

Итоговый функционал sub.osipenkov.ru (Composer + Cursor)

3. Docker: база данных, API, фронтенд, nginx

3.1. docker-compose.yml

- Сервисы: `db` (PostgreSQL), `api` (NestJS), `web` (Next.js), `nginx`.
- Общая сеть, например `subnet`.
- Порты nginx: `80`, `443`.
- Volumes: данные БД, конфиг nginx, SSL-сертификаты (после выпуска), при необходимости — `adminer.htpasswd`.

3.2. Переменные окружения

- ****API (`apps/api/.env`)**:**
 - `DATABASE_URL=postgresql://sub_user:sub_password@db:5432/sub_db`
 - `TELEGRAM_BOT_TOKEN=...` (от BotFather)
 - `JWT_SECRET=...` (сгенерировать: `openssl rand -hex 32`)
 - `TELEGRAM_CHANNEL_USERNAME=@your_channel`
- ****Web (`apps/web/.env`)**:**
 - `NEXT_PUBLIC_TELEGRAM_BOT_NAME=your_bot_username` (без @)

3.3. Запуск

```
```bash
cd ~/my-project
Создать пустой .env в apps/web, если нужен для docker-compose
touch apps/web/.env
docker compose up -d db api web nginx
docker compose ps
```
```

Проверка: `http://IP/api/health` → `{"status":"ok"}` (до настройки HTTPS — по IP).

4. SSL (HTTPS) на VPS

4.1. DNS

- А-запись: `sub.example.com` → IP сервера (`1.2.3.4`).

- Для Let's Encrypt лучше временно убрать АААА, если она ведёт не на ваш сервер.

4.2. Выпуск сертификата (certbot)

```
```bash
cd ~/my-project
docker compose down
sudo certbot certonly --standalone -d sub.example.com --email your@email.com --
agree-tos --non-interactive
```
```

Сертификаты появятся в `/etc/letsencrypt/live/sub.example.com/`
(`fullchain.pem`, `privkey.pem`).

4.3. nginx: HTTPS и редирект

- В `docker-compose.yml` у nginx: порты `80`, `443` и volumes на `fullchain.pem` и `privkey.pem`.
- В `nginx.conf`:
 - Сервер на 80: `return 301 https://$host$request_uri;`
 - Сервер на 443: `ssl_certificate`, `ssl_certificate_key`, `location /` → проху на web, `location /api/` → проху на api.

4.4. Запуск с HTTPS

```
```bash
docker compose up -d db api web nginx
```
```

Проверка: `https://sub.example.com/api/health` → `{"status":"ok"}`.

5. Авторизация через Telegram

5.1. BotFather

- Создать бота, получить токен.
- В настройках бота указать `**Domain**`: `sub.example.com` (без https и пути).

5.2. Бэкенд

- Модуль Auth: проверка подписи данных Telegram Login Widget (HMAC-SHA256 с токеном бота).

- Таблица `users`: `id`, `telegram_id`, `username`, `first_name`, `last_name`, `photo_url`, `bot_subscribed`, `channel_subscribed`, `last_login_at` и т.д.
- Эндпоинт `POST /api/auth/telegram`: принять данные от виджета, проверить подпись, создать/обновить пользователя, выдать JWT.

5.3. Фронтенд (главная)

- Страница с Telegram Login Widget (скрипт с `data-telegram-login`, `data-onauth`).
- В `onTelegramAuth(user)` — запрос `POST /api/auth/telegram` с телом от Telegram, сохранение JWT (например в `localStorage`), редирект на `/lk`.

5.4. Важно

- Виджет работает только на домене, указанном в BotFather (не по IP).
- Подпись для Login Widget считается по правилам Telegram (не как для WebApp).

6. Личный кабинет и JWT

- JWT Guard в NestJS: чтение токена из `Authorization: Bearer <token>`.
- `GET /api/me`: по JWT взять `sub` (id пользователя), вернуть данные из `users`.
- Страница `/lk`: при загрузке запрос к `/api/me` и `/api/subscription/status`; при 401 — очистить токен и редирект на `^`.
- Меню ЛК: пункты «Подписка в Telegram», «Бесплатный контент» (и при необходимости «Платный контент» позже).

7. Telegram-бот: вебхук и подписки

7.1. Вебхук

- В nginx: `location /telegram/webhook` → проху на `http://api:3001/api/telegram/webhook` (учёт префикса API).
- Установить вебхук (один раз):

```
```bash
curl -s "https://api.telegram.org/bot<TOKEN>/setWebhook" -d
"url=https://sub.example.com/telegram/webhook"
```
```

- В API: `POST /api/telegram/webhook` — разбор `body.message`, `body.my_chat_member`, `body.chat_member`.

7.2. Обработка событий

- `message.text === '/start'`: создать/обновить пользователя, `bot_subscribed = true`, отправить приветствие через Bot API (`sendMessage`).
- `my_chat_member` (приватный чат с ботом): при `new_chat_member.status` = `kicked`/`left` → `bot_subscribed = false`, при `member` → `true`.
- `chat_member` (канал): при отписке/подписке обновлять `channel_subscribed` в БД (если бот админ канала).

