

## ЛР 4 REACT и NODE.JS

### Тема:

1. Библиотека для создания пользовательских интерфейсов React;
2. Веб-фреймворк Express (Node.js/JavaScript)

### Цель работы:

Закрепить знания и навыки по работе с библиотекой JavaScript для создания пользовательских интерфейсов React, фреймворком Express (Node.js).

Пример: <https://www.bezkoder.com/react-node-express-mongodb-mern-stack/>

### Справочная информация по React:

Документация: <https://ru.react.js.org>, Начало работы – React

[Руководство по React \(metanit.com\)](#)

[Getting started with React - Learn web development | MDN \(mozilla.org\)](#)

[Быстрый старт React с примерами кода \(reactdev.ru\)](#)

[React Reference Overview – React](#)

[Учебник. React для начинающих | Microsoft Learn](#)

### Справочная информация по Node.js/ Express:

[https://developer.mozilla.org/ru/docs/Learn/Server-side/Express\\_Nodejs](https://developer.mozilla.org/ru/docs/Learn/Server-side/Express_Nodejs)

[Node.js | Введение и начало работы \(metanit.com\)](#)    [Node.js | Начало работы с Express \(metanit.com\)](#)

[Guides | Node.js \(nodejs.org\)](#)

[Полное руководство по Node.js — Изучите Node для начинающих ↗ Node.js с примерами кода \(nodejsdev.ru\)](#)

[Полный справочник по Node.js \(questu.ru\)](#)

### Задание.

Разработать веб-сайт с использованием Node.js(Express) для реализации сервера и React для клиента.

## Постановка задачи:

- Определить и реализовать **минимум четыре модели** в соответствии с предметной областью предыдущих ЛР (использовать встроенные валидаторы, подходящие типы данных). В качестве базы данных всем использовать **MongoDB** (<https://metanit.com/web/nodejs/6.1.php>, [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\\_Nodejs/mongoose](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose) ).
- Реализовать механизмы авторизации/аутентификации с помощью логина/пароля ([Аутентификация Node.js с примерами кода \(nodejsdev.ru\)](#)). На отметку от 7 баллов - через аккаунты Google или любой другой доступный;
- Для авторизованного юзера реализовать CRUD (create, read, update, delete) операции и возможность поиска, сортировки записей [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\\_Nodejs/mongoose](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose); Использовать формы, созданные с использованием React, для добавления записей. <https://metanit.com/web/react/3.3.php>
- Для неавторизованного пользователя обеспечить только просмотр, поиск, сортировку и авторизацию;
- Обеспечить наполнение данными для демонстрации не менее 10 записей в списке товаров/услуг/объектов/клиентов.
- **отображать** тайм зону пользователя, текущую дату, дату добавления/изменения данных в таблицы для тайм зоны пользователя и для UTC.
- Для создания всех страниц (**минимум четыре – выбрать любые из предыдущих ЛР**(среди страниц обязательно должен быть каталог, просмотр инфо об объекте из каталога), **переключение между ними с помощью панели навигации**) использовать только React.
- Создать компоненты (**см. индивидуальное задание по вариантам**)
  - с использованием декларативных функций, стрелочных функций и классов.
  - Применить props, значения по умолчанию.
  - Использовать разработанные компоненты внутри других компонентов.
  - Добавить обработчики событий (минимум семь) – в функциональных компонентах, компонентах классах, с передачей параметров в обработчик события. <https://metanit.com/web/react/2.5.php>

- Продемонстрировать работу со state (<https://metanit.com/web/react/2.4.php>)
- Продемонстрировать использование любого из хуков.

### **React компоненты (минимальные требования для всех):**

1. Функциональный компонент с декларативной функцией - для основной логики
2. Функциональный компонент со стрелочной функцией - для презентационных компонентов
3. Классовый компонент - для компонентов с сложным жизненным циклом
4. Минимум 7 обработчиков событий разных типов (onClick, onChange, onSubmit, etc.)
5. Использование state через useState/setState
6. Применение хуков (useEffect, useContext, useReducer и др.)

- Настроить стили css (см. индивидуальное задание по вариантам) (без <https://www.bootstrapcdncdn.com/>);

### **CSS требования:**

- Flexbox/Grid layout
- CSS transitions/animations
- Псевдоклассы (:hover, :focus, :active)
- Адаптивный дизайн
- CSS variables для темизации
- Семантические классы
- Валидация форм как на стороне сервера, так и на стороне клиента;
- Ограничить использование API проекта для неавторизованных запросов.
- **На отметку от 7 баллов – выполнить индивидуальные задания по добавлению AI API (хотя бы одно API) и асинхронности (хотя бы один вариант использования).**

### **Доп. Ссылки AI API:**

OpenAI ChatGPT API - AI-консультант по выбору ....

Документация: <https://platform.openai.com/docs/api-reference>

Endpoint: <https://api.openai.com/v1/chat/completions>

### **Google Vision AI** - определение ..... животных по фото

Документация: <https://cloud.google.com/vision/docs>

Endpoint: <https://vision.googleapis.com/v1/images:annotate>

### **Dialogflow ES** - чат-бот для бронирования

Документация: <https://cloud.google.com/dialogflow/docs>

Endpoint: <https://dialogflow.googleapis.com/v2/projects/{project}/agent/sessions/{session}:detectIntent>

### **IBM Watson Tone Analyzer** - анализ отзывов

Документация: <https://cloud.ibm.com/apidocs/tone-analyzer>

Endpoint: <https://api.us-south.tone-analyzer.watson.cloud.ibm.com/v3/tone>

### **Microsoft Azure Custom Vision** - анализ упражнений

Документация: <https://docs.microsoft.com/azure/cognitive-services/custom-vision-service/>

Endpoint: <https://{{endpoint}}/customvision/v3.0/Prediction/{{projectId}}/classify/iterations/{{publishedName}}/image>

### **Hugging Face Inference API** - генерация тренировок

Документация: <https://huggingface.co/docs/api-inference/>

Endpoint: [https://api-inference.huggingface.co/models/{{model\\_name}}](https://api-inference.huggingface.co/models/{{model_name}})

### **OpenAI DALL-E** - генерация изображений направлений

Документация: <https://platform.openai.com/docs/api-reference/images>

Endpoint: <https://api.openai.com/v1/images/generations>

### **Google Places API + AI** - рекомендации мест

Документация: <https://developers.google.com/maps/documentation/places/web-service>

Endpoint: <https://maps.googleapis.com/maps/api/place/nearbysearch/json>

### **Google Cloud Video Intelligence** - анализ парковки

Документация: <https://cloud.google.com/video-intelligence/docs>

Endpoint: <https://videointelligence.googleapis.com/v1/videos:annotate>

### **AWS Rekognition** - распознавание номеров

Документация: <https://docs.aws.amazon.com/rekognition/>

Endpoint: <https://rekognition.{region}.amazonaws.com>

### **Google Routes API** - оптимизация маршрутов

Документация: <https://developers.google.com/maps/documentation/routes>

Endpoint: <https://routes.googleapis.com/directions/v2:computeRoutes>

### **IBM Watson NLP** - анализ грузовых накладных

Документация: <https://cloud.ibm.com/apidocs/natural-language-understanding>

Endpoint: https://api.us-south.natural-language-understanding.watson.cloud.ibm.com/v1/analyze

