Отлично — так и **должно быть**. Виртуальное окружение (venv) **никогда не включают** в репозиторий, потому что оно:

* содержит тонны ненужных бинарников,
* специфично для ОС и Python-версии,
* легко воссоздаётся по requirements.txt.

**Что делать, если venv нет:**

На втором компьютере ты просто создаёшь новое окружение и **восстанавливаешь зависимости**:

**Шаги:**

1. **Создай venv:**
2. python -m venv venv
3. **Активируй его:**
   * Windows:
   * venv\Scripts\activate
   * macOS/Linux:
   * source venv/bin/activate
4. **Установи зависимости:**
5. pip install -r requirements.txt

Если requirements.txt ещё не было — на первом компе создай его командой:

pip freeze > requirements.txt

Хочешь, я помогу составить .gitignore для Django и Python-проекта?

Да, всё верно:

### ✅ requirements.txt создаётся ****внутри активированного виртуального окружения****

Он фиксирует все зависимости, установленные в этом окружении, командой:

pip freeze > requirements.txt

### 📁 Где должен лежать requirements.txt?

Обычно — **в корне проекта**, рядом с manage.py, README.md и .gitignore.

Пример структуры:

myproject/

│

├── manage.py

├── requirements.txt ← вот тут

├── .gitignore

├── myapp/

│ └── ...

└── myproject/

└── settings.py

Когда другой человек (или ты сам на другом компе) клонируешь репозиторий, он сможет развернуть проект, используя:

python -m venv venv

source venv/bin/activate # или venv\Scripts\activate на Windows

pip install -r requirements.txt

Хочешь — покажу, как зафиксировать только необходимые зависимости, без лишнего мусора из pip freeze?

Если ты уже запушил Django-проект на GitHub и хочешь продолжить работу на другом компьютере, **создавать приложение заново не нужно**. Просто:

### 🔧 Шаги для запуска Django-проекта на втором компьютере:

1. **Клонируй репозиторий:**
2. git clone https://github.com/твой-пользователь/твой-проект.git
3. cd твой-проект
4. **Создай и активируй виртуальное окружение:**
5. python -m venv venv
6. source venv/bin/activate # на Linux/macOS
7. venv\Scripts\activate # на Windows
8. **Установи зависимости из requirements.txt:**
9. pip install -r requirements.txt
10. **Настрой .env (если используешь):** Создай .env файл и пропиши переменные окружения (например, SECRET\_KEY, DEBUG, DATABASE\_URL и т. д.).
11. **Примени миграции и собери статику (если нужно):**
12. python manage.py migrate
13. python manage.py collectstatic
14. **Запусти сервер:**
15. python manage.py runserver

### ✅ У тебя уже всё будет:

* Все файлы приложений и настроек.
* База (если SQLite) или можно подключиться к удалённой (если PostgreSQL и т.п.).
* Только не забудь про .env или local\_settings.py, если они в .gitignore.

Хочешь — могу помочь с созданием requirements.txt или .env.