

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
[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).
- 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; 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>

- Продemonстрировать работу со state (<https://metanit.com/web/react/2.4.php>)
- Продemonстрировать 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.bootstrapcdn.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}

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	Microsoft Azure Custom Vision - классификация товаров для животных Hugging Face API - чат-консультант по уходу за питомцами API: https://azure.microsoft.com	1. PetProductMatcher (функциональный с useContext) - подбор товаров для питомца 2. InventoryManager (кл	Яркий и дружелюбный дизайн с пастельными тонами. Интерактивные элементы с	XMLHttpRequest: Отправка больших заказов поставщикам с прогрессом setTimeout: 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.	t.com/services/cognitive-services/custom-vision-service , https://huggingface.co/inference-api	ассовый) - управление складскими запасами 3. SupplierDashboard (стрелочная 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 Promise: 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.	Amazon Rekognition - распознавание лиц для регистрации Google Dialogflow - чат-бот для службы поддержки API: https://aws.amazon.com/rekognition , https://cloud.google.com/dialogflow	1. FaceCheckIn (функция) - онлайн с useMemo) - система регистрации по лицу 2. RoomAllocation (класс) - распределение номеров 3. ServiceManager (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.	XMLHttpRequest: Отправка отзывов с отслеживанием прогресса и возможностью повтора setTimeout: Организовать автоматическое обновление статуса номеров каждые 30 секунд Promise: 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.	Google MediaPipe - анализ позы во время упражнений OpenAI API - генерация персональных тренировок API: https://google.github.io/mediapipe , https://openai.com	1. PoseAnalyzer (функциональный с useCallback) - анализ правильности выполнения упражнений 2. GroupSchedule (классовый) - расписание групповых занятий 3. ProgressTracker (стрелочная функция) - отслеживание прогресса клиентов Обработчики: onPoseCapture, onScheduleUpdate, onProgressLog, onPayment, onMembership, onTrainerAssign, onHealthData	Энергичный спортивный дизайн с яркими акцентами. Анимации движения. Карточки тренировок с градиентами.	XMLHttpRequest: Загрузка видео тренировок с отображением прогресса загрузки setTimeout: Реализовать таймер обратного отсчета для начала групповых занятий Promise: 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 отдыхать. При этом	Google Places API + AI - рекомендации мест OpenAI DALL-E - генерация изображений направлений API: https://developers.google.com/maps/documentation	1. DestinationFinder (функциональный с useReducer) - интеллектуальный подбор направлений 2. BookingSystem (класс	Яркий travel-дизайн с изображениями направлений. Интерактивная карта. Слайдеры с фотографиями.	XMLHttpRequest: 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)
		ему демонстрируются различные варианты, включающие страну проживания, особенности местного климата, имеющиеся отели разного класса. Наряду с этим, обсуждается возможная длительность пребывания и стоимость путевки. 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. Фирма работает с несколькими отелями в нескольких странах. Путевки продаются на одну, две или четыре недели. Стоимость путевки зависит от длительности тура и отеля.	ation/places , https://openai.com/dall-e	овый) - система бронирования туров 3. TravelPlanner (стрелочная функция) - планирование маршрута Обработчики: onDestinationSelect, onDatePick, onHotelBook, onItinerary, onBudgetSet, onReview, onTravelAlert		setTimeout: Автоматическое сохранение черновика заявки на тур каждые 2 минуты Promise: Цепочка для подбора тура: поиск вариантов → проверка availability → расчет стоимости
5.	Автостоянка	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. Уникальный номер парковочного места имеет числовой формат с ограничением не более 999.	AWS Rekognition - распознавание номерных знаков Google Cloud Video Intelligence - мониторинг парковки API: https://aws.amazon.com/rekognition , https://cloud.google.com/video-intelligence	1. LicensePlateReader (функциональный с useRef) - автоматическое распознавание номеров 2. ParkingMonitor (классовый) - мониторинг занятости мест 3. BillingManager (стрелочная функция) - управление платежами Обработчики: onPlateDetect, onSpaceReserve, onPayment, onAccessControl, onViolation,	Технический минималистичный дизайн. Схема парковки с интерактивными элементами. Статусные индикаторы.	XMLHttpRequest: Мониторинг статуса камер видеонаблюдения с периодическим опросом setTimeout: Автоматическое снятие брони парковочного места при неоплате в течение 15 минут Promise: Процесс въезда: распознавание номера → проверка прав →