### Асинхронные операции (На отметку от 7 баллов):

#### XMLHttpRequest (можно заменить на Fetch API) использовать для:

- Операций с отслеживанием прогресса (upload/download)
- Работы с legacy API или специфическими протоколами
- Реализации кастомных обработчиков прогресса
- 

#### setTimeout использовать для:

- Задержек в UI/UX (автоскрытие уведомлений)
- Периодических обновлений данных
- Реализации таймеров и обратного отсчета
- сохранять состояние таймеров в localStorage

#### Promise использовать для:

- Цепочек асинхронных операций с зависимостями
- Обработки нескольких параллельных запросов
- Реализации сложной бизнес-логики с асинхронными шагами
- использовать async/await для лучшей читаемости

### Индивидуальные задания по вариантам (использовать тот вариант, который был в прошлом году)

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
1.	Зоомагазин	фирма, занимается продажей зоотоваров. Основная часть деятельности, находящейся в Вашем ведении, связана с работой с поставщиками. Фирма имеет определенный набор поставщиков, по каждому из которых известны название, адрес	<b>Microsoft Azure Custom Vision</b> - классификация товаров для животных <b>Hugging Face API</b> - чат-консультант по уходу за питомцами API: <a href="https://azure.microsoft.com/">https://azure.microsoft.com/</a>	1. <b>PetProductMatcher</b> (функциональный с useContext) - подбор товаров для питомца 2. <b>InventoryManager</b> (кл	Яркий и дружелюбный дизайн с пастельными тонами. Интерактивные элементы с	<b>XMLHttpRequest:</b> Отправка больших заказов поставщикам с прогрессом <b>setTimeout:</b> Автоматическое

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
		и телефон. У этих поставщиков Вы приобретаете товары. Товар наряду с названием характеризуется артикулом и ценой (считаем цену постоянной). Некоторые из поставщиков могут поставлять одинаковые товары (один и тот же артикул). Каждый факт покупки зоотовара у поставщика фиксируется, причем обязательными для запоминания являются дата покупки и количество приобретенных товаров. Цена товара может меняться от поставки к поставке. Поставщики заранее ставят Вас в известность о дате изменения цены и о его новом значении.	<a href="https://aws.amazon.com/services/cognitive-services/custom-vision-service">https://aws.amazon.com/services/cognitive-services/custom-vision-service</a> , <a href="https://huggingface.co/inference-api">https://huggingface.co/inference-api</a>	ассовый) - управление складскими запасами <b>3.</b> <b>SupplierDashboard</b> (с трелочная функция) - дашборд поставщиков Обработчики: onProductMatch, onInventoryUpdate, onSupplierRate, onDeliveryTrack, onPetProfile, onRecommendation, onSubscription	анимациями. Карточки товаров с тенями.	скрытие уведомлений о добавлении товара в корзину через 3 секунды <b>Promise:</b> Цепочка заказа: проверка наличия → расчет доставки → отправка на сборку
2.	Гостиница	гостиница предоставляет номера клиентам на определенный срок. Каждый номер характеризуется вместимостью, комфортностью (люкс, полулюкс, обычный) и ценой. Вашими клиентами являются различные лица, о которых Вы собираете определенную информацию (фамилия, имя, отчество и некоторый комментарий). Сдача номера клиенту производится при наличии свободных мест в номерах, подходящих клиенту по указанным выше параметрам. При поселении фиксируется дата поселения. При выезде из гостиницы для каждого места запоминается дата освобождения.	<b>Amazon Rekognition</b> - распознавание лиц для регистрации <b>Google Dialogflow</b> - чат-бот для службы поддержки API: <a href="https://aws.amazon.com/rekognition">https://aws.amazon.com/rekognition</a> , <a href="https://cloud.google.com/dialogflow">https://cloud.google.com/dialogflow</a>	<b>1.</b> <b>FaceCheckIn</b> (функциональный с useMemo) - система регистрации по лицу <b>2.</b> <b>RoomAllocation</b> (классовый) - распределение номеров <b>3.</b> <b>ServiceManager</b> (стрелочная функция) - управление дополнительными услугами Обработчики: onFaceScan, onRoomAssign, onServiceRequest, onCheckIn, onCheckOut,	Элегантный дизайн в стиле hospitality. Спокойная цветовая гамма (бежевый, золотой, белый). Плавные transition эффекты.	<b>XMLHttpRequest:</b> Отправка отзывов с отслеживанием прогресса и возможностью повтора <b>setTimeout:</b> Организовать автоматическое обновление статуса номеров каждые 30 секунд <b>Promise:</b> Создать цепочку для бронирования: проверка доступности → создание брони → отправка подтверждения

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
				onCleaningSchedule, onGuestFeedback		
3.	Фитнес-клуб	Фитнес-клуб организует спортивный досуг клиентов в группах. Группа состоит из нескольких человек (клиентов), один и тот же клиент может записаться в несколько различных групп. При записи клиента в группу им единоразово вносится оплата за весь цикл занятий, проводимых для данной группы. Каждое занятие проводится в одной группе одним или несколькими инструкторами. Для одной группы организуется, как правило, несколько последовательных занятий. Один и тот же инструктор может принимать участие в проведении различных занятий в различных группах. По каждому занятию фиксируется: группа, в которой оно проводится, время начала и окончания занятия, инструкторы, принимающие участие в проведении занятия. Составить список клиентов, занимающихся в определенной группе. Подсчитать количество занятий, проведенных в каждой из групп за определенный период. Определить стоимость оказанных услуг каждому клиенту за весь период посещения им фитнес-клуба.	<b>Google MediaPipe</b> - анализ позы во время упражнений <b>OpenAI API</b> - генерация персональных тренировок API: <a href="https://google.github.io/mediapipe">https://google.github.io/mediapipe</a> , <a href="https://openai.com">https://openai.com</a>	1. <b>PoseAnalyzer</b> (функциональный с useCallback) - анализ правильности выполнения упражнений 2. <b>GroupSchedule</b> (классовый) - расписание групповых занятий 3. <b>ProgressTracker</b> (строковая функция) - отслеживание прогресса клиентов Обработчики: onPoseCapture, onScheduleUpdate, onProgressLog, onPayment, onMembership, onTrainerAssign, onHealthData	Энергичный спортивный дизайн с яркими акцентами. Анимации движения. Карточки тренировок с градиентами.	<b>XMLHttpRequest:</b> Загрузка видео тренировок с отображением прогресса загрузки <b>setTimeout:</b> Реализовать таймер обратного отсчета для начала групповых занятий <b>Promise:</b> Организовать запись на тренировку: проверка абонемента → бронирование места → уведомление тренера
4.	Турагентство	компания работает с клиентами, продавая им путевки. у каждого клиента, пришедшего к Вам, собираются некоторые стандартные данные – фамилия, имя, отчество, адрес, телефон. После этого Ваши сотрудники выясняют у клиента, куда он хотел бы поехать отдохнуть. При этом	<b>Google Places API + AI</b> - рекомендации мест <b>OpenAI DALL-E</b> - генерация изображений направлений API: <a href="https://developers.google.com/maps/document">https://developers.google.com/maps/document</a>	1. <b>DestinationFinder</b> (функциональный с useReducer) - интеллектуальный подбор направлений 2. <b>BookingSystem</b> (класс	Яркий travel-дизайн с изображениями направлений. Интерактивная карта. Слайдеры с фотографиями.	<b>XMLHttpRequest:</b> Отправка больших файлов документов (сканы паспортов) с индикацией прогресса

