Пакет документов для этапов 0–1 (без кода)

Ниже — готовые черновики файлов каталога `docs/`, разложенные по 12 холдам. Можно копировать как есть в репозиторий — это текстовые спецификации без кода. При необходимости я вынесу каждый файл в отдельный документ.

File: docs/CONTRIBUTING.md

Цель: единые правила ведения репозитория и качества.

1) Ветвление

- `main` — стабильные релизы.
- `develop` — интеграция фич перед релизами.
- `feature/<кратко-о-фиче>` — работа над задачей.
- `hotfix/<issue-номер>` — срочные исправления.

2) Коммиты (Conventional Commits)

- `feat:` новая функциональность
- `fix:` исправление багов
- `docs:` документация
- `chore:` сервисные изменения
- `refactor:` без изменения поведения
- `perf:` производительность
- `test:` тесты
- `revert:` откат

3) Pull Request (PR)

- Заполнить шаблон: цель, изменения, риски, как тестировать.
- Линк на issue и метки (labels).
- Обязателен 1+ ревьюер.
- CI «зелёный», линтеры/форматирование пройдены.

4) Качество и стиль

- Единые правила в ESLint/Prettier (описаны в `docs/architecture.md`).
- Не добавлять секреты и ключи в репозиторий.

File: docs/architecture.md

Цель: зафиксировать архитектурные договорённости фронта и интеграции с Supabase.

1) Технологии и окружения

- Клиент: React (TypeScript), сборка: Vite.
- Окружения: `dev`, `stage`, `prod`.
- `.env*`: клиентские переменные только с префиксом `VITE_` (примерные ключи см. в `.env.example`).

2) Архитектурные принципы

- **Feature-based**: `src/features/<feature>` + `src/shared/{ui,api,types,hooks,lib}`.
- Импорт «вверх по фичам» запрещён; общее — только из `shared`.
- Минимум глобального состояния; локальный state или специализированные сторы per-фича.

3) Роутинг (MVP)

- `/` — витрина слотов (карточки игр, фильтры — позже).
- `/profile` — профиль пользователя (почта/телефон/пароль, баланс — заглушка).
- `/admin` — заглушка (будущее: роли/права/аудит/настройки).

4) Данные и сетевой слой

- Единая обёртка Supabase SDK в `shared/api`.
- Ошибки: единый обработчик (уведомления/логгирование).
- Кэширование запросов — позже (по мере появления реальных данных).

5) Производительность и доступность

- Code-splitting по маршрутам, lazy components.
- A11y как стандарт: семантика, фокус-стили, контраст, альтернативный текст.

File: docs/ui-guidelines.md

Цель: единый визуальный язык и минимальный набор примитивов.

1) Дизайн-токены (черновик)

- Цвета: базовая нейтральная палитра (тёмный текст, светлый фон), 1 акцент.
- Тени: мягкие, без «жёстких» границ.
- Радиусы: большие (например, 12–16) для карточек/кнопок.
- Отступы: шкала 4/8/12/16/24/32.
- Типографика: читабельные размеры (заголовки/тело/подписи), единый ритм.

2) Примитивы UI (тонкий слой)

- **Button**, **Card**, **Input**, **Label**, **Heading**, **Skeleton**.
- Состояния: default/hover/active/disabled/loading; размеры: sm/md/lg.

3) Паттерны (MVP)

- Витрина слотов: сетка карточек, карточка = обложка, название, метки.

- Профиль: 2-колоночный layout (слева настройки, справа детали/баланс).
- Desktop-first; мобильная адаптация — после MVP.

File: docs/security.md

Цель: базовые правила безопасности и приватности данных.

1) Аутентификация и токены

- Сессии по умолчанию через Supabase Auth.
- **Access** короткоживущий + **Refresh** для продления; не хранить чувствительные данные в JWT.
- Переход на cookie-based (HttpOnly, Secure, SameSite) — в будущих итерациях.

2) Ключи и секреты

- `anon`-ключ используется только на клиенте.
- `service_role` — только сервер-сайд (никогда не в клиенте/репозитории).