№	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 паркуются 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.	Google Routes API - оптимизация маршрутов IBM Watson NLP - анализ грузовых накладных API: https://developers.google.com/maps/documentation/routes , https://www.ibm.com/cloud/watson-natural-language-understanding	1. RouteOptimizer (функциональный с useMemo) - оптимизация маршрутов доставки 2. CargoTracker (классовый) - отслеживание грузов 3. DriverManager (стрелочная функция) - управление водителями Обработчики: onRouteCalculate, onCargoTrack, onDriverAssign, onDeliveryConfirm, onDocumentScan, onFuelLog, onVehicleMaintenance	Профессиональный корпоративный дизайн. Интерактивные карты. Таблицы с сортировкой.	XMLHttpRequest : Отслеживание местоположения груза через периодические запросы к GPS-трекеру setTimeout : Автоматическое уведомление о задержке доставки при превышении планового времени Promise : 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	Clarifai API - распознавание деталей по фото OpenAI API - чат-помощник по подбору запчастей	1. PartRecognition (функциональный с useState) - загрузка фото детали для распознавания 2.	Технический дизайн с индустриальной цветовой схемой (серый, синий, оранжевый). Карточки	XMLHttpRequest : 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>	<p>API: https://clarifai.com, https://openai.com</p>	<p>SupplierCatalog (классовый) - каталог поставщиков с фильтрацией</p> <p>3. PriceTracker (стрелочная функция) - отслеживание изменения цен</p> <p>Обработчики: onImageUpload, onSearch, onFilter, onPriceChange, onSupplierSelect, onPartAdd, onError</p>	<p>деталей с hover-эффектами.</p> <p>Progress bar для загрузки изображений.</p>	<p>setTimeout: Автоматическое обновление цен каждые 5 минут</p> <p>Promise: Цепочка оформления заказа: проверка наличия → резервирование → отправка уведомления поставщику</p>
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>	<p>Google Vision AI - распознавание животных по фото</p> <p>IBM Watson NLP - анализ отзывов посетителей</p> <p>API: https://cloud.google.com/vision, https://www.ibm.com/cloud/watson-natural-language-understanding</p>	<p>1. AnimalRecognition (функциональный с useReducer) - идентификация животных по фото</p> <p>2. HabitatManager (классовый) - управление помещениями и вольерами</p> <p>3. FeedingSchedule (стрелочная функция) - расписание кормления</p> <p>Обработчики: onAnimalIdentify, onHabitatUpdate, onFeedingTime, onStaffAssign, onFoodOrder,</p>	<p>Естественная цветовая палитра (зеленый, коричневый, бежевый).</p> <p>Анимации плавных переходов.</p> <p>Карточки животных с градиентными фонами.</p>	<p>XMLHttpRequest: Загрузка видео наблюдений за животными с прогрессом</p> <p>setTimeout: Автоматическое напоминание о времени кормления животных</p> <p>Promise: Бронирование экскурсии: проверка расписания → бронирование гида → формирование группы</p>

№	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.	Google Cloud Vision - анализ изображений экспонатов OpenAI GPT - генерация описаний экспонатов API: https://cloud.google.com/vision , https://openai.com	1. ExhibitScanner (функциональный с useRef) - сканирование и анализ экспонатов 2. TourPlanner (классовый) - планирование экскурсионных маршрутов 3. ConservationTracker (стрелочная функция) - отслеживание состояния экспонатов Обработчики: onExhibitScan, onTourCreate, onConservationUpdate, onVisitorFlow, onAudioGuide, onLightingControl, onSecurityCheck	Классический музейный дизайн с темными тонами и акцентами. Типографика с засечками. Слайдеры для галереи.	XMLHttpRequest: Загрузка аудиогидов с отслеживанием прогресса setTimeout: Автоматическое переключение слайдов в виртуальной экспозиции Promise: 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.	Google Vision AI - распознавание товаров и штрих-кодов API: https://cloud.google.com/vision OpenAI API - консультант по средствам и их применению API: https://openai.com	1. ProductScanner (функциональный с useCallback) - сканер товаров и штрих-кодов 2. InventoryOptimizer (классовый) - оптимизация складских запасов 3. SafetyAdvisor (стрелочная функция) - консультант по безопасному	Чистый гигиенический дизайн с синими и белыми тонами. Четкая категоризация товаров. Предупреждающие цветовые коды.	XMLHttpRequest: Отправка данных инвентаризации с отслеживанием прогресса setTimeout: Автоматическое применение скидок к товарам с истекающим сроком годности Promise: 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.	Google Vision AI - распознавание книг по обложкам API: https://cloud.google.com/vision OpenAI API - рекомендательная система и отзывы API: https://openai.com	1. BookRecommender (функциональный с useMemo) - интеллектуальная рекомендация книг 2. ReadingTracker (классовый) - отслеживание прогресса чтения 3. ReviewAnalyzer (стрелочная 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.	XMLHttpRequest : Загрузка превью книг и отрывков с индикацией прогресса setTimeout : Автоматическое сохранение позиции чтения в электронной книге Promise : 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.	Google Vision AI - анализ внешнего состояния автомобилей или распознавание автомобилей по фото для подбора аналогов API: https://cloud.google.com/vision	1. CarConfigurator (функциональный с useReducer) - configurator автомобилей с опциями 2.	Премиальный дизайн с хромовыми акцентами. 3D визуализации автомобилей. Интерактивные configurators.	XMLHttpRequest : Загрузка configurator автомобиля с интерактивным выбором опций setTimeout : Автоматическое

