# Инструкция по развертыванию FindMyPet на PythonAnywhere

## Содержание

1. [Введение](#введение)
2. [Подготовка](#подготовка)
3. [Настройка backend](#настройка-backend)
4. [Настройка frontend](#настройка-frontend)
5. [Настройка базы данных](#настройка-базы-данных)
6. [Запуск приложения](#запуск-приложения)
7. [Решение проблем](#решение-проблем)

## Введение

Данная инструкция описывает процесс развертывания приложения FindMyPet на платформе PythonAnywhere. Приложение состоит из: - Backend на Flask (Python) - Frontend на React.js - База данных SQLite (или MySQL)

## Подготовка

1. Зарегистрируйтесь на [PythonAnywhere](https://www.pythonanywhere.com/)
2. Выберите подходящий тариф (для начала подойдет бесплатный)
3. Заранее подготовьте GitHub репозиторий проекта

## Настройка backend

### 1. Создание виртуального окружения

1. Откройте консоль Bash на PythonAnywhere и выполните:

cd  
mkdir findmypet  
cd findmypet  
python -m venv venv  
source venv/bin/activate

### 2. Клонирование репозитория

git clone https://github.com/marguberk/findmypet.git repo  
cd repo

### 3. Установка зависимостей backend

cd backend  
pip install -r requirements.txt

### 4. Настройка WSGI файла

1. В PythonAnywhere перейдите в раздел Web
2. Создайте новое веб-приложение, выбрав “Manual configuration” и версию Python 3.9 или выше
3. В разделе “Code” установите путь к вашему приложению: /home/yourusername/findmypet/repo/backend
4. Настройте WSGI файл (нажмите на ссылку со путем к WSGI файлу) и замените его содержимое:

import sys  
import os  
  
# Добавляем путь к приложению  
path = '/home/yourusername/findmypet/repo/backend'  
if path not in sys.path:  
 sys.path.append(path)  
  
# Активируем виртуальное окружение  
activate\_this = '/home/yourusername/findmypet/venv/bin/activate\_this.py'  
with open(activate\_this) as file\_:  
 exec(file\_.read(), dict(\_\_file\_\_=activate\_this))  
  
# Импортируем приложение  
from run import app as application

### 5. Настройка статических файлов

В разделе “Static files” PythonAnywhere добавьте: - URL: /static/ - Path: /home/yourusername/findmypet/repo/backend/app/static

И для файлов загрузки: - URL: /uploads/ - Path: /home/yourusername/findmypet/repo/backend/app/uploads

## Настройка frontend

### 1. Сборка проекта React

**ВАЖНО:** На PythonAnywhere нет Node.js и npm по умолчанию, поэтому сборку фронтенда необходимо делать на локальной машине!

Соберите фронтенд на вашей локальной машине (не на PythonAnywhere):

cd frontend/findmypet-client  
npm install  
npm run build

После сборки у вас появится папка build с готовыми статическими файлами.

### 2. Загрузка файлов сборки

После сборки на локальной машине, существует два способа загрузить файлы на PythonAnywhere:

#### Вариант 1: Через Git

Добавьте папку build в репозиторий, закоммитьте и выполните git push, затем на PythonAnywhere выполните git pull.

# На локальной машине  
git add frontend/findmypet-client/build  
git commit -m "Add build files"  
git push  
  
# На PythonAnywhere  
cd /home/yourusername/findmypet/repo  
git pull

#### Вариант 2: Через ZIP архив

1. На локальной машине запакуйте папку build в ZIP архив
2. Загрузите архив на PythonAnywhere через раздел Files
3. Распакуйте архив на PythonAnywhere:

cd /home/yourusername/findmypet/repo  
unzip build.zip -d frontend\_build

### 3. Настройка статических файлов для фронтенда

В разделе “Static files” PythonAnywhere добавьте: - URL: / - Path: /home/yourusername/findmypet/repo/frontend\_build или /home/yourusername/findmypet/repo/frontend/findmypet-client/build (в зависимости от метода загрузки)

## Настройка базы данных

### 1. SQLite (простой вариант)

Обновите конфигурацию в файле /home/yourusername/findmypet/repo/backend/config.py:

SQLALCHEMY\_DATABASE\_URI = 'sqlite:////home/yourusername/findmypet/instance/pets.db'

### 2. MySQL (для продакшена)

1. Создайте базу данных MySQL в разделе Databases на PythonAnywhere
2. Обновите конфигурацию в файле /home/yourusername/findmypet/repo/backend/config.py:

SQLALCHEMY\_DATABASE\_URI = 'mysql://yourusername:database\_password@yourusername.mysql.pythonanywhere-services.com/yourusername$pets'

### 3. Инициализация базы данных

cd /home/yourusername/findmypet/repo/backend  
python -c "from app import create\_app, db; app=create\_app(); app.app\_context().push(); db.create\_all()"

## Запуск приложения

1. Перейдите в раздел Web в PythonAnywhere
2. Нажмите кнопку “Reload yourusername.pythonanywhere.com”
3. Проверьте работу приложения, перейдя по URL: https://yourusername.pythonanywhere.com

## Решение проблем

### Логи ошибок

В случае проблем проверьте: - Error log на странице Web в PythonAnywhere - Server log для более подробной информации

### Проблемы с CORS

Если возникают проблемы с CORS, обновите настройки в файле /home/yourusername/findmypet/repo/backend/app/\_\_init\_\_.py:

# Настройка CORS  
CORS(app, resources={r"/api/\*": {"origins": "https://yourusername.pythonanywhere.com"}})

### Проблемы с npm и Node.js

PythonAnywhere не предоставляет Node.js и npm в бесплатном тарифе. Есть несколько решений:

1. **Рекомендуемый способ**: Собирайте React-приложение локально и загружайте готовую сборку
2. **Альтернативный способ**: При необходимости, можно установить Node.js в домашнюю директорию:

# Установка Node.js в домашнюю директорию (может не работать на бесплатном тарифе)  
mkdir -p ~/node  
cd ~/node  
wget https://nodejs.org/dist/v16.20.0/node-v16.20.0-linux-x64.tar.xz  
tar xf node-v16.20.0-linux-x64.tar.xz  
echo 'export PATH=$HOME/node/node-v16.20.0-linux-x64/bin:$PATH' >> ~/.bashrc  
source ~/.bashrc

### Проблемы с загрузкой файлов

Убедитесь, что директория для загрузки файлов существует и имеет правильные разрешения:

mkdir -p /home/yourusername/findmypet/repo/backend/app/uploads  
chmod 755 /home/yourusername/findmypet/repo/backend/app/uploads

### Обновление приложения

Для обновления приложения:

cd /home/yourusername/findmypet/repo  
git pull  
# Перезагрузите приложение через веб-интерфейс PythonAnywhere

Для обновления фронтенда после внесения изменений повторите процесс сборки на локальной машине и загрузки готовых файлов на PythonAnywhere.