**План изменений**

**1) Страница Main**

**UI/UX**

* Добавить нижнюю навигацию (fixed bottom bar): **Главная · Календарь · Профиль · Анализы**.  
  Активная вкладка подсвечена; обработать safe-area (mobile).
* Кнопку/иконку «Обнулить всё» (в шапке или в меню «⋯»): ведёт на страницу подтверждения «Обнуление данных».
* Блок «Прогресс»:
  + Переключатель режима дозы: **Минимальная / Оптимальная / Максимальная** (табы или segmented control).
  + Большое кольцо = текущий выбранный режим; рядом два **маленьких** кольца для других режимов (тап по маленькому — делает его главным).
  + Свайп по большому кольцу тоже переключает режим.
  + Таргет по умолчанию (изменяемый в настройках курса):  
    **min**=120 мг/кг, **opt**=135 мг/кг, **max**=150 мг/кг.  
    target\_mg = weight\_kg \* dose\_per\_kg\_selected.
  + Прогресс = sum(intakes.mg) / target\_mg.

**Фронт**

* Ввести роутер или свой стейт-роутинг: main | calendar | profile | labs.
* Компоненты: <BottomNav/>, <ProgressRings mode=... rings=[min,opt,max] />, страница <ResetAll/>.
* Состояние: doseMode (persist в БД и/или localStorage), rings = {min,opt,max} с вычислением процентов.
* Защита: если нет weight\_kg → в блоке «Прогресс» показать плейсхолдер «Укажите вес в профиле».

**Бэк**

* Эндпоинты:
  + GET /api/progress?mode=min|opt|max → {target\_mg, taken\_mg, percent}
  + POST /api/intakes → {date, mg} (логирование «выпито»)
  + POST /api/dose-mode → {mode} (сохраняем выбор)
  + POST /api/reset → обнуление данных пользователя (см. ниже).

**Страница «Обнуление данных»**

* Текст с перечислением, что будет удалено (интейки, напоминания, локальные кэши прогресса).
* Двойное подтверждение: чекбокс «Я понимаю…» + поле ввода ОБНУЛИТЬ.
* Бэк: POST /api/reset удаляет из intakes, сбрасывает прогресс/настройки, но **не** удаляет профиль.

**2) Напоминание «нужно выпить сегодня»**

**UX**

* Тумблер «Напоминать ежедневно» + время (time-picker) на странице Main/План или Настройки курса.
* Кнопка «Отправить тестовое напоминание».

**Бэк (Telegram WebApp/Бот)**

* Храним chat\_id из initData WebApp.
* Таблица расписаний для напоминаний (время, часовой пояс, включено).
* Планировщик (например, node-cron / BullMQ): раз в минуту проверяем «кому слать сейчас».
* Отправка через Bot API: текст + кнопка «Отметить дозу» (deep-link в мини-апп с параметром startapp=log\_today).

**Минимальный набросок**

// сохраняем настройки

POST /api/reminder { enabled: true, time: "21:00", tz: "Europe/Moscow" }

// крон

cron.schedule('\* \* \* \* \*', async () => {

const due = await db.findUsersToNotify(nowInTheirTZ);

for (const u of due) {

await bot.sendMessage(u.chat\_id, 'Пора принять капсулу 💊', {

reply\_markup: {

inline\_keyboard: [[{ text:'Отметить дозу', web\_app: { url: WEBAPP\_URL + '?startapp=log\_today' } }]]

}

});

}

});

**3) Профиль: нельзя дальше без веса и аллергий**

**UX**

* Поля: **Вес (кг)** — обязательное (валидировать: 30–200).  
  **Аллергии** — обязательное: chips-ввод (или «нет»).
* Кнопка «Дальше/Сохранить» **disabled**, пока форма не валидна.
* Подсветка ошибок инлайн.

**Бэк**

* Серверная валидация: если пусто — 400 {field:"weight\_kg"} / {field:"allergies"}.
* Возвращать детальную ошибку, чтобы фронт показал конкретно где.

**4) Страница/документ «Политика безопасности и отказ от ответственности»**

**Где показывать**

* Ссылка в профиле/онбординге и на странице warnings.html Чекбокс «Согласен» обязателен при старте.
* Храним версию и время согласия.

