

# Техническое задание (ТЗ)

---

## 1. Общая информация о проекте

**Название проекта:** (условно) «On-line/Off-line школа 3D-дизайна»

**Краткое описание:**

Веб-приложение, позволяющее:

1. Осуществлять запись на групповые и индивидуальные курсы (онлайн и офлайн).
2. Приобретать видео-курсы и просматривать их на сайте без возможности скачивания.
3. Хранить и просматривать библиотеку ресурсов (текстуры, HDRI, 3D-модели, видео-уроки).
4. Публиковать работы пользователей (портфолио), с возможностью оценки, комментариев, лайков/дизлайков комментариев.
5. Осуществлять поиск по всему контенту сайта.
6. Управлять пользователями и контентом (админ-панель).

**Цели проекта:**

- Упростить процесс записи на курсы и покупки видеоматериалов без участия менеджера (полноценное online-оформление).
- Обеспечить удобный просмотр обучающих видео прямо на сайте, без скачивания.
- Создать сообщество для обмена опытом, комментариями и оценками проектов.
- Обеспечить гибкую и удобную систему управления контентом и данными через headless CMS (Strapi) и/или Supabase.

---

## 2. Требования к функционалу

### 2.1 Пользовательские роли и доступ

#### 1. Гость (неавторизованный пользователь)

- Может просматривать общедоступные разделы сайта (главная страница, информация о курсах, портфолио пользователей, часть ресурсов и т. д.).
- Может искать по сайту, но не может скачивать платный контент или смотреть купленные видео-уроки.
- Может просматривать комментарии и рейтинги работ.
- Для покупки/записи на курсы требуется регистрация и авторизация.

#### 2. Зарегистрированный пользователь

- Имеет личный профиль (аватар, имя, фамилия, дата рождения, контакты и т. п.).
- Может редактировать свою личную информацию.
- Может покупать видео-курсы, бронировать место в групповых курсах или записываться на индивидуальные курсы.
- Может участвовать в комментировании чужих работ, ставить оценки (1–5) и лайки/дизлайки комментариям.

- Может загружать собственные проекты (изображения, описание, итоговые 3D-сцены и т. п.) в свою галерею.
- Может получать подтверждающие документы (чек, электронная квитанция) после оплаты.

### 3. Администратор (или Модератор)

- Имеет доступ к админ-панели для управления пользователями, курсами, видео-курсами, ценами, библиотекой ресурсов, публикуемыми работами (утверждение/снятие, модерирование комментариев, удаление спама и т. д.).
  - Может управлять контентом в Strapi (если используется связка Next.js + Strapi) или в Supabase (если там организована БД и часть бизнес-логики).
- 

## 2.2 Регистрация и аутентификация

### 1. Регистрация:

- Поля: имя, фамилия, email, пароль, дата рождения (не обязательно сразу, но может быть запрошена на этапе редактирования профиля), Instagram (необязательно), прочие поля по необходимости.
- Проверка на уникальность email, валидация полей.
- Опционально: подтверждение email (отправка ссылки на активацию).

### 2. Авторизация:

- По email и паролю.
- Возможность сброса пароля через email.
- Использование JWT или другой механизм (Supabase Auth / Strapi Auth и т. д.).

### 3. Личный кабинет:

- Отображение личных данных пользователя (имя, фамилия, email, аватар и т. д.).
  - Возможность редактировать профиль.
  - История покупок (курсы/видео-курсы).
  - Приобретённые видеокурсы с прямой ссылкой на просмотр.
- 

## 2.3 Функционал курсов

### 1. Онлайн-курс

- Название: «red event (online)»
- Количество учеников в группе: 12
- Длительность: 4 месяца, занятия через день по 2 часа
- Возрастное ограничение: 16–35 лет
- Цена: 120 000 Р
- Механизм записи/бронирования места:
  - Пользователь кликает «записаться», выбирает удобную дату (или видит уведомление, что курс начнётся по факту набора 12 человек).
  - Оплата (полная или частичная, по выбору).

- После оплаты генерируется чек/квитанция, пользователь получает уведомление в личном кабинете и/или на почту.

