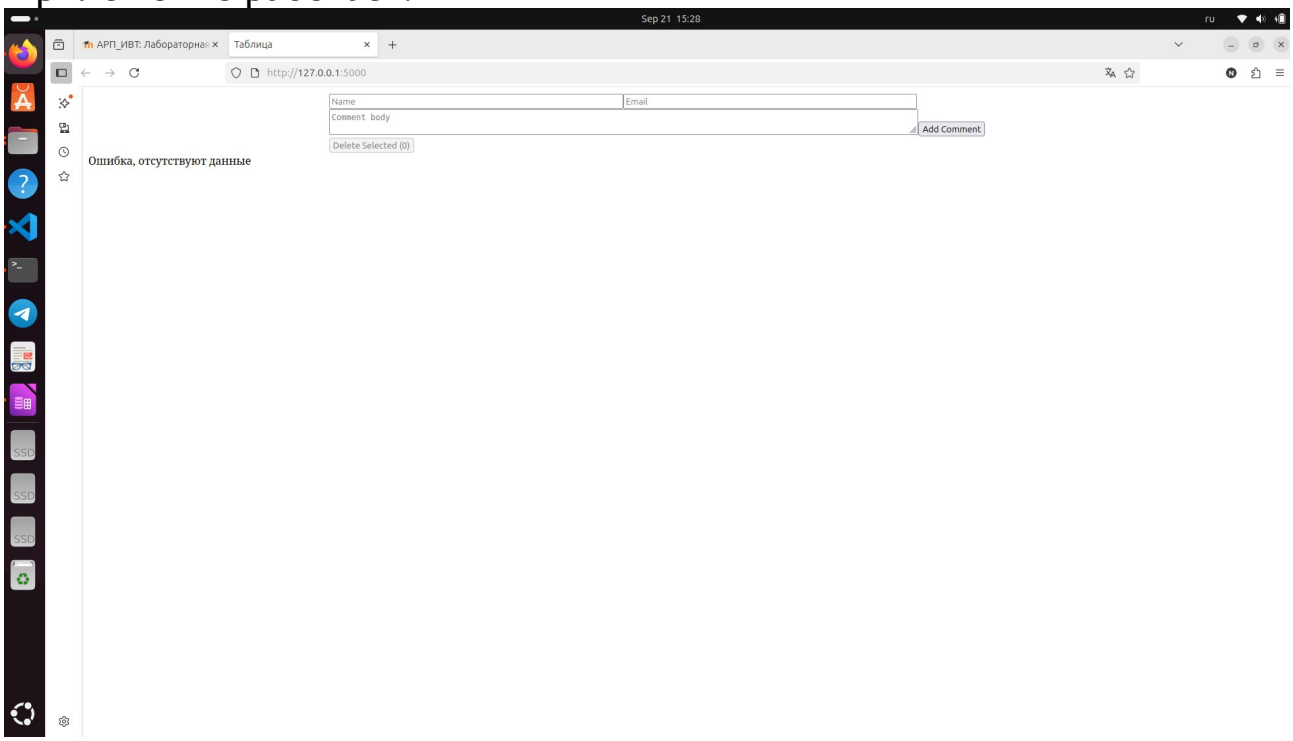
Создаём контейнер:

