**КОНКУРСНОЕ ЗАДАНИЕ КОМПЕТЕНЦИИ**

**«ВЕБ-ТЕХНОЛОГИИ»**

2023 г.

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](#_Toc126846333)

[1.1. ОБЩИЕ СВЕДЕНИЯ О ТРЕБОВАНИЯХ КОМПЕТЕНЦИИ 3](#_Toc126846334)

[1.2. ПЕРЕЧЕНЬ ПРОФЕССИОНАЛЬНЫХ ЗАДАЧ СПЕЦИАЛИСТА ПО КОМПЕТЕНЦИИ «ВЕБ-ТЕХНОЛОГИИ» 3](#_Toc126846335)

[1.3. ТРЕБОВАНИЯ К СХЕМЕ ОЦЕНКИ 8](#_Toc126846336)

[1.4. СПЕЦИФИКАЦИЯ ОЦЕНКИ КОМПЕТЕНЦИИ 10](#_Toc126846337)

[1.5. КОНКУРСНОЕ ЗАДАНИЕ 10](#_Toc126846338)

[1.5.1. Разработка/выбор конкурсного задания (https://disk.yandex.ru/i/XJfQr1jBmZ2rew) 10](#_Toc126846339)

[1.5.2. Структура модулей конкурсного задания (инвариант/вариатив) 11](#_Toc126846340)

[Модуль А. Разработка интерфейса пользователя 11](#_Toc126846341)

[Модуль Б. Разработка Веб-приложения на стороне клиента 13](#_Toc126846342)

[Модуль В. Разработка Веб-приложения на стороне сервера 18](#_Toc126846343)

[Модуль Г. Разработка ИР с использованием готовых решений 37](#_Toc126846344)

[2. СПЕЦИАЛЬНЫЕ ПРАВИЛА КОМПЕТЕНЦИИ 40](#_Toc126846345)

[2.1. Личный инструмент конкурсанта 40](#_Toc126846346)

[2.2. Материалы, оборудование и инструменты, запрещенные на площадке 40](#_Toc126846347)

[3. ПРИЛОЖЕНИЯ 41](#_Toc126846348)

**ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ**

*1. ИР – 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:  − 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:  − собирать, анализировать и оценивать информацию;  − использовать навыки грамотности для толкования стандартов и требований;  − составлять тестовую документации для тестирования новых функциональностей продукта  − 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 | **Техническая поддержка и администрирование информационных ресурсов** | 7 |
| Специалист должен знать и понимать:  − 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. |
| Специалист должен уметь:  − 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 | **Разработка интерфейса пользователя** | 26 |
| Специалист должен знать и понимать:  − 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;  − 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; |
| Специалист должен уметь:  − 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 | **Разработка на стороне клиента** | 25 |
| Специалист должен знать и понимать:  − основные принципы паттерновой разработки веб-приложений;  − ECMAScript (JavaScript);  − принципы, особенности и способы использования открытых фреймворков;  − принципы разработка кода с использованием открытых библиотек;  − различные интерфейсы взаимодействия с объектами браузера |
| Специалист должен уметь:  − 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 | **Разработка веб приложения на стороне сервера** | 25 |
| Специалист должен знать и понимать:  − 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. |
| Специалист должен уметь:  − 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 | **Разработка информационных ресурсов с использованием готовых решений** | 10 |
| Специалист должен знать и понимать:  − 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. |
| Специалист должен уметь:  − 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** |  | **A** | **Б** | **В** | **Г** |  |
| **1** | 2 | 1 | 2.5 | 1,5 | 7 |
| **2** | 2 | 2 | 2 | 1 | 7 |
| **3** | 20 | 1 |  | 5 | 26 |
| **4** |  | 23 |  | 2 | 25 |
| **5** |  |  | 22 | 3 | 25 |
| **6** |  |  |  | 10 | 10 |
| **Итого баллов за критерий/модуль** | | 24 | 27 | 26,5 | 22,5 | **100** |

### 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

### Общая продолжительность Конкурсного задания[[1]](#footnote-1): 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.

Обязательная к выполнению часть (инвариант) выполняется всеми регионами без исключения на всех уровнях чемпионатов.