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
		<p>ему демонстрируются различные варианты, включающие страну проживания, особенности местного климата, имеющиеся отели разного класса. Наряду с этим, обсуждается возможная длительность пребывания и стоимость путевки.</p> <p>В случае если удалось договориться, и найти для клиента приемлемый вариант, Вы регистрируете факт продажи путевки (или путевок, если клиент покупает сразу несколько путевок), фиксируя дату отправления.</p> <p>Фирма работает с некоторыми отелями в нескольких странах. Путевки продаются на одну, две или четыре недели. Стоимость путевки зависит от длительности тура и отеля.</p>	<a href="#">action/places</a> , <a href="https://openai.com/dall-e">https://openai.com/dall-e</a>	<p>овый) - система бронирования туров</p> <p><b>3.</b> <b>TravelPlanner</b> (стрелочная функция) - планирование маршрута</p> <p>Обработчики:</p> <ul style="list-style-type: none"> <li>onDestinationSelect,</li> <li>onDatePicker,</li> <li>onHotelBook,</li> <li>onItinerary,</li> <li>onBudgetSet,</li> <li>onReview,</li> <li>onTravelAlert</li> </ul>		<p><b>setTimeout:</b> Автоматическое сохранение черновика заявки на тур каждые 2 минуты</p> <p><b>Promise:</b> Цепочка для подбора тура: поиск вариантов → проверка availability → расчет стоимости</p>
5.	Автостоянка	<p>В городе существует круглосуточная отапливаемая автостоянка "AutoCar" с установленной автоматизированной системой наблюдения, которая предоставляет гарантии безопасности автомобиля посетителя на своей территории, за счет того, что контролирует все выходы из комплекса автостоянки и сохраняет время автовладельцев на прогрев автомобиля, также в добавок ко всему закрытая автостоянка дает возможность избежать загрязнения от природных погодных условий. Клиент оплачивает место автостоянки ежемесячно.</p> <p>Уникальный номер парковочного места имеет числовой формат с ограничением не более 999.</p>	<p><b>AWS Rekognition</b> - распознавание номерных знаков</p> <p><b>Google Cloud Video Intelligence</b> - мониторинг парковки</p> <p>API: <a href="https://aws.amazon.com/rekognition">https://aws.amazon.com/rekognition</a>, <a href="https://cloud.google.com/video-intelligence">https://cloud.google.com/video-intelligence</a></p>	<p><b>1.</b> <b>LicensePlateReader</b> (функциональный с useRef) - автоматическое распознавание номеров</p> <p><b>2.</b> <b>ParkingMonitor</b> (классовый) - мониторинг занятости мест</p> <p><b>3.</b> <b>BillingManager</b> (стрелочная функция) - управление оплатами</p> <p>Обработчики:</p> <ul style="list-style-type: none"> <li>onPlateDetect,</li> <li>onSpaceReserve,</li> <li>onPayment,</li> <li>onAccessControl,</li> <li>onViolation,</li> </ul>	<p>Технический минималистичный дизайн. Схема парковки с интерактивными элементами. Статусные индикаторы.</p>	<p><b>XMLHttpRequest:</b> Мониторинг статуса камер видеонаблюдения с периодическим опросом</p> <p><b>setTimeout:</b> Автоматическое снятие брони парковочного места при неоплате в течение 15 минут</p> <p><b>Promise:</b> Процесс въезда: распознавание номера → проверка прав →</p>

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
		У каждого уникального парковочного места есть своя цена. Клиент паркуется только на своем парковочном месте исходя из номера автомобиля; У одного клиента может быть несколько автомобилей; У одного автомобиля может быть несколько клиентов.		onMaintenance, onCustomerNotify		открытие шлагбаума
6.	Грузоперевозки	Автоматизация работы компании, занимающейся перевозкой грузов посредством автотранспорта.	<b>Google Routes API</b> - оптимизация маршрутов <b>IBM Watson NLP</b> - анализ грузовых накладных API: <a href="https://developers.google.com/maps/documentation/routes">https://developers.google.com/maps/documentation/routes</a> , <a href="https://www.ibm.com/cloud/watson-natural-language-understanding">https://www.ibm.com/cloud/watson-natural-language-understanding</a>	<b>1.</b> <b>RouteOptimizer</b> (функциональный с <code>useMemo</code> ) - оптимизация маршрутов доставки <b>2.</b> <b>CargoTracker</b> (классовый) - отслеживание грузов <b>3.</b> <b>DriverManager</b> (стрелочная функция) - управление водителями Обработчики: <code>onRouteCalculate</code> , <code>onCargoTrack</code> , <code>onDriverAssign</code> , <code>onDeliveryConfirm</code> , <code>onDocumentScan</code> , <code>onFuelLog</code> , <code>onVehicleMaintenance</code>	Профессиональный корпоративный дизайн. Интерактивные карты. Таблицы с сортировкой.	<b>XMLHttpRequest:</b> Отслеживание местоположения груза через периодические запросы к GPS-трекеру <b>setTimeout:</b> Автоматическое уведомление о задержке доставки при превышении планового времени <b>Promise:</b> Цепочка оформления заказа: расчет маршрута → проверка документов → назначение водителя
7.	Автомагазин	фирма, занимается продажей запасных частей для автомобилей. Основная часть деятельности, находящейся в Вашем ведении, связана с работой с поставщиками. Фирма имеет определенный набор поставщиков, по каждому из	<b>Clarifai API</b> - распознавание деталей по фото <b>OpenAI API</b> - чат-помощник по подбору запчастей	<b>1.</b> <b>PartRecognition</b> (функциональный с <code>useState</code> ) - загрузка фото детали для распознавания <b>2.</b>	Технический дизайн с индустриальной цветовой схемой (серый, синий, оранжевый). Карточки	<b>XMLHttpRequest:</b> Отслеживание прогресса загрузки больших каталогов запчастей