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
			OpenAI API - консультант по выбору автомобиля API: https://openai.com	TestDriveScheduler (классовый) - планирование тест-драйвов 3. TradeInEvaluator (стрелочная функция) - оценка автомобилей с пробегом Обработчики: onCarConfigure, onTestDriveSchedule, onTradeInEvaluate, onFinanceCalculate, onColorSelect, onFeatureCompare, onDelivery		обновление акций и специальных предложений Promise: 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.	Google Vision AI - анализ фотографий объектов недвижимости API: https://cloud.google.com/vision OpenAI API - генерация описаний объектов и маркетинговых материалов, чат-бот для первичной консультации API: https://openai.com	1. PropertyMatcher (функциональный с useContext) - подбор объектов по критериям клиента 2. DealFlow (классовый) - управление потоком сделок 3. MarketAnalyzer (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.	XMLHttpRequest: Загрузка документов по сделке (договоры, акты) setTimeout: Автоматическое напоминание о просроченных платежах по аренде Promise: 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	Google Cloud Healthcare API - анализ медицинских назначений Microsoft Azure Cognitive Services - распознавание рецептов API: https://cloud.google.com/healthcare-api , https://azure.microsoft.com/services/cognitive-services	1. PrescriptionReader (функциональный с useState) - сканирование и анализ рецептов 2. DrugInventory (классовый) - управление запасами лекарств 3. InteractionChecker (стрелочная функция) - проверка взаимодействий препаратов Обработчики: onPrescriptionScan, onDrugSearch, onInteractionCheck, onInventoryAlert, onExpiryTrack, onOrderPlace, onCustomerConsult	Медицинский чистый дизайн. Четкая типографика. Цветовые коды для категорий лекарств.	XMLHttpRequest : Отслеживание статуса доставки лекарств курьерской службой setTimeout : Автоматическое напоминание о необходимости рецепта для определенных препаратов Promise : 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: фию, дату рождения, адрес и т.д. Медицинский центр предоставляет определенные типы услуг согласно фиксированному прайс-листу. Каждому типу относится несколько видов услуг, каждая из которой характеризуется названием и стоимостью. Известно, что один клиент в разное время может совершать несколько заказов. Также известно, что за одно	Google Cloud Healthcare NLP - анализ медицинских записей IBM Watson Assistant - виртуальный медицинский ассистент API: https://cloud.google.com/healthcare-api , https://www.ibm.com/cloud/watson-assistant	1. MedicalRecordAnalyzer (функциональный с useContext) - анализ медицинских карт 2. AppointmentScheduler (классовый) - система записи на прием 3. TreatmentPlanner (стрелочная функция) - планирование лечения Обработчики:	Спокойный медицинский дизайн. Доступные интерфейсы. Четкая визуальная иерархия.	XMLHttpRequest : Загрузка медицинских изображений и результатов анализов setTimeout : Автоматическое напоминание о приеме лекарств по расписанию Promise : 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	Google Vision AI - анализ эмоций зрителей OpenAI API - рекомендация фильмов API: https://cloud.google.com/vision , https://openai.com	1. EmotionAnalyzer (функциональный с useReducer) - анализ реакций зрителей 2. MovieRecommender (классовый) - система рекомендаций фильмов 3. ShowtimeManager (стрелочная функция) - управление сеансами Обработчики: onEmotionDetect, onMovieRecommend, onShowtimeSelect, onTicketBook, onSeatSelect, onTrailerPlay, onRating	Кинематографический дизайн с темной темой. Эффекты затемнения. Анимации появления.	XMLHttpRequest : Загрузка трейлеров фильмов с отображением прогресса буферизации setTimeout : Автоматическое освобождение забронированных билетов при неоплате Promise : Цепочка покупки билетов: выбор места → бронирование → оплата → отправка билета
17.	Автосервис	Автоматизация работы компании, занимающейся ремонтом автомобилей	Clarifai API - диагностика по фото повреждений Google Cloud AutoML - предсказание стоимости ремонта API: https://clarifai.com , https://cloud.google.com/automl	1. DamageAssessor (функциональный с useRef) - оценка повреждений по фото 2. RepairEstimator (классовый) - расчет стоимости ремонта 3. ServiceScheduler (стрелочная функция) - планирование	Технический дизайн с индустриальными элементами. Диагностические диаграммы. Статусные индикаторы ремонта.	XMLHttpRequest : Мониторинг статуса ремонта через периодические запросы setTimeout : Автоматическое уведомление о готовности автомобиля Promise : Процесс