Количество модулей из вариативной части, выбирается регионом самостоятельно в зависимости от материальных возможностей площадки соревнований и потребностей работодателей региона в соответствующих специалистах. В случае если ни один из модулей вариативной части не подходит под запрос работодателя конкретного региона, то вариативный (е) модуль (и) формируется регионом самостоятельно под запрос работодателя. При этом, время на выполнение модуля (ей) и количество баллов в критериях оценки по аспектам не меняются.

*Таблица №4*

**Матрица конкурсного задания**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Обобщенная трудовая функция | Трудовая функция | Нормативный документ/ЗУН | Модуль | Константа/вариатив | ИЛ | КО |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Проектирование разделов ИР | ПС: 06.035 Разработчик Web и мультимедийных приложений | **Модуль 1 - Разработка интерфейса пользователя** | Константа | Весь ИЛ, на выбор участника | КО 1 |
| Техническая поддержка процессов создания (модификации) и сопровождения информационных ресурсов | Верстка страниц ИР |
| Техническая поддержка процессов создания (модификации) и сопровождения информационных ресурсов | Проверка и отладка программного кода | ПС: 06.035 Разработчик Web и мультимедийных приложений | **Модуль 2 - Разработка веб приложения на стороне клиента** | Константа | Весь ИЛ, на выбор участника | КО 2 |
| Техническая поддержка процессов создания (модификации) и сопровождения информационных ресурсов | Кодирование на языках веб-программирования |
| Техническая поддержка процессов создания (модификации) и сопровождения информационных ресурсов | Тестирование ИР с точки зрения логической целостности (корректность ссылок, работа элементов форм) |
| Техническая поддержка процессов создания (модификации) и сопровождения информационных ресурсов | Проверка и отладка программного кода | ПС: 06.035 Разработчик Web и мультимедийных приложений | **Модуль 3 - Разработка веб приложения на стороне сервера** | Константа | Весь ИЛ, на выбор участника | КО 3 |
| Техническая поддержка процессов создания (модификации) и сопровождения информационных ресурсов | Кодирование на языках веб-программирования |
| Техническая поддержка процессов создания (модификации) и сопровождения информационных ресурсов | Тестирование интеграции ИР с внешними сервисами и учетными системами |
| Техническая поддержка процессов создания (модификации) и сопровождения информационных ресурсов | Работа с системой контроля версий |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Установка и настройка прикладного программного обеспечения и модулей |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Проведение и регламентация работ по резервному копированию и развертыванию резервной копии ИР |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Обеспечение безопасной и бесперебойной работы сайта |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Сбор предварительных данных для выявления требований к ИР | ПС: 06.035 Разработчик Web и мультимедийных приложений | **Модуль 4 - Разработка ИР с использованием готовых решений** | Вариатив | Весь ИЛ, на выбор участника | КО 4 |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Определение первоначальных требований заказчика к ИР и возможности их реализации |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Планирование коммуникаций с заказчиком в рамках типовых регламентов организации |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Установка и настройка прикладного программного обеспечения и модулей |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Разработка процедур интеграции программных модулей |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Осуществление интеграции программных модулей и компонент и верификации выпусков программного продукта |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Управление доступом к данным и определение уровней прав пользователей ИР |
| Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов | Регистрация и обработка запросов Заказчика в службе технической поддержки в соответствии с трудовым заданием |

#### 1.5.2. Структура модулей конкурсного задания (инвариант/вариатив)

##### Модуль А. Разработка интерфейса пользователя

**Технологии этого модуля:** граф. Дизайн, HTML5, CSS3

**Время на выполнение:** 6 часов

**Задания:** В современном мире с каждым днём появляется всё больше и больше новых технологий. В настоящее время сложно представить организацию, у которой нет веб-сайта.

К вам обратилась компания «Amfiteatr» - занимающаяся организацией бронирования и продажей билетов на спектакли в театры России. Вам необходимо использовать все имеющиеся навыки в верстке чтобы создать Landing Page, а также остальные страницы. Используйте анимацию для привлечения внимания посетителя к акцентам.

Ваша задача – сверстать страницы веб-сайта со следующими именами:

● Главная страница (Landing Page) - index.html

● Страница входа в личный кабинет - login.html

● Страница регистрации в личном кабинете - register.html

● Страница личного кабинета - profile.html

● Страница покупки - purchase.html

● Страница бронирования - booking.html

● Страница банковские карты - bank\_card.html

**Главная страница** должна содержать логотип и название компании, поиск по сайту, авторизация или регистрация, основную информацию о спектаклях. Также заказчик хочет, чтоб на главной странице была реализована фильтрация спектакля: жанр, дата, стоимость, билет со скидкой, пушкинская карта.

