

КОНКУРСНОЕ ЗАДАНИЕ

(Веб-дизайн и разработка)

Конкурсное задание включает в себя следующие разделы:

1. Формы участия в конкурсе.
2. Общее время на выполнение задания.
3. Задание для конкурса.
4. Модули задания и необходимое время.
5. Критерии оценки.
6. Приложения к Конкурсному заданию.

1. **ФОРМЫ УЧАСТИЯ В КОНКУРСЕ:** Индивидуальный конкурс.
2. **ОБЩЕЕ ВРЕМЯ НА ВЫПОЛНЕНИЕ ЗАДАНИЯ:** 8 ч.
3. **ЗАДАНИЕ ДЛЯ КОНКУРСА.**

Конкурсное задание состоит из четырёх модулей. Результат выполнения каждого модуля не влияет на выполнение задания других модулей.

4. **МОДУЛИ ЗАДАНИЯ И НЕОБХОДИМОЕ ВРЕМЯ** (Таблица 1).

Таблица 1.

Наименование модуля		Соревновательный день (С1, С2, С3)	Время на задание
A	Проектирование	С1	3 часа
B	Вёрстка и дизайн	С1	3 часа
C	Программирование на стороне клиента	С1	3 часа
D	Программирование на стороне сервера	С2	3 часа

Модуль 1: Проектирование.

ВВЕДЕНИЕ

К вам обратился заказчик, который знает, чего он хочет, но в силу своей профессии он не совсем компетентен в веб-технологиях на данный момент. Поэтому он не знает, как лучше сделать тот или иной функционал.

Ваша задача разработать ТЗ на основе требований заказчика. При разработке ТЗ вам нужно опираться на несколько факторов:

1. В дальнейшем разработчик должен успеть разработать проект на основе вашего ТЗ за 12 часов, 4 из которых у него будет на верстку и дизайн проекта и 8 на серверное и клиентское программирование.

2. Требования, описанные в ТЗ должны быть современными и удобными для пользователей.

3. ТЗ должно содержать: пояснительную часть, схему базы данных, модель предметной области, сценарии работы пользователя в информационной системе, макеты интерфейсов (UX-дизайн, от 5 штук), требования к:

3.1. Фреймворкам и языкам программирования.

3.2. Графическому оформлению.

3.3. Срокам выполнения.

3.4. Аппаратному-техническому обеспечению.

3.5. Производительности системы (frontend, backend).

3.6. Взаимодействию с внешними системами (в случае их наличия).

3.7. Описание взаимодействия подсистем (компонентов).

3.8. Информационной безопасности.

3.9. Доступности системы для людей с ограниченными возможностями.

4. Оформление текста ТЗ:

4.1. Шрифт: Обычный текст: Times New Roman. Обычный текст: 12 пт, Заголовки: 16 пт, 14 пт.

4.2. Межстрочный интервал: 1.5

4.3. Абзацный отступ: 1 см.

4.4. Поля: слева 3 см, справа 1.5 см, сверху и снизу 2см.

5. Решения, описанные в тексте ТЗ, должны быть обоснованы.

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

«Центр социальной поддержки населения» (далее – Центр) является некоммерческой организацией социальной сферы. Центр создан для оказания услуг в сфере социальной поддержки, оказываемой:

- детям (различные пособия семьям с детьми; услуги, предоставляемые семьям и детям);
- многодетным семьям (выдача удостоверений многодетной семьи, различные пособия, денежные выплаты, оплата услуг ЖКХ и др.);
- ветеранам и инвалидам боевых действий (различные денежные выплаты, комплексы услуг);
- малоимущим семьям;
- гражданам за особые заслуги перед Отечеством;
- гражданам, имеющим право на льготы (учащимся, пенсионерам, работникам спецгосучреждений и др.);
- молодым семьям (программы улучшения жилищных условий и др.);

Деятельность Центра занятости осуществляется в соответствии с действующим законодательством Российской Федерации, Административными регламентами Федеральной службы по труду и занятости и иными нормативными правовыми актами Российской Федерации.

Сотрудники Центра обратились к вам с просьбой создать веб-сервис для организации событий Центра, направленных на обеспечение социальных гарантий в области занятости населения.

