Руководство администратора — Farmer Markets

📘 Документация к приложению Farmer Markets

1. Назначение документа

Документ описывает полную процедуру развертывания, первичной настройки, эксплуатации, безопасного и стабильного сопровождения приложения **Farmer Markets** на Windows, Linux и macOS. Все операции максимально автоматизированы; для неизбежных ручных шагов приведены пошаговые инструкции.

2. Архитектура и состав

Бэкенд: **Django 5** (приложение `markets`), встроенная аутентификация, группы и права.

База данных: **PostgreSQL** (в Docker).

Веб-интерфейсы: **Django** (основной), **Streamlit** (демо/исторический).

Оркестрация: **Docker Compose**.

Скрипты запуска: бат/ш для Windows/Linux/macOS (см. папку `exe/`).

Автоинициализация БД + ролей при старте контейнеров выполняется через `entrypoint.sh` (миграции, роли, загрузка данных, опционально — автосоздание суперпользователя из ENV) .

README с обзором возможностей, структурой проекта и командами уже в репозитории (актуализировано) .

3. Предварительные требования

Windows: Docker Desktop.

Linux/macOS: Docker Engine + Docker Compose.
Проверь:

docker --version

docker compose version # либо docker-compose -v на старых установках

4. Подготовка окружения

1. Клонируй репозиторий и перейди в корень проекта:

git clone https://github.com/nzhivolun/farmer_markets_project.git

cd farmer_markets_project

`entrypoint.sh` при запуске Django берёт эти переменные и создаёт суперпользователя «если его ещё нет»

5. Автоматизированное развертывание

5.1 Windows (PowerShell / cmd)

Django-стек:

exe\start_app-django.bat

Скрипт поднимет контейнеры, выполнит миграции, инициализирует роли и (при наличии ENV) создаст суперпользователя.

Streamlit-стек (демо):

exe\start_app-streamlit.bat

5.2 Linux/macOS

Сначала дать права на исполнение

chmod +x exe/*.sh

Django-стек:

./exe/start_app-django.sh

Streamlit-стек (демо):

./exe/start_app-streamlit.sh

5.3 Проверка статуса

docker ps --format 'table {{.Names}}\t{{.Ports}}'

Ожидаемо увидеть:

`farmer_db` — PostgreSQL (порт 5432 проброшен),

`farmer_django` — Django (порт 8502 проброшен).

6. Первичный доступ

Сайт (Django): `http://127.0.0.1:8502/`

Админка Django: `http://127.0.0.1:8502/admin/`
(адреса также печатает `entrypoint.sh` при запуске) .

Если переменные `DJANGO_SUPERUSER_*` не заданы, суперпользователь **не создаётся** автоматически (это видно в логах `entrypoint.sh`). В этом случае можно:

Либо перезапустить контейнер с ENV,

Либо создать вручную внутри контейнера:

:: Windows (уже работающий контейнер)

docker compose exec -it farmer_django sh -lc "python /app/web/manage.py createsuperuser"

(Путь `manage.py` и команда соответствуют логике entrypoint; там же показывается использование `createsuperuser --noinput` при наличии ENV) .

7. Конфигурация и роли/права

Группы и права инициализируются отдельной Django-командой (`manage.py init_roles`), вызов организован из `entrypoint.sh` и выполняется идемпотентно (безопасно при повторах) .

Авторизация/шаблоны входа/выхода уже готовы (`login.html`, `logged_out.html`) и интегрированы в тему (кнопка «Принять условия», ссылки на Sign In/Up и т.д.) .

8. Локализация (переводы)

Файлы `.po` компилируются в `.mo` простым скриптом `tools/compile_messages.py` (использует лёгкий модуль `polib`):

python tools/compile_messages.py

Скрипт ищет стандартные пути (`web/locale/ru/LC_MESSAGES/django.po` и т.д.) и создаёт `.mo` рядом (UTF-8) .

9. Обновление версии/деплой

1. Обнови код:

git pull

2. Пересобери и перезапусти:

docker compose down

docker compose up --build -d

3. Миграции выполнятся автоматически при старте (см. `entrypoint.sh`) .

10. Устранение неполадок

`docker-compose не найден`: установи `docker compose` (для Linux) и перезапусти терминал.

Контейнер перезапускается: посмотри логи `farmer_django` и `farmer_db`; проверь переменные окружения и доступность БД.

`createsuperuser` просит ввод и «закрывается» на Windows: запускай в интерактивном режиме, как показано выше (через `docker compose exec -it ... sh -lc "python /app/web/manage.py createsuperuser"`).