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
				ремонт 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.	Google Cloud Vision - анализ серийных номеров оборудования API: https://cloud.google.com/vision OpenAI API - генерация диагностических отчетов API: https://openai.com	1. DeviceDiagnostic (функциональный с useState) - автоматическая диагностика оборудования 2. RepairTracker (классовый) - отслеживание статуса ремонта 3. ClientPortal (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.	XMLHttpRequest : Загрузка диагностических отчетов и логов оборудования setTimeout : Автоматическое обновление статуса заявки в реальном времени Promise : 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.	Google Vision AI - распознавание ингредиентов по фото API: https://cloud.google.com/vision OpenAI API - генерация рецептов и	1. PizzaCustomizer (функциональный с useReducer) - конструктор пиццы с ингредиентами 2. OrderTracker (классов	Яркий итальянский стиль с красными и желтыми акцентами. Анимации добавления	XMLHttpRequest : Отслеживание статуса заказа и местоположения курьера setTimeout : Автоматическая отмена заказа

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
			рекомендаций API: https://openai.com	ый) - отслеживание заказов и доставки 3. MenuManager (стрелочная функция) - управление меню и акциями Обработчики: onPizzaBuild, onOrderTrack, onMenuUpdate, onIngredientSelect, onDeliveryAssign, onPromoApply, onFeedback	ингредиентов. Карточки пицц с фотографиями.	при длительном ожидании курьера Promise: 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.	AWS Rekognition - проверка документов и водительских прав API: https://aws.amazon.com/rekognition Google Routes API - планирование оптимальных маршрутов API: https://developers.google.com/maps/documentation/routes	1. DocumentVerifier (функциональный с useRef) - верификация документов клиентов 2. FleetManager (классовый) - управление автопарком 3. RentalCalculator (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.	XMLHttpRequest: Загрузка документов для аренды (права, паспорт) с валидацией setTimeout: Автоматическое напоминание о возврате автомобиля за 2 часа до окончания аренды Promise: Цепочка бронирования: проверка доступности → бронирование → подготовка документов
21.	Клининговая компания	Клининговая компания — организация, которая предлагает достаточно	Google Cloud Vision - оценка состояния помещений до/после	1. CleanlinessAssessor (функциональный с	Чистый минималистичный дизайн с	XMLHttpRequest: 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: https://cloud.google.com/vision OpenAI API - генерация индивидуальных планов уборки API: https://openai.com	useContext) - оценка чистоты помещений 2. SchedulePlanner (классовый) - планирование графиков уборки 3. QuoteGenerator (стрелочная функция) - расчет стоимости услуг Обработчики: onAssessment, onSchedulePlan, onQuoteGenerate, onTeamAssign, onSupplyOrder, onQualityCheck, onClientReview	бело-синей палитрой. Интуитивные формы заказа. Индикаторы выполнения задач.	выполненной уборке setTimeout: Автоматическое напоминание клиенту за день до визита клинеров Promise: 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.	Amazon Rekognition - анализ кожи и рекомендации процедур API: https://aws.amazon.com/rekognition OpenAI API - консультант по уходу за кожей API: https://openai.com	1. SkinAnalyzer (функциональный с useMemo) - анализ состояния кожи 2. AppointmentBooker (классовый) - система записи на процедуры 3. TreatmentPlanner (стрелочная функция) - планирование курса процедур Обработчики: onSkinAnalyze, onAppointmentBook, onTreatmentPlan, onProductRecommend, onProgressTrack,	Элегантный дизайн с пастельными тонами. Плавные анимации переходов. Карточки процедур с изображениями.	XMLHttpRequest: Загрузка фотографий "до/после" процедур setTimeout: Автоматическое напоминание о следующем визите по расписанию процедур Promise: Цепочка записи: консультация → выбор процедуры → бронирование → подготовка