## 2. Офлайн-курс

- Название: «red event (off-line)»
- Количество учеников в группе: 12
- Длительность: 4 месяца, занятия через день по 2 часа
- Возрастное ограничение: 16–45 лет
- Цена: 100 000 ₽
- Механизм записи:
  - Аналогичен онлайн-курсу (запись, оплата, чек).

## 3. Индивидуальное обучение

- 1 ученик + преподаватель, 2 месяца, занятия каждый день по 2 часа
- Цена: 170 000 ₽
- Возраст: 16–40 лет
- Механизм записи:
  - Пользователь выбирает даты, оплачивает.
  - «Обучение начнётся после оплаты, по мере готовности».

## 4. Динамическое управление:

- Все данные по курсам (цена, расписание, количество мест) должны храниться и редактироваться через админку (Strapi или Supabase).
- 

## 2.4 Функционал видео-курсов

### 1. Список видео-курсов:

- «red event» (6 ч, 1280×1080, звук) – 40 000 ₽
- «3Ds Max» (4 ч, 1280×1080) – 20 000 ₽
- «Revit» (2 ч, 1280×1080) – 20 000 ₽
- Возможность добавлять/редактировать курсы через админку.

### 2. Процесс покупки:

- Пользователь выбирает видео-курс, добавляет в корзину (или сразу «Купить»).
- Переходит к оформлению, производит оплату (интеграция с платёжным шлюзом).
- После успешной оплаты:
  - Должен формироваться чек (PDF или электронная квитанция).
  - В личном кабинете появляется доступ к курсу для **онлайн-просмотра** (без скачивания).

### 3. Просмотр видео:

- Встроенный плеер в приложении.
- Запись хранится на защищённом хранилище (Supabase Storage / облачное хранилище с доступом по токенам).

- Проверка доступа у пользователя на клиенте или на сервере (чтобы не могли смотреть те, кто не купил).
- 

## 2.5 Библиотека ресурсов

### 1. Контент:

- Текстуры
- HDRI
- 3D-модели
- Видео-уроки (бесплатные уроки или демо)

### 2. Поисковая строка:

- Поиск по названию, ключевым словам, типу ресурса.
- Возможность сортировки (самые новые, популярные, по рейтингу).

### 3. Ограничение доступа (опционально):

- Часть контента может быть бесплатной и доступной всем.
- Часть – доступна только зарегистрированным пользователям или покупателям курсов.

### 4. Загрузка нового контента администратором:

- Через админ-панель (Strapi / Supabase) с указанием названия, описания, категории, загрузкой файлов и т. д.
- 

## 2.6 Галерея (портфолио) пользователей

### 1. Профиль пользователя:

- Аватар
- Личные данные
- Список загруженных работ (превью-картинки, заголовок, краткое описание, при клике – просмотр полной работы).

### 2. Просмотр проекта:

- Картинка обложки + дополнительные изображения (или 3D-модели, если реализована превью).
- Оценка работы (1–5).
- Комментарии (другие пользователи могут комментировать).
- Лайк/дизлайк на комментарии других пользователей.

### 3. Механика загрузки:

- Зарегистрированный пользователь может загрузить проект, заполнить поля (название, описание, изображения/файлы).
- После проверки администратором работа становится доступна на сайте (или автоматически без модерации – решается на уровне политики проекта).

---

## 2.7 Система рейтинга и комментариев

### 1. Рейтинг:

- Возможность оценивать работу (1–5 звёзд).
- Подсчёт средней оценки.

### 2. Комментарии:

- Добавление комментария под работой/видеокурсом/ресурсом.
- Возможность лайкнуть/дизлайкнуть чужой комментарий.
- При большом количестве комментариев – сортировка (по дате, по количеству лайков).

### 3. Модерация:

- В админ-панели администратор видит все жалобы (если реализован функционал «пожаловаться») или может удалять негативный контент.
- 

## 2.8 Поиск по всему сайту

### 1. Поисковая строка (на видном месте):

- Поиск по курсам, видео-курсам, ресурсам, проектам пользователей, имени/фамилии пользователя и т. д.

