Отлично, принял — давай обновим ТЗ с учётом новых важных уточнений:

**🧾 Обновлённое техническое задание**

**Название проекта:** ONLINE PRINTER  
**Автор:** Эркинов Амантур Эркинович  
**Дата:** 05.04.2025

**1. Цель проекта**

Разработка сети автономных принтеров, доступных для онлайн-печати документов через веб-интерфейс. Пользователь выбирает ближайший принтер на карте, загружает файл, оплачивает онлайн и получает распечатку на месте.

**2. Ключевые особенности MVP**

**📍 Геолокация и карта:**

* Отображение всех доступных принтеров на интерактивной карте
* Информация о каждом принтере: статус (работает / нет), типы бумаги, цветность, загруженность
* Возможность выбора ближайшего принтера

**📄 Печать:**

* Загрузка документов онлайн через веб-сайт
* Поддержка форматов: PDF, DOCX, PNG, JPG
* Печать A4 (цветная); A3 — опционально в будущем
* Предпросмотр документа перед печатью
* Очередь печати, статус выполнения

**💳 Оплата:**

* Только онлайн (через сайт)
* Поддержка: Kaspi QR, Apple Pay, Google Pay, карта
* Оплата до начала печати, автоматическая разблокировка задания после оплаты

**🧾 Учет:**

* Принтеры привязаны к уникальному ID / локации
* Учёт количества напечатанных страниц, затраченной бумаги
* Учет дохода по каждому принтеру
* Автоматическая очистка загруженных файлов после печати

**3. Оборудование терминала**

| **Компонент** | **Описание** |
| --- | --- |
| Одноплатный ПК | Raspberry Pi 5 / Orange Pi 5 / Radxa Rock / Intel NUC |
| Принтер | Цветной принтер A4 (USB/Wi-Fi); поддержка драйверов на Linux |
| Модем | 4G USB-модем с SIM-картой (автономный интернет) |
| USB-порт | (опционально для будущего — приём с флешки) |
| Корпус | Компактный корпус с выводом бумаги и охлаждением |
| Источник питания | Возможность автозапуска при питании (настройка BIOS «Power on AC loss») |

**4. Архитектура сервиса**

**🌐 Веб-сайт:**

* Пользователь загружает документ, выбирает принтер на карте, оплачивает
* После оплаты файл отправляется на сервер принтера

**🖨 Терминал:**

* Работает мини-сервер (Python + Flask/Django)
* Принимает файл, отправляет на печать
* Связывается с облаком (через API) для получения статуса оплаты
* Удаляет файл после печати

**🔒 Безопасность:**

* HTTPS
* Шифрование документов при передаче
* Файлы хранятся на принтере временно (не дольше 10 мин)

**5. Технологии и ПО**

| **Компонент** | **Инструмент** |
| --- | --- |
| Серверная часть | Django REST + Celery (обработка печати) |
| Фронтенд | React / Vue (опционально), Bootstrap, Leaflet (для карты) |
| БД | PostgreSQL / SQLite |
| Печать | CUPS (Common Unix Printing System) |
| ОС | Ubuntu Server / Raspberry Pi OS |
| Контейнеризация | Docker (опционально) |
| Хостинг | DigitalOcean / Render / Hetzner (для центрального сервера) |

**6. Ожидаемый результат**

* Сеть из минимум 2–3 онлайн-принтеров
* Полнофункциональный веб-сайт с загрузкой, оплатой, картой
* Надежная онлайн-печать с минимальными затратами
* Гибкость масштабирования под сотни устройств в городе

**🚀 Дальнейшие шаги**

1. ✅ У тебя уже готов сайт с картой — отлично!
2. ⏳ Настроим базу данных с принтерами, API, очередь печати
3. ⚙️ Установим софт на один Raspberry Pi и протестируем
4. 🧩 Подготовим визуальную архитектуру системы и интерфейса
5. 💸 После тестов можно подавать на гранты, искать партнёров, ставить по городу

Хочешь, сейчас подготовлю:

* 📄 PDF-документ с этим ТЗ
* 🖼 Визуальную схему архитектуры (как сайт ↔ сервер ↔ терминал работает)
* 💻 Пример интерфейса (макет загрузки + оплата + карта)

С чего начнем?