**Отчет о выполнении лабораторной работы №5**

Якухин Иван, 6303-010302D

*Идея: приложение с карточками для запоминания английских слов*

1. **Упаковка в Docker**

Для упаковки в Docker контейнеры в backend и frontend частях приложения были созданы dockerfile файлы, описывающие установку зависимостей и сборку проектов.

Сервер:

FROM python:3.10  
  
WORKDIR /app  
  
COPY requirements.txt .  
RUN pip install -r requirements.txt  
  
COPY . .  
  
EXPOSE 8000  
  
CMD ["gunicorn", "--bind", "0.0.0.0:8000", "cardiogram.wsgi:application"]

Клиент:

FROM node:20 as build  
WORKDIR /app  
COPY package\*.json ./  
RUN npm ci  
COPY . .  
RUN npm run build  
FROM nginx:alpine  
COPY --from=build /app/build /usr/share/nginx/html  
COPY --from=build /app/build/static /usr/share/nginx/html/static  
COPY nginx.conf /etc/nginx/conf.d/default.conf  
EXPOSE 80  
CMD ["nginx", "-g", "daemon off;"]

1. **Настройка docker-compose**

Для согласования работы всех компонентов был создан docker-compose.yml файл, в котором прописаны особенности работы базы данных, серверной и клиентской части.

services:  
 db:  
 image: postgres:latest  
 container\_name: cardiogram-database  
 restart: always  
 volumes:  
 - postgres\_data:/var/lib/postgresql/data  
 env\_file:  
 - .env  
  
 backend:  
 build: cardiogram/  
 container\_name: cardiogram-backend  
 ports:  
 - "8000:8000"  
 depends\_on:  
 - db  
 restart: always  
 env\_file:  
 - .env  
  
 frontend:  
 build: cardiogram-front/  
 container\_name: cardiogram-frontend  
 ports:  
 - "80:80"  
 depends\_on:  
 - backend  
 restart: always  
  
volumes:  
 postgres\_data:

1. **Проверка доступности**

После сборки проекта командой docker-compose up --build была протестирована доступность всех элементов сайта, найденные ошибки были исправлены, проект пересобран до полностью рабочей версии.