Веб-сервис должен:

- предоставлять полную информацию о видах социальной поддержки: категория граждан (несовершеннолетний, многодетная семья, малоимущая семья, гражданин, имеющий право на льготы, молодая семья), виды социальной поддержки (пособия, услуги (уход, временное проживание, трудоустройство), выплаты, государственные соц. программы), необходимая документация;
- позволять пользователям находить необходимую информацию по социальной поддержке;
- предоставлять администраторам возможность управлять пользователями;

Основные ролевые группы и их состав:

1) Пользователи сервиса:

- Незарегистрированный Посетитель. Пользователь, который не имеет учетную запись и не авторизовался в системе. Функциональные возможности: просмотр информационных страниц, взаимодействие с формами регистрации и авторизации.

- Зарегистрированный Посетитель. Пользователь, который имеет учетную запись в системе и авторизовался в системе. Функциональные возможности: просмотр текстовых информационных страниц, поиск видов социальной поддержки, подача заявки, редактирование данных профиля.

2) Пользователи системы управления:

- Сотрудник Центра. Функциональные возможности: регистрация на сервисе, авторизация в системе, добавление и редактирование видов социальной поддержки, настройка, добавление фотографий, просмотр списка зарегистрированных посетителей сервиса. Сотрудник Центра может добавлять информацию только после одобрения администратором. Информация о видах социальной поддержки, добавленная или отредактированная Сотрудником Центра, должна пройти согласование администратором.

- Администратор системы. Функциональные возможности: изменение роли Посетитель/Сотрудник, формирование списка заявок на конкретный вид социальной поддержки для рассылки информационных сообщений на электронную почту пользователей, просмотр и одобрение информации о видах социальной поддержки, получение статистики о Посетителе/Сотруднике (количество, оценка работы сотрудника, список отзывов о Центре).

Ваш проект должен корректно открываться со следующих устройств:

- мобильный телефон 375 x 820px;
- компьютеры с шириной экрана от 1200px.

Следует обратить внимание, что дизайн приложения должен быть удобен и понятен для использования, соответствовать современным тенденциям, а также учитывать особенности платформ, для которых создается сайт. При разработке дизайна необходимо использовать брендбук Центра. Также необходимо позаботиться о защите проекта: от взлома, от несанкционированного доступа к административным функциям и т.д.

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Разработанное ТЗ должно быть сохранено на сервере в папке xxxx-m1, где xxxx – ваш логин, все дополнительные материалы, используемые в ТЗ должны быть так же сохранены в виде файлов в папке xxxx-m1/files.

Модуль 2: Вёрстка и дизайн.

ВВЕДЕНИЕ

Вам предоставляется готовое ТЗ на основе которого вам необходимо сверстать все необходимые веб-страницы. На каждой странице должен присутствовать логотип Центра социальной поддержки населения. Дизайн проекта должен быть в цветовой гамме Центра социальной поддержки населения (см. брендбук)

Примените все свои навыки верстки чтобы сверстать страницы в соответствии с современными требованиями. Уделите внимание юзабилити, анимации и интерактивности, чтобы использование сервиса было удобными и простым.

В дальнейшем программисты будут реализовывать требуемый по ТЗ функционал опираясь на сверстанные вами веб-страницы.

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ниже приведен список веб-страниц и их структуры, которые вы должны сверстать:

- Страница регистрации – главная страница для неавторизованного пользователя
 - Страница авторизации
 - Главная страница
 - Личный кабинет Посетителя
 - Личные данные
 - Моя социальная поддержка
 - Личный кабинет Сотрудника
 - Личные данные
 - Личный кабинет Администратора
 - Меры социальной поддержки
 - Управление пользователями (Посетитель/Сотрудник)
 - Страница создания Сотрудника
 - Страница Виды социальной поддержки
- Ваша верстка должна быть валидной.

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Сверстанные веб-страницы должны быть размещены на сервере в папке xxxx-m2, и доступны по адресу <http://xxxx-m2.wsr.ru>, где xxxx – ваш логин.

В процессе работы вы можете пользоваться следующими библиотеками и фреймворками:

- Bootstrap 3
- Bootstrap 4

Модуль 3: Программирование на стороне клиента.

