



ВСЕРОССИЙСКОЕ
ЧЕМПИОНАТНОЕ
ДВИЖЕНИЕ
ПО ПРОФЕССИОНАЛЬНОМУ
МАСТЕРСТВУ

СОГЛАСОВАНО

Главный эксперт

Н.Ф. Насыров

СОГЛАСОВАНО

Индустриальный партнер

АО «НПП «Сигнал»

Директор по КПО

П.С. Жданова

М.П.



КОНКУРСНОЕ ЗАДАНИЕ КОМПЕТЕНЦИИ «Веб-технологии»

(региональный этап)

Чемпионата по профессиональному мастерству

«Профессионалы» в 2024г.

Конкурсное задание разработано экспертным сообществом и утверждено Менеджером компетенции, в котором установлены нижеследующие правила и необходимые требования владения профессиональными навыками для участия в соревнованиях по профессиональному мастерству.

Конкурсное задание включает в себя следующие разделы:

1. ОСНОВНЫЕ ТРЕБОВАНИЯ КОМПЕТЕНЦИИ	3
1.1. ОБЩИЕ СВЕДЕНИЯ О ТРЕБОВАНИЯХ КОМПЕТЕНЦИИ.....	3
1.2. ПЕРЕЧЕНЬ ПРОФЕССИОНАЛЬНЫХ ЗАДАЧ СПЕЦИАЛИСТА ПО КОМПЕТЕНЦИИ «ВЕБ-ТЕХНОЛОГИИ».....	3
1.3. ТРЕБОВАНИЯ К СХЕМЕ ОЦЕНКИ	7
1.4. СПЕЦИФИКАЦИЯ ОЦЕНКИ КОМПЕТЕНЦИИ	8
1.5. КОНКУРСНОЕ ЗАДАНИЕ	8
1.5.1. Разработка/выбор конкурсного задания	8
1.5.2. Структура модулей конкурсного задания (инвариант/вариатив).....	8
Модуль А. Разработка интерфейса пользователя	9
Модуль Б. Разработка Веб-приложения на стороне клиента.....	15
Модуль В. Разработка Веб-приложения на стороне сервера.....	31
Модуль Г. Разработка ИР с использованием готовых решений.....	49
2. СПЕЦИАЛЬНЫЕ ПРАВИЛА КОМПЕТЕНЦИИ	52
2.1. Личный инструмент конкурсанта.....	52
2.2. Материалы, оборудование и инструменты, запрещенные на площадке	52
3. ПРИЛОЖЕНИЯ.....	53

ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ

1. *IP – Информационный ресурс*
2. *SSH - SSH (от англ. secure shell — безопасная оболочка) — это защищённый сетевой протокол для удалённого управления сервером через интернет.*
3. *FTP – (File Transfer Protocol), или «протокол передачи файлов» это набор процедур или правил, позволяющих электронным устройствам взаимодействовать между собой.*
4. *CMS – (Content Management System) — это система создания и управления сайтом. Это визуально удобный интерфейс, с помощью которого можно добавлять и редактировать содержимое сайта.*

1. ОСНОВНЫЕ ТРЕБОВАНИЯ КОМПЕТЕНЦИИ

1.1. ОБЩИЕ СВЕДЕНИЯ О ТРЕБОВАНИЯХ КОМПЕТЕНЦИИ

Требования компетенции (ТК) «Веб-технологии» определяют знания, умения, навыки и трудовые функции, которые лежат в основе наиболее актуальных требований работодателей отрасли.

Целью соревнований по компетенции является демонстрация лучших практик и высокого уровня выполнения работы по соответствующей рабочей специальности или профессии.

Требования компетенции являются руководством для подготовки конкурентоспособных, высококвалифицированных специалистов / рабочих и участия их в конкурсах профессионального мастерства.

В соревнованиях по компетенции проверка знаний, умений, навыков и трудовых функций осуществляется посредством оценки выполнения практической работы.

Требования компетенции разделены на четкие разделы с номерами и заголовками, каждому разделу назначен процент относительной важности, сумма которых составляет 100.

1.2. ПЕРЕЧЕНЬ ПРОФЕССИОНАЛЬНЫХ ЗАДАЧ СПЕЦИАЛИСТА ПО КОМПЕТЕНЦИИ «ВЕБ-ТЕХНОЛОГИИ»

Таблица №1

Перечень профессиональных задач специалиста

№ п/п	Раздел	Важность в %
1	Тестирование информационных ресурсов	7
	Специалист должен знать и понимать: <ul style="list-style-type: none">– способы решения возникающих проблем, анализ проблемной ситуации возникшей в ходе решения профессиональных задач, пути их решения с учетом этических норм и правил, опираясь на профессиональную этику;– принципы, лежащие в основе сбора и представления информации;– основные приемы и методы визуального представления информации (черновое макетирование страниц, объектно-событийное моделирование, создание блок-схем и др.);– английский язык в рамках чтения и понимания официальной технической документации по используемым технологиями и языкам программирования.	
	Специалист должен уметь: <ul style="list-style-type: none">– собирать, анализировать и оценивать информацию;– использовать навыки грамотности для толкования стандартов и требований;– составлять тестовую документации для тестирования новых функциональностей продукта	

	<ul style="list-style-type: none"> – проводить ручное тестирование новых функциональностей – проводить регрессионное ручное тестирование – вести баг-репорты – составлять отчеты по итогам тестирования – автоматизировать регрессионное тестирование – общаться с заказчиком, командой разработки и тестирования 	
2	<p>Техническая поддержка и администрирование информационных ресурсов</p>	7
	<p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> – принципы и практики, которые позволяют продуктивно работать, в том числе в команде; – аспекты систем, которые позволяют повысить продуктивность и выработать оптимальную стратегию; – основные принципы выбора технологий и инструментария для решения поставленных задач (проектов); - основные подходы к планированию и документированию проекта. 	
	<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> – формировать архитектуру проекта (программного продукта) в соответствии с последними отраслевыми решениями; – выбирать технологии и инструменты для решения поставленных задач; – планировать график рабочего дня с учетом требований; – планировать задачи, учитывать временные ограничения и сроки; – решать распространенные задачи веб-дизайна и разработки кода; – формировать тестовые наборы, применять инструменты автоматического тестирования; – производить отладку кода программ и находить ошибки; – оптимально использовать компьютерное оборудование и программное обеспечение для повышения эффективности своей работы; – использовать менеджеры пакетов при разработке проекта; - использовать систему контроля версий. 	
3	<p>Разработка интерфейса пользователя</p>	26
	<p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> – структуру и общепринятые элементы веб-страниц различных видов и назначений; – основные принципы организации контента веб-приложения; – основные правила выбора цвета, работы с типографикой и композицией; 	

	<ul style="list-style-type: none"> – принципы и методы создания и адаптации графики для использования ее на веб-сайтах; – ограничения, которые накладывают мобильные устройства и разрешения экранов при использовании их для просмотра веб-сайтов; – принципы построения эстетичного и креативного дизайна; – методы обеспечения доступа к страницам веб-сайтов аудитории с ограниченными возможностями; – World Wide Web Consortium (W3C) стандарты HTML и CSS; – методы верстки веб-сайтов и их стандартную структуру; – Web accessibility initiative (WAI) стандарт доступности активных Интернет-приложений для людей с ограниченными возможностями; – основные принципы применения соответствующих CSS правил и селекторов для получения ожидаемого результата; – лучшие практики для Search Engine Optimization (SEO) и интернет-маркетинга; 	
	<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> – создавать, использовать и оптимизировать изображения для веб-сайтов; – выбирать дизайнерское решение, которое будет наиболее подходящим для целевого рынка; – принимать во внимание влияние каждого элемента, который добавляется в проект во время разработки дизайна; – использовать все требуемые элементы при разработке дизайна; – создавать «отзывчивый» дизайн, который будет отображаться корректно на различных устройствах и при разных разрешениях; – создавать html-страницы сайта на основе предоставленных графических макетов их дизайна; – корректно использовать CSS для обеспечения единого дизайна в разных браузерах; – создавать адаптивные веб-страницы, которые способны оставаться функциональными на различных устройствах при разных разрешениях; – создавать веб-сайты полностью соответствующие текущим стандартам W3C (http://www.w3.org); – создавать и модифицировать веб-интерфейсы с учетом принципов Search Engine Optimization; – использовать препроцессоры. 	
4	<p>Разработка на стороне клиента</p> <p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> – основные принципы паттерновой разработки веб-приложений; – ECMAScript (JavaScript); – принципы, особенности и способы использования открытых фреймворков; 	25