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
23.	Фабрика игрушек	Фабрика производит игрушки нескольких видов и моделей. Все виды игрушек изготавливаются по заказам оптовых продавцов.	Google Cloud Vision - контроль качества продукции API: https://cloud.google.com/vision OpenAI DALL-E - генерация дизайнов игрушек API: https://openai.com/dall-e	1. QualityInspector (функциональный с useReducer) - автоматический контроль качества 2. ProductionTracker (классовый) - отслеживание производственного процесса 3. DesignStudio (стрелочная функция) - создание дизайнов игрушек Обработчики: onQualityCheck, onProductionTrack, onDesignCreate, onMaterialOrder, onShipment, onClientApprove, onInventory	Яркий детский дизайн с разноцветными элементами. Интерактивные элементы производства. Визуализация производственной линии.	XMLHttpRequest : Мониторинг статуса производственной линии через API оборудования setTimeout : Автоматическое обновление статуса заказов в производственной очереди Promise : 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.	Google Vision AI - распознавание десертов по фото для каталога API: https://cloud.google.com/vision OpenAI API - генерация рецептов и описаний десертов API: https://openai.com	1. CakeDesigner (функциональный с useState) - конструктор тортов и десертов 2. OrderManager (классовый) - управление заказами и доставкой 3. RecipeBook (стрелочная функция) - база рецептов с поиском	Сладкая пастельная палитра с акцентами шоколадного и кремового. Анимации украшений. Карточки десертов с аппетитными фото.	XMLHttpRequest : Загрузка изображений готовых изделий для каталога setTimeout : Автоматическое уведомление о готовности индивидуального заказа Promise : 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.	Google Cloud Vision - распознавание продуктов по фото API: https://cloud.google.com/vision OpenAI API - рекомендации рецептов на основе покупок API: https://openai.com	1. ProductRecognizer (функциональный с useCallback) - распознавание продуктов по изображению 2. DeliveryTracker (классовый) - отслеживание доставки в реальном времени 3. SmartCart (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.	XMLHttpRequest : Отслеживание местоположения курьера в реальном времени setTimeout : Автоматическое применение акций к товарам в корзине Promise : Процесс доставки: сбор заказа → упаковка → доставка → подтверждение получения
26.	Агентство недвижимости	Автоматизация работы компании, занимающейся продажей и арендой объектов недвижимости.	Google Vision AI - анализ фотографий объектов недвижимости API: https://cloud.google.com/vision OpenAI API - генерация	1. PropertyEvaluator (функциональный с useContext) - оценка стоимости недвижимости 2.	Элегантный дизайн с золотыми и темно-синими акцентами. Галереи фотографий	XMLHttpRequest : Загрузка документов по сделке (договоры, акты) setTimeout : Автоматическое