ВВЕДЕНИЕ

Компания Tehmex разработала API для проверки компьютеров в сборе, чтобы предотвратить продажу машин с несовместимыми частями. Чтобы сделать это доступным для пользователей, ваша задача сейчас состоит в том, чтобы *создать веб-сайт*, отображаемый на стороне клиента, который использует этот API, то есть с использованием фреймворков JavaScript.

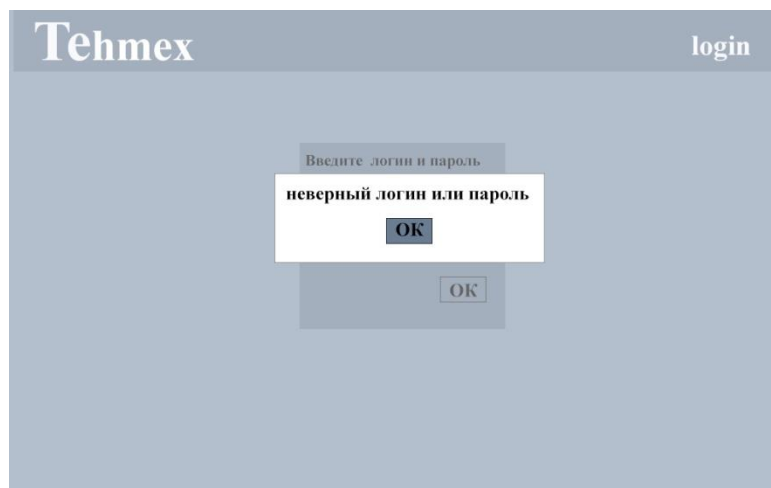
ОПИСАНИЕ ПРОЕКТА И ЗАДАЧИ

Запросы, которые будут использоваться

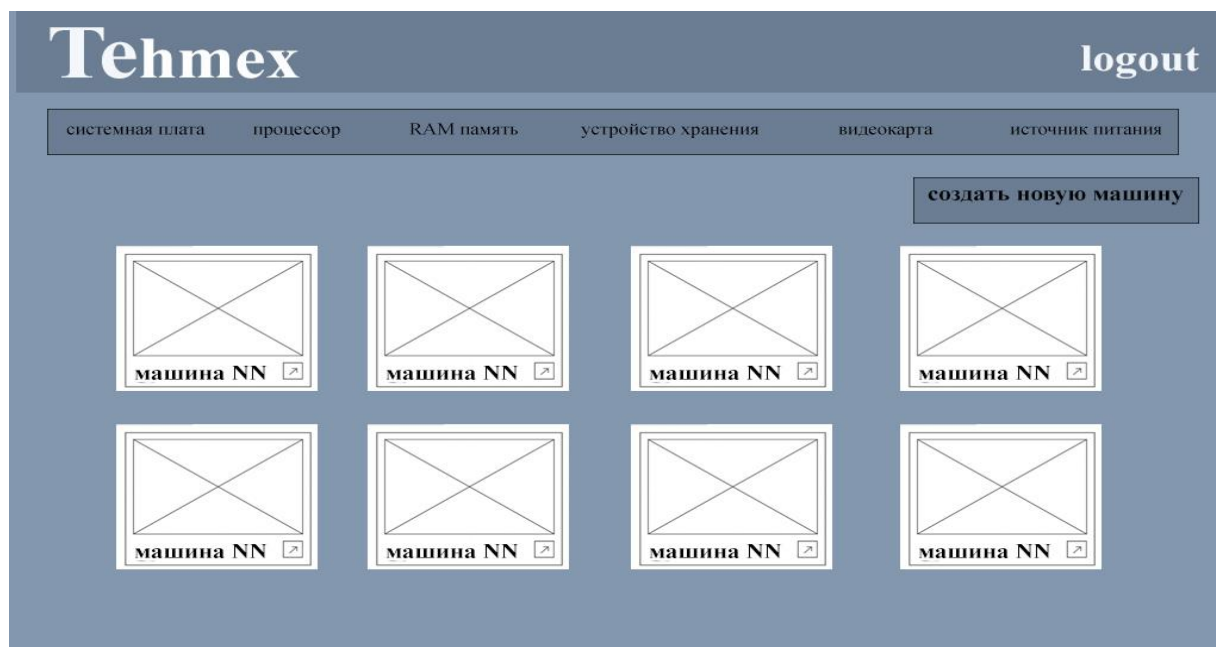
Типы частей, предоставляемых API, и их соответствующие конечные точки (связанные с <http://zabmv-m3.wsr.ru/tehmex/api>):