7.3. Подписки в ЛК

- `GET /api/subscription/status`: вернуть `bot_subscribed`, `channel_subscribed` из БД.
- `POST /api/subscription/check`: через Bot API `getChatMember` проверить подписку на канал, обновить БД, вернуть актуальные флаги.

8. Бесплатный контент

8.1. БД

Таблица `free_content`:

- `id`, `title`, `description`, `type` (text, link, photo, document, video), `payload`, `is_active`, `created_at`.
- Создание таблицы — при первом обращении к API (или миграцией).

8.2. API

- `GET /api/free-content/list`: список записей с `is_active = true` (без поля `payload` в ответе).
- `POST /api/free-content/send`: тело `{ contentId }`. Проверить JWT, подписки; взять запись из `free_content`; в зависимости от `type` вызвать `sendMessage`, `sendPhoto`, `sendDocument`, `sendVideo` (для text/link — `parse_mode: 'HTML'` при необходимости).

8.3. Добавление контента

- Через Adminer или pgAdmin: вставка в `free_content`. Для `type = text` или `link` в `payload` — готовый HTML-текст (можно с эмодзи и ссылками).

- Для `photo`/`document`/`video` по URL возможны ограничения Telegram; надёжный вариант — отправлять файл боту вручную и использовать `file_id` в `payload`.

9. Adminer (GUI для БД) и защита паролем

9.1. Сервис Adminer в docker-compose

- Образ `adminer`, переменная `ADMINER_DEFAULT_SERVER: db`.
- В nginx: `location /adminer/` → проху на `http://adminer:8080/`.

9.2. Basic Auth для /adminer/

- Сгенерировать пароль: `openssl passwd -apr1` (ввести пароль, сохранить хеш).
- Создать файл на хосте: `echo 'login:hash' > infra/nginx/adminer.htpasswd` (подставить свой логин и хеш).
- В docker-compose: volume
`./infra/nginx/adminer.htpasswd:/etc/nginx/conf.d/adminer.htpasswd:ro`.
- В nginx: в `location /adminer/` добавить `auth_basic` и `auth_basic_user_file /etc/nginx/conf.d/adminer.htpasswd`.

Важно: на хосте `adminer.htpasswd` должен быть именно ****файлом****, не директорией. После `git pull` при необходимости пересоздать файл на сервере и выполнить `docker compose down && docker compose up -d ...`.

9.3. Вход в Adminer

- URL: `https://sub.example.com/adminer/`.
- Сначала Basic Auth (логин/пароль из htpasswd).
- Затем форма Adminer: System = PostgreSQL, Server = `db`,
Username/Password/Database — из docker-compose (например `sub_user`,
`sub_password`, `sub_db`).





10. Фронтенд: главная, ЛК, адаптивность

10.1. Главная страница

- Заголовок, логотип, описание, виджет Telegram Login.
- Текст про согласие с условиями и политикой конфиденциальности; ссылки на `/terms` и `/privacy` (открывать в новой вкладке при желании).

- Фавикон в `public/favicon.png` и подключение в `_app.jsx` через `<link rel="icon" ... /></Head>`.

10.2. Личный кабинет (/lk)

- Сворачиваемое боковое меню: в свёрнутом виде — иконки (например  Подписка,  Контент,  Выйти), в развёрнутом — текст.
- На мобильных (например `width < 768px`): по умолчанию свёрнутое меню; кнопка  открывает меню (можно оверлеем).
- Контент: блок «Подписка в Telegram» (кнопки: бот, канал, «Проверить подписку»); блок «Бесплатный контент» — таблица на десктопе, на мобильных — карточки с кнопкой «Отправить».
- Отступ контента от левого меню на мобильных задать достаточным (например `marginLeft` = ширина свёрнутого сайдбара + отступ`).

10.3. Кнопки и стили

- Кнопки «Отправить» в блоке бесплатного контента: цвет фона `#009688`, текст белый.
- Единый стиль (тёмная тема, шрифты) на главной и в ЛК.

11. Метрика, dataLayer, документы

11.1. Яндекс Метрика

- В `_app.jsx` подключить счётчик (Next.js `Script` или вставка кода из интерфейса Метрики).
- Указать ID счётчика и при необходимости `ecommerce: "dataLayer"` в `um(..., 'init', { ... })`.

11.2. dataLayer после авторизации

- На странице `/lk` после успешной загрузки данных пользователя (ответ `/api/me`):

```
```js
window.dataLayer = window.dataLayer || [];
window.dataLayer.push({
 event: 'tg_authorization',
 user_id: meData.id, // id из БД
 username: meData.username || `user_${meData.telegram_id}`,
});
```

'''

### ### 11.3. Документы

- Страницы `/terms` и `/privacy`: заголовок, разделы, текст условий использования и политики конфиденциальности, ссылка «Вернуться на главную».
- На главной — короткий текст про согласие с этими документами и ссылки на `/terms` и `/privacy`.

---