### 2. Фильтры:

- Категория (3D-модели, курсы, видео-курсы, текстуры, автор и т. д.).
- Сортировка (по дате добавления, по рейтингу).

### 3. Реализация:

- Если используется Supabase, можно организовать полнотекстовый поиск по PostgreSQL.
  - Если используется Strapi, использовать встроенные возможности поиска или сторонние плагины.
- 

## 3. Выбор технологического стека

### 3.1 Next.js

#### • Frontend:

- React + TypeScript (Next.js) для реализации клиентской части.
- Роутинг: страницы курсов, страницы пользователя, страница видео-плеера, админ-панель (если часть админки делается в рамках приложения; хотя админку чаще ведут отдельно в Strapi).

#### • UI-библиотека:

- **ShadCN** (коллекция компонентов на базе Tailwind CSS) для унифицированного стиля и быстрого прототипирования.

## 3.2 Headless CMS / Backend

### 1. Supabase:

- Хранение данных (PostgreSQL).
- Авторизация (Supabase Auth).
- Хранение файлов (Supabase Storage) – для пользовательских изображений, превью, видео.
- Роль сервера для API-запросов (например, отправка и проверка платежей, формирование чеков).

### 2. Strapi:

- Альтернативная или дополнительная система, если хочется удобной CMS с UI для менеджмента контента (курсы, видео-курсы, библиотека).
- Можно подключить Strapi к той же базе PostgreSQL.
- Даёт удобный интерфейс для администраторов и контент-менеджеров.

*(Обычно выбирают что-то одно: либо Supabase как полный бекенд, либо Strapi для контента + любую базу. Если проект крупный, можно совместить: Supabase для аутентификации, Storage, а Strapi – для управления контентом. Но при старте MVP можно ограничиться чем-то одним.)*

## 3.3 Платёжная интеграция

- Использовать любой доступный провайдер для региона (Яндекс Касса / CloudPayments / Stripe / PayPal и т. д.).
- При оплате формировать заказ и привязывать статус оплаты в базе (Supabase или Strapi).

---

## 4. Структура приложения и основные модули

### 1. Главная страница:

- Промо информации, ссылки на основные разделы (курсы, видео-курсы, ресурсы, галерея, «о нас» и т. д.).

### 2. Страница «Курсы»:

- Список доступных курсов (online, offline, индивидуальные).
- Клик по курсу => детальная страница курса с описанием, ценой, кнопкой «Записаться»/«Бронь».

### 3. Страница «Видео-курсы»:

- Перечень видео-курсов, описание, цена, кнопка «Купить».
- При покупке – переход к оформлению.
- При оплате – доступ к странице просмотра.

### 4. Страница «Ресурсы»:

- Список/каталог (текстуры, HDRI, 3D-модели, видео-уроки).
- Поиск, фильтры.

#### 5. Страница «Галерея» (работ пользователей):

- Список работ (миниатюры).
- Возможность перейти в подробности, посмотреть оценки и комментарии.

#### 6. Личный кабинет:

- Профиль, редактирование данных.
- История заказов.
- Мои курсы и видео-курсы (доступ для просмотра приобретённых).
- Загрузка работ в галерею.

#### 7. Страница «О нас» (просто информационная).

#### 8. Админ-панель:

- Управление пользователями (роль, бан, удаление).
- Управление курсами и видео-курсами (создание, редактирование, цены, описания).
- Управление контентом (ресурсы, галерея пользователей – одобрить/удалить).
- Просмотр статистики (количество продаж, записи на курс и т. д.).
- Может быть реализована средствами Strapi (если выбрана эта CMS) или кастомно на Next.js + Supabase.

---

## 5. Требования к дизайну и UX

- Использование Tailwind CSS + ShadCN для быстрой стилизации.
- Адаптивная верстка (мобильная, планшет, десктоп).
- Акцент на простоте взаимодействия при покупке/записи (максимум 2–3 шага до оплаты).
- Удобная и быстрая система поиска и фильтрации.
- Минимальное количество всплывающих окон; стремиться к ясной навигации (многоуровневое меню или «бургер»-меню на мобильных устройствах).