- системная плата: MBoard
- процессор: CPU
- RAM память: RAM
- устройство хранения: S-devices
- видеокарта: G-cards
- источник питания: P-supplies
- ПК: PC

На вашем веб-сайте должен отображаться экран входа в систему всякий раз, когда неаутентифицированный пользователь пытается получить к нему доступ. Если учетные данные неверны, пользователю должно быть показано сообщение в модальном формате.



Кроме того, после успешного входа в систему должен появиться список готовых машин (конечных компьютеров), которые следует выделить, а отдельные части разделить на вкладки. При выборе направляющей машины должна быть кнопка «Новый». Вы можете следовать предложенному ниже макету, но не стесняйтесь предлагать лучший дизайн.



При нажатии на кнопку «Создать» на вкладке «Готовые машины» интерфейс экрана должен измениться и отобразить список в левой части экрана с типами деталей (см. В приложении «Приложение EndPoints.pdf»). При выборе «типа детали» детали выбранного типа должны быть представлены следующим образом:

- в списке (по умолчанию): только ваши изображения и имена;
- при наведении указателя мыши на элемент: название и бренд должны отображаться во всплывающем или подобном виде рядом с изображением и названием.

По умолчанию должен быть выбран тип детали.

Пример:



Монтаж компьютеров

Первоначально при сборке ПК будут отображаться области вставки деталей без каких-либо размещенных на них деталей. Единственные части, которые могут быть размещены в это время - это источники или материнские платы, поскольку другие части зависят от них. Когда в машине есть эти две части, для каждой новой размещенной части текущее состояние машины будет повторно проверяться через API (через запрос на проверку совместимости).

- POST (verify-compatibility): Проверьте совместимость между двумя или более частями, не обязательно всей машиной, однако всегда должны быть материнская плата и блок питания:
 - Обязательные параметры:
 - **MBoard Id:** ID материнской платы
 - **P-supplies Id:** ID источника питания
 - Необязательные параметры:
 - **CPU Id:** ID процессора
 - **RAM Id:** ID RAM
 - **RAM Amount:** объем оперативной памяти
 - **S-devices:** массив объектов JSON, содержащий:
 - S-devicesId: IDs запоминающих устройств
 - amount: количество устройств данной модели
 - **G-cards Id:** ID видеокарты
 - **G-cards Amount:** количество видеокарт
 - Ответ:
 - 200/OK: действительна вся модель машины (не содержит несовместимости и части работают вместе).
 - 400/BAD REQUEST: количество деталей было недопустимым или обнаружены несовместимости.
 - 401/UNAUTHORIZED: если токен не указан в заголовке авторизации.
 - 403/FORBIDDEN: если в заголовке авторизации указан неверный токен.
 - Другое: ошибка должна быть возвращена в формате JSON.

Вставка деталей выполняется перетаскиванием из списка на машину (в области, соответствующей детали). Завершение процесса создания машины возможно только в том случае, если на машине есть хотя бы одна деталь каждого

типа. Когда указатель мыши находится над областью вставки одной из частей, можно будет удалить эту часть, нажав кнопку «Удалить», которая должна появиться в правом верхнем углу.

При установке новой материнской платы все детали должны быть удалены из компьютера, чтобы избежать несовместимости и позволить процессу сборки перезапуститься. Когда вставляется какая-либо другая деталь, необходимо удалить ту часть, которая была на месте, то есть деталь необходимо заменить.

Когда создается новая машина, пользователь должен иметь возможность вставить изображение, и оно должно быть отправлено в API в формате base64.

Проверка смонтированных компьютеров

Если API определяет несовместимость на машине, последняя перемещенная часть должна иметь эффект встряхивания и удаляться переходным способом до появления списка, в дополнение к наличию красного фильтра и максимальной непрозрачности 60% (чтобы показать пользователю, что это было «Недействительное действие»). Изображение, представленное ниже, иллюстрирует процесс вставки данной несовместимой детали в машину, всегда соблюдая проверку API.