**Структура разделов (готовый каркас текста)**

* **1. Общие положения.** Назначение приложения; не является мед. изделием.
* **2. Обработка данных.** Какие данные собираем (tg id, вес, аллергии, приёмы), цель, срок хранения.
* **3. Правовые основания и права пользователя.** Доступ/удаление/отзыв согласия.
* **4. Безопасность.** Шифрование в транзите, ограничение доступа, резервные копии.
* **5. Дисклеймер (не медсовет).** Информация носит справочный характер; всегда консультируйтесь с врачом; экстренные ситуации — 112/911.
* **6. Ограничения использования.** 18+, противопоказания, ответственность пользователя за вводимые данные.
* **7. Ответственность.** Команда не несёт ответственности за решения пользователя, сбои связи, интеграции сторонних сервисов.
* **8. Контакты.** Email/бот для запросов по данным.
* **9. Версия документа.** v1.0 от 26.08.2025.

(Могу сразу набросать полный текст — скажи, если нужно включить сюда.)

**5) База данных / миграции**

**Новые/изменённые поля**

* Таблица **users/profiles** (куда у тебя сейчас сохраняются профильные данные):
  + weight\_kg NUMERIC(5,2) NOT NULL
  + allergies TEXT[] NOT NULL DEFAULT '{}'::text[]
  + consent\_privacy\_version TEXT, consent\_privacy\_at TIMESTAMPTZ
  + consent\_disclaimer\_version TEXT, consent\_disclaimer\_at TIMESTAMPTZ
  + tz TEXT DEFAULT 'Europe/Moscow' (или хранить реальный TZ пользователя)
  + chat\_id BIGINT (для напоминаний)
* Таблица **plans** (добавить логику доз):
  + dose\_mode TEXT NOT NULL DEFAULT 'opt' -- 'min'|'opt'|'max'
  + dose\_per\_kg\_min SMALLINT NOT NULL DEFAULT 100
  + dose\_per\_kg\_opt SMALLINT NOT NULL DEFAULT 140
  + dose\_per\_kg\_max SMALLINT NOT NULL DEFAULT 200
  + remind\_enabled BOOL NOT NULL DEFAULT FALSE
  + remind\_time TIME WITH TIME ZONE
* Таблица **intakes** (если ещё нет): журнал приёмов
* CREATE TABLE IF NOT EXISTS derma.intakes(
* id BIGSERIAL PRIMARY KEY,
* patient\_id BIGINT NOT NULL REFERENCES derma.patients(id) ON DELETE CASCADE,
* taken\_at DATE NOT NULL,
* mg INTEGER NOT NULL CHECK (mg > 0),
* created\_at TIMESTAMPTZ DEFAULT now()
* );
* CREATE UNIQUE INDEX IF NOT EXISTS intakes\_unique\_day
* ON derma.intakes(patient\_id, taken\_at);
* Вьюха для прогресса (опционально):
* CREATE OR REPLACE VIEW derma.v\_progress AS
* SELECT p.patient\_id,
* COALESCE(SUM(i.mg),0) AS taken\_mg
* FROM derma.plans p
* LEFT JOIN derma.intakes i USING (patient\_id)
* GROUP BY 1;

**Сброс данных пользователя**

-- в транзакции

DELETE FROM derma.intakes WHERE patient\_id = $1;

UPDATE derma.plans

SET dose\_mode='opt', remind\_enabled=FALSE, remind\_time=NULL

WHERE patient\_id=$1;

-- профиль не удаляем

**6) Эндпоинты (итог)**

* GET /api/progress?mode=...
* POST /api/intakes {date, mg}
* POST /api/dose-mode {mode}
* POST /api/reminder {enabled, time, tz}
* POST /api/reset
* POST /api/consent {privacy\_version, disclaimer\_version}

**7) Мини-примеры фронта**

**Нижняя навигация**

<nav class="bottom-nav">

<button data-route="main">Главная</button>

<button data-route="calendar">Календарь</button>

<button data-route="profile">Профиль</button>

<button data-route="labs">Анализы</button>

</nav>

document.querySelectorAll('.bottom-nav [data-route]')

.forEach(b => b.onclick = () => app.navigate(b.dataset.route));

**Валидация профиля**

saveBtn.disabled = !(weightValid() && allergiesValid());

**8) Definition of Done (кратко)**

* Bottom-nav на всех страницах, корректный роутинг.
* Страница «Обнуление данных» с двойным подтверждением; POST /api/reset работает.
* «Прогресс»: три кольца (одно большое + два малых), переключение режимов, корректные проценты.
* Напоминания: включение/выключение, выбор времени, отправка тестового; ежедневные пуши через бота.
* Профиль: нельзя продолжить без веса и аллергий; серверная и клиентская валидация.
* Страница «Политика безопасности и отказ от ответственности» + запись согласия (версия/время).
* Миграции применены; существующие пользователи не ломаются.

Если хочешь, могу сразу выдать готовые тексты для «Политики…» и короткие Express-хендлеры для /api/reset, /api/reminder, /api/intakes.