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
		<p>которых известны название, адрес и телефон. У этих поставщиков Вы приобретаете детали.</p> <p>Каждая деталь наряду с названием характеризуется артикулом и ценой (считаем цену постоянной). Некоторые из поставщиков могут поставлять одинаковые детали (один и тот же артикул). Каждый факт покупки запчастей у поставщика фиксируется, причем обязательными для запоминания являются дата покупки и количество приобретенных деталей.</p> <p>Цена детали может меняться от поставки к поставке. Поставщики заранее ставят Вас в известность о дате изменения цены и о его новом значении.</p>	API: <a href="https://clarifai.com">https://clarifai.com</a> , <a href="https://openai.com">https://openai.com</a>	<p><b>SupplierCatalog</b> (классовый) - каталог поставщиков с фильтрацией</p> <p><b>3.</b></p> <p><b>PriceTracker</b> (стрелочная функция) - отслеживание изменения цен</p> <p>Обработчики:</p> <ul style="list-style-type: none"> <li>onImageUpload,</li> <li>onSearch, onFilter,</li> <li>onPriceChange,</li> <li>onSupplierSelect,</li> <li>onPartAdd, onError</li> </ul>	деталей с hover-эффектами. Progress bar для загрузки изображений.	<b>setTimeout:</b> Автоматическое обновление цен каждые 5 минут <b>Promise:</b> Цепочка оформления заказа: проверка наличия → резервирование → отправка уведомления поставщику
8.	Зоопарк	<p>хранение информации о деятельности зоопарка, т.е. о помещениях, животных, сотрудниках;</p> <p>доступ к информации, какое животное в каком месте находится и какой сотрудник за ним закреплен;</p> <p>учет расхода кормов разных видов.</p>	<b>Google Vision AI</b> - распознавание животных по фото <b>IBM Watson NLP</b> - анализ отзывов посетителей API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> , <a href="https://www.ibm.com/cloud/watson-natural-language-understanding">https://www.ibm.com/cloud/watson-natural-language-understanding</a>	<p><b>1.</b></p> <p><b>AnimalRecognition</b> (функциональный с useReducer) - идентификация животных по фото</p> <p><b>2.</b></p> <p><b>HabitatManager</b> (классовый) - управление помещениями и вольерами</p> <p><b>3.</b></p> <p><b>FeedingSchedule</b> (стрелочная функция) - расписание кормления</p> <p>Обработчики:</p> <ul style="list-style-type: none"> <li>onAnimalIdentify,</li> <li>onHabitatUpdate,</li> <li>onFeedingTime,</li> <li>onStaffAssign,</li> <li>onFoodOrder,</li> </ul>	Естественная цветовая палитра (зеленый, коричневый, бежевый). Анимации плавных переходов. Карточки животных с градиентными фонами.	<b>XMLHttpRequest:</b> Загрузка видео наблюдений за животными с прогрессом <b>setTimeout:</b> Автоматическое напоминание о времени кормления животных <b>Promise:</b> Бронирование экскурсии: проверка расписания → бронирование гида → формирование группы

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
				onVisitorCount, onMaintenance		
9.	Музей	хранение информации о деятельности музея, т.е. о залах, экспонатах, сотрудниках; доступ к информации, какой экспонат в каком месте находится и какой сотрудник за ним закреплен в качестве смотрящего; учет проводимых в музее экскурсий.	<b>Google Cloud Vision</b> - анализ изображений экспонатов <b>OpenAI GPT</b> - генерация описаний экспонатов API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> , <a href="https://openai.com">https://openai.com</a>	<p>1. <b>ExhibitScanner</b> (функциональный с useRef) - сканирование и анализ экспонатов</p> <p>2. <b>TourPlanner</b> (классовый) - планирование экскурсионных маршрутов</p> <p>3. <b>ConservationTracker</b> (стрелочная функция) - отслеживание состояния экспонатов</p> <p>Обработчики:</p> <ul style="list-style-type: none"> <li>onExhibitScan,</li> <li>onTourCreate,</li> <li>onConservationUpdate,</li> <li>onVisitorFlow,</li> <li>onAudioGuide,</li> <li>onLightingControl,</li> <li>onSecurityCheck</li> </ul>	Классический музейный дизайн с темными тонами и акцентами. Типография с засечками. Слайдеры для галереи.	<b>XMLHttpRequest</b> : Загрузка аудиогидов с отслеживанием прогресса <b>setTimeout</b> : Автоматическое переключение слайдов в виртуальной экспозиции <b>Promise</b> : Процесс покупки билета: проверка квот → выбор времени → оплата
10.	Магазин бытовой химии	Вы являетесь сотрудником коммерческого отдела компании, продающей различные товары через Интернет. Вашей задачей является отслеживание финансовой составляющей работы компании.	<b>Google Vision AI</b> - распознавание товаров и штрих-кодов API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI API</b> - консультант по средствам и их применению API: <a href="https://openai.com">https://openai.com</a>	<p>1. <b>ProductScanner</b> (функциональный с useCallback) - сканер товаров и штрих-кодов</p> <p>2. <b>InventoryOptimizer</b> (классовый) - оптимизация складских запасов</p> <p>3. <b>SafetyAdvisor</b> (стрелочная функция) - консультант по безопасному</p>	Чистый гигиенический дизайн с синими и белыми тонами. Четкая категоризация товаров. Предупреждающие цветовые коды.	<b>XMLHttpRequest</b> : Отправка данных инвентаризации с отслеживанием прогресса <b>setTimeout</b> : Автоматическое применение скидок к товарам с истекающим сроком годности <b>Promise</b> : Цепочка заказа:

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
				использованию Обработчики: onProductScan, onInventoryOptimize, onSafetyAdvise, onCompositionCheck, onAllergyAlert, onUsageGuide, onDisposal		проверка наличия → расчет доставки → отправка на сборку
11.	Книжный магазин	Вы являетесь сотрудником коммерческого отдела компании, продающей различные товары через Интернет. Вашей задачей является отслеживание финансовой составляющей работы компании.	<b>Google Vision AI</b> - распознавание книг по обложкам API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI API</b> - рекомендательная система и рецензии API: <a href="https://openai.com">https://openai.com</a>	1. <b>BookRecommender</b> (функциональный с useMemo) - интеллектуальная рекомендация книг 2. <b>ReadingTracker</b> (классовый) - отслеживание прогресса чтения 3. <b>ReviewAnalyzer</b> (строчная функция) - анализ отзывов и рецензий Обработчики: onBookRecommend, onReadingTrack, onReviewAnalyze, onWishlist, onBookClub, onQuoteShare, onCollection	Классический книжный дизайн с темно-зелеными и коричневыми тонами. Элегантная типография. Карточки книг с рейтингами.	<b>XMLHttpRequest:</b> Загрузка превью книг и отрывков с индикацией прогресса <b>setTimeout:</b> Автоматическое сохранение позиции чтения в электронной книге <b>Promise:</b> Процесс рекомендаций: анализ предпочтений → подбор книг → формирование подборки
12.	Автосалон	Вы являетесь сотрудником коммерческого отдела компании, продающей автомобили. Вашей задачей является отслеживание финансовой составляющей работы компании.	<b>Google Vision AI</b> - анализ внешнего состояния автомобилей или распознавание автомобилей по фото для подбора аналогов API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a>	1. <b>CarConfigurator</b> (функциональный с useReducer) - конфигуратор автомобилей с опциями 2.	Премиальный дизайн с хромовыми акцентами. 3D визуализации автомобилей. Интерактивные конфигураторы.	<b>XMLHttpRequest:</b> Загрузка конфигуратора автомобиля с интерактивным выбором опций <b>setTimeout:</b> Автоматическое