	<ul style="list-style-type: none"> – принципы разработка кода с использованием открытых библиотек; – различные интерфейсы взаимодействия с объектами браузера 	
	<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> – создавать и модифицировать JavaScript код для улучшения функциональности и интерактивности сайта; – манипулировать элементами страницы веб-приложения; – разрабатывать анимацию для повышения доступности и визуальной привлекательности веб-приложения; – применять открытые библиотеки и фреймворки; – тестировать веб-приложение. 	
5	<p>Разработка веб приложения на стороне сервера</p>	25
	<p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> – процедурные и объектно-ориентированные языки PHP, Python, Node.js; – основные принципы и правила использования открытых библиотек и фреймворков; – распространенные модели организации и хранения данных; – основные принципы создания баз данных; – основные принципы обмена данными между клиентом и сервером; – методы работы с протоколами SSH/(s)FTP при подключении к серверам; – способы разработки программного кода в соответствии с паттернами проектирования; – основные принципы обеспечения безопасности веб-приложения. 	
	<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> – разрабатывать процедурный и объектно-ориентированный программный код; – разрабатывать веб-сервисы с применением PHP, Python, Node.js в соответствии с техническим заданием; – создавать библиотеки и модули для выполнения повторяющихся задач; – разрабатывать веб-приложения с доступом к различным базам данных; – создавать SQL (Structured Query Language) запросы и конструкции; – обеспечивать безопасность (устойчивость веб-приложения к атакам и взломам); – интегрировать существующий и создавать новый программный код с API (Application Programming Interfaces); – использовать открытые библиотеки и фреймворки; 	
6	<p>Разработка информационных ресурсов с использованием готовых решений</p>	10
	<p>Специалист должен знать и понимать:</p>	

<ul style="list-style-type: none"> – преимущества и ограничения системы управления контентом с открытым исходным кодом; – методы работы с плагинами/модулями; – способы реализации функциональных возможностей CMS; – основные принципы организации контента веб-приложения; – понимать необходимость поддержания и обновления для плагинов CMS и соответствующих модулей для безопасности системы; – основные принципы интеграции с внешними веб-приложениями. 	
<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> – устанавливать, настраивать и модифицировать систему управления контентом; – устанавливать, настраивать и обновлять плагины/модули CMS; – создать пользовательские темы/шаблоны для системы управления контентом; – создавать пользовательские плагины/модули и шаблоны/темы; – использовать встроенные методы и средства CMS при разработке веб-приложения. 	

1.3. ТРЕБОВАНИЯ К СХЕМЕ ОЦЕНКИ

Сумма баллов, присуждаемых по каждому аспекту, должна попадать в диапазон баллов, определенных для каждого раздела компетенции, обозначенных в требованиях и указанных в таблице №2.

Таблица №2

Матрица пересчета требований компетенции в критерии оценки

Критерий/Модуль						Итого баллов за раздел ТРЕБОВАНИЙ КОМПЕТЕНЦИИ
Разделы ТРЕБОВАНИЙ КОМПЕТЕНЦИИ	1	А	Б	В	Г	
	2	2	1	2,5	1,5	7
	3	2	2	2	1	7
	4	20	1		5	26
	5		23		2	25
	6			22	3	25
	Итого баллов за критерий/модуль		24	27	26,5	22,5

1.4. СПЕЦИФИКАЦИЯ ОЦЕНКИ КОМПЕТЕНЦИИ

Оценка Конкурсного задания будет основываться на критериях, указанных в таблице №3:

Таблица №3

Оценка конкурсного задания

Критерий		Методика проверки навыков в критерии
А	Разработка интерфейса пользователя	Проверка результата выполнения модуля после его завершения
Б	Разработка Веб-приложения на стороне клиента	Проверка результата выполнения модуля после его завершения
В	Разработка Веб-приложения на стороне сервера	Проверка результата выполнения модуля после его завершения
Г	Разработка ИР с использованием готовых решений	Проверка результата выполнения модуля после его завершения

1.5. КОНКУРСНОЕ ЗАДАНИЕ

Возрастной ценз: от 16 лет

Общая продолжительность Конкурсного задания¹: 15 ч.

Количество конкурсных дней: 3 дня

Вне зависимости от количества модулей, КЗ должно включать оценку по каждому из разделов требований компетенции.

Оценка знаний участника должна проводиться через практическое выполнение Конкурсного задания. В дополнение могут учитываться требования работодателей для проверки теоретических знаний / оценки квалификации.

1.5.1. Разработка/выбор конкурсного задания

Конкурсное задание состоит из 4 модулей, включает обязательную к выполнению часть (инвариант) - 3 модуля, и вариативную часть - 1 модуль. Общее количество баллов конкурсного задания составляет 100.

Матрица конкурсного задания представлена в Приложении 1.

1.5.2. Структура модулей конкурсного задания (инвариант/вариатив)

¹ Указывается суммарное время на выполнение всех модулей КЗ одним конкурсантом.

Модуль А. Разработка интерфейса пользователя

Технологии этого модуля: граф. дизайн, HTML5, CSS3, GIT

Время на выполнение: 6 часов

В современном мире с каждым днём появляется всё больше и больше новых технологий, которые позволяют решать повседневные проблемы. К вам обратился заказчик, который просит разработать сервис для планирования онлайн встреч.

Сайт предназначен для участников команды, которые хотят запланировать встречу. Проблема в том, что лидер не знает, когда каждый из участников команды может присоединиться.

Саму встречу создает лидер команды, указывает день (дни) встречи и временные интервалы для этой встречи. Лидер должен иметь возможность планировать встречу на несколько дней. После этого ему доступен инструмент приглашения участников команды. Каждый из участников может указать временные интервалы, когда он может присоединиться к встрече. Информация о присоединившихся отображается лидеру.

Подобных ресурсов достаточное количество, а заказчик хочет в кратчайшие сроки занять лидирующие позиции в этой нише (максимально увеличить количество пользователей, планирующих свои встречи с помощью сервиса). Для этого вам необходимо использовать в дизайне сверстанных страниц элементы, концепции и приемы, позволяющие привлечь внимание, создать положительный образ ресурса, реализовать современные концепции дизайна веб-страниц.

Подразумевается, что пользователями сайта являются IT-специалисты со всего мира, которые ценят свое время, и которые обращаются к ресурсу в первую очередь с помощью смартфонов. Вам следует явно отразить эту специфику в дизайне.

Общая информация и требования

- Страницы должны корректно отображаться как на мобильных устройствах, так и на десктопных - в том числе на страницах управления встречами, их просмотра и т.д.
- Оцениваются как верстка, так и дизайн. Однако вы можете сконцентрироваться только на дизайне.
- Также при верстке страниц вам следует учитывать существующие практики UI/UX, чтобы использование веб-ресурса было максимально удобным и продуктивным, а также творчески применять предоставленные медиаданные.
- Использование любых фреймворков и библиотек (Bootstrap, Tailwind, например) запрещено.

ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!

В ходе разработки необходимо использовать систему контроля версий GIT. Должно быть не менее 3 коммитов с осмысленными комментариями!

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – создать дизайн-макеты и сверстать следующие страницы веб-сайта (реализация функциональных возможностей на данном этапе не требуется):

1. Главная страница – index.html, index.png
2. Страница управления встречами (лидером) – leader.html, leader.png
3. Страница просмотра встречи (для участников встречи) – meet.html, meet.png
4. Страница встречи (управление участником) – meet-edit.html, meet-edit.png
5. Личный кабинет – account.html, account.png
6. Встреча не найдена – 404.html, 404.png

Сверстаный веб-сайт должен быть доступен по адресу <http://nnnnnnnn-m1.wsr.ru>, где nnnnnnnn - email участника (указан на индивидуальной карточке).

Роли пользователей сервиса

Гость / неавторизованный пользователь

- Аутентификация
- Просмотр информации о встрече по полученной ссылке
- Создание встречи

Авторизованный пользователь

- Просмотр встреч в личном кабинете, на которые он регистрировался или которые создавал (т.е. выступал в роли лидера команды)
- Изменение временных слотов для встреч, на которые регистрировался
- Просмотр информации и управление своими встречами, которые он создавал ранее
- Выход из профиля

Точки входа

1. Главная страница (страница 1)

- Через форму аутентификации – по email и паролю – переход в ЛК (*страница 5*), где отображаются встречи, созданные пользователем, или встречи, на которые он зарегистрировался.