---

## 6. Интеграции и дополнительные сервисы

#### 1. Email-рассылки:

- Подтверждение регистрации, уведомления о покупке, сброс пароля, напоминания о старте курса.

#### 2. СМС-уведомления (опционально):

- Подтверждение бронирования, оповещение о старте курса.

#### 3. Соцсети:

- Интеграция с Instagram (если нужно тянуть аватар/посты).

#### 4. Аналитика:

- Подключение сервисов (Google Analytics, Яндекс.Метрика и т. п.).

#### 5. Уведомления внутри сайта (уведомление о комментариях, лайках и т. п.) – опционально.

---

## 7. Требования к производительности и безопасности

#### 1. Производительность:

- Серверный рендеринг Next.js (оптимизирует SEO и скорость загрузки).
- Кеширование запросов (SWR или React Query) при повторных заходах.

#### 2. Безопасность:

- Хранение паролей в зашифрованном виде (если не используется встроенный Auth провайдер).
  - HTTPS-соединения.
  - Защита от XSS, CSRF (встроенные инструменты Next.js + валидация на стороне сервера).
  - Ограничение прав доступа по ролям (admin, user, guest).
- 

## 8. План разработки (основные этапы)

#### 1. Анализ и детализация требований

- Финализировать структуру базы данных (курсы, пользователи, видео-курсы, заказы, комментарии и т. д.).
- Решить, используем ли одновременно Strapi и Supabase, или что-то одно.

#### 2. Прототипирование UI

- Проработать страницы в Figma (или другом инструменте), основываясь на макете из PDF.

#### 3. Настройка окружения

- Инициализация Next.js-проекта (TypeScript, ShadCN).
- Настройка Supabase (Auth, Storage, DB) или Strapi (модели данных).

#### 4. Реализация основных пользовательских сценариев

- Регистрация/Авторизация.
- Просмотр курсов и запись.
- Покупка видео-курсов и просмотр.
- Галерея/портфолио (загрузка, комментарии, рейтинги).
- Библиотека ресурсов.

#### 5. Интеграция платежной системы

- Модуль оплаты, формирование чека.



## 6. Админ-панель

- Управление контентом, модерация, изменение цен.

## 7. Тестирование

- Тест кейсы: функциональные, нагрузочные, UX/UI.

## 8. Развёртывание и поддержка

- Хостинг (Vercel, Netlify или другой для фронтенда Next.js).
- Серверная часть (Strapi – на собственном сервере/VPS или Supabase в облаке).

---

# 9. Риски и возможные улучшения

### 1. Выбор между Supabase и Strapi

- Если проект небольшой, можно обойтись одним решением. Две системы одновременно усложняют архитектуру, но дают большую гибкость.

### 2. Правовая сторона оплаты

- Нужно уточнить, какие методы оплаты легальны для вашего региона, как формировать корректные чеки (ККТ в РФ).

### 3. Контент и авторские права

- При загрузке работ пользователями нужно соглашение (пользовательское соглашение, политика конфиденциальности).

### 4. Масштабируемость

- Возможность вынести heavy-контент (большие файлы, видео) на CDN или внешние сторедж-сервисы для снижения нагрузки.

---

# 10. Заключение

Данный документ описывает структуру, ключевые функции и предполагаемый технический стек будущего веб-приложения. В ходе проекта могут вноситься изменения и дополнения, однако основные модули и логика работы останутся неизменными. Рекомендуется на старте проекта согласовать и детализировать схему базы данных, а также уточнить, кто будет заниматься админкой и контент-менеджментом (через Strapi или через кастомный интерфейс + Supabase).

**Основной результат:** приложение, которое позволит:

- Полностью автоматизировать запись на курсы (онлайн/офлайн/индивидуальные).
- Продавать видео-уроки с онлайн-просмотром.
- Вести библиотеку бесплатных и платных ресурсов.
- Создать сообщество 3D-дизайнеров с возможностью публиковать и оценивать работы.

Это ТЗ может служить базовой отправной точкой для планирования задач и распределения обязанностей среди команды разработчиков, дизайнеров и контент-менеджеров.