При вставке запоминающего устройства, уже имеющегося в машине, это движение должно считаться недопустимым и должен применяться эффект ошибки, описанный выше.

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Вы должны предоставить функциональный веб-сайт, доступный через сервер и по URL-адресу `xxxxx-m3.wsr.ru/alatech`, где `xxxxx` – ваш логин.

В API вам заведено 10 тестовых пользователей: `xxxxx-N`, где `xxxxx` - ваш логин, `N` - порядковый номер от 1 до 10. Пароль предоставлен вам на карточке участника вместе с доступом к серверу и является одинаковым для всех тестовых пользователей.

Модуль 4: Программирование на стороне сервера.

ВВЕДЕНИЕ

Компания по продаже и сборке компьютеров расширяет свой бизнес. Чтобы гарантировать, что ПК, проданные вашим клиентам, имеют совместимые детали, им нужна веб-служба, которая проверяет совместимость деталей.

Сервис должен уметь:

- зарегистрировать деталь;
- сборка ПК из деталей, совместимых друг с другом;
- предупреждать клиента о несовместимости деталей.

Регистрация всех ПК должна быть динамической для пользователя.

ГЛОССАРИЙ

- Socket: разъем, расположенный на материнской плате, который определяет электрический интерфейс для передачи данных к процессору и от него.
- TDP (Thermal Design Power): это максимальное количество тепла, выделяемого компьютерным чипом или компонентом (которым может быть процессор, видеокарта и т. д.), которое система охлаждения компьютера была спроектирована для рассеивания без какой-либо перегрузки.
- RAM (Random Access Memory): тип памяти в компьютерах для быстрой и случайной обработки. Обычно используется для активных данных во время работы пользователя.
- SLI/Crossfire: технологии, используемые видеокартами Nvidia и AMD, соответственно, для совместной работы и увеличения общей вычислительной мощности компьютера.
- PCI Express (Peripheral Component Interconnect Express): стандарт высокоскоростного последовательного соединения, присутствующий на материнских платах.

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧИ

Вы должны создать *API* с использованием *RНР-фреймворка* и базы данных.

Вы получите файлы *.json* с исходными данными, файл создания базы данных и изображения каждого из различных типов объектов, которые должны быть зарегистрированы в вашей системе, чтобы разрешить выполнение тестов.

Описание формата доставки

При сдаче тестового проекта должны быть соблюдены следующие требования:

- Ваш API должен быть доступен по адресу `xxxxx-m1.wsr.ru/Tehmex/api`. Где `xxxxx` - ваш логин.

Причины несовместимости

Ваш API должен сообщать о несовместимости при создании или редактировании компьютеров в следующих случаях:

- Тип сокета материнской платы отличается от типа сокета процессора;
- TDP процессора превышает максимальный TDP, поддерживаемый материнской платой;
- Тип ОЗУ на материнской плате отличается от типа ОЗУ;
- Количество RAM-памяти больше, чем количество слотов на материнской плате (на машине всегда должно быть, как минимум 1 RAM);
- Количество видеокарт больше, чем количество слотов PCI Express на материнской плате (на машине всегда должна быть как минимум 1 видеокарта);
- Количество запоминающих устройств типа SATA больше, чем количество слотов SATA на материнской плате;
- Количество запоминающих устройств типа M2 больше, чем количество слотов M2 на материнской плате;
- Общая сумма устройств хранения равна нулю (то есть должно быть, как минимум 1 устройство SATA или 1 устройство M2);
- Количество видеокарт меньше 1, и модель видеокарты не поддерживает SLI / Crossfire;
- Мощность блока питания меньше минимального значения мощности видеокарты, умноженного на количество видеокарт.

ДЕТАЛИ API

Аутентификация

По соображениям безопасности, маршруты входа и изображений должны быть общедоступными. При доступе к любому другому адресу, который не соответствует ранее упомянутым, API должен возвращать код состояния **HTTP 401 Unauthorized**, если заголовок **Authorization** отсутствует в запросе. Если заголовок есть, но он недействителен, должен быть возвращен код состояния **HTTP 403 Forbidden**.

