КОНКУРСНОЕ ЗАДАНИЕ
КОМПЕТЕНЦИИ

Веб-технологии

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:** 12 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 (C1, C2, C3)	Время на задание
A	Проектирование	C1	3 часа
B	Вёрстка и дизайн	C1	3 часа
C	Программирование на стороне клиента	C2	3 часа
D	Программирование на стороне сервера	C2	3 часа

Модуль A: 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

Риелторская компания «Мой риелтор» (далее – Компания) является коммерческой организацией. Компания занимается предоставлением услуг по покупке/продаже недвижимости, а также аренде. Деятельность Компании осуществляется в соответствии с действующим законодательством Российской Федерации, Федеральным законом о риелторской деятельности и иными нормативными правовыми актами Российской Федерации.

Сотрудники Компании обратились к вам с просьбой создать веб-сервис для организации процессов Компании, направленных на покупку, продажу и аренду жилой недвижимости.

Веб-сервис должен:

- предоставлять полную информацию об объектах недвижимости: адрес, планировка, продажа/аренда, планировка, ближайшая инфраструктура (больницы, школы, садики, остановки общественного транспорта);
- позволять покупателям находить подходящую недвижимость, давать обратную связь, подписываться на информационную рассылку;
- позволять владельцам создавать объявления для продажи и сдачи в аренду;
- предоставлять администраторам возможность управлять пользователями, объявлениями;

Основные ролевые группы и их состав:

1) Пользователи сервиса:

- Незарегистрированный пользователь. Пользователь, который не имеет учетную запись и не авторизовался в системе. Функциональные возможности: просмотр информационных страниц, взаимодействие с формами регистрации и авторизации, поиск и просмотр доступных объявлений, взаимодействие с формой подписки на информационную рассылку.

- Зарегистрированный пользователь. Пользователь, который имеет учетную запись в системе и авторизовался в системе. Функциональные возможности: просмотр текстовых информационных страниц, поиск и просмотр

доступных объявлений, создание и управление объявлением, редактирование данных профиля, добавление отзывов на объявление.

2) Пользователи системы управления:

- Администратор системы. Функциональные возможности: формирование списка доступной недвижимости для рассылки информационных сообщений на электронную почту подписчиков и пользователей, просмотр и одобрение информации о недвижимости, получение статистики о пользователях (количество, список отзывов о недвижимости, список объявлений).

Ваш проект должен корректно открываться со следующих устройствах:

- 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 Компании. 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 B: Вёрстка и дизайн.

ВВЕДЕНИЕ

Вам предоставляет готовое ТЗ на основе которого вам необходимо сверстать все необходимые веб-страницы. На каждой странице должен присутствовать логотип Риелторской компании. Дизайн проекта должен быть в цветовой гамме Компании (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:

- Главная страница со списком объявлений, фильтрами и поиском
- Страница О компании
- Страница с Правовой информацией
- Страница регистрации
- Страница авторизации
- Личный кабинет Пользователя
    - Личные данные
    - Мои Объявления
- Личный кабинет Администратора
    - Объявления
    - Отзывы
    - Управление пользователями
- Страница создания работодателя
- Страница объявления

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 4
- Bootstrap 5

Модуль C: Программирование на стороне клиента.

ВВЕДЕНИЕ

Todoist создал API для создания и ведения канбан-досок. 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.