- Через форму создания встречи – по обязательным e-mail и паролю. После создания встречи происходит переход на *страницу 2*

2. Страница просмотра встречи (для участников встречи) – meet.html, (*страница 3*) – через форму аутентификации – указываются email и пароль (участник может в дальнейшем авторизоваться на этой странице или на главной странице и скорректировать свои временные слоты):

– Если введены email и пароль лидера встречи, то происходит переход к *странице 2*.

– Если введены email и пароль (не лидера встречи), то происходит переход к *странице 4*

3. Страница управления встречами (лидером) – leader.html – по прямой ссылке, отображение *страницы 2* для управления встречами

1. Главная страница – [index.html](#)

Представляет собой Landing Page с описанием и демонстрацией возможностей сервиса. Также на главной странице лидер команды может создать встречу.

Основные элементы пользовательского интерфейса формы создания встречи на главной странице

- Название встречи - обязательное
- Описание встречи - необязательное
- e-mail - обязательное
- Пароль
- Дни - до 5 дней для планирования, выбирать можно произвольный интервал - до 30 дней вперед, начиная с текущей даты. Указать прошедшие даты невозможно. Выбрать даты позже 30 дней также нельзя.
- Интервал времени проведения встреч - по умолчанию с 9 до 18 часов. Интервал применяется сразу ко всем дням, указанным лидером. Начальное и конечное значения временного диапазона могут изменены.
- Элемент, подтверждающий согласие с политикой конфиденциальности - обязательное для заполнения
- Кнопка/ссылка "Создать", для создания встречи. После нажатия/клика по кнопке происходит переадресация на страницу управления встречей лидером команды - **Страница 2.**

Основные элементы пользовательского интерфейса формы аутентификации на главной странице:

- email - обязательное
- пароль - обязательное
- Кнопка/ссылка "Войти", после нажатия/клика по кнопке происходит переадресация на страницу личного кабинета - **Страница 5.**

Также на главной странице должны быть представлены следующие блоки/разделы (названия могут быть изменены, но тематика блока/раздела должна быть понятна):

1. Шапка сайта.

1.1. Логотип.

Вам предоставлен существующий логотип компании logo.png, однако заказчик просит разработать и использовать на сайте новый логотип.

Учтите, что новый логотип должен содержать основные цвета сайта, сохранять преемственность с существующим, но, тем не менее, иметь отличия от предоставленного.

1.2. Меню навигации

1.2.1. Создание встречи (ссылка на раздел на главной странице)

1.2.2. Личный кабинет (ссылка на соответствующую страницу)

1.2.3. Войти (ссылка на раздел на главной странице)

1.2.4. О нас (ссылка на раздел на главной странице)

1.2.5. Отзывы (ссылка на раздел на главной странице)

1.2.6. О системе (ссылка на раздел на главной странице)

1.2.7. Контакты (ссылка на футер)

- О нас - показатели, которых достиг проект - необходимо использовать всю предоставленную информацию из предоставленного текстового файла media/text/about.txt.
- О системе (необходимо предусмотреть область для демонстрации видео о работе сервиса – media/video/about.mp4. Текст для этого блока/раздела можно взять из текущего документа)
- Отзывы – не менее трех, представлены в виде слайдсета. На каждом слайде присутствуют фото автора, текст отзыва, дата отзыва
- Футер с информацией о компании, контактами (информация в файле media/text/footer.txt), ссылками на пользовательское соглашение и политику конфиденциальности – файлы предоставлены. При клике по соответствующей ссылке файлы должны скачиваться на устройство пользователя.

Допускается наличие других блоков/разделов, а также добавление информации в описанные блоки.

При переходе к мобильной версии главной страницы должно быть реализовано не менее трех структурных изменений контента.

Отобразите варианты сообщений, если пользователь указал некорректные данные, не заполнил обязательные поля и т.д. (например, в форме аутентификации, создания встречи).

2. Страница управления встречами (лидером) – leader.html

Страница управления встречами - лидер может просматривать информацию о встрече и вносить изменения.

- Доступна только для лидера - вход по:
 - ссылке вида **{host}/leader.html**
 - по “секретной ссылке”, расположенной на *странице 5*
 - email и паролю со *страницы 3*

Также переход на эту страницу осуществляется сразу после создания встречи лидером (со *страницы 1*).

Информация на странице и соответствующие элементы:

- Отображается ссылка в формате - **{host}/leader.html**.
- Отображается ссылка для приглашения участников в формате **{host}/meet.html**
- Возможность указать лидеру временные интервалы, в которые он может принять участие на встречах. Временные интервалы лидера должны визуально выделяться.
- Доступен просмотр информации об участниках, их выбранных временных интервалов по дням. Визуально видно, когда большая часть участников может присоединиться к встрече.
- Элемент для блокировки встречи, чтобы участники после этого могли только просматривать встречу.
- Поле для ввода списка адресов электронной почты участников команды и кнопка/ссылка отправки приглашения.

- Элемент для удаления информации о пользователях, которые зарегистрировались на эту встречу.
- Поле/область для загрузки изображения - обязательно для заполнения, допустимые форматы загружаемого изображения - png, jpg, jpeg. *Пояснение - Загруженное изображение в дальнейшем будет использоваться в качестве фонового изображения для страниц 3 и 4*
Присутствует кнопка/ссылка “Отправить фоновое изображение”
Присутствует кнопка/ссылка “Удалить фоновое изображение”
- Поле/область для загрузки до 2 файлов в формате pdf с информацией для предстоящей встречи.
Присутствует кнопка/ссылка “Отправить pdf-файл/файлы”
Присутствует кнопка/ссылка “Удалить pdf-файлы”
- Кнопка/ссылка для удаления встречи.

3. Страница просмотра встречи (для участников встречи) – meet.html

- На странице отображается общая информация о встрече (дата, время, email лидера команды, который создал встречу)
- Доступен просмотр информации об участниках, их выбранных временных интервалов по дням.
- В качестве фона страницы используется изображение. Тип размещения изображения - “замостить” (многократное повторение одного изображения по горизонтали и по вертикали)
- Форма для ввода email и пароля, а также кнопка/ссылка “Далее”

Учтите, что пользователи могут распечатать содержимое страницы, чтобы иметь печатную копию информации о встрече.

При подготовке к печати вам необходимо изменить отображение контента (структуру и оформление). При этом вся информация должна быть в черно-белом варианте отображения (допускаются градации серого), единственная цветная область (фрагмент/элемент) - логотип, измененный гарнитура шрифта

4. Страница встречи (управление участником) – meet-edit.html

Переход на страницу осуществляется со *страницы 3*.

Участники могут указать свои временные интервалы для участия.

Участник может посмотреть, кто и когда готов подключиться. На странице должна выводиться информация по всем дням, которые мог указать лидер при создании встречи – потенциально до 5 дней.

Дополнительная информация и требования к странице:

- Доступна информация о встрече (дата, время, email лидера команды).
- Доступен просмотр информации об участниках, их выбранными временными интервалами по дням.
- В качестве фона страницы используется изображение. Тип размещения изображения - “замостить” (многократное повторение одного изображения по горизонтали и по вертикали)

- Кнопка/ссылка "Сохранить" для сохранения изменений.

5. Личный кабинет – account.html

Доступ к странице доступен со *страницы 1* при указании email и пароля в форме аутентификации.

На странице отображается следующая информация, а также есть возможность выполнить описанные действия:

- Доступна информация о запланированных ранее встречах (как прошедших, так и предстоящих)
 - Для каждой встречи, для которой пользователь является лидером, отображается ссылка для приглашения участников в формате **{host}/leader.html**
 - Для каждой встречи, для которой пользователь является лидером, присутствуют ссылка/кнопка для просмотра более подробной информации и редактирования каждой встречи - переход на *страницу 2* по ссылке вида **{host}/leader.html**
 - Для каждой встречи, для которой пользователь является лидером, присутствуют ссылки/кнопки для удаления записи о каждой встрече. После удаления ссылка на встречу не отображается в ЛК лидера, пользователю отображается *страница 6* с сообщением об удаленной встрече по ссылке вида **{host}/404.html**.
 - Для каждой встречи, для которой пользователь не является лидером (т.е. он был приглашен другим лидером), отображается ссылка в формате **{host}/meet.html** с возможностью перехода на *страницу 4*
- Реализована кнопка/ссылка для выхода из профиля

6. Встреча не найдена – 404.html

Отображается сообщение, что встреча не найдена.

