**Результаты и демонстрация по этапам**

**Этап 1 — Архитектура и DevOps (VDS)**  
**Результат:**

* Развёрнут VDS, настроены Docker/Docker Compose, reverse-proxy (nginx), логирование.
* Выпущены и автопродлеваются HTTPS-сертификаты (Let’s Encrypt/Certbot).
* Базовый IaC-набор (Terraform/описание) для воспроизводимого окружения: сеть, DNS/записи, хост.
* CI/CD для фронта и бэка (автосборка, автодеплой в staging/prod).  
  **Демонстрация/приёмка:** доступ к серверу и репозиторию, прогон пайплайна «commit -> сборка -> деплой», проверка автопродления сертификата и доступности сайта по HTTPS. (Let’s Encrypt/Certbot; Terraform/IaC — как бенчмарк практики.)

**Этап 2 — Backend (Strapi, моделирование)**  
**Результат:**

* Смоделированы типы контента (категории/товары/атрибуты/медиа).
* Включены публичные API: REST (+фильтры/пагинация) и/или GraphQL, настроены роли/права.
* Черновая документация по схемам и эндпоинтам.  
  **Демонстрация/приёмка:** Postman-коллекция/Swagger для REST и интроспекция/примеры для GraphQL; показываем реальные ответы API из staging.

**Этап 3 — Импорт Oasis + медиа + ISR/реиндекс**  
**Результат:**

* Скрипты импорта каталога/медиа, лог импорта с отчётом об ошибках.
* Настроена регенерация страниц без полного билда (Next.js ISR), фоновая переиндексация.  
  **Демонстрация/приёмка:** запускаем импорт, на глазах появляются новые позиции; меняем запись и показываем, что страница обновилась без полного «next build».

**Этап 4 — Поиск (Meilisearch)**  
**Результат:**

* Развёрнут Meilisearch, созданы индексы, настроены атрибуты для сортировок и фасетов.
* Включены facet-search и опечатки (typo tolerance) с заданными порогами.  
  **Демонстрация/приёмка:** интерактивные примеры: запрос с 1–2 опечатками, проверка счётчиков фасетов и facet-search; бенчмарк нескольких запросов.

**Этап 5 — Frontend (Next.js) — разбит на 8 подэтапов**  
5.1 Архитектура/layout/мета/OG/Schema.org — базовый каркас, head-теги, shared-компоненты.  
5.2 Главная — сборка блоков, баннер/листы.  
5.3 PLP/категории + фильтры/пагинация/сортировки — связка с поиском/фасетами.  
5.4 PDP (модель/вариация) + галерея/атрибуты — карточка товара.  
5.5 Поиск/подсказки/404/empty-states — UX сценарии.  
5.6 Статические страницы (оферта/политика/контакты).  
5.7 Портфолио.  
5.8 Оптимизация изображений (next/image, CDN), базовая web-perf.  
**Демонстрация/приёмка:** на каждый подпункт — ссылка на staging, чек-лист сценариев и короткий реторн-видео/скринкаст; для страниц каталога/товаров дополнительно показываем, что обновления подтягиваются через ISR.

**Этап 6 — Корзина и оформление (без онлайн-оплаты)**  
**Результат:**

* Полный флоу: добавление в корзину, адрес/контакты, подтверждение, создание заказа в бекенде, сервисные письма.
* Проброс основных e-commerce событий (view\_item, add\_to\_cart, begin\_checkout, purchase-stub) в GA4.  
  **Демонстрация/приёмка:** создаём тест-заказ, показываем запись в админке/БД и доставку писем; в GA4 DebugView видно события.

**Этап 7 — Аналитика/правовые**  
**Результат:**

* GA4/Tag Manager настроены, события e-commerce подключены (минимум: просмотры, добавления, чекаут).
* SEO-блок: Product structured data (JSON-LD), каноникал/мета, базовая карта сайта.
* Страницы оферты/политики/контактов.  
  **Демонстрация/приёмка:** валидаторы (GA4 DebugView/Realtime, Rich Results Test), просмотр JSON-LD в исходнике, проверка sitemap и robots.

**Этап 8 — Тестирование**  
**Результат:**

* Чек-листы и отчёт UAT (pass/fail по сценариям), заведённые баг-репорты и фикс-лог.  
  **Демонстрация/приёмка:** предоставляем чек-листы, прогоняем «живые» сценарии; критерии приёмки оформлены как AC/DoD (прозрачные pass/fail).

**Этап 9 — Запуск и стабилизация**  
**Результат:**

* Прод-релиз, проверка доменов/сертов/редиректов, дымовые тесты, мониторинг.
* Сводный отчёт по инцидентам (если были) и их устранению.  
  **Демонстрация/приёмка:** чек-лист запуска, доступ к мониторингу/логам, контрольные метрики после релиза (доступность, скорость ответа). (HTTPS и инфраструктурные практики — как на Этапе 1.)