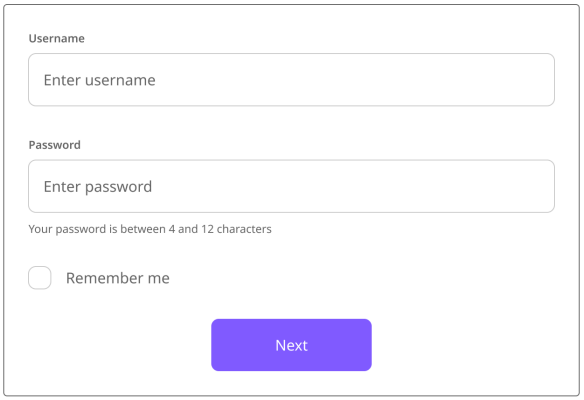
ОПИСАНИЕ ПРОЕКТА И ЗАДАЧИ

Запросы, которые будут использоваться

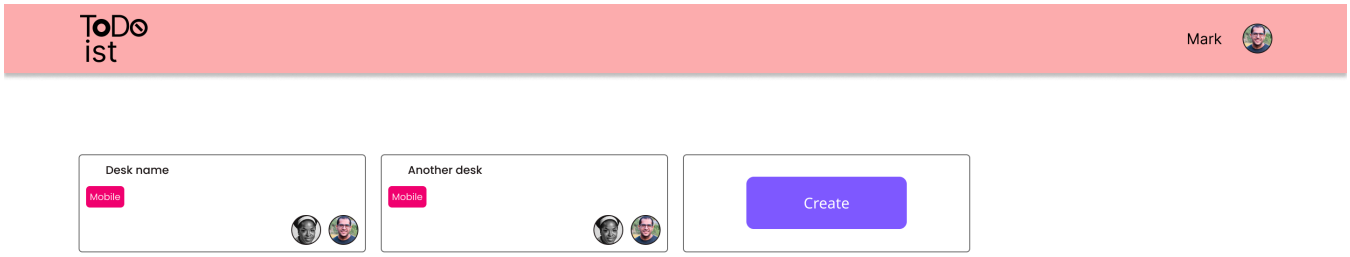
Канбан доски и элементы, предоставляемых API, и их соответствующие конечные точки (связанные с http://kanban.wsr.ru/kanban/api):

- доска: boards
- шаблон: templates
- колонка: columns
- карточка: cards
- участник: users

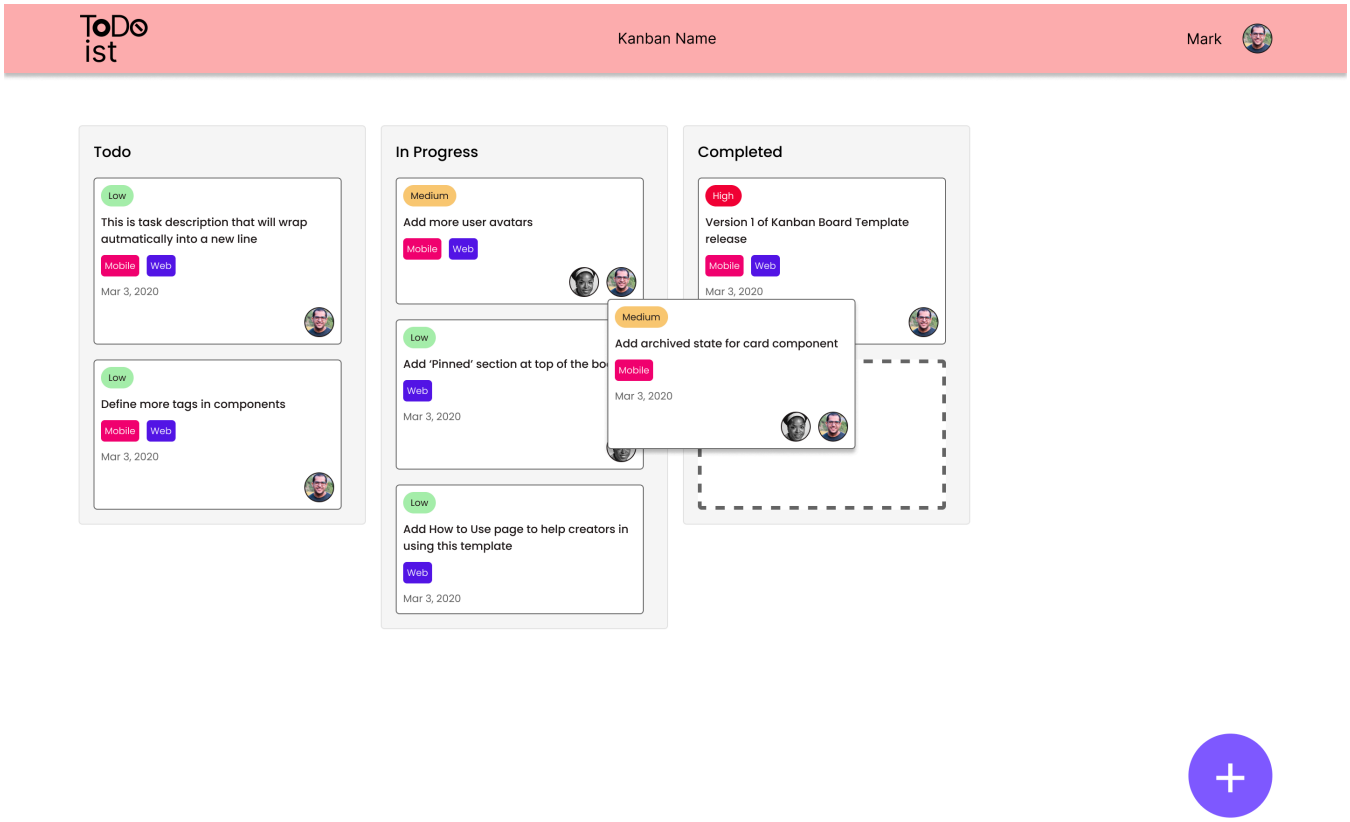
На вашем веб-сайте должен отображаться экран входа в систему всякий раз, когда неаутентифицированный пользователь пытается получить к нему доступ. Если учетные данные неверны, пользователю должно быть показано сообщение в модальном формате.