3) RLS «с первого дня»

- Для всех пользовательских таблиц: включить RLS и явные `policy`.
- Политики проверяют идентичность по user id, роли и владению записью.

4) Защита приложения

- Минимизировать XSS: экранирование, безопасные рендеры, отсутствие инлайновых опасных вставок.
- Rate limiting/anti-abuse — для чувствительных действий (логин, бонусы).

5) Аудит и хранение данных

- `audit_logs`: кто/что/когда (аутентификация, профиль, операции с балансом — позже).
- Политика хранения и удаления данных (retention) — будет определена для рынков.

File: docs/rbac.md

Цель: определить роли и права (RBAC) для MVP и расширения.

1) Роли (черновой состав)

- **Owner** — полный доступ, стратегические настройки.
- **Admin** — управление пользователями/ролями/играми/промо, просмотр аудита.
- **Moderator** — поддержка пользователей (просмотр профилей, ограниченные действия).
- **User** — игрок.

2) Группы прав (пример)

- `user_mgmt` (просмотр/блокировка/сброс 2FA)

- `role_mgmt` (создание/назначение ролей) — только Admin/Owner
- `slots_mgmt` (каталог игр, параметры)
- `promo_mgmt` (промокоды/реферальные правила)
- `finance_view` (просмотр транзакций) / `finance_approve` (подтверждение выводов) — только Admin/Owner
- `settings_write` (системные параметры)
- `audit_view` (логи)

3) Маппинг ролей → прав (MVP-минимум)

- Owner: всё
- Admin: всё кроме «Owner-только»
- Moderator: `user_mgmt` (ограниченно), `audit_view`
- User: нет административных прав

File: docs/product-brief.md

Цель: сфокусировать MVP и определить рамки.

1) Видение

Масштабируемая платформа слотов с минималистичным UI, честными механиками и основой для дальнейших интеграций.

2) Целевая аудитория (черновик)

Взрослые пользователи, интересующиеся слотами (региональные и юридические ограничения будут уточнены позже).

3) Ценность

Простая и понятная витрина игр, удобный профиль, быстрый старт; фундамент для дальнейшего роста.

4) Допущения/ограничения

- Desktop-first; мобильные — позже.
- Простые механики слотов на старте.
- Платежи и KYC — после MVP.

5) MVP-объём

- Регистрация/логин (email/телефон), профиль (смена пароля/почты/телефона), баланс (заглушка), витрина слотов, карточка игры (заглушка).

6) Успех (черновые метрики)

- Завершённые регистрации/доля входов
- Просмотры карточек игр/переходы к запуску
- Базовое удержание (возврат через 7 дней)

File: docs/compliance.md

Цель: ориентиры честности, приватности и ответственной игры.

1) Честность игр

- Принципы RNG/сертификации — план на будущие релизы.
- Методики контроля RTP и тестовые процедуры.

2) Ответственная игра

- Самоисключение, лимиты, напоминания о сессиях — план внедрения.

3) Приватность/данные

- GDPR-подход: минимизация данных, права пользователя, хранение и удаление.

4) Платежи (будущее)

- Соответствие PCI DSS при работе с PSP.

File: docs/data-model.md

Цель: перечислить сущности и доступ к ним на уровне идей (без DDL).

Сущности MVP и рядом

- `users` — учётная запись; **чтение**: владелец/админ; **запись**: владелец (свой профиль)/ админ.
- `roles`, `permissions`, `user_roles` — роли/права; **чтение**: админ/модератор; **запись**: админ.
- `audit_logs` — логи действий; **чтение**: админ/модератор; **запись**: система.
- `games` — каталог слотов (метаданные); **чтение**: все; **запись**: админ.
- `game_rounds` — результаты раундов; **чтение**: владелец записи/админ; **запись**: система.
- `bets` — ставки; **чтение**: владелец/админ; **запись**: система/владелец в момент создания.
- `promos` — промокоды; **чтение**: все; **запись**: админ.
- `referrals` — связи «кто кого пригласил»; **чтение**: владелец/админ; **запись**: система.
- (позже) `wallets`, `transactions` — кошельки и операции; **чтение**: владелец/админ; **запись**: система/админ.