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
			<b>OpenAI API</b> - консультант по выбору автомобиля API: <a href="https://openai.com">https://openai.com</a>	<b>TestDriveScheduler</b> (классовый) - планирование тест-драйвов <b>3.</b> <b>TradeInEvaluator</b> (стрелочная функция) - оценка автомобилей с пробегом Обработчики: onCarConfigure, onTestDriveSchedule, onTradeInEvaluate, onFinanceCalculate, onColorSelect, onFeatureCompare, onDelivery		обновление акций и специальных предложений <b>Promise:</b> Цепочка тест-драйва: проверка доступности → бронирование → напоминание
13.	Риэлтерское агентство	Вы являетесь сотрудником коммерческого отдела риэлтерского агентства, продающей объекты недвижимости через Интернет. Вашей задачей является отслеживание финансовой составляющей работы компании.	<b>Google Vision AI</b> - анализ фотографий объектов недвижимости API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a>  <b>OpenAI API</b> - генерация описаний объектов и маркетинговых материалов, чат-бот для первичной консультации API: <a href="https://openai.com">https://openai.com</a>	<b>1.</b> <b>PropertyMatcher</b> (функциональный с useContext) - подбор объектов по критериям клиента <b>2.</b> <b>DealFlow</b> (классовый) - управление потоком сделок <b>3.</b> <b>MarketAnalyzer</b> (стрелочная функция) - анализ рынка недвижимости Обработчики: onPropertyMatch, onDealFlow, onMarketAnalyze, onViewingArrange, onDocumentPrepare, onClientCommunicate, onCommission	Профессиональный дизайн с корпоративной цветовой схемой. Интерактивные карты районов. Галереи объектов с фильтрами.	<b>XMLHttpRequest:</b> Загрузка документов по сделке (договоры, акты) <b>setTimeout:</b> Автоматическое напоминание о просроченных платежах по аренде <b>Promise:</b> Цепочка сделки: проверка документов → подписание → регистрация → расчет комиссии

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
14.	Аптека	Автоматизация работы аптеки, основным направлением которой является продажа медикаментов разных видов	<b>Google Cloud Healthcare API</b> - анализ медицинских назначений <b>Microsoft Azure Cognitive Services</b> - распознавание рецептов API: <a href="https://cloud.google.com/healthcare-api">https://cloud.google.com/healthcare-api</a> , <a href="https://azure.microsoft.com/services/cognitive-services">https://azure.microsoft.com/services/cognitive-services</a>	1. <b>PrescriptionReader</b> (функциональный с useState) - сканирование и анализ рецептов 2. <b>DrugInventory</b> (классовый) - управление запасами лекарств 3. <b>InteractionChecker</b> (строковая функция) - проверка взаимодействий препаратов Обработчики: onPrescriptionScan, onDrugSearch, onInteractionCheck, onInventoryAlert, onExpiryTrack, onOrderPlace, onCustomerConsult	Медицинский чистый дизайн. Четкая типография. Цветовые коды для категорий лекарств.	<b>XMLHttpRequest:</b> Отслеживание статуса доставки лекарств курьерской службой <b>setTimeout:</b> Автоматическое напоминание о необходимости рецепта для определенных препаратов <b>Promise:</b> Цепочка заказа рецептурных препаратов: проверка рецепта → резервирование → уведомление врача
15.	Медицинский центр	Клиентом медицинского центра является любой человек, который хотя бы раз воспользовался его услугами. При первом обращении клиента сотрудник центра заносит в базу данных всю необходимую информацию: фио, дату рождения, адрес и т.д. Медицинский центр предоставляет определенные типы услуг согласно фиксированному прайс-листу. Каждому типу относится несколько видов услуг, каждая из которых характеризуется названием и стоимостью. Известно, что один клиент в разное время может совершать несколько заказов. Также известно, что за одно	<b>Google Cloud Healthcare NLP</b> - анализ медицинских записей <b>IBM Watson Assistant</b> - виртуальный медицинский ассистент API: <a href="https://cloud.google.com/healthcare-api">https://cloud.google.com/healthcare-api</a> , <a href="https://www.ibm.com/cloud/watson-assistant">https://www.ibm.com/cloud/watson-assistant</a>	1. <b>MedicalRecordAnalyzer</b> (функциональный с useContext) - анализ медицинских карт 2. <b>AppointmentScheduler</b> (классовый) - система записи на прием 3. <b>TreatmentPlanner</b> (строковая функция) - планирование лечения Обработчики:	Спокойный медицинский дизайн. Доступные интерфейсы. Четкая визуальная иерархия.	<b>XMLHttpRequest:</b> Загрузка медицинских изображений и результатов анализов <b>setTimeout:</b> Автоматическое напоминание о приеме лекарств по расписанию <b>Promise:</b> Процесс записи к врачу: проверка расписания →

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
		обращение клиент может заказать несколько услуг. Один пациент может иметь несколько заболеваний, которые лечит один и/или разные врачи. Один врач может работать с несколькими пациентами.		onRecordAnalyze, onAppointmentBook, onTreatmentPlan, onTestResult, onPrescription, onBilling, onFollowUp		бронирование → SMS-напоминание
16.	Кинотеатр	Автоматизация работы кинотеатра	<b>Google Vision AI</b> - анализ эмоций зрителей <b>OpenAI API</b> - рекомендация фильмов API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> , <a href="https://openai.com">https://openai.com</a>	1. <b>EmotionAnalyzer</b> (функциональный с useReducer) - анализ реакций зрителей 2. <b>MovieRecommender</b> (классовый) - система рекомендаций фильмов 3. <b>ShowtimeManager</b> (строковая функция) - управление сеансами Обработчики: onEmotionDetect, onMovieRecommend, onShowtimeSelect, onTicketBook, onSeatSelect, onTrailerPlay, onRating	Кинематографический дизайн с темной темой. Эффекты затемнения. Анимации появления.	<b>XMLHttpRequest:</b> Загрузка трейлеров фильмов с отображением прогресса буферизации <b>setTimeout:</b> Автоматическое освобождение забронированных билетов при неоплате <b>Promise:</b> Цепочка покупки билетов: выбор места → бронирование → оплата → отправка билета
17.	Автосервис	Автоматизация работы компании, занимающейся ремонтом автомобилей	<b>Clarifai API</b> - диагностика по фото повреждений <b>Google Cloud AutoML</b> - предсказание стоимости ремонта API: <a href="https://clarifai.com">https://clarifai.com</a> , <a href="https://cloud.google.com/automl">https://cloud.google.com/automl</a>	1. <b>DamageAssessor</b> (функциональный с useRef) - оценка повреждений по фото 2. <b>RepairEstimator</b> (классовый) - расчет стоимости ремонта 3. <b>ServiceScheduler</b> (строковая функция) - планирование	Технический дизайн с индустриальным и элементами. Диагностические диаграммы. Статусные индикаторы ремонта.	<b>XMLHttpRequest:</b> Мониторинг статуса ремонта через периодические запросы <b>setTimeout:</b> Автоматическое уведомление о готовности автомобиля <b>Promise:</b> Процесс