Оформление страницы соответствует дизайн-решениям других страниц.

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Ваш HTML/CSS должен быть валидным. Используйте лучшие практики верстки в соответствии с заданием.

Вам нужно использовать не менее одного из представленных шрифтов.

Оценка будет производиться в браузере Google Chrome.

Сверстанный веб-сайт должен быть доступен по адресу <http://nnnnnn-m1.wsr.ru/>, где xxxxxx - email участника (указан на индивидуальной карточке).

Сохраните сверстанные страницы и соответствующие дизайн-макеты со следующими именами:

1. Главная страница – index.html, index.png
2. Страница управления встречами (лидером) – leader.html, leader.png
3. Страница просмотра встречи (для участников встречи) – meet.html, meet.png
4. Страница встречи (управление участником) –meet-edit.html, meet-edit.png
5. Личный кабинет– account.html, account.png

6. Встреча не найдена – 404.html, 404.png

Все страницы указанные выше должны быть доступны к просмотру по соответствующим адресам: <http://nnnnnn-m1.wsr.ru/index.html>, <http://nnnnnn-m1.wsr.ru/leader.html> и т.д.

Макеты расположены в отдельной папке.

На страницах должны использоваться анимации и микроанимации. Созданные анимированные эффекты должны быть уместны и способствовать формированию положительного впечатления о сайте.

Страницы должны корректно отображаться как на мобильных устройствах, так и на десктопных - в том числе на страницах управления встречами, их просмотра и т.д.

ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!

Страницы, расположенные в других местах или с ошибками в названии проверяться не будут!

Работы, выполненные с использованием CSS-фреймворков, проверяться не будут!

В ходе разработки необходимо использовать систему контроля версий GIT. Должно быть не менее 3 коммитов с осмысленными комментариями!

Модуль Б. Разработка Веб-приложения на стороне клиента

Технологии этого модуля: Клиентское программирование REST API

Время на выполнение: 3 часа

К вам обратился заказчик, который просит разработать сервис для планирования онлайн встреч.

Сайт предназначен для участников команды, которые хотят запланировать встречу. Проблема в том, что лидер не знает, когда каждый из участников команды может присоединиться.

Саму встречу создает лидер команды, указывает день (дни) встречи и временные интервалы для этой встречи. Лидер должен иметь возможность планировать встречу на несколько дней. После этого ему доступен инструмент приглашения участников команды. Каждый из участников может указать временные интервалы, когда он может присоединиться к встрече. Информация о присоединившихся отображается лидеру.

Заказчик предоставляет вам полностью готовую верстку со всеми страницами и рабочее API. Вам необходимо использовать все имеющиеся навыки в клиентской разработке для создания Single Page Application, далее SPA.

На страницах должны использоваться анимации и микроанимации, реализованные с помощью JS. Переходы между экранами должны быть анимированы. Созданные

анимированные эффекты должны способствовать формированию положительного впечатления о сайте.

Заказчик хочет, чтобы API можно было легко поддерживать, поэтому использование JavaScript фреймворков (библиотек) будет плюсом. Вам доступны библиотеки/фреймворки Vue, React, Angular.

ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – реализовать SPA приложение, которое будет работать с уже разработанным API. Для вашего удобства во всех URL будет использоваться переменная {host} которая обозначает хост адрес API вида: [http:// nnnnnn.wsr.ru/api](http://nnnnnn.wsr.ru/api)

Ваше SPA должно состоять из следующих экранов (сверстанные страницы предоставляются):

1. Главная страница – index.html (для SPA и API {host}/)
2. Страница управления встречами (лидером) – leader.html (для SPA и API {host}/yuyu/xxxx)
3. Страница просмотра встречи (для участников встречи) – meet.html (для SPA и API {host}/yuyu)
4. Страница встречи (управление участником) – meet-edit.html (для SPA и API {host}/yuyu/edit)
5. Личный кабинет – account.html (для SPA и API {host}/account)
6. Встреча не найдена – 404.html (для SPA и API {host}/404)

Приложение должно обладать следующими функциональными возможностями.

Вся валидация (кроме размера и типа файлов) проводится средствами JS/HTML!

Отправка некорректных данных на сервер невозможна.

Успешные действия, а также ответы сервера должны отображаться на странице (например, при успешной отправке запроса, получении ответа сервера о возникшей ошибке и т.д.)

Аутентификация

Запрос для входа пользователя в систему. При отправке запроса необходимо передать объект со следующими свойствами:

- email - обязательное;
- пароль - обязательное;

После нажатия/клика по кнопке/ссылке "Войти" происходит переадресация на страницу личного кабинета лидера команды - *Страница 5*. В случае ошибки аутентификации/авторизации появляется сообщение с текстом ошибки.

Request	Response
<p>URL: {{host}}/api/user/login Method: POST</p> <p>Headers -Content-Type: application/json</p> <p>Body: <pre>{ "email": "user@user.ru", "password": "Password1" }</pre></p>	<p>----- Successful ----- Status: 200 Content-Type: application/json Body: <pre>{ "data": { "token": "bGUXVw4Fxdq_ErSekIaZnFAFXoKHTmU4" } }</pre></p> <p>-----Unauthorized----- Status: 401 Content-Type: application/json Body: <pre>{ "error": { "code": 401, "message": "Unauthorized" } }</pre></p>

Создание встречи

Запрос для создания новой встречи в системе. При отправке запроса необходимо передать объект со следующими свойствами:

- title - название встречи, обязательное;
- description - описание встречи, необязательное;
- e-mail - обязательное, формат email;
- password - пароль, обязательное, от 7 до 10 символов, допускаются только символы латинского алфавита и цифры, обязательно использование символа “*”;
- dates - массив дней. До 5 дней для планирования, выбирать можно произвольный интервал - до 30 дней вперед, начиная с текущей даты. Должна быть указана минимум 1 дата. Даты не должны быть одинаковыми. Указать прошедшие даты невозможно. Выбрать даты позже 30 дней также нельзя;
- start - начало интервала времени проведения встреч - по умолчанию с 9 часов.
- end - окончание интервала времени проведения встреч - по умолчанию с 18 часов;
- confirm - подтверждение согласие с политикой конфиденциальности, обязательное для заполнения.

Интервалы применяется сразу ко всем дням, указанным лидером. Начальное и конечное значения временного диапазона могут изменены.

Отправка запроса на создание встречи происходит по нажатию на кнопку/ссылку "Создать". Если все данные указаны корректно, после нажатия/клика по кнопке происходит

переадресация на страницу управления встречами лидером команды - **Страница 2**, иначе отображаются сообщения об ошибочно заполненных полях, данные не отправляются.

Request	Response
URL: {host}/api/meet Method: POST Body: <pre>{ "title": "title", "description": "description", "email": "user1@user.ru", "password": "Password*", "dates": ["02-03-2024",], "start": "09:00", "end": "18:00", "confirm": 1 }</pre>	<pre>----- Successful ----- Status: 200 Content-Type: application/json Body: { "data": { "meet": { "hash": "Co72ZvZK4wHbdMDZaumL_zoWm- aZX_cP", "user": { "id": 23, "leaderHash": "uP2tYXXAFB9Tyau_Op2DEmN qpJVII1km" } } } }</pre>

2. Страница управления встречами (лидером)

Страница управления встречами - лидер может просматривать информацию о встрече и вносить изменения.

- Доступна только для лидера - вход по:
 - “секретной ссылке” вида {host}/yyyy/edit
 - по “секретной ссылке”, расположенной на *странице 5*
 - email и паролю со *страницы 3*

Также переход на эту страницу осуществляется сразу после создания встречи лидером (*со страницы 1*).

Информация на странице и соответствующие элементы:

- Отображается “секретная ссылка” в формате - {host}/yyyy/xxxx, где yyyy - идентификатор встречи, xxxx - идентификатор лидера команды.
- У всех встреч, созданных одним пользователем (по паре email-пароль) идентификатор xxxx одинаковый
- Отображается ссылка для приглашения участников в формате {host}/yyyy, где yyyy - идентификатор встречи, полученный со стороны сервера
- Возможность указать лидеру временные интервалы, в которые он может принять участие на встречах. Временные интервалы лидера должны визуально выделяться.
- Возможность изменить свои временные интервалы, когда лидер может принять участие.
- Доступен просмотр информации об участниках, их выбранных временных интервалов по дням. Визуально видно, когда большая часть участников может присоединиться к встрече.