📂 Структура проекта

```
```
FARMER_MARKETS_DJANGO/ # корень проекта
├── app/ # консоль + Streamlit (исторический код)
│ ├── main.py # консольное меню (CLI)
│ ├── app_streamlit.py # запуск Streamlit-приложения
│ ├── ui_markets_streamlit.py # страницы/виджеты Streamlit
│ ├── markets.py # логика рынков (консоль)
│ ├── reviews.py # работа с отзывами (процедурный подход)
│ ├── reviews_oop.py # ООП-менеджер отзывов
│ ├── categories.py # работа с категориями (консоль)
│ ├── utils.py # простые утилиты (валидация, пагинация и т.п.)
│ └── db.py # подключение к БД (минимальная обёртка)
│
├── docs/ # документация (для пользователя и администратора)
│ ├── Admin_User_Guide.docx
│ ├── Admin_User_Guide.pdf
│ ├── User_Guide.docx
│ └── User_Guide.pdf
│
├── exe/ # кроссплатформенные скрипты
│ ├── create_django_superuser.bat # Windows: создать суперпользователя в
контейнере
│ ├── create_django_superuser.sh # Linux/Mac: создать суперпользователя в
контейнере
│ ├── start_app-django.bat # Windows: запуск стека Django
│ ├── start_app-django.sh # Linux/Mac: запуск стека Django
│ ├── start_app-streamlit.bat # Windows: запуск Streamlit-версии
│ ├── start_app-streamlit.sh # Linux/Mac: запуск Streamlit-версии
│ ├── start_app.bat # Windows: общий старт-скрипт
│ └── start_app.sh # Linux/Mac: общий старт-скрипт
│
```
```

```
├── setup/ # подготовка и инициализация БД (локальные сценарии)
│ ├── setup_db.py # создание схемы/таблиц вне Django (для учебных целей)
│ ├── config.py # локальная конфигурация подключения
│ ├── init.sql # SQL-схема/данные
│ └── Export.csv # пример исходных данных
│
├── tools/ # дополнительные инструменты/скрипты (опционально)
├── venv/ # локальное виртуальное окружение (может отсутствовать)
│
├── web/ # Django-версия (основное веб-приложение)
│ ├── fm_project/ # django-проект (settings, urls и т.д.)
│ ├── locale/ # переводы (po/mo)
│ ├── markets/ # django-приложение (views, models, urls, templates)
│ ├── static/ # статика для Django
│ ├── templates/ # базовые шаблоны (base.html и т.д.)
│ ├── .env # локальные переменные окружения (НЕ коммитить)
│ ├── .env.example # шаблон .env
│ ├── manage.py # точка входа Django
│ └── requirements.txt # зависимости Python
│
├── docker/ # дополнительные docker-скрипты (если нужны)
├── docker-compose.yml # оркестрация: Postgres + Web
├── Dockerfile # образ приложения
├── entrypoint.sh # entrypoint контейнера (ожидание БД, миграции, init)
└── README.md
```

🔐 Безопасность

- Пароли и логины не хардкодятся в коде — читаются из переменных окружения.
- Пользователь БД — обычный (`app_user`), а не суперпользователь (`postgres`).
- Все операции по инициализации инкапсулированы в `entrypoint.sh`.

📊 Структура базы данных

Создаётся 5 таблиц:

Таблица	Назначение
`locations`	Локации (город, штат, индекс и т.п.)
`markets`	Основная информация о рынке
`reviews`	Отзывы пользователей
`categories`	Категории товаров (например, "Meat", "Fruits")
`market_categories`	Связь рынков с категориями (многие ко многим)

❗ Частые ошибки и способы решения

Ошибка	Возможная причина	Решение
`Docker не найден!`	Docker не установлен или не добавлен в PATH	Установить Docker Desktop
`docker-compose не найден!`	Не установлен `docker-compose`	Установить отдельно или использовать встроенный (Docker Compose v2)
`Ошибка: контейнер farmer_app не запущен!`	Контейнер не стартовал	Проверить `docker ps`, перезапустить `docker-compose up --build`
`psycopg2.OperationalError: connection refused`	Контейнер с базой ещё не готов	Убедиться, что есть пауза (sleep) перед подключением или перезапустить
`FATAL: database "farmer_markets" does not exist`	Скрипт `setup_db.py` не выполнился	Проверить логи, запустить вручную: `docker exec -it farmer_app python setup/setup_db.py`
`Ошибка при загрузке данных`	Проблема с CSV, отсутствует файл или неверная кодировка	Убедиться, что файл `Export.csv` на месте, в правильной папке и UTF-8

🐧 Запуск в WSL (Windows Subsystem for Linux)

Если ты используешь **WSL (Ubuntu в Windows)**:

🔧 Подготовка

1. Убедись, что Docker Desktop установлен на Windows и включена интеграция с WSL:

- Открой Docker Desktop → Settings → Resources → WSL Integration → включи Ubuntu.

2. Проверь, что из WSL доступны команды:

docker --version

docker-compose –version

Перейди в каталог проекта:

cd /mnt/d/Обучение/ИТМО/Разработчик\ Python/03\ Базовый\ Питон/farmer_markets_project

Путь может отличаться — проверь как примонтирован диск (`/mnt/c`, `/mnt/d` и т.п.).

🚀 Запуск в WSL

chmod +x start_app.sh

./start_app.sh

После этого откроется интерактивное консольное приложение прямо в WSL.

🛠️ Проблемы в WSL и их решение

Ошибка	Причина	Решение
`Cannot connect to the Docker daemon`	Docker не доступен из WSL	Проверить, включена ли интеграция Docker Desktop с WSL
`Permission denied` при запуске `.sh`	Нет прав на выполнение	Выполнить `chmod +x start_app.sh`
`File not found: /app/setup/init.sql`	Нарушена структура папок	Убедиться, что проект собран строго по описанной структуре