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
				ремонтов Обработчики: onDamageAssess, onRepairEstimate, onSchedule, onPartOrder, onCustomerNotify, onQualityCheck, onInvoice		диагностики: сбор данных → анализ → формирование отчета → расчет стоимости
18.	Сервисный центр	Предприятие занимается настройкой, ремонтом и обслуживанием компьютерной техники и периферийных устройств, а также оказывает настройку рабочих мест клиента. Руководством принято решение создать базу данных для автоматизации работы: хранения, поиска и ознакомления с информацией о клиентах, договорах, услугах.	<b>Google Cloud Vision</b> - анализ серийных номеров оборудования API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI API</b> - генерация диагностических отчетов API: <a href="https://openai.com">https://openai.com</a>	1. <b>DeviceDiagnostic</b> (функциональный с useState) - автоматическая диагностика оборудования 2. <b>RepairTracker</b> (классовый) - отслеживание статуса ремонта 3. <b>ClientPortal</b> (стрелочная функция) - личный кабинет клиента Обработчики: onDeviceScan, onRepairUpdate, onClientNotify, onPartOrder, onDiagnosticRun, onReportGenerate, onPayment	Технический дизайн с синими и серыми тонами. Четкие информационные блоки. Индикаторы прогресса ремонта.	<b>XMLHttpRequest:</b> Загрузка диагностических отчетов и логов оборудования <b>setTimeout:</b> Автоматическое обновление статуса заявки в реальном времени <b>Promise:</b> Цепочка приема заявки: регистрация → диагностика → согласование ремонта
19.	Пиццерия	Вы являетесь сотрудником коммерческого отдела компании, продающей различные товары через Интернет. Вашей задачей является отслеживание финансовой составляющей работы компании.	<b>Google Vision AI</b> - распознавание ингредиентов по фото API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI API</b> - генерация рецептов и	1. <b>PizzaCustomizer</b> (функциональный с useReducer) - конструктор пиццы с ингредиентами 2. <b>OrderTracker</b> (классов	Яркий итальянский стиль с красными и желтыми акцентами. Анимации добавления	<b>XMLHttpRequest:</b> Отслеживание статуса заказа и местоположения курьера <b>setTimeout:</b> Автоматическая отмена заказа

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
			рекомендаций API: <a href="https://openai.com">https://openai.com</a>	ый) - отслеживание заказов и доставки <b>3.</b> <b>MenuManager</b> (стрелочная функция) - управление меню и акциями Обработчики: onPizzaBuild, onOrderTrack, onMenuUpdate, onIngredientSelect, onDeliveryAssign, onPromoApply, onFeedback	ингредиентов. Карточки пицц с фотографиями.	при длительном ожидании курьера <b>Promise:</b> Процесс заказа: проверка ингредиентов → приготовление → упаковка → доставка
20.	Прокат автомобилей	Вы являетесь руководителем коммерческой службы в фирме, занимающейся прокатом автомобилей. Вашей задачей является отслеживание финансовых показателей работы пункта проката.	<b>AWS Rekognition</b> - проверка документов и водительских прав API: <a href="https://aws.amazon.com/rekognition">https://aws.amazon.com/rekognition</a> <b>Google Routes API</b> - планирование оптимальных маршрутов API: <a href="https://developers.google.com/maps/documentation/routes">https://developers.google.com/maps/documentation/routes</a>	1. <b>DocumentVerifier</b> (функциональный с useRef) - верификация документов клиентов <b>2.</b> <b>FleetManager</b> (классовый) - управление автопарком <b>3.</b> <b>RentalCalculator</b> (стрелочная функция) - расчет стоимости аренды Обработчики: onDocumentVerify, onCarSelect, onRentalCalculate, onInsurance, onMaintenance, onDamageAssess, onReturnInspect	Профессиональный дизайн с корпоративной цветовой схемой. Четкие формы бронирования. Интерактивный выбор автомобилей.	<b>XMLHttpRequest:</b> Загрузка документов для аренды (права, паспорт) с валидацией <b>setTimeout:</b> Автоматическое напоминание о возврате автомобиля за 2 часа до окончания аренды <b>Promise:</b> Цепочка бронирования: проверка доступности → бронирование → подготовка документов
21.	Клининговая компания	Клининговая компания — организация, которая предлагает достаточно	<b>Google Cloud Vision</b> - оценка состояния помещений до/после	1. <b>CleanlinessAssessor</b> (функциональный с	Чистый минималистичный дизайн с	<b>XMLHttpRequest:</b> Отправка фотоотчетов о

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
		большой спектр клининговых услуг, такие, как профессиональная уборка помещений, офисов, коттеджей.	уборки API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a>  <b>OpenAI API</b> - генерация индивидуальных планов уборки API: <a href="https://openai.com">https://openai.com</a>	useContext) - оценка чистоты помещений <b>2.</b> <b>SchedulePlanner</b> (классовый) - планирование графиков уборки <b>3.</b> <b>QuoteGenerator</b> (строковая функция) - расчет стоимости услуг Обработчики: onAssessment, onSchedulePlan, onQuoteGenerate, onTeamAssign, onSupplyOrder, onQualityCheck, onClientReview	бело-синей палитрой. Интуитивные формы заказа. Индикаторы выполнения задач.	выполненной уборке <b>setTimeout:</b> Автоматическое напоминание клиенту за день до визита клинеров <b>Promise:</b> Процесс заказа уборки: расчет стоимости → выбор времени → подтверждение → напоминание
22.	Косметологический центр	Клиентом косметологического центра является любой человек, который хотя бы раз воспользовался его услугами. При первом обращении клиента сотрудник косметологического центра заносит в базу данных всю необходимую информацию.	<b>Amazon Rekognition</b> - анализ кожи и рекомендации процедур API: <a href="https://aws.amazon.com/rekognition">https://aws.amazon.com/rekognition</a>  <b>OpenAI API</b> - консультант по уходу за кожей API: <a href="https://openai.com">https://openai.com</a>	<b>1.</b> <b>SkinAnalyzer</b> (функциональный с useMemo) - анализ состояния кожи <b>2.</b> <b>AppointmentBooker</b> (классовый) - система записи на процедуры <b>3.</b> <b>TreatmentPlanner</b> (строковая функция) - планирование курса процедур Обработчики: onSkinAnalyze, onAppointmentBook, onTreatmentPlan, onProductRecommend, onProgressTrack,	Элегантный дизайн с пастельными тонами. Плавные анимации переходов. Карточки процедур с изображениями.	<b>XMLHttpRequest:</b> Загрузка фотографий "до/после" <b>setTimeout:</b> Автоматическое напоминание о следующем визите по расписанию процедур <b>Promise:</b> Цепочка записи: консультация → выбор процедуры → бронирование → подготовка