- Кнопка/ссылка для удаления встречи.

Удаление информации об участниках

Запрос на удаление участника из встречи - временные интервалы, указанные этими пользователями ранее, после этого не отображаются на странице.

Request	Response
<p>URL: {{host}}/api/meet/{{meetHash}}/leaderHash}}/user/{{userID}}</p> <p>Method: DELETE</p>	<p>----- Successful ----- Status: 204</p> <p>----- User not found ----- Status: 404 Content-Type: application/json Body: <pre>{ "error": { "code": 404, "message": "Не найдено" } }</pre></p> <p>----- Meet not found ----- Status: 404 Content-Type: application/json Body: <pre>{ "error": { "code": 404, "message": "Не найдено" } }</pre></p> <p>----- Leader hash is invalid ----- Status: 403 Content-Type: application/json Body: <pre>{ "error": { "code": 403, "message": "Недоступно для вас" } }</pre></p>

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера;
- {{userID}} - id удаляемого участника.

Удаление файлов

Запрос на удаление файла встречи (изображения и/или pdf).

Request	Response
URL: {{host}}/api/meet/ {{meetHash}}/{{leaderHash}}/ file/{{filename}} Method: DELETE	<pre> ----- Successful ----- Status: 204 -----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } } -----Leader hash is invalid----- Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Недоступно для вас" } } </pre>

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера;
- {{filename}} - имя удаляемого файла.

Блокировка встречи

Запрос на блокировку встречи.

Request	Response
URL: {{host}}/api/meet/ {{meetHash}}/{{leaderHash}} Method: PATCH	<p>----- Successful ----- Status: 204</p> <p>-----Meet is already block----- Status: 409 Content-Type: application/json Body: { "error": { "code": 409, "message": "Встреча уже заблокирована" } }</p> <p>-----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p> <p>-----Leader hash is invalid----- Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Недоступно для вас" } }</p>

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера.

Загрузка файлов

Запрос на загрузку файлов. При отправке запроса необходимо передать объект с одним из свойств:

- `img` - изображение, обязательно для заполнения, допустимые форматы загружаемого изображения - `png`, `jrg`, `jreg`. Загруженное изображение в дальнейшем будет использоваться в качестве фонового изображения для *страниц 3 и 4*.
- `files` - объект с файлами, до 2 файлов в формате `pdf` с информацией для предстоящей встречи. Отправка файлов происходит одним полем.

Request	Response
<p>URL: {{host}}/api/meet/{{meetHash}}/{{leaderHash}}</p> <p>Method: POST</p> <p>Headers --Content-Type: Form-data</p> <p>Body: { img: file }</p> <p>или</p> <p>Body: { files: [file1, file2] }</p>	<p>----- Successful -----</p> <p>Status: 204</p> <p>-----Validation error-----</p> <p>Status: 422 Content-Type: application/json Body: { "error": { "code": 422, "message": "Validation error", "errors": { "upload_img": ["Файл «175715-nacionalnyj_park_glejsher-ozero_tu_medisin-lavina_ozero-gora_rejnolds-ozero_makdonald-7680x4320.jpg» слишком большой. Размер не должен превышать 2.00 МиБ."], "upload_files": ["Файлов слишком много. Максимально можно сохранить 2 файла"] } } }</p> <p>-----Meet not found-----</p> <p>Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p> <p>-----Leader hash is invalid-----</p> <p>Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Недоступно для вас" } }</p>

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера.

Отправка приглашений участникам по почте

Запрос на отправку приглашений участникам по почте. При отправке запроса необходимо передать объект с адресами электронных почт (саму отправку сообщений проводить не требуется)

Если данные отправлены успешно, то выводится соответствующее сообщение

Request	Response
URL: {{host}}/api/meet/{{meetHash}}/{{leaderHash}}/invite Method: POST Headers - Content-type: application/json Body: { "user1@user.ru", "user2@user.ru", "user3@user.ru" }	----- Successful ----- Status: 204 -----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } } -----Leader hash is invalid----- Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Недоступно для вас" } }

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера.

3. Страница просмотра встречи (для участников встречи)

Страница доступна по ссылке-приглашению от лидера или по ссылке, полученной на e-mail.

- На странице отображается общая информация о встрече (дата, время, email лидера команды);
- Доступен просмотр информации об участниках, их выбранных временных интервалов по дням;
- Если ранее лидером был загружен графический файл, то он используется в качестве фонового изображения страницы. Тип размещения - "замостить".

Получение информации о встрече

Запрос на получение информации о встрече.

Request	Response
<p>URL: <code>{{host}}/api/meet/ {{meetHash}}</code> Method: GET</p>	<pre> ----- Successful ----- Status: 200 Content-Type: application/json Body: { "meet": { "title": "title", "description": "description", "dates": ["02-03-2024",], "start": "09:00", "end": "18:00", "interval": 60, "block": 0, "users": { "leader@gmail.com": { "id": 5, "availables": [[1,1,1,1,1,0,0,0,0],] }, "participant@gmail.com": { "id": 6, "availables": [[0,0,0,1,1,0,0,0,0],] } }, "leader": { "id": 5, "login": "leader@gmail.com" }, "img": "background.jpg", "files": [], } } -----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } } </pre>

- `{{meetHash}}` - идентификатор встречи.

Регистрация на встречу

Запрос на присоединение пользователя к встрече. При отправке запроса необходимо передать объект со следующими свойствами:

- `email` - обязательно;
- `password` - обязательно.

Request	Response
<p>URL: <code>{{host}}/api/meet/{{meetHash}}/login</code> Method: POST</p> <p>Headers -Content-type: application/json</p> <p>Body: <pre>{ "email": "user@user.ru", "password": "Password1" }</pre></p>	<p>----- Successful ----- Status: 200 Content-Type: application/json Body: <pre>{ "data": { "user": { "id": 25, "token": "sdwwkeqOWewnsdsawdsawWobnf" } } }</pre></p> <p>-----Unauthorized----- Status: 401 Content-Type: application/json Body: <pre>{ "error": { "code": 401, "message": "Unauthorized" } }</pre></p> <p>-----Forbidden----- Status: 403 Content-Type: application/json Body: { <pre> "error": { "code": 403, "message": "Недоступно для вас" } }</pre></p> <p>-----Meet is block-----</p>

	Status: 400 Content-Type: application/json Body: { "error": { "code": 400, "message": "Встреча заблокирована" } } -----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }
--	--

- {{meetHash}} - идентификатор встречи.

4. Страница управления встречами (участником)

Переход на страницу осуществляется со *страницы 3*.

Участники могут указать свои временные интервалы для участия. Участник может получить доступ к этой странице повторно, если ранее указал и email, и пароль (на *странице 3*).

Участник может просмотреть, кто и когда готов подключиться. На странице должна выводиться информация по всем дням, которые указал лидер.

Дополнительная информация и требования к странице:

- Доступна информация о встрече (дата, время, лидер команды).
- Доступен просмотр информации об участниках, их выбранными временными интервалами по дням.
- Если ранее лидером был загружен графический файл, то он используется в качестве фонового изображения страницы. Тип размещения - "замостить"
- Кнопка "Сохранить" для сохранения изменений - опционально, если данные по временным интервалам не отправляются автоматически.

Изменение интервалов пользователями

Запрос на изменение интервалов пользователя. При отправке запроса необходимо передать объект с интервалами пользователя, где 1 - пользователь может принять участие, 0 - не может принять участие.

Request	Response
---------	----------

<p>URL: {{host}}/api/meet/ {{meetHash}}/user/{{userID}}</p> <p>Method: PATCH</p> <p>Headers -Content-type: application/json</p> <p>Body: { "dates": ["02-03-2024",], "availables": [[1,1,1,1,1,0,0,0],] }</p>	<p>----- Successful -----</p> <p>Status: 204</p> <p>-----Meet not found-----</p> <p>Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p> <p>-----User not found-----</p> <p>Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p>
--	---

- {{meetHash}} - идентификатор встречи.
- {{userID}} - id пользователя .

5. Страница личного кабинета

Доступ к странице доступен со *страницы 1* при указании email и пароля в форме аутентификации.