Кроме того, после успешного входа в систему должен появиться список уже созданных канбан досок и кнопа «Создать». Вы можете следовать предложенному ниже макету, но не стесняйтесь предлагать лучший дизайн.



При нажатии на кнопку «Создать» должно отобразиться модальное окно, с формой создания доски канбан с полями, которые указаны в файле Kanban-api.pdf. Доска создается по готовому шаблону, с предопределенными колонками, изображающими процесс работы. В шаблоне указываются колонки и их порядок. При клике на существующую доску пользователь должен переходить к ней (интерфейс ниже).



Настройка колонки

Колонку на доске можно настраивать, изменять заголовок, указывать максимальное значение.

Создание карточки

Карточка содержит задачу, теги, крайний срок выполнения, участников и приоритет выполнения. Новая задача попадает в крайнюю левую колонку по умолчанию.

Перенос карточки

Переносить карточку из колонки в колонку разрешено слева направо использую drag-and-drop, при этом неограниченное количество должно быть только в крайних левых и правых колонках. Все промежуточные колонки имеют ограничения, настраиваемые пользователем. Если колонка уже содержит максимальное значение, то должна быть анимация встряхивания и возврат карточки в изначальную колонку. Анимация ошибки так же должна быть, если карточку переносят в колонку, которая идет не следующей по порядку.

Настройки пользователя

Можно настраивать пользователя, менять аватар и ФИО.

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Вы должны предоставить функциональный веб-сайт, доступный через сервер и по URL-адресу xxxxx-m3.wsr.ru/kanban, 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 управлению проектами создает новый сервис, для ведения канбан досок. Чтобы упростить разработку и использование, а так же дальнейшее улучшение, вам требуется создать сервис, доступный по REST API.

Сервис должен уметь:

- создавать канбан доску;
- создавать и переносить карточки;
- настраивать колонки доски;
- предупреждать клиента о невозможности перенести карточку,
- управлять данными своего пользователя.

  Доска должна быть динамической для пользователя.

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧИ

Вы должны создать API с использованием PHP или python фреймворка и базы данных.

Вы получите файлы .json с исходными данными, файл создания базы данных, 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-m4.wsr.ru/kanban/api. Где xxxxx - ваш логин.

  Причины отказа создания и переноса карточки

Ваш API должен сообщать о невозможности перенести карточку в следующих случаях:

- Карточка создается не в первый по порядку колонке;
- Карточка переносится не в следующую по порядку колонку;
- Карточка переносится в колонку, где уже присутствует максимальное для колонки количество других карточек;
- Карточку можно переносить только в рамках одной канбан доски;
- Карточку может переносить только ее участник.