Информация о спектаклях должна представляться в виде карточки в которой размещено: иллюстрация, название спектакля, краткое описание, дата и время, театр и стоимость (если есть скидка, то обозначить это), кнопка купить (при нажатии на кнопку открывается страница покупки), кнопка забронировать (при нажатии на кнопку открывается страница бронирование). Если есть возможность приобрести билет на спектакль по пушкинской карте, то обозначить это.

**Для регистрации пользователь** должен заполнить следующие поля: email, пароль, повторить пароль, а также подтвердить согласие с условиями пользовательского соглашения. Регистрация возможна с помощью социальных сетей VK и OK.

**Для входа** на сайт пользователь должен ввести: email и пароль. Если пользователь забыл пароль у него должна быть возможность его восстановить. Для входа на сайт можно использовать аккаунты в социальных сетях VK и OK. При нажатии на кнопку «Войти» пользователь переходит на страницу «Личный кабинет».

**В личном кабинете** у пользователя должна отображаться следующая информация: email (информация о пользователе), кнопка «выйти», информация о приобретенном и/или забронированном билете на спектакль (название спектакля, дата и время, место, ряд, стоимость, его статус(покупка или бронь). Если билет забронирован, реализуйте возможность перехода на страницу банковские карты для приобретения. Предусмотрите возможность перехода на главную страницу

**Страница покупки билета** на спектакль должна содержать: слайдер изображений, связанных со спектаклем, его название, описание, в каком театре будет проходить спектакль, стоимость билета. Пользователь должен иметь возможность выбрать дату и время, посмотреть занятые и свободные места на спектакль, выбрать ряд и место (возможность одновременной покупки 2 и более билетов), купить билет. При нажатии на кнопку «Купить билет» открывается страница «банковские карты». Осуществлена возможность перехода на главную страницу и личный кабинет.

**Страница бронирования** должна отображать следующую информацию: Фото, название спектакля, время, дата, стоимость, театр, выбор ряда, выбор места. Кнопка забронировать. Возможность перехода на главную страницу и личный кабинет.

На **странице банковские карты** необходимо отобразить информацию о привязанных банковских картах клиента: (Номер карты (должен быть не явным, отображаются последние 4 цифры), Тип карты (Visa, MasterCard, Мир), данные о пользователе (фамилия имя), срок годности(месяц/год) стоимость билета, кнопка оплатить.

Необходимо создать логотип компании в формате PNG.

Компания не хочет разбираться со сторонними авторскими правами на материалы, поэтому вы можете использовать только то, что предоставляет заказчик в медиафайлах или ваши личные дизайнерские разработки.

Ваш HTML/CSS должен быть валидным.

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

Проверяются только работы, загруженные на сервер! Страницы расположенные в других местах или с ошибками в названии проверяться не будут!

Оценка будет производиться в браузере Яндекс.Браузер.

Использование любых фреймворков и библиотек (bootstrap, например) запрещено.

##### Модуль Б. Разработка Веб-приложения на стороне клиента

**Технологии этого модуля:** Клиентское программирование

**Время на выполнение**: 3 часа

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: <http://yddfzaez-m3.wsr.ru>

Ваше SPA должно состоять из следующих экранов:

* Главная
* Вход в личный кабинет
* Регистрация в личном кабинете
* Личный кабинет
* Список турниры
* Регистрация команды
* Список матчей

Приложение должно обладать следующим функционалом:

1. Домашний экран – на данном экране располагается статическая информация и список игровых дисциплин.

При нажатии на иконку с логотипом игры пользователю должен отобразиться экран с турнирами по данной игровой дисциплине.

1. Экран с турнирами – на данном экране пользователю должен предоставляться список найденных турниров в зависимости от указанных данных. Каждый турнир должен содержать:
   1. Название игровой дисциплины
   2. Призовой фонд
   3. Взнос для участия
   4. Дата проведения турнира
   5. Время сеанса
   6. Возрастное ограничение

Пользователь должен выбрать 1 турнир, который подходит ему и нажать на кнопку “Зарегистрировать команду” для перехода на экран с регистрацией команды.