На странице отображается следующая информация, а также есть возможность выполнить описанные действия:

- Доступна информация о запланированных ранее встречах (как прошедших, так и предстоящих)
 - Для каждой встречи, для которой пользователь является лидером, отображается ссылка для приглашения участников в формате **{host}/yyyy**, где уууу - идентификатор встречи
 - Для каждой встречи, для которой пользователь является лидером, присутствуют ссылка/кнопка для просмотра более подробной информации и редактирования каждой встречи - переход на *страницу 2* по ссылке вида **{host}/yyyy/xxxx**
 - Для каждой встречи, для которой пользователь является лидером, присутствуют ссылки/кнопки для удаления записи о каждой встрече. После удаления ссылка на встречу не отображается в ЛК лидера, пользователю отображается *страница 6* с сообщением об удаленной встрече.
 - Для каждой встречи, для которой пользователь не является лидером (т.е. он был приглашен другим лидером), отображается ссылка в формате **{host}/yyyy**, где уууу - идентификатор встречи с возможностью перехода на *страницу 4*
- Реализована кнопка/ссылка для выхода из профиля

Получение информации о встречах пользователя

Request	Response
<p>URL: {{host}}/api/profile Method: GET Headers -Authorization: Bearer {{token}}</p>	<pre> ----- Successful ----- Status: 200 Content-Type: application/json Body: { "user": { "id": 2, "email": "leader@email.com", "identifier": "-ewqQw" "meets_leader": [{ "id": 1 "title": "title", "hash": "xxxxx", "block": 0 },], "meets_partic": [{ "id": 2 "title": "title", "hash": "xxxxx", "block": 0 },],] } -----Unauthorized----- Status: 401 Content-Type: application/json Body: { "error": { "code": 401, "message": "Unauthorized" } } </pre>

Удаление встречи

Запрос на удаление встречи.

Request	Response
---------	----------

<p>URL: {{host}}/api/meet/ {{meetHash}}/{{hashLeader}}</p> <p>Method: DELETE</p>	<p>----- Successful ----- Status: 204</p> <p>-----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p> <p>-----Hash leader is invalid----- Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Не доступно для вас" } }</p>
--	---

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера.

Выход из личного кабинета

Запрос на выход из личного кабинета.

Request	Response
<p>URL: {{host}}/api/logout Method: GET</p> <p>Headers -Authorization: Bearer {{token}}</p>	<p>----- Successful ----- Status: 204</p> <p>-----Unauthorized----- Status: 401 Content-Type: application/json Body: { "error": { "code": 401, "message": "Unauthorized" } }</p>

Дополнительные запросы

Получение изображения встречи

Запрос на получении изображении встречи.

Request	Response
URL: <code>{{host}}/api/meet/{{meetHash}}/file/{{filename}}</code> Method: GET	----- Successful ----- Status: 200 Файл отдается на скачивание -----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }

- `{{meetHash}}` - идентификатор встречи;
- `{{filename}}` - имя файла.

Проверка идентификатора лидера встречи

Запрос на проверку идентификатора лидера встречи.

Request	Response
---------	----------

<p>URL: <code>{{host}}/api/meet/{{meetHash}}/{{hashLeader}}</code> Method: GET</p>	<p>----- Successful ----- Status: 204</p> <p>-----Meet not found----- Status: 404 Content-Type: application/json Body: <pre>{ "error": { "code": 404, "message": "Не найдено" } }</pre></p> <p>-----Hash leader is invalid----- Status: 403 Content-Type: application/json Body: <pre>{ "error": { "code": 403, "message": "Недоступно для вас" } }</pre></p>
---	---

- `{{meetHash}}` - идентификатор встречи;
- `{{hashLeader}}` - идентификатор лидера.

Вся валидация (кроме размера и типа файлов) проводится средствами JS/HTML!

Отправка некорректных данных на сервер невозможна.

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Разработанное приложение должно быть доступно по адресу `http://nnnnnn-m2.wsr.ru/`, где `nnnnnn` - логин участника (указан на индивидуальной карточке).

Проверка работы будет производиться в браузере Google Chrome.

На страницах должны использоваться анимации и микроанимации, реализованные с помощью JS. Созданные анимированные эффекты должны способствовать формированию положительного впечатления о сайте.

ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!

Модуль В. Разработка Веб-приложения на стороне сервера

Технологии этого модуля: REST API

Время на выполнение: 3 часа

К вам обратился заказчик, который просит разработать сервис для планирования онлайн встреч.

Сайт предназначен для участников команды, которые хотят запланировать встречу. Проблема в том, что лидер не знает, когда каждый из участников команды может присоединиться.

Вам необходимо использовать все имеющиеся навыки в серверной разработке для создания REST API.

Заказчик хочет, чтобы api можно было легко поддерживать, поэтому использование фреймворков будет плюсом.

Доступные фреймворки: Yii2, Laravel, Django

ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!

В медиафайлах вам предоставляется sql дамп с готовой базой данных. База данных уже содержит набор данных, которые НЕ должны быть изменены! Структуру БД менять также запрещается. Любое изменение или удаление предоставленных данных в БД будет влиять на вашу оценку.

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – реализовать REST API, которое будет отвечать требованиям заказчика.

Для вашего удобства, во всех URL будет использоваться переменная {host} которая обозначает адрес `http://nnnnnn-m3.wsr.ru/`, где xxxxxx - email участника.

В случае ошибок связанных с валидацией данных во всех запросах необходимо возвращать следующее тело ответа:

```
{
  "error": {
    "code": <code>,
    "message": <message>,
    "errors": {
      <key>: [ <error message>]
    }
  }
}
```

Обратите внимание, что вместо `<code>` и `<message>` необходимо указывать соответствующее значение, определенное в описании ответа на соответствующий запрос. В свойстве `error.errors` необходимо перечислить те свойства, которые не прошли валидацию, а в их значениях указать массив с ошибками валидации.

Например, если отправить пустой запрос на сервер, где проверяется следующая валидация:

- `phone` – обязательно поле
- `password` – обязательное поле

то тело ответа должно быть следующим

```
{
  "error": {
    "code": 422,
    "message": "Validation error",
    "errors": {
      phone: [ "field phone can not be blank" ],
      password: [ "field password can not be blank" ]
    }
  }
}
```

Учтите, что `code` и `message` могут быть определены иначе, если в запросе указано иное. В значениях свойств `errors` вы можете использовать любые сообщения об ошибках (если не указана конкретная ошибка), но они должны описывать возникшую проблему.

Аутентификация

Запрос для входа пользователя в систему. При отправке запроса необходимо передать объект со следующими свойствами:

- `email` - обязательное;
- `пароль` - обязательное;

Request	Response
<p>URL: <code>{{host}}/api/user/login</code> Method: POST</p> <p>Headers -Content-Type: application/json</p> <p>Body: <pre>{ "email": "user@user.ru", "password": "Password1" }</pre></p>	<p>----- Successful -----</p> <p>Status: 200 Content-Type: application/json Body: <pre>{ "data": { "token": "bGUXVw4Fxdq_ErSekIaZnFAFXoKHTmU4" } }</pre></p> <p>----- Validation Error -----</p> <p>Status: 422 Content-Type: application/json Body: <pre>{</pre></p>

	<pre> "error": { "code": 422, "message": "Validation error", "errors": { "email": ["Необходимо заполнить «Email»."] } } } } -----Unauthorized----- Status: 401 Content-Type: application/json Body: { "error": { "code": 401, "message": "Unauthorized" } } </pre>
--	--

Создание встречи

Запрос для создание новой встречи в системе. При отправке запроса необходимо передать объект со следующими свойствами:

- title - название встречи, обязательное;
- description - описание встречи, необязательное;
- e-mail - обязательное, формат email;
- password - пароль, обязательное, от 7 до 10 символов, допускаются только символы латинского алфавита и цифры, обязательно использование символа “*”;
- dates - массив дней. До 5 дней для планирования, выбирать можно произвольный интервал - до 30 дней вперед, начиная с текущей даты. Должна быть указана минимум 1 дата. Даты не должны быть одинаковыми. Указать прошедшие даты невозможно. Выбрать даты позже 30 дней также нельзя;
- start - начало интервала времени проведения встреч - по умолчанию с 9 часов.
- end - окончание интервала времени проведения встреч - по умолчанию с 18 часов;
- confirm - подтверждение согласие с политикой конфиденциальности, обязательное для заполнения;

Интервалы применяется сразу ко всем дням, указанным лидером. Начальное и конечное значения временного диапазона могут изменены.