№	Web сайт	Описание (что должно быть обеспечено)	AI API	React компоненты/ обработчики (пример)	CSS требования	Асинхронность (пример)
			описаний объектов API: https://openai.com	DealManager (классовый) - управление сделками и документами 3. VirtualTour (стрелочная функция) - виртуальные туры по объектам Обработчики: onPropertyEvaluate, onDealManage, onTourStart, onDocumentUpload, onClientMatch, onViewingSchedule, onContract	объектов. Интерактивные планы помещений.	напоминание о просроченных платежах по аренде Promise: 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.	Google Document AI - обработка страховых документов API: https://cloud.google.com/document-ai OpenAI API - консультант по страховым продуктам API: https://openai.com	1. RiskCalculator (функциональный с useMemo) - расчет страховых рисков 2. PolicyManager (классовый) - управление страховыми полисами 3. ClaimProcessor (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.	XMLHttpRequest: Загрузка сканов документов для оформления страховки setTimeout: Автоматическое продление страховки при согласии клиента Promise: 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.	Google Vision AI - контроль качества мебели API: https://cloud.google.com/vision OpenAI API - дизайнер мебели и планировщик помещений API: https://openai.com	1. FurnitureDesigner (функциональный с useRef) - 3D конструктор мебели 2. ProductionScheduler (классовый) - планирование производства 3. QualityController (стрелочная функция) - контроль качества продукции Обработчики: onFurnitureDesign, onProductionSchedule, onQualityControl, onMaterialCalculate, onClientConsult, onShipmentTrack, onInstallation	Естественные деревянные тона в дизайне. 3D превью мебели. Интерактивные планировки помещений.	XMLHttpRequest : Мониторинг статуса заказа на производстве setTimeout : Автоматическое уведомление о готовности мебели к отгрузке Promise : Цепочка заказа: проектирование → производство → доставка → сборка
29.	Ремонт одежды	Автоматизация работы фирмы, занимающейся ремонтом одежды или обуви.	Google Vision AI - анализ повреждений одежды API: https://cloud.google.com/vision OpenAI API - рекомендации по ремонту и уходу API: https://openai.com	1. DamageAnalyzer (функциональный с useState) - анализ повреждений одежды 2. RepairEstimator (классовый) - расчет стоимости и сроков ремонта 3. MaterialManager (стрелочная функция) - управление материалами и фурнитурой Обработчики:	Текстильный дизайн с узорами и фактурами. Четкие формы приема заказов. Визуализация процессов ремонта.	XMLHttpRequest : Загрузка фотографий повреждений для оценки стоимости ремонта setTimeout : Автоматическое напоминание о готовности заказа Promise : 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.	Ветеринар ная клиника	Клиентом ветеринарной клиники является любой человек, который хотя бы раз воспользовался его услугами для своего домашнего питомца.	Google Vision AI - анализ медицинских снимков животных API: https://cloud.google.com/vision OpenAI API - консультант по уходу за животными API: https://openai.com IBM Watson NLP - анализ симптомов и истории болезни API: https://www.ibm.com/cloud/watson-natural-language-understanding	1. PetHealthAnalyzer (функциональный с useReducer) - анализ здоровья питомцев 2. VaccineTracker (классовый) - отслеживание вакцинации и процедур 3. EmergencyAlert (стрелочная функция) - система экстренных уведомлений Обработчики: onHealthAnalyze, onVaccineTrack, onEmergencyAlert, onAppointmentBook, onPrescription, onPetProfile, onReminder	Дружелюбный ветеринарный дизайн с мягкими тонами. Интуитивные карточки пациентов. Четкая система уведомлений.	XMLHttpRequest : Загрузка результатов анализов и рентгеновских снимков setTimeout : Автоматическое напоминание о следующем визите к ветеринару Promise : Цепочка приема: запись → осмотр → диагностика → назначение лечения
31.	Театр	Автоматизация работы театра.	Google Vision AI - анализ заполняемости зала или распознавание эмоций зрителей во время спектакля API: https://cloud.google.com/vision OpenAI API - генерация описаний спектаклей и рецензий или	1. SeatSelector (функциональный с useContext) - интерактивный выбор мест 2. PerformanceManager (классовый) - управление репертуаром и	Театральный дизайн с бархатными текстурами и золотыми акцентами. Эффектные шрифты. Анимации занавеса.	XMLHttpRequest : Загрузка видео репетиций и дополнительных материалов setTimeout : 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: https://openai.com	расписанием 3. CastingDirector (стре лочная функция) - база данных актеров и ролей Обработчики: onSeatSelect, onPerformanceManage, onCasting, onTicketBook, onRehearsalSchedule, onCostume, onReview		Promise: 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([]);
  // Стил: чат-интерфейс с градиентным фоном
};

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

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

```

```

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

```

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

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

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

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

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

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

jsx

// Монитор парковки (функциональный компонент)

```
const ParkingMonitor = ({ cameraFeeds }) => {  
  const [occupancy, setOccupancy] = useState({});
```

// Стил: схема парковки с реальными обновлениями
};

// Распознаватель номеров (классовый компонент)

```
class LicensePlateRecognizer extends React.Component {
```

// Стил: интерфейс с preview изображения
}