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
				onReminderSet, onReview		
23.	Фабрика игрушек	Фабрика производит игрушки нескольких видов и моделей. Все виды игрушек изготавливаются по заказам оптовых продавцов.	<b>Google Cloud Vision</b> - контроль качества продукции API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI DALL-E</b> - генерация дизайнов игрушек API: <a href="https://openai.com/dall-e">https://openai.com/dall-e</a>	<p>1. <b>QualityInspector</b> (функциональный с useReducer) - автоматический контроль качества</p> <p>2. <b>ProductionTracker</b> (классовый) - отслеживание производственного процесса</p> <p>3. <b>DesignStudio</b> (стрелочная функция) - создание дизайнов игрушек Обработчики: onQualityCheck, onProductionTrack, onDesignCreate, onMaterialOrder, onShipment, onClientApprove, onInventory</p>	Яркий детский дизайн с разноцветными элементами. Интерактивные элементы производства. Визуализация производственной линии.	<b>XMLHttpRequest:</b> Мониторинг статуса производственной линии через API оборудования <b>setTimeout:</b> Автоматическое обновление статуса заказов в производственной очереди <b>Promise:</b> Процесс заказа: расчет материалов → планирование производства → контроль качества
24.	Кондитерская	Вы являетесь сотрудником коммерческого отдела компании, продающей различные товары через Интернет. Вашей задачей является отслеживание финансовой составляющей работы компании.	<b>Google Vision AI</b> - распознавание десертов по фото для каталога API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI API</b> - генерация рецептов и описаний десертов API: <a href="https://openai.com">https://openai.com</a>	<p>1. <b>CakeDesigner</b> (функциональный с useState) - конструктор тортов и десертов</p> <p>2. <b>OrderManager</b> (классовый) - управление заказами и доставкой</p> <p>3. <b>RecipeBook</b> (стрелочная функция) - база рецептов с поиском</p>	Сладкая пастельная палитра с акцентами шоколадного и кремового. Анимации украшений. Карточки десертов с аппетитными фото.	<b>XMLHttpRequest:</b> Загрузка изображений готовых изделий для каталога <b>setTimeout:</b> Автоматическое уведомление о готовности индивидуального заказа <b>Promise:</b> Цепочка заказа

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
				Обработчики: onCakeDesign, onOrderManage, onRecipeSearch, onIngredientCheck, onDeliverySchedule, onCustomization, onTasting		торта: консультация → дизайн → приготовление → уведомление
25.	Продукты с доставкой	Вы являетесь сотрудником коммерческого отдела компании, продающей различные товары через Интернет. Вашей задачей является отслеживание финансовой составляющей работы компании.	<b>Google Cloud Vision</b> - распознавание продуктов по фото API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI API</b> - рекомендации рецептов на основе покупок API: <a href="https://openai.com">https://openai.com</a>	1. <b>ProductRecognizer</b> (функциональный с useCallback) - распознавание продуктов по изображению 2. <b>DeliveryTracker</b> (классовый) - отслеживание доставки в реальном времени 3. <b>SmartCart</b> (стрелочная функция) - умная корзина с рекомендациями Обработчики: onProductRecognize, onDeliveryTrack, onCartUpdate, onRecipeSuggest, onSubstitution, onExpiryAlert, onRestock	Свежий дизайн с зелеными и оранжевыми акцентами. Четкая категоризация продуктов. Интуитивная навигация.	<b>XMLHttpRequest:</b> Отслеживание местоположения курьера в реальном времени <b>setTimeout:</b> Автоматическое применение акций к товарам в корзине <b>Promise:</b> Процесс доставки: сбор заказа → упаковка → доставка → подтверждение получения
26.	Агентство недвижимости	Автоматизация работы компании, занимающейся продажей и арендой объектов недвижимости.	<b>Google Vision AI</b> - анализ фотографий объектов недвижимости API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI API</b> - генерация	1. <b>PropertyEvaluator</b> (функциональный с useContext) - оценка стоимости недвижимости 2.	Элегантный дизайн с золотыми и темно-синими акцентами. Галереи фотографий	<b>XMLHttpRequest:</b> Загрузка документов по сделке (договоры, акты) <b>setTimeout:</b> Автоматическое

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
			описаний объектов API: <a href="https://openai.com">https://openai.com</a>	<b>DealManager</b> (классов ый) - управление сделками и документами <b>3.</b> <b>VirtualTour</b> (стрелочн ая функция) - виртуальные туры по объектам Обработчики: onPropertyEvaluate, onDealManage, onTourStart, onDocumentUpload, onClientMatch, onViewingSchedule, onContract	объектов. Интерактивные планы помещений.	напоминание о просроченных платежах по аренде <b>Promise:</b> Цепочка сделки: проверка документов → подписание → регистрация → расчет комиссии
27.	Страховая фирма	Компания имеет различные филиалы по всей стране. Каждый филиал характеризуется названием, адресом и телефоном. Деятельность компании организована следующим образом: к Вам обращаются различные лица с целью заключения договора о страховании.	<b>Google Document AI</b> - обработка страховых документов API: <a href="https://cloud.google.com/document-ai">https://cloud.google.com/document-ai</a> <b>OpenAI API</b> - консультант по страховым продуктам API: <a href="https://openai.com">https://openai.com</a>	<b>1.</b> <b>RiskCalculator</b> (функциональный с useMemo) - расчет страховых рисков <b>2.</b> <b>PolicyManager</b> (класс овый) - управление страховыми полисами <b>3.</b> <b>ClaimProcessor</b> (стрелочная функция) - обработка страховых случаев Обработчики: onRiskCalculate, onPolicyManage, onClaimProcess, onDocumentAnalyze, onPaymentTrack, onRenewal, onFraudDetect	Консервативный корпоративный дизайн. Четкие таблицы и формы. Цветовые коды для разных типов страховок.	<b>XMLHttpRequest:</b> Загрузка сканов документов для оформления страховки <b>setTimeout:</b> Автоматическое продление страховки при согласии клиента <b>Promise:</b> Процесс оформления полиса: расчет рисков → оформление → оплата → отправка полиса