Request	Response
URL: {host}/api/meet Method: POST Body: <pre>{ "title": "title", "description": "description", "email": "user1@user.ru", "password": "Password*", "dates": ["02-03-2024",], "start": "09:00", "end": "18:00", "confirm": 1 }</pre>	<pre>----- Successful ----- Status: 200 Content-Type: application/json Body: { "data": { "meet": { "hash": "Co72ZvZK4wHbdMDZaumL_zoWm- aZX_cP", "user": { "id": 23, "leaderHash": "uP2tYXXAFB9Tyau_Op2DEmNqpJ VII1km" } } } } ----- Validation error ----- Status: 422 Content-Type: application/json Body: { "error": { "code": 422, "message": "Validation error", "errors": { "key" => "value" }, } }</pre>

2. Страница управления встречами (лидером)

Примечание: У всех встреч, созданных одним пользователем (по паре email-пароль) идентификатор **xxxx** должен быть одинаковым

Удаление информации об участниках

Запрос на удаление участника из встречи.

Request	Response
---------	----------

<p>URL: {{host}}/api/meet/ {{meetHash}}/{leaderHash}}/ user/{{userID}}</p> <p>Method: DELETE</p>	<p>----- Successful ----- Status: 204</p> <p>----- User not found ----- - Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p> <p>-----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p> <p>-----Leader hash is invalid----- Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Недоступно для вас" } }</p>
--	---

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера;
- {{userID}} - id удаляемого участника.

Удаление файлов

Запрос на удаление файла встречи.

Request	Response
---------	----------

<p>URL: {{host}}/api/meet/ {{meetHash}}/{{leaderHash}}/ file/{{filename}} Method: DELETE</p>	<pre> ----- Successful ----- Status: 204 -----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } } -----Leader hash is invalid----- Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Недоступно для вас" } } </pre>
---	--

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера;
- {{filename}} - имя удаляемого файла.

Блокировка встречи

Запрос на блокировку встречи.

Request	Response
URL: {{host}}/api/meet/ {{meetHash}}/{{leaderHash}} Method: PATCH	<p>----- Successful ----- Status: 204</p> <p>-----Meet is already block----- Status: 409 Content-Type: application/json Body: { "error": { "code": 409, "message": "Встреча уже заблокирована" } }</p> <p>-----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p> <p>-----Leader hash is invalid----- Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Недоступно для вас" } }</p>

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера.

Загрузка файлов

Запрос на загрузку файлов. При отправке запроса необходимо передать объект с одним из свойств:

- `img` - изображение, обязательно для заполнения, допустимые форматы загружаемого изображения - `png`, `jrg`, `jreg`.
- `files` - объект с файлами, до 2 файлов в формате `pdf` с информацией для предстоящей встречи. Отправка файлов происходит одним полем.

Request	Response
<p>URL: {{host}}/api/meet/{{meetHash}}/{{leaderHash}}</p> <p>Method: POST</p> <p>Headers --Content-Type: Form-data</p> <p>Body: { img: file }</p> <p>или</p> <p>Body: { files: [file1, file2] }</p>	<p>----- Successful -----</p> <p>Status: 204</p> <p>-----Validation error-----</p> <p>Status: 422 Content-Type: application/json Body: { "error": { "code": 422, "message": "Validation error", "errors": { "upload_img": ["Файл «175715-nacionalnyj_park_glejsher-ozero_tu_medisin-lavina_ozero-gora_rejnolds-ozero_makdonald-7680x4320.jpg» слишком большой. Размер не должен превышать 2.00 МиБ."], "upload_files": ["Файлов слишком много. Максимально можно сохранить 2 файла"] } } }</p> <p>-----Meet not found-----</p> <p>Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p> <p>-----Leader hash is invalid-----</p> <p>Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Недоступно для вас" } }</p>

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера.

Отправка приглашений участникам по почте

Запрос на отправку приглашений участникам по почте. При отправке запроса необходимо передать объект с адресами электронных почт (саму отправку сообщений проводить не требуется)

Request	Response
<p>URL: {{host}}/api/meet/{{meetHash}}/{{leaderHash}}/invite Method: POST</p> <p>Headers -Content-type: application/json</p> <p>Body: { "user1@user.ru", "user2@user.ru", "user3@user.ru" }</p>	<p>----- Successful ----- Status: 204</p> <p>-----Validation error----- Status: 422 Content-Type: application/json Body: { "error": { "code": 422, "message": "Validation error", "errors": { "email1": ["Является не корректным E-mail адресом",], }, }, } }</p> <p>-----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" }, } }</p> <p>-----Leader hash is invalid----- Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Недоступно для вас" }, } }</p>

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера.

3. Страница просмотра встречи (для участников встречи)

Получение информации о встрече

Request	Response
<p>URL: {{host}}/api/meet/ {{meetHash}}</p> <p>Method: GET</p>	<pre> ----- Successful ----- Status: 200 Content-Type: application/json Body: { "meet": { "title": "title", "description": "description", "dates": ["02-03-2024",], "start": "09:00", "end": "18:00", "interval": 60, "block": 0, "users": { "leader@gmail.com": { "id": 5, "availables": [[1,1,1,1,1,0,0,0,0],] }, "participant@gmail.com": { "id": 6, "availables": [[0,0,0,1,1,0,0,0,0],] } }, "leader": { "id": 5, "login": "leader@gmail.com" }, "img": "background.jpg", "files": [], } } -----Meet not found----- Status: 404 </pre>

	Content-Type: application/json Body: <pre>{ "error": { "code": 404, "message": "Не найдено" } }</pre>
--	---

- `{{meetHash}}` - идентификатор встречи.

Регистрация на встрече

Запрос на присоединение пользователя к встрече. При отправке запроса необходимо передать объект со следующими свойствами:

- `email` - обязательно;
- `password` - обязательно.

Request	Response
<p>URL: <code>{{host}}/api/meet/{{meetHash}}/login</code> Method: POST</p> <p>Headers -Content-type: application/json</p> <p>Body: <pre>{ "email": "user@user.ru", "password": "Password1" }</pre></p>	<p>----- Successful ----- Status: 200 Content-Type: application/json Body: <pre>{ "data": { "user": { "id": 25, "token": "sdwwkeqOWewnsdsawdsawWobnf" } } }</pre></p> <p>-----Unauthorized----- Status: 401 Content-Type: application/json Body: <pre>{ "error": { "code": 401, "message": "Unauthorized" } }</pre></p> <p>-----Forbidden----- Status: 403 Content-Type: application/json</p>

	<pre> Body: { "error": { "code": 403, "message": "Недоступно для вас" } } -----Meet is block----- Status: 400 Content-Type: application/json Body: { "error": { "code": 400, "message": "Встреча заблокирована" } } ----- Validation Error ----- ----- Status: 422 Content-Type: application/json Body: { "error": { "code": 422, "message": "Validation error", "errors": { "email": ["Необходимо заполнить «Email»."] } } } -----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } } </pre>
--	--

- {{meetHash}} - идентификатор встречи.

4. Страница управления встречами (участником) Изменение интервалов пользователями

Запрос на изменение интервалов пользователя. При отправке запроса необходимо передать объект с интервалами пользователя, где 1 - пользователь может принять участие, 0 - не может принять участие.

Request	Response
<p>URL: {{host}}/api/meet/{{meetHash}}/user/{{userID}}</p> <p>Method: PATCH</p> <p>Headers -Content-type: application/json</p> <p>Body: { "dates": ["02-03-2024",], "availables": [[1,1,1,1,1,0,0,0,0],] }</p>	<p>----- Successful -----</p> <p>Status: 204</p> <p>-----Meet not found-----</p> <p>Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p> <p>-----User not found-----</p> <p>Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p>

- {{meetHash}} - идентификатор встречи.
- {{userID}} - id пользователя .

5. Страница личного кабинета

Получение информации о встречах пользователя

Request	Response
---------	----------

<p>URL: {{host}}/api/profile Method: GET Headers -Authorization: Bearer {{token}}</p>	<pre> ----- Successful ----- Status: 200 Content-Type: application/json Body: { "user": { "id": 2, "email": "leader@email.com", "identifier": "-ewqQw" "meets_leader": [{ "id": 1 "title": "title", "hash": "xxxxx", "block": 0 },], "meets_partic": [{ "id": 2 "title": "title", "hash": "xxxxx", "block": 0 },]] } -----Unauthorized----- Status: 401 Content-Type: application/json Body: { "error": { "code": 401, "message": "Unauthorized" } } </pre>
--	--

Удаление встречи

Запрос на удаление встречи.