Аутентификация пользователя в API должна выполняться с помощью токена, который будет отправляться на каждый запрос в заголовке авторизации. Этот токен должен быть:

- зашифрованный;
- формат: Bearer {token}

Информация об объектах

После аутентификации пользователя он должен иметь доступ ко всем маршрутам, которые позволят ему манипулировать следующими объектами (каждый объект должен иметь свой маршрут) для сборки ПК (которые будут отдельным маршрутом):

- Список материнских плат
 - Должен иметь имя, URL-адрес изображения, бренд, тип сокета, тип RAM, количество слотов RAM, максимальный TDP, количество слотов SATA, количество слотов M2, количество слотов PCI Express.
- Список процессоров
 - Должен содержать имя, URL изображения, бренд, тип сокета, количество физических ядер, базовую частоту, максимальную частоту, объем кеша, TDP.
- Список памяти RAM
 - Должно быть указано имя, URL изображения, бренд, объем памяти, тип оперативной памяти, частота.
- Список запоминающих устройств
 - Должен иметь имя, URL-адрес изображения, марку, тип устройства (HDD или SSD), объем памяти, тип ввода (SATA или M2).
- Список видеокарт
 - Должен иметь имя, URL-адрес изображения, марку, объем памяти, тип памяти (GDDR5 или GDDR6), минимальный блок питания, поддерживает SLI / Crossfire.
- Список источников питания
 - Должен иметь имя, URL изображения, марку, мощность (в ваттах), рейтинг 80 Plus (белый, бронзовый, серебряный, золотой, платиновый, титановый или неприменимо).
- Список брендов
 - Должно быть имя
- Управление всей машиной

- Должен иметь имя, URL изображения, материнскую плату, процессор, память (минимум 1 блок), устройства хранения (минимум 1 блок), видеокарты (минимум 1 блок), источник питания.

Все продукты должны иметь фотографии (фотографии представлены в медиа), и они должны быть доступны через запрос, указанную в разделе API ниже.

API

API всегда должен получать и возвращать данные в формате JSON. Адрес API: xxxxx-m1.wsr.ru/ Tehmex/api. Где xxxxx - ваш логин.

POST (<i>login</i>)	
Аутентификация пользователя	
Request	Response
Content-type: application/json Body: <pre>{ username: string < имя пользователя для аутентификации > password: string < пароль пользователя для аутентификации > }</pre>	1. Success: Status: 200/OK Content-type: application/json Body: <pre>{ token: string <token который разрешает пользователю доступ к конечным точкам системы.> }</pre> 2. Validation error: Status: 400/BAD REQUEST Content-type: application/json Body: <pre>{ message: “Неверные учетные данные” }</pre> 3. Если для пользователя уже активна аутентификация: Status: 403/FORBIDDEN Content-type: application/json Body: <pre>{ message: “Пользователь уже аутентифицирован” }</pre>

DELETE (<i>logout</i>)
Удаление аутентификации пользователя

Request	Response
Authorization: Bearer string <token >	<p>1. Выход подтвержден: Status: 200/OK Content-type: application/json Body: { message: “Успешный выход” }</p> <p>2. Если токен отсутствует в запросе: Status: 401/UNAUTHORIZED Content-type: application/json Body: { message: “Необходима аутентификация” }</p> <p>3. Если токен не соответствует действительной аутентификации: Status: 403/FORBIDDEN Content-type: application/json Body: { message: “Неверный токен” }</p>

<p>GET (motherboards, processors, ram-memories, storage-devices, graphic-cards, power-supplies, machines, brands)</p> <p>Списки материнских плат, процессоров, оперативной памяти, запоминающих устройств, видеокарт, блоков питания и целых машин соответственно.</p>	
Request	Response
<p>header: Authorization string < token ></p> <p>Parametrs: Параметры строки запроса, оба необязательные:</p> <ul style="list-style-type: none"> • pageSize: int < определяет размер каждой страницы поиска > (по умолчанию 10) • page: int < определяет текущую страницу поиска > (по умолчанию 1) 	<p>1. Fluxo normal: Status: 200/OK Content-type: application/json Body: [{ <свойства, описанные в подразделе «Информация об объектах» для указанной сущности> },]</p>