На данном экране должна быть предусмотрена фильтрация сеансов следующими способами:

* Фильтрация по дате проведения – должны отображаться сеансы, дата которых находится внутри выбранного диапазона.
* Также должна работать сортировка по следующим параметрам:
* Наибольший призовой фонд — турниры должны отсортировываться по величине призового фонда
* Самый ближайший – турниры должны отсортировываться по дате начала (наиболее ранние в начале)

Получить список турниров можно используя следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/tournaments  **Method:** GET  **Query string (GET parameters):**  - disciplineId (необязательно) | **------------------------- Successful --------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {     “data”: {       “tournaments”: [          {              “id”: 1              “disciplineName”: {                  “id”: 1,                  “name”: “CS:GO”              },              “prizeMoney”: 5000,              “contribution”: 100,              “date”: “2022-11-11”,              “ageLimit”: 14          }       ]     }  }  **---------------------------- No results -----------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {     “data”: {       “tournaments”: []     }  } |

1. Экран регистрации команды – на данном экране пользователь должен видеть информацию о выбранном турнире, а также иметь возможность добавить игроков в команду.

Если пользователь авторизован, то в команде капитаном изначально должен быть указан авторизованный пользователь.

Добавить игрока можно нажав на кнопку “Добавить игрока”. Для каждого игрока необходимо заполнить следующие поля:

* Имя
* Фамилия
* Дата рождения
* Номер телефона
* Игровой никнейм
* Фотография

Также должна быть возможность игрока клиента, но нельзя игрока клиента, если он единственный.

Для подтверждения регистрации необходимо нажать на кнопку “Подтвердить”. После этого пользователь должен быть перенаправлен на страницу с информацией о команде.

Для создания команды используйте следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/teams  **Method:** POST  **Headers**  **- Content-Type:** application/json  **Body:**  {     “disciplineId”: 1,     “title”: “NaVi”,     “logoUrl”: “{host}/uploads/logos/logo.png”,     “players”: [        {           “firstName”: “Петр”,           “lastName”: “Петров”,           “birthDate”: “2003-05-28”,           “phone”: “8-800-555-35-35”,           “nickname”: “XyL[i]gan4eG”,           “photoUrl”: “{host}/uploads/photos/1.jpg”        },        {...}     ]  } | **------------------------- Successful --------------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {     “data”: {       “teamId”: <id нового пользователя>     }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  } |

**Регистрация команды на турнир**

При регистрации на сервер передается id команды:

* teamId - обязательное поле, в таблице должна существовать запись с таким id

Зарегистрировать команду на турнир может только её участник, который является капитаном, в ином случае выводится соответствующая ошибка.

Каждый игрок в команде должен подходить под возрастное ограничение турнира. Если хотя бы один игрок не подходит, выводится ошибка.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/tournaments/{id}/teams  **Method:** POST  **Headers**  **- Content-Type:** application/json  **Body:**  {     “teamId”: 1  } | **------------------------- Successful --------------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {     “data”: {       “teamId”: <id нового пользователя>     }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  }  **------------------------ Forbidden --------------------------**  **Status:** 403  **Content-Type:** application/json  **Body:**  {      "error": {          "code": 403,          "message": "Player is not allowed to register team"      }  }  **----------------- Age limit is not passed ------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {      "error": {          "code": 422,          "message": "Age limit is not passed"       }  } |

**Участники турнира**

Запрос на получение списка участников турнира по его id.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/tournaments/{id}/teams  **Method:** GET | **------------------------- Successful --------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          “teams”: [              {                  “id”: 1,                  “title”: “NaVi”,                  “logoUrl”: “{host}/uploads/logos/logo.png”,                  “players”: [                      {                          “firstName”: “Петр”,                          “lastName”: “Петров”,                          “nickname”: “XyL[i]gan4eG”,                          “photoUrl”: “{host}/uploads/photos/1.jpg”                      },                      {...}                   ]              }          ]      }  }  **---------------------- Validation error -----------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  } |

**Список матчей в турнире**

Получить список матчей можно по id турнира.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/tournaments/{id}/matches  **Method:** GET | Status: 200  Content-Type: application/json  Body:  {  "data": {  "games": [  {  “id”: 1,  “time”: “15:00”,  “teams”: [  {  “id”: 1,  “title”: “NaVi”,  “logoUrl”: “{host}/uploads/logos/1.png”  },  {  “id”: 2,  “title”: “7Б класс”,  “logoUrl”: “{host}/uploads/logos/2.png”  }  ]  },  {  “id”: 2,  “time”: “17:00”,  “teams”: [  {  “id”: 3,  “title”: “PROTeam”,  “logoUrl”: “{host}/uploads/logos/3.png”  },  {  “id”: 4,  “title”: “Пирожки”,  “logoUrl”: “{host}/uploads/logos/4.png”  }  ]  },  ]  }  } |

**Загрузка логотипа команды на сервер**

В тело запроса необходимо передать обязательное поле image, изображение размером до 4Мб, разрешенные форматы: png, jpeg, bmp.