Request	Response
---------	----------

<p>URL: {{host}}/api/meet/ {{meetHash}}/{{hashLeader}}</p> <p>Method: DELETE</p>	<p>----- Successful ----- Status: 204</p> <p>-----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }</p> <p>-----Hash leader is invalid----- Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Не доступно для вас" } }</p>
--	---

- {{meetHash}} - идентификатор встречи;
- {{leaderHash}} - идентификатор лидера.

Выход из личного кабинета

Запрос на выход из личного кабинета.

Request	Response
<p>URL: {{host}}/api/logout Method: GET</p> <p>Headers -Authorization: Bearer {{token}}</p>	<p>----- Successful ----- Status: 204</p> <p>-----Unauthorized----- Status: 401 Content-Type: application/json Body: { "error": { "code": 401, "message": "Unauthorized" } }</p>

Дополнительные запросы

Получение изображения встречи

Запрос на получении изображении встречи.

Request	Response
URL: <code>{{host}}/api/meet/{{meetHash}}/file/{{filename}}</code> Method: GET	----- Successful ----- Status: 200 Файл отдается на скачивание -----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } }

- `{{meetHash}}` - идентификатор встречи;
- `{{filename}}` - имя файла.

Проверка идентификатора лидера встречи

Запрос на проверку идентификатора лидера встречи.

Request	Response
---------	----------

<pre> URL: {{host}}/api/meet/ {{meetHash}}/{{hashLeader}} Method: GET </pre>	<pre> ----- Successful ----- Status: 204 -----Meet not found----- Status: 404 Content-Type: application/json Body: { "error": { "code": 404, "message": "Не найдено" } } -----Hash leader is invalid----- Status: 403 Content-Type: application/json Body: { "error": { "code": 403, "message": "Недоступно для вас" } } </pre>
--	--

- {{meetHash}} - идентификатор встречи;
- {{hashLeader}} - идентификатор лидера.

Заказчик допускает возможность изменения базы данных в будущем, поэтому вам необходимо подготовить свой вариант схемы базы данных и сохранить его в корне с модулем. Сохраните файл с названием DB.png.

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Разработанное API должно быть доступно по адресу <http://nnnnnn-m3.wsr.ru/>, где xxxxxx - логин участника (указан на индивидуальной карточке).

Форматы запросов и ответов, а также форматы дат и времени должен соответствовать примерам из задания.

Проверяются только работы, загруженные на сервер.

В медиафайлах вам предоставляется sql дамп с готовой базой данных. База данных уже содержит набор данных, которые НЕ должны быть изменены! Структуру БД менять также запрещается. Любое изменение или удаление предоставленных данных в БД будет влиять на вашу оценку.

Модуль Г. Разработка ИР с использованием готовых решений

Технологии модуля: HTML5, CSS3, CMS WordPress, JavaScript, граф. дизайн

Время на выполнение: 3 часа

К вам обратился заказчик, который просит разработать промо-сайт для сервиса планирования онлайн встреч. На сайте должна быть представлена основная информация о сервисе, а также дополнительные возможности в соответствии с требованиями ниже.

Разработку нового сайта необходимо произвести с использованием системы управления контентом WordPress.

ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – разработать сайт согласно предоставленному дизайн-макету главной страницы сайта (media/img/design.png). Остальные страницы должны быть оформлены с учетом дизайн-решений главной страницы.

Вся информация должна меняться через панель администрирования

Структура главной страницы

Главная страница должна содержать секции/блоки в соответствии с дизайн-макетом. Цвета элементов также должны быть в строгом соответствии с предоставленным дизайн-макетом. Возможные блоки/секции/страницы:

- *Меню* (навигация по странице или к отдельной странице). Примеры пунктов меню представлены ниже
 - Логотип со ссылкой на главную страницу
 - О нас (ссылка на раздел на главной странице)
 - О системе (ссылка на раздел на главной странице)
 - О проекте (ссылка на соответствующую страницу)
 - Обратная связь (ссылка на соответствующую страницу)
 - Новости (ссылка на соответствующую страницу)
 - Контакты (ссылка на футер)
- *Слайдер* (не менее трех слайдов с использованием стороннего плагина).
- *О нас* (краткая информация из медиа).
- *О системе* (текст и видео из медиа-данных)
- *Новости* (4 блока с возможностью пагинации остальных новостей, с возможностью перехода на отдельную страницу новостей). Элементы новостной карточки:

- Главное изображение;
 - Название;
 - Краткое описание;
 - Автор
- *Подвал сайта (футер):*
 - Продублированное меню;
 - Кликабельный номер телефона;
 - Текущее время

Страница О проекте

Описание возможностей портала из медиа – необходимо использовать все медиаданные

Страница со всеми новостями проекта

На странице со всеми новостями отображаются все новости с пагинацией и фильтрацией по категориям:

- Возможности
- СМИ
- Развитие

Каждая новость отображается на отдельной странице и содержит следующую информацию:

- Главное изображение;
- Дополнительные изображения;
- Полное название;
- Категория;
- Описание;
- Автор
- Ссылки на дополнительные материалы ([http external links](#)).

Все данные должны быть взяты из предоставленных медиа ресурсов.

Страница Обратная связь

На данной странице должна находиться форма для связи с редакцией портала (ФИО, e-mail, сообщение и номер телефона, кнопка Отправить), реализованная с помощью **плагина Contact Form 7**. Все поля формы являются обязательными для заполнения.

Форма должна быть стилизована с учетом дизайн-макета главной страницы.

Также вам необходимо установить и настроить предоставленный плагин для SEO.

Плагин «Звонок диспетчеру»

Вам необходимо разработать плагин. Плагин должен выводиться в футере, на всех страницах и выводить номер телефона диспетчера. При наведении курсора мыши на номер

телефона должен быть выведен всплывающий блок, содержащий: заголовок (указанный в настройках плагина), номер телефона (указанный в настройках плагина), кнопку «позвонить», с реализованной возможностью осуществления звонка непосредственно со страницы. Плагин должен иметь возможность активации/деактивации. Реализация клиентской части плагина должна быть выполнена средствами JS.

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Разработанный веб-сайт должен быть доступен по адресу <http://nnnnnn-m4.wsr.ru/>, где xxxxxx - логин участника (указан на индивидуальной карточке).

Доступ к панели администратора: логин **rc_spb_2024**, пароль **you_are_the_best**. Панель администратора должна быть доступна по адресу, заданному по умолчанию в WordPress.

Вся информация (например, заголовки, текст, меню и т.д.) должна редактироваться в панели управления администратором сайта без знаний программирования, верстки или доступа к файловой системе сервера.

2. СПЕЦИАЛЬНЫЕ ПРАВИЛА КОМПЕТЕНЦИИ²

2.1. Личный инструмент конкурсанта

Конкурсанту разрешается использовать собственные:

- клавиатуру на любом языке. Если конкурсант пользуется своей клавиатурой, и она выходит из строя, организатор предоставляет ему замену.;
- языковые файлы для клавиатуры;
- мышь;
- графический планшет;
- наушники;
- аудиофайлы с музыкальными композициями (не более 30 файлов в формате mp3). Файлы предоставляются на флеш-носителях в день С-1 техническому эксперту на проверку.

Все оборудование не должно содержать встроенной памяти.

2.2. Материалы, оборудование и инструменты, запрещенные на площадке

Всё оборудование, принесенное конкурсантами, может быть проверено экспертами на наличие внутренних запоминающих устройств. В случае обнаружения материалы будут изыматься.

Экспертам допускается использовать персональные компьютеры, но в специальной зоне. В помещениях для проведения оценки использование любых электронных устройств запрещено, кроме специально организованных для оценки.

Также запрещено приносить:

- дополнительные программы и библиотеки, не предусмотренные инфраструктурным листом;
- мобильные телефоны;
- фото/видео устройства;
- карты памяти и другие носители информации;
- внутренние устройства памяти в собственном оборудовании.

² Указываются особенности компетенции, которые относятся ко всем возрастным категориям и чемпионатным линейкам без исключения.

3. ПРИЛОЖЕНИЯ

Приложение №1 Инструкция по заполнению матрицы конкурсного задания

Приложение №2 Матрица конкурсного задания

Приложение №3 Инфраструктурный лист

Приложение №4 Критерии оценки

Приложение №5 План застройки

Приложение №6 Инструкция по охране труда и технике безопасности по компетенции «Веб-технологии».

Приложение № 7 Медиа файлы