	<p><свойства, описанные в подразделе «Информация об объектах» для указанной сущности></p> <p>},</p> <p>...</p> <p>]</p> <p>2. Если токен отсутствует в запросе: Status: 401/UNAUTHORIZED Content-type: application/json Body: { message: “Необходима аутентификация” }</p> <p>3. Если токен не соответствует действительной аутентификации: Status: 403/FORBIDDEN Content-type: application/json Body: { message: “Неверный токен” }</p>
--	--

GET (search/{category}?q={q}) Поиск товаров	
Request	Response
<p>Authorization: Bearer string <token></p> <p>Параметры: Обязательные:</p> <ul style="list-style-type: none"> ● q: string < поисковая строка, которая должна быть применена к имени искомых элементов > ● category: string < тип части для поиска (должен соответствовать имени части категории, например материнские платы или процессоры)> <p>необязательные:</p> <ul style="list-style-type: none"> ● pageSize: int < определяет размер каждой страницы поиска > (по умолчанию 10) ● page: int < определяет текущую страницу поиска > (по умолчанию 1) 	<p>1. Успешный ответ:</p> <ul style="list-style-type: none"> - перечисленные продукты - Общее количество элементов для всех групп не должно превышать значение pageSize. <p>Status: 200/OK Content-type: application/json Body: [{ < свойства, описанные в подразделе «Информация об объектах» для указанной сущности > }, { < свойства, описанные в подразделе «Информация об объектах» для указанной сущности > }, ...]</p>

	<p>сущности > }, ...]</p> <p>2. Если токен отсутствует в запросе: Status: 401/UNAUTHORIZED Content-type: application/json Body: { message: “Необходима аутентификация” }</p> <p>3. Если токен не соответствует действительной аутентификации: Status: 403/FORBIDDEN Content-type: application/json Body: { message: “Неверный токен” }</p>
--	---

POST (machines)	
Создать новую модель машины.	
Request	Response
<p>Authorization: Bearer string <token></p> <p>Body: объект JSON, который должен содержать все следующие свойства:</p> <p>Свойства JSON:</p> <ul style="list-style-type: none"> ● motherboardId: ID материнской платы ● powerSupplyId: ID источника питания ● processorId: ID процессора ● ramMemoryId: ID RAM ● ramMemoryAmount: объем оперативной памяти ● storageDevices: массив объектов JSON, содержащий: <ul style="list-style-type: none"> ○ storageDeviceId: IDs запоминающих устройств ○ amount: количество устройств данной модели ● graphicCardId: ID видеокарты ● graphicCardAmount: количество 	<p>1. Успешно созданная машина: Status: 201/CREATED Content-type: application/json Body: [{ “id”: < ID созданной машины>, ... }]</p> <p>2. Изображение не предоставлено, количество деталей недействительно или обнаружены несовместимости: Status: 400/BAD REQUEST Content-type: application/json Body: {</p>

видеокарт	<pre> “<свойство с ошибкой>”: “<описание ошибки или несовместимости >”, “<свойство с ошибкой>”: “<описание ошибки или несовместимости >”, ... </pre> <p>3. Если токен отсутствует в запросе: Status: 401/UNAUTHORIZED Content-type: application/json Body: <pre> { message: “Необходима аутентификация” } </pre> </p> <p>4. Если токен не соответствует действительной аутентификации: Status: 403/FORBIDDEN Content-type: application/json Body: <pre> { message: “Неверный токен” } </pre> </p>
-----------	---

PUT (machines/{id}) Изменить существующую модель машины.	
Request	Response
<p>Authorization: Bearer string <token></p> <p>Body: <pre> { < то же, что и POST (machines)> } </pre> Если imageBase64 не указан, изображение машины (свойство imageUrl) должно оставаться неизменным.</p> <p>Обязательные параметры: - id: Идентификатор машины</p>	<p>1. Модель машины успешно обновлена: Status: 200/OK Content-Type: application/json Body: <pre> { < свойства машины, описанные в подразделе «Информация об объектах»> } </pre> </p> <p>2. Количество деталей недействительно или обнаружены несовместимости: Status: 400/BAD REQUEST Content-Type: application/json Body: <pre> { “<свойство с ошибкой>”: “<описание ошибки </pre> </p>