Загруженное изображение должно сохраниться в папке uploads/logos.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/images/logo  **Method:** POST  **Headers**  **- Content-Type:** FormData | **------------------------------ Success -------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          “imageUrl”: “<ссылка на изображение>”      }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  } |

**Загрузка фотографии игрока на сервер**

В тело запроса необходимо передать обязательное поле image, изображение размером до 4Мб, разрешенные форматы: png, jpeg, bmp.

Загруженное изображение должно сохраниться в папке uploads/photos.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/images/photo  **Method:** POST  **Headers**  **- Content-Type:** FormData | **------------------------------ Success -------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          “imageUrl”: “<ссылка на изображение>”      }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  } |

1. Регистрация – неавторизованный пользователь должен иметь возможность зарегистрироваться в системе на странице с регистрацией.  
   Для регистрации пользователя предусмотрен следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/register  **Method:** POST  **Headers**  **- Content-Type:** application/json  **Body:**  {     “firstName”: “Иван”,     “lastName”: “Иванов”,     “birthDate”: “2003-06-25”,     “email”: “ivan\_ivanov@mail.ru”,     “phone”: “8-900-123-45-67”,     “nickname”: “488lalka228”,     “password”: “paSSword”,     “passwordConfirmation”: “paSSword”  } | **------------------------- Successful --------------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {     “data”: {       “userId”: <id нового пользователя>     }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  } |

1. Вход в личный кабинет – неавторизованный пользователь должен иметь возможность войти в систему на странице со входом.  
   Для аутентификации пользователя предусмотрен следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/login  **Method:** POST  **Headers**  **- Content-Type:** application/json  **Body:**  {     “phone”: “+7-900-012-34-56”,     “password”: “paSSword”  } | **------------------------- Successful --------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {     “data”: {       “token”: <сгенерированный token>     }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  }  **----------------------- Unauthorized -------------------------**  **Status:** 401  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 401,       “message”: “Unauthorized”,       “errors”: {          “auth”: “Incorrect login or password”       }     }  } |

1. Должна быть возможность выйти из личного кабинета.
2. Необходимо позаботиться об уведомлении пользователей о каких-либо действиях (ошибки валидации, подтверждения и т.п).

**ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА**

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

**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 обратилась компания «CyberCutlet» - занимающаяся организацией и проведением киберспортивных турниров. 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.

В медиа файлах 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://xxxxxx-m2.wsr.ru/, где xxxxxx - логин 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 – обязательное 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.

В сервисе предусмотрено 3 группы пользователей:

* гость
* игрок
* администратор

Регистрация

Запрос для регистрации нового пользователя в системе. При отправке запроса необходимо передать объект со следующими свойствами:

* firstName – обязательное поле, строка
* lastName – обязательное поле, строка
* birthDate – обязательное поле, дата в формате YYYY-MM-DD
* photoUrl – обязательное поле, валидный URL
* email – обязательное и уникальное поле, строка, email
* phone – обязательное и уникальное поле, строка, номер телефона в формате +7-xxx-xxx-xx-xx
* nickname - обязательное поле и уникальное поле, строка
* password – обязательное поле, строка длиной не менее 8 символов, подтверждён

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/register  **Method:** POST  **Headers**  **- Content-Type:** application/json  **Body:**  {     “firstName”: “Иван”,     “lastName”: “Иванов”,     “birthDate”: “2003-06-25”,     “photoUrl”: “{host}/uploads/photos/1.jpg”     “email”: “ivan\_ivanov@mail.ru”,     “phone”: “8-900-123-45-67”,     “nickname”: “488lalka228”,     “password”: “paSSword”,     “passwordConfirmation”: “paSSword”  } | **------------------------------ Successful -------------------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {     “data”: {       “userId”: <id нового пользователя>     }  }  **--------------------------- Validation error ---------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  } |

