отлично, я посмотрел структуру **DocuGenius** (Node/TS: server+client, Express + node-telegram-bot-api, in-memory storage, mock Google Drive). Ниже — готовые доработки **под этот проект**, без пересборки на Python.

я сделал для тебя **пакет патчей**: новые модули + вставки в существующие файлы (телеграм-бот, маршруты) и список зависимостей. Забирай архив и распакуй в корень репо (с сохранением путей).

Супер. Собрал **единый Git-патч**, который применяет все правки разом: новые файлы + изменения в routes.ts и telegramBot.ts + добавляет зависимости (googleapis, qrcode) в package.json.

📦 **Единый патч:**  
docugenius\_all\_in\_one.patch

На всякий случай оставил и целиком собранный проект:

* DocuGenius\_patched\_v2.zip — можно просто распаковать поверх.

**Как применить патч (Git)**

Выполни из корня репозитория **DocuGenius** (где лежат server/, package.json):

# 1) Скопируй docugenius\_all\_in\_one.patch в корень репо

# 2) Применяй патч

git apply --whitespace=fix docugenius\_all\_in\_one.patch

# 3) Установи зависимости

npm i

# 4) Запусти

npm run dev

Если что-то локально уже менялось и будут конфликты — можно так:

git apply --reject --whitespace=fix docugenius\_all\_in\_one.patch

# после — вручную принять .rej (обычно минимально)

**Что включает патч**

**Новые файлы**

* server/services/langDetect.ts — авто-детект KAZ/RUS
* server/services/funnel.ts — «воронка» из 3 вопросов + резюме
* server/services/qr.ts — генерация PNG QR
* server/integrations/google/sheets.ts — поиск/цены/лиды в Google Sheets
* server/integrations/whatsapp/{base.ts,factory.ts,wa360dialog.ts} — каркас WA (360dialog)
* README\_PATCH.txt

**Изменённые файлы**

* server/routes.ts — REST:
  + GET /api/catalog/price/:sku
  + GET /api/catalog/search?q=…
  + POST /api/leads → создаёт лид и возвращает **QR (PNG)**
  + POST /api/payments/qr/callback (мок, проверка x-signature)
  + POST /api/whatsapp/webhook — логика под 360dialog: /price, /find, /order, 2GIS, “перезвон”, общий **funnel**
* server/services/telegramBot.ts — команды и обработчики:
  + /start → клавиатура: **🧭 Навигация (2GIS)**, **📞 Перезвонить** + запуск воронки
  + /stop, /price <SKU>, /find <q>, /promo, /order
  + fallback-воронка (KAZ/RUS) на обычные сообщения
* package.json — добавлены зависимости:
  + "googleapis": "^136.0.0", "qrcode": "^1.5.4"

**Переменные окружения (проверь в Secrets/.env)**

TELEGRAM\_BOT\_TOKEN=<...>

TELEGRAM\_OPERATORS\_GROUP\_ID=<ID группы операторов> # опционально

# Каталог / цены (Google Sheets)

PRICES\_SHEET\_ID=<ID таблицы с ценами>

PRICES\_RANGE=Sheet1!A:Z

# Лиды (Google Sheets) — опционально

LEADS\_SHEET\_ID=<ID таблицы для лидов>

LEADS\_RANGE=Sheet1!A:Z

# Перезвоны (можно reuse LEADS\_SHEET\_ID)

CALLBACKS\_SHEET\_ID=<ID таблицы для перезвонов>

CALLBACKS\_RANGE=Sheet1!A:Z

# Google Service Account

GOOGLE\_CREDENTIALS\_JSON\_BASE64=<base64 JSON сервис-аккаунта>

# Ссылки

PROMO\_URL=https://example.com

TWO\_GIS\_URL=https://2gis.kz/astana/search/YourAddress

# Платежи (мок колбэка)

PAYMENT\_CALLBACK\_SECRET=<секрет>

# WhatsApp (каркас 360dialog)

WA\_PROVIDER=360dialog

WA\_API\_URL=https://waba.360dialog.io/v1

WA\_API\_KEY=<api-key>

WA\_PHONE\_NUMBER\_ID=<number-id>

**Быстрый smoke-тест**

* /start → клавиатура + первый вопрос воронки (язык соответствует входному)
* /price <SKU> и /find <q> → ответы из Google Sheets
* /order → создаётся лид (если задан LEADS\_SHEET\_ID), присылается **QR**
* Нажать **🧭 Навигация (2GIS)** → присылает TWO\_GIS\_URL
* Нажать **📞 Перезвонить** → запись в CALLBACKS\_\* и уведомление в группу (если задана)
* POST /api/whatsapp/webhook с сообщениями 360dialog → отрабатывают команды и воронка

Если нужен ещё **отправка QR в WhatsApp** (картинкой) — добавлю sendImage() в WA-клиент и хукну в /order под WA.