	<p>или несовместимости >”, “<свойство с ошибкой>”: “<описание ошибки или несовместимости >”, ... } 3. Если токен отсутствует в запросе: Status: 401/UNAUTHORIZED Content-type: application/json Body: { message: “Необходима аутентификация” } 4. Если токен не соответствует действительной аутентификации: Status: 403/FORBIDDEN Content-type: application/json Body: { message: “Неверный токен” }</p>
--	---

DELETE (machines/{id}) Удалите существующую модель машины.	
Request	Response
Authorization: Bearer string <token> Обязательные параметры: <ul style="list-style-type: none"> • id: int < Идентификатор машины, подлежащий удалению > 	<p>1. Модель машины успешно удалена: Status: 204/NO CONTENT</p> <p>2. Если модель машины не существует: Status: 404/NOT FOUND Content-type: application/json Body: { message: “Модель машины не найдена” } 3. Если токен отсутствует в запросе: Status: 401/UNAUTHORIZED Content-type: application/json Body: { message: “Необходима аутентификация” }</p>

	<p>4. Если токен не соответствует действительной аутентификации: Status: 403/FORBIDDEN Content-type: application/json Body: <pre>{ message: "Неверный токен" }</pre></p>
--	--

POST (verify-compatibility)	
<p>Проверьте совместимость двух или более деталей, не обязательно машины, однако всегда должны быть материнская плата и блок питания.</p>	
Request	Response
<p>Authorization: Bearer string <token></p> <p>Body: объект JSON с некоторыми обязательными свойствами (отмеченными *) и другими необязательными:</p> <p>Propriedades do JSON:</p> <ul style="list-style-type: none"> ● MBoard Id*: ID материнской платы ● P-supplies Id*: ID источника питания ● CPU Id: ID процессора ● RAM Id: ID RAM ● RAM Amount: объем оперативной памяти ● S-devices: массив объектов JSON, содержащий: <ul style="list-style-type: none"> ○ S-devicesId: IDs запоминающих устройств ○ amount: количество устройств данной модели ● G-cards Id: ID видеокарты ● G-cards Amount: количество видеокарт 	<p>1. Проверка прошла успешно: Status: 200/NO CONTENT Body: <pre>{ message: "Действующая машина" }</pre></p> <p>2. Обнаружены несовместимости: Status: 400/BAD REQUEST Content-Type: application/json Body: <pre>{ "<свойство с ошибкой>": "<описание ошибки или несовместимости >", "<свойство с ошибкой>": "<описание ошибки или несовместимости >", ... }</pre></p> <p>3. Если токен отсутствует в запросе: Status: 401/UNAUTHORIZED Content-type: application/json Body: <pre>{ message: "Необходима аутентификация" }</pre></p> <p>4. Если токен не соответствует действительной аутентификации: Status: 403/FORBIDDEN Content-type: application/json</p>

	Body: <pre>{ message: "Неверный токен" }</pre>
--	--

GET (images/{id}) Возвращает изображение с данными «id» (это значение, возвращаемое в «imageUrl» в списках API).	
Request	Response
Параметры: Обязательный: <ul style="list-style-type: none"> id: int < Идентификатор изображения > 	1. Нормальный поток: Status: 200/OK Content-type: image/string < строка, являющаяся значением, соответствующим типу изображения > Body: Само изображение 2. Если изображение не существует: Status: 404/NOT FOUND Content-type: application/json Body: <pre>{ message: "Изображение не найдено" }</pre>

5. КРИТЕРИИ ОЦЕНКИ.

Таблица 2.

Критерий		Баллы		
		Судейские аспекты	Объективная оценка	Общая оценка
A	Организация работы и управление	4.00	3.00	7.00
B	Коммуникация и навыки межличностного общения	4.00	3.00	7.00
C	Дизайн	20.00	3.00	23.00
D	Вёрстка	3.00	20.00	23.00
E	Программирование на стороне клиента	0.00	10.00	10.00
F	Программирование на стороне сервера	0.00	10.00	10.00
G	Проектирование	6.00	2.00	8.00
Итого		37.00	51.00	88.00

6. ПРИЛОЖЕНИЯ К ЗАДАНИЮ.

- Модуль 2: *T3.docx, бренд-бук.pdf, media_m2.zip*
- Модуль 3: *Routes.docx, media_m3.zip*
- Модуль 4: *media_m4.zip*