**Аутентификация**

Запрос для аутентификации пользователя в системе. При отправке запроса необходимо передать объект с номером телефона и паролем. Если клиент отправил корректные данные, то необходимо вернуть сгенерированный bearer токен, а иначе сообщение об ошибке.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/login  **Method:** POST  **Headers**  **- Content-Type:** application/json  **Body:**  {     “phone”: “+7-900-012-34-56”,     “password”: “paSSword”  } | **---------------------------------- Successful ----------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {     “data”: {       “token”: <сгенерированный token>     }  }  **----------------------------- Validation error ------------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  }  **------------------------------- Unauthorized --------------------------------**  **Status:** 401  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 401,       “message”: “Incorrect login or password”     }  } |

**Создание команды**

При регистрации команды капитан должен передать на сервер название команды, ссылку на изображение логотипа, а также список игроков. Пользователь, регистрирующий команду, автоматически становится её капитаном.

После успешного создания команды в ответе должна вернуться сгенерированный инвайт-код из 16 латинских символов, по которому смогут вступить в команду другие участники.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/teams  **Method:** POST  **Headers**  **- Content-Type:** application/json  - **Authorization:** Bearer Token  **Body:**  {     “title”: “NaVi”,     “logoUrl”: “{host}/uploads/logos/logo.png”  } | **------------------------- Successful --------------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {     “data”: {       “inviteCode”: <сгенерированный инвайт-код>     }  }  **---------------------- Validation error ---------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  }  **---------------------- Unauthorized ------------------------**  **Status:** 401  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 401,       “message”: “Unauthorized”     }  } |

**Вступление в команду по инвайт-коду**

Данный запрос позволяет вступить в команду любому авторизованному пользователю.

В каждой команде есть 4 места (не считая капитана) + 1 запасной участник. После того, как в команде не останется мест, инвайт-код должен быть недоступен, а в ответ должно выводиться соответствующее сообщение.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/teams/invite/{code}  **Method:** POST  **Headers**  **- Content-Type:** application/json  - **Authorization:** Bearer Token | **--------------------------------- Successful -------------------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {     “data”: {       “status”: “success”     }  }  **------------------------------ Validation error --------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Invalid invite code”     }  }  **-------------------------------- Unauthorized ---------------------------**  **Status:** 401  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 401,       “message”: “Unauthorized”     }  }  **---------------------------- Object not found ----------------------------**  **Status:** 404  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 404,       “message”: “Object not found”     }  } |

**Список турниров**

При отправке запроса возможно передать необязательный параметр disciplineId, который должен содержать id игровой дисциплины. Если параметр указан, в списке должны выводиться турниры, относящиеся только к данной игровой дисциплине.

Заказчик хочет добавить возможность фильтрации турниров с возможностью вывода только предстоящих турниров.

Также должна быть возможность сортировки турниров по дате проведения и призовому фонду от большего значения к меньшему и наоборот. По умолчанию сортировка выполняется по id турнира.

У каждого турнира может быть 3 статуса: предстоящий, в процессе проведения, завершенный. В случае, если турнир находится в процессе проведения, необходимо вывести ссылку на трансляцию.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/tournaments  **Method:** GET  **Query string (GET parameters):**  - onlyUpcoming (обязательно, true|false)  - disciplineId (необязательно)  - orderBy (необязательно)  - orderDirection (обязательно при наличии orderBy) | **-------------------------------- Successful --------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {     “data”: {       “tournaments”: [          {              “id”: 1              “discipline”: {                  “id”: 1,                  “name”: “CS:GO”              },              “prizeMoney”: 5000,              “date”: “2023-04-11”,              “ageLimit”: 14,              “maxTeams”: 6,              “availiableSpots”: 0,              “status”: “processing”,              “streamUrl”: “http://twitch.tv/cyber\_cutlet”          },          {              “id”: 2              “discipline”: {                  “id”: 2,                  “name”: “Dota 2”              },              “prizeMoney”: 5              “date”: “2023-04-18”,              “ageLimit”: 6,              “maxTeams”: 10,              “availiableSpots”: 10,              “status”: “upcoming”          },       ]     }  }  **----------------------------------- No results ------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {     “data”: {       “tournaments”: []     }  } |

**Регистрация команды на турнир**

При регистрации на сервер передается id команды:

teamId - обязательное поле, в таблице должна существовать запись с таким id

Зарегистрировать команду на турнир может только её участник, который является капитаном, в ином случае выводится соответствующая ошибка.