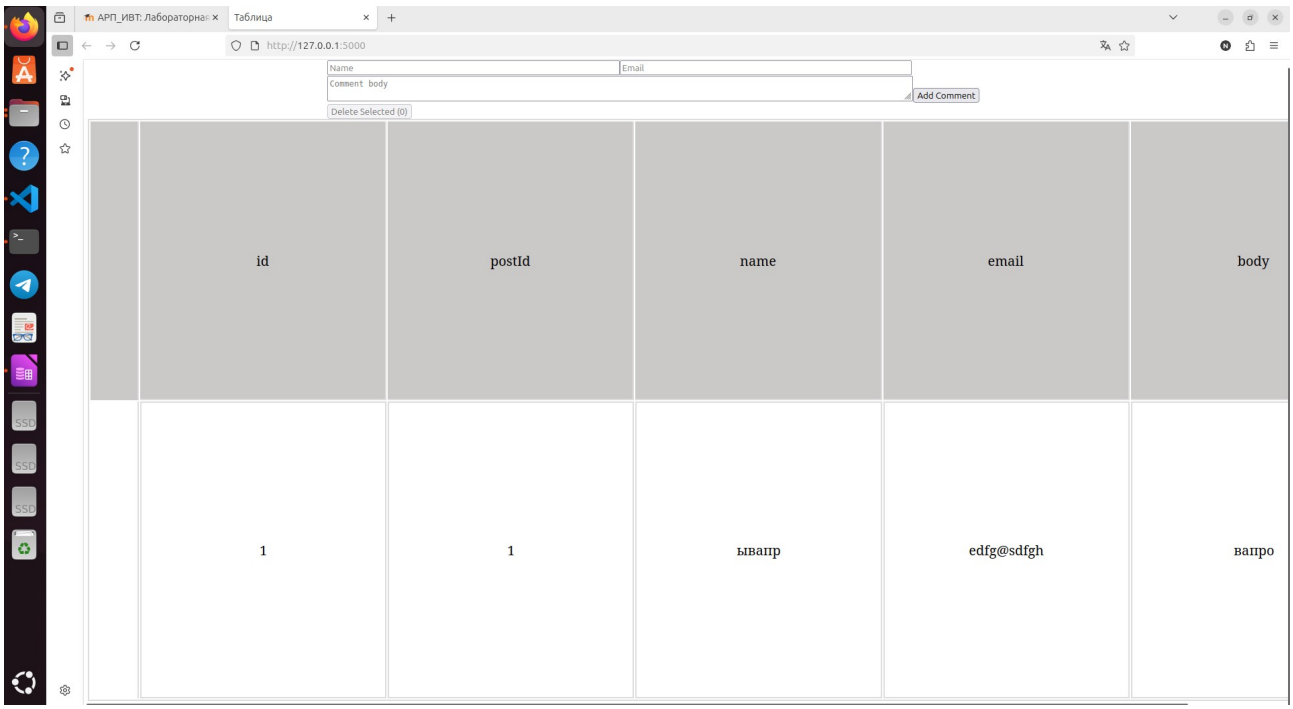
```
gastello123@gastello123:~/Desktop/my-simple-distributed-application$ sudo docker create my-simple-distributed-application
a62598d32a4cc2ae7644a9d63eebc2390508dd72539be3736c24047e2bdef81f
```

Запускаем контейнер:

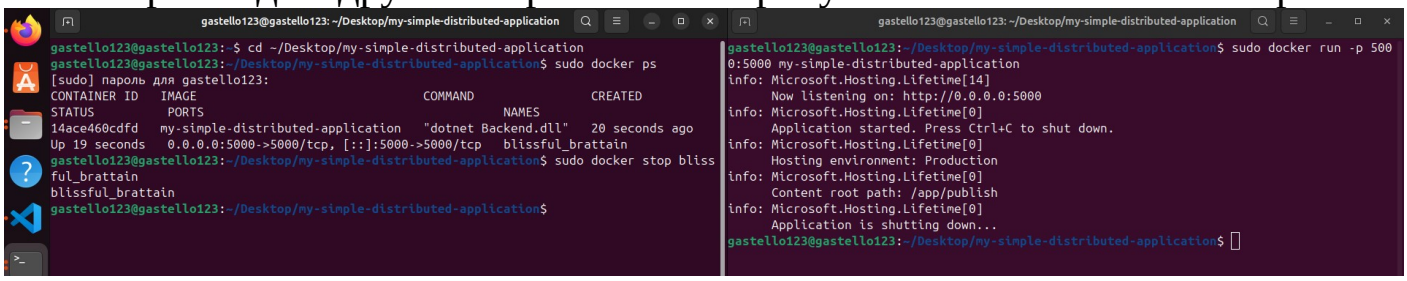
```
gastello123@gastello123:~/Desktop/my-simple-distributed-application$ sudo docker run -p 5000:5000 my-simple-distributed-application
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://0.0.0.0:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /app/publish
```

Приложение работает:





Теперь зайдя с другого терминала попробуем остановить контейнер:



Приложение сразу же закроется и во фронте, попробовав обновить страницу или что-то ещё получим