Принципы RLS в словах

- Фильтрация по `auth.uid()` для пользовательских записей.
- Разрешение администраторов по роли/флагу.
- Сервисные операции — через сервер-сайд ключ и отдельные политики.

File: docs/mvp-checklist.md

Цель: объективно зафиксировать «MVP готов».

Чек-лист

- Репозиторий настроен (ветки, PR, линтеры, правила коммитов).
- Каркас клиента и маршруты `/`, `/profile`, `/admin` (заглушка) описаны в архитектуре.
- В Supabase создан проект, включён Auth; `.env.example` заполнен ключами и URL (без секретов).
- RLS включён для таблиц MVP; базовые политики описаны.
- Профиль: смена пароля, почта/телефон — сценарии описаны.
- Витрина слотов: карточки игр — сценарии описаны; одна демонстрационная игра — как заглушка в требованиях.
- Аудит ключевых действий определён (события, кто пишет/читает).

File: docs/holds.md

Цель: дорожная карта «холдов» и критерии готовности.

Hold 1 — Репозиторий и правила

- **Цель:** единый каркас и дисциплина коммитов.
- **Что сделать:** ветки, правила PR, линтеры, конвенции.
- **Готово когда:** PR проходит CI, есть `CONTRIBUTING.md`.

Hold 2 — Каркас сборки/окружения

- **Цель:** Vite+TS, `.env`, префикс `VITE_`.
- **Что сделать:** зафиксировать в `architecture.md`.
- **Готово когда:** документ утверждён.

Hold 3 — Архитектура фич и роутинг

- **Цель:** структура `features/`, `shared/`, маршруты.
- **Что сделать:** схема в `architecture.md`.
- **Готово когда:** схема согласована.

Hold 4 — UI-примитивы и токены

- **Цель:** единый вид, минимум компонентов.
- **Что сделать:** правила в `ui-guidelines.md`.
- **Готово когда:** токены и примитивы описаны.

Hold 5 — Supabase проект

- **Цель:** готовность к Auth/БД.
- **Что сделать:** шаги в `security.md` и `.env.example`.
- **Готово когда:** подход к ключам и токенам зафиксирован.

Hold 6 — RLS с первого дня

- **Цель:** защита данных политиками.
- **Что сделать:** чек-лист включения RLS и policy.
- **Готово когда:** список правил согласован.

Hold 7 — RBAC и аудит

- **Цель:** роли/права и события логирования.
- **Что сделать:** `rbac.md`, события в `security.md` / `data-model.md`.
- **Готово когда:** маппинг ролей и событий утверждён.

Hold 8 — Профиль и баланс (MVP)

- **Цель:** сценарии пользователя.
- **Что сделать:** флоу в `product-brief.md`.
- **Готово когда:** сценарии согласованы.

Hold 9 — Безопасность токенов

- **Цель:** минимизация рисков XSS/угонов.
- **Что сделать:** правила JWT/хранения в `security.md`.
- **Готово когда:** принципы утверждены.

Hold 10 — Честность/соответствие

- **Цель:** план требований (RNG/RTP/RG).
- **Что сделать:** `compliance.md` (черновик требований/процедур).
- **Готово когда:** чек-лист готов.

Hold 11 — План данных

- **Цель:** перечень сущностей/операций.
- **Что сделать:** `data-model.md`.
- **Готово когда:** роли/операции согласованы.

Hold 12 — Итоговый чек-лист MVP

- **Цель:** единая точка «готово».
- **Что сделать:** `mvp-checklist.md`.
- **Готово когда:** все пункты отмечены.

> Готов дорабатывать любой файл: добавлять разделы, примеры UI-паттернов, диаграммы процессов и уточнения по рынкам/комплаенсу — скажи, какие блоки приоритетны.