Каждый игрок в команде должен подходить под возрастное ограничение турнира. Если хотя бы один игрок не подходит, выводится ошибка.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/tournaments/{id}/teams  **Method:** POST  **Headers**  **- Content-Type:** application/json  - **Authorization:** Bearer Token  **Body:**  {     “teamId”: 1  } | **-------------------------------------- Successful -----------------------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {     “data”: {       “status”: “success”     }  }  **---------------------------------- Validation error -------------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  }  **-------------------------------------- Forbidden ----------------------------------**  **Status:** 403  **Content-Type:** application/json  **Body:**  {      "error": {          "code": 403,          "message": "Player is not allowed to register team"      }  }  **----------------------------- Age limit is not passed ----------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {      "error": {          "code": 422,          "message": "Age limit is not passed"       }  }  **--------------------------------- Object not found -------------------------------**  **Status:** 404  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 404,       “message”: “Object not found”     }  } |

**Участники турнира**

Запрос на получение списка участников турнира по его id.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/tournaments/{id}/teams  **Method:** GET | **------------------------- Successful --------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          “teams”: [              {                  “id”: 1,                  “title”: “NaVi”,                  “logoUrl”: “{host}/uploads/logos/logo.png”,                  “players”: [                      {                          “firstName”: “Петр”,                          “lastName”: “Петров”,                          “nickname”: “XyL[i]gan4eG”,                          “photoUrl”: “{host}/uploads/photos/1.jpg”                      },                      {...}                   ]              }          ]      }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  } |

**Список матчей в турнире**

Получить список матчей можно по id турнира.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/tournaments/{id}/matches  **Method:** GET | **-------------------------------------- Successful -----------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          “teams”: [              {                  “id”: 1,                  “title”: “Outsiders”,                  “logoUrl”: “{host}/uploads/logos/logo.png”,                  “players”: [                      {                          “firstName”: “Петр”,                          “lastName”: “Петров”,                          “nickname”: “XyL[i]gan4eG”,                          “photoUrl”: “{host}/uploads/photos/1.jpg”                      },                      {...}                   ]              }          ]      }  }  **----------------------------------- Validation error ------------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  }  **---------------------------------- Object not found -----------------------------**  **Status:** 404  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 404,       “message”: “Object not found”     }  } |

**Список матчей в турнире**

Получить список матчей можно по id турнира.

Данный запрос доступен только если была проведена жеребьевка, в ином случае выводится соответствующая ошибка.

Список матчей должен быть отсортирован по времени проведения.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/tournaments/{id}/games  **Method:** GET | **---------------------------- Successful -----------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          "games": [              {                  “id”: 1,                  “time”: “15:00”,                  “teams”: [                      {                          “id”: 1,                          “title”: “Outsiders”,                          “logoUrl”: “{host}/uploads/logos/1.png”                      },                      {                          “id”: 2,                          “title”: “7Б класс”,                          “logoUrl”: “{host}/uploads/logos/2.png”                      }                  ]              },              {                  “id”: 2,                  “time”: “17:00”,                  “teams”: [                      {                          “id”: 3,                          “title”: “PROTeam”,                          “logoUrl”: “{host}/uploads/logos/3.png”                      },                      {                          “id”: 4,                          “title”: “Пирожки”,                          “logoUrl”: “{host}/uploads/logos/4.png”                      }                  ]              },          ]      }  }  **----------------------------------- Validation error ------------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Tournament grid is not created”     }  }  **------------------------ Object not found -------------------------**  **Status:** 404  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 404,       “message”: “Object not found”     }  } |

**Результаты матча**

Запрос на получение результатов матча по его id.

Данный запрос должен быть доступен только после завершения матча.

В ответе необходимо вернуть id победителя.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/games/{id}  **Method:** GET | **-------------------------------------- Successful -----------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          “winnerId”: <id команды, победившей в матче>      }  }  **----------------------------------- Validation error ------------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Game is not over”     }  }  **---------------------------------- Object not found -----------------------------**  **Status:** 404  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 404,       “message”: “Object not found”     }  } |

**Результаты турнира**

Запрос на получение результатов турнира по его id.

Данный запрос должен быть доступен только после завершения турнира.