ДЕТАЛИ API

Аунтификация

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 работы с канбан доской (которые будут отдельным маршрутом):

- Список канбан досок, только те, в которых пользователь является участником
  - Должен иметь имя, список участников.
- Список шаблонов канбан досок
  - Должен содержать название шаблона, названия и прядок колонок и максимальное количество карточек в них.
- Список колонок канбан доски
  - Должен содержать имя, очередь на доске, максимальное количество карточек (для крайних левой и правой это значение должно равняться 0).
- Список карточек
  - Должна быть указана задача, колонка, приоритет выполнения, список тегов, список участников, крайняя дата выполнения.
- Список участников
  - Должен иметь имя пользователя, URL-адрес аватар, ФИО.

Участники 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-m4.wsr.ru/kanban/api. Где xxxxx - ваш логин. Обязательные параметры отмечены *.

POST (*login*) **Аутентификация пользователя**

Request	Response

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

DELETE (*logout*) **Удаление аутентификации пользователя**	
Request	Response

header: Authorization: Bearer string <token >	**1. Выход подтвержден:**
	Status: 200/OK
	Content-type: application/json
	Body:
	{
	message: “Успешный выход”
	}
	
	2. Если токен отсутствует в запросе:
	Status: 401/UNAUTHORIZED
	Content-type: application/json
	Body:
	{

	message: “Необходима аутентификация”
	}
	
	3. Если токен не соответствует действительной аутентификации:
	Status: 403/FORBIDDEN
	Content-type: application/json
	Body:
	{
	message: “Неверный токен”
	}

GET (*boards, templates, columns, cards, users*) **Списки канбан досок, шаблонов, колонок, карточек и участников соответственно.**	
Request	Response

header: Authorization: Bearer string < token **>**
 	
Parametrs:
Параметры строки запроса:
● *board: int < id канбан доски> (везде, кроме **boards и templates**)
● pageSize: int < определяет размер каждой страницы поиска > (по умолчанию 10)
● page: int < определяет текущую страницу поиска > (по умолчанию 1)
	1. Успешный ответ:
	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>
	},
	...
	]
	
	2. Если токен отсутствует в запросе:
	Status: 401/UNAUTHORIZED
	Content-type: application/json
	Body:
	{
	message: “Необходима аутентификация”
	}
	
	3. Если токен не соответствует действительной аутентификации:

	Status: 403/FORBIDDEN
	Content-type: application/json
	Body:
	{
	message: “Неверный токен”
	}

GET (search/{category}?q={q}) **Поиск доски, карточки (название, теги), участника**	
Request	Response

header: Authorization: Bearer string <token>
 	
Параметры:
Обязательные:
● q: string < поисковая строка, которая должна быть применена к имени искомых элементов >
● category: string < объект поиска >
 	
необязательные:
● pageSize: int < определяет размер каждой страницы поиска > (по умолчанию 10)
● page: int < определяет текущую страницу поиска > (по умолчанию 1)
	1. Успешный ответ:
	- перечисленные объекты
	- Общее количество элементов для всех групп не должно превышать значение pageSize.
	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 >
	},
	...
	]
	
	2. Если токен отсутствует в запросе:
	Status: 401/UNAUTHORIZED
	Content-type: application/json
	Body:
	{
	message: “Необходима аутентификация”
	}
	
	3. Если токен не соответствует действительной аутентификации:
	Status: 403/FORBIDDEN
	Content-type: application/json

	Body:
	{
	message: “Неверный токен”
	}

POST (boards) **Создать новую доску.**	
Request	Response

header: Authorization: Bearer string <token>
 	
Body: объект JSON, который должен содержать все следующие свойства:
 	
Свойства JSON:
● name: название доски
● columns: массив колонок
○ name: название
○ maxCards: максимальное количество карточек в колонке
○ order: порядок, число по возрастающей
● users: массив участников
○ userId: ID участника
	1. Успешно созданная доска:
	Status: 201/CREATED
	Content-type: application/json
	Body:
	{
	“id”: < ID созданной доски>,
	…
	}
	
	2. Поля не заполнены, пустой массив колонок, maxCards для первой и последней по порядку колонке не равно 0, порядок колонок имеет одинаковое значение:
	Status: 400/BAD REQUEST
	Content-type: application/json
	Body:
	{
	“<свойство с ошибкой>”: “<описание ошибки 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”
	}