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
28.	Мебельная фабрика	Фабрика производит мебель нескольких видов (кухонную, кабинетную, офисную). Все виды мебели изготавливаются по заказам оптовых продавцов.	<b>Google Vision AI</b> - контроль качества мебели API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI API</b> - дизайнер мебели и планировщик помещений API: <a href="https://openai.com">https://openai.com</a>	1. <b>FurnitureDesigner</b> (функциональный с useRef) - 3D конструктор мебели 2. <b>ProductionScheduler</b> (классовый) - планирование производства 3. <b>QualityController</b> (строковая функция) - контроль качества продукции Обработчики: onFurnitureDesign, onProductionSchedule, onQualityControl, onMaterialCalculate, onClientConsult, onShipmentTrack, onInstallation	Естественные деревянные тона в дизайне. 3D превью мебели. Интерактивные планировки помещений.	<b>XMLHttpRequest:</b> Мониторинг статуса заказа на производстве <b>setTimeout:</b> Автоматическое уведомление о готовности мебели к отгрузке <b>Promise:</b> Цепочка заказа: проектирование → производство → доставка → сборка
29.	Ремонт одежды	Автоматизация работы фирмы, занимающейся ремонтом одежды или обуви.	<b>Google Vision AI</b> - анализ повреждений одежды API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI API</b> - рекомендации по ремонту и уходу API: <a href="https://openai.com">https://openai.com</a>	1. <b>DamageAnalyzer</b> (функциональный с useState) - анализ повреждений одежды 2. <b>RepairEstimator</b> (классовый) - расчет стоимости и сроков ремонта 3. <b>MaterialManager</b> (строковая функция) - управление материалами и фурнитурой Обработчики:	Текстильный дизайн с узорами и фактурами. Четкие формы приема заказов. Визуализация процессов ремонта.	<b>XMLHttpRequest:</b> Загрузка фотографий повреждений для оценки стоимости ремонта <b>setTimeout:</b> Автоматическое напоминание о готовности заказа <b>Promise:</b> Процесс приема: оценка → согласование

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
				onDamageAnalyze, onRepairEstimate, onMaterialManage, onClientConsult, onQualityCheck, onPickupNotify, onFeedback		цены → ремонт → уведомление
30.	Ветеринарная клиника	Клиентом ветеринарной клиники является любой человек, который хотя бы раз воспользовался его услугами для своего домашнего питомца.	<b>Google Vision AI</b> - анализ медицинских снимков животных API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI API</b> - консультант по уходу за животными API: <a href="https://openai.com">https://openai.com</a> <b>IBM Watson NLP</b> - анализ симптомов и истории болезни API: <a href="https://www.ibm.com/cloud/watson-natural-language-understanding">https://www.ibm.com/cloud/watson-natural-language-understanding</a>	1. <b>PetHealthAnalyzer</b> (функциональный с useReducer) - анализ здоровья питомцев 2. <b>VaccineTracker</b> (классовый) - отслеживание вакцинации и процедур 3. <b>EmergencyAlert</b> (стречная функция) - система экстренных уведомлений Обработчики: onHealthAnalyze, onVaccineTrack, onEmergencyAlert, onAppointmentBook, onPrescription, onPetProfile, onReminder	Дружелюбный ветеринарный дизайн с мягкими тонами. Интуитивные карточки пациентов. Четкая система уведомлений.	<b>XMLHttpRequest</b> : Загрузка результатов анализов и рентгеновских снимков <b>setTimeout</b> : Автоматическое напоминание о следующем визите к ветеринару <b>Promise</b> : Цепочка приема: запись → осмотр → диагностика → назначение лечения
31.	Театр	Автоматизация работы театра.	<b>Google Vision AI</b> - анализ заполняемости зала или распознавание эмоций зрителей во время спектакля API: <a href="https://cloud.google.com/vision">https://cloud.google.com/vision</a> <b>OpenAI API</b> - генерация описаний спектаклей и рецензий или	1. <b>SeatSelector</b> (функциональный с useContext) - интерактивный выбор мест 2. <b>PerformanceManager</b> (классовый) - управление репертуаром и занавесами.	Театральный дизайн с бархатными текстурами и золотыми акцентами. Эффектные шрифты. Анимации занавеса.	<b>XMLHttpRequest</b> : Загрузка видео репетиций и дополнительных материалов <b>setTimeout</b> : Автоматическое освобождение забронированных билетов

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
			рекомендация спектаклей по предпочтениям API: <a href="https://openai.com">https://openai.com</a>	расписанием <b>3.</b> <b>CastingDirector</b> (стречная функция) - база данных актеров и ролей Обработчики: onSeatSelect, onPerformanceManage, onCasting, onTicketBook, onRehearsalSchedule, onCostume, onReview		<b>Promise:</b> Процесс бронирования: выбор спектакля → выбор места → бронирование → оплата

**Примеры: React компоненты и стили:**

```
jsx
// AI консультант (функциональный компонент с хуком useState)
const PetConsultant = ({ onRecommendation }) => {
  const [messages, setMessages] = useState([]);
  // Стиль: чат-интерфейс с градиентным фоном
};
```

```
// Классовый компонент для загрузки фото
class PhotoUploader extends React.Component {
  // Стиль: drag-and-drop зона с анимацией
}
```

```
// Декларативный компонент рекомендаций
function PetRecommendations({ pets, onSelect }) {
  // Стиль: карточки с hover-эффектом и тенями
}
```

```
jsx
// Чат-бот (функциональный компонент с useReducer)
```

```
const BookingAssistant = ({ userId }) => {
  const [state, dispatch] = useReducer(chatReducer, initialState);
  // Стиль: bubble чат с плавными анимациями
};
```

```
// Анализатор отзывов (классовый компонент)
class ReviewAnalyzer extends React.Component {
  // Стиль: дашборд с графиками и цветовой кодировкой
}
```

```
jsx
// Генератор тренировок (функциональный компонент)
const WorkoutGenerator = ({ userLevel, goals }) => {
  const [workoutPlan, setWorkoutPlan] = useState(null);
  // Стиль: интерактивные карточки упражнений
};
```

```
// Анализатор формы (классовый компонент)
class FormAnalyzer extends React.Component {
  // Стиль: видео плеер с оверлеем рекомендаций
}
```

```
jsx
// Планировщик путешествий (функциональный компонент)
const TravelPlanner = ({ budget, preferences }) => {
  const [itinerary, setItinerary] = useState([]);
  // Стиль: интерактивная карта с маркерами
};
```

```
// Генератор изображений (классовый компонент)
class DestinationVisualizer extends React.Component {
  // Стиль: галерея с lazy loading
}
```

```
jsx
// Монитор парковки (функциональный компонент)
const ParkingMonitor = ({ cameraFeeds }) => {
  const [occupancy, setOccupancy] = useState({});
  // Стиль: схема парковки с реальными обновлениями
};

// Распознаватель номеров (классовый компонент)
class LicensePlateRecognizer extends React.Component {
  // Стиль: интерфейс с preview изображения
}
```