PUT (boards/{id}) **Изменить существующую доску канбан.**	
Request	Response

Authorization: Bearer string <token>
 	
Body:
{
< то же, что и **POST (boards)**>
}
 	
Обязательные параметры:
- id: Идентификатор доски
	1. Доска канбан успешно обновлена:
	Status: 200/OK
	Content-Type: application/json
	Body:
	{
	< свойства доски, описанные в подразделе «Информация об объектах»>
	}
	
	2. Поля не заполнены, пустой массив колонок, maxCards для первой и последней по порядку колонке не равно 0, порядок колонок имеет одинаковое значение:
	Status: 400/BAD REQUEST
	Content-Type: application/json
	Body:
	{
	“<свойство с ошибкой>”: “<описание ошибки 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”
	}

DELETE (boards/{id}) **Удалите существующую доску канбан.**	
Request	Response

Authorization: Bearer string <token>
 	
Обязательные параметры:
● id: int < Идентификатор доски, подлежащий удалению >
	1. Модель машины успешно удалена:
	Status: 204/NO CONTENT
	
	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”
	}
	4. Если токен не соответствует действительной аутентификации:
	Status: 403/FORBIDDEN
	Content-type: application/json
	Body:
	{
	message: “Неверный токен”
	}

POST (cards) **Создание карточки**	
Request	Response
Authorization: Bearer string <token>	**1. Проверка прошла успешно:**

Body: объект JSON с некоторыми обязательными свойствами (отмеченными *) и другими необязательными:
 	
Содержание JSON:
● boardId*: ID доски
● name*: название задачи
● tags: список строк, тегов
● users*: массив участников
○ userId: ID участника
● priority: приоритет выполнения, выбор из low, medium, high, critical
● deadline: крайний срок выполнения, формат 2023-04-10
	Status: 201/CREATED
	Body:
	{
	id: “ID созданной карточки”,
	…
	}
	
	2. Пользователь не является участником доски, deadline указано значение в прошлом, priority отличается от :
	Status: 400/BAD REQUEST
	Content-Type: application/json
	Body:
	{
	“<свойство с ошибкой>”: “<описание ошибки 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”
	}

POST (cards/{id}/move) **Перенос каточки в колонку**	
Request	Response
Authorization: Bearer string <token> **Body:** объект JSON	**1. Проверка прошла успешно:** **Status:** 200/NO CONTENT **Body:** { id: “ID карточки”,

Содержание JSON:
● columnId*: ID колонки
 	
Обязательные параметры:
● id: int < Идентификатор карточки>
	…
	}
	
	2. Пользователь не является участником карточки, колонка не является следующей по порядку, колонка содержит максимальное число карточек:
	Status: 400/BAD REQUEST
	Content-Type: application/json
	Body:
	{
	“<свойство с ошибкой>”: “<описание ошибки 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”
	}
	
	5. Если карточки не существует:
	Status: 404/NOT FOUND
	Content-type: application/json
	Body:
	{
	message: “Карточка не найдена”
	}

GET (images/{id}) **Возвращает изображение с данными «id» (это значение, возвращаемое в «imageUrl» в списках API).**	
Request	Response

Параметры:
Обязательный:
• 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:
	{
	message: “Изображение не найдено”
	}

5. КРИТЕРИИ ОЦЕНКИ.

Таблица 2.

Критерий		Баллы		
		Судейские аспекты	Объективная оценка	Общая оценка
A	Организация работы и управление	1.00	5.00	6.00
B	Коммуникация и навыки межличностного общения	2.50	3.50	6.00
C	Дизайн	16.10	2.00	18.10
D	Вёрстка	2.00	20.00	22.00
E	Программирование на стороне клиента	2.00	13.50	15.50
F	Программирование на стороне сервера	0.00	11.90	11.90
G	Проектирование	6.00	2.50	8.50
	Итого	29.60	58.40	88.00

6. ПРИЛОЖЕНИЯ К ЗАДАНИЮ.

- Модуль B: ***T3.docx, бренд-бук.pdf, media_m2.zip***
- Модуль C: ***Routes.docx, media_m3.zip***
- Модуль D: ***media_m4.zip***