В ответе необходимо вернуть команды, занявшие три призовых места. У каждой команды нужно вернуть место, приз.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/games/{id}  **Method:** GET | **-------------------------------------- Successful -----------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          “teams”: [              {                  “id”: 2,                  “title”: “7Б класс”,                  “logoUrl”: “{host}/uploads/logos/logo2.png”,                  “place”: 1,                  “prize”: 2500              },              {                  “id”: 2,                  “title”: “Пирожки”,                  “logoUrl”: “{host}/uploads/logos/logo4.png”,                  “place”: 2,                  “prize”: 1500              },              {                  “id”: 1,                  “title”: “Outsiders”,                  “logoUrl”: “{host}/uploads/logos/logo.png”,                  “place”: 3,                  “prize”: 1000              }          ]      }  }  **----------------------------------- Validation error ------------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Tournament is not over”     }  }  **---------------------------------- Object not found -----------------------------**  **Status:** 404  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 404,       “message”: “Object not found”     }  } |

**Генерация турнирной сетки**

Данный запрос доступен только для турниров, на которых не осталось свободных мест для регистрации.

Первая генерация должна в случайном порядке распределить команды между собой по матчам. Время начала каждого матча должно быть на 2 часа позже предыдущего. Время первого матча в сетке по умолчанию устанавливается на 15:00.

Последующая генерация сетки так же случайно распределяет победителей предыдущих матчей в новый этап.

Новая генерация недоступна, если не все матчи из предыдущей сетки были завершены, и если турнир был завершен. В этих случаях должны выводиться соответствующие ошибки.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/tournaments/{id}/grid  **Method:** POST  **Headers**  **- Content-Type:** application/json  - **Authorization:** Bearer Token | **-------------------------------------- Successful -----------------------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {     “data”: {       “status”: “success”     }  }  **---------------------------------- Validation error -------------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  }  **-------------------------------------- Forbidden ----------------------------------**  **Status:** 403  **Content-Type:** application/json  **Body:**  {      "error": {          "code": 403,          "message": "Player is not allowed to register team"      }  }  **----------------------------- Age limit is not passed ----------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {      "error": {          "code": 422,          "message": "Age limit is not passed"       }  }  **--------------------------------- Object not found -------------------------------**  **Status:** 404  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 404,       “message”: “Object not found”     }  } |

**Смена статуса турнира**

Данный запрос доступен только для турниров, в которых нет предстоящих матчей.

При смене статуса на “completed” (завершен), необходимо передать массив с призерами турнира от первого места к третьему. При этом нужно распределить между победителем и призерами призовый фонд. 1-ое место получает 50%, 2-ое - 30%, 3-е - 20% от суммы призового фонда.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/tournaments/{id}/status  **Method:** PATCH  **Headers**  **- Content-Type:** application/json  - **Authorization:** Bearer Token  **Body:**  {     “status”: “completed”,     “places”: [1, 4, 2]  } | **-------------------------------------- Successful -----------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {     “data”: {       “status”: “success”     }  }  **---------------------------------- Validation error -------------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  }  **-------------------------------------- Forbidden ----------------------------------**  **Status:** 403  **Content-Type:** application/json  **Body:**  {      "error": {          "code": 403,          "message": "Player is not allowed to register team"      }  }  **----------------------------- Age limit is not passed ----------------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {      "error": {          "code": 422,          "message": "Age limit is not passed"       }  }  **--------------------------------- Object not found -------------------------------**  **Status:** 404  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 404,       “message”: “Object not found”     }  } |

**Загрузка логотипа команды на сервер**

В тело запроса необходимо передать обязательное поле image, изображение размером до 4Мб, разрешенные форматы: png, jpeg, bmp.

Загруженное изображение должно сохраниться в папке uploads/logos.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/images/logo  **Method:** POST  **Headers**  **- Content-Type:** FormData | **------------------------------ Success -------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          “imageUrl”: “<ссылка на изображение>”      }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  } |

**Загрузка фотографии игрока на сервер**

В тело запроса необходимо передать обязательное поле image, изображение размером до 4Мб, разрешенные форматы: png, jpeg, bmp.

Загруженное изображение должно сохраниться в папке uploads/photos.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/images/photo  **Method:** POST  **Headers**  **- Content-Type:** FormData | **------------------------------ Success -------------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          “imageUrl”: “<ссылка на изображение>”      }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {     “error”: {       “code”: 422,       “message”: “Validation error”,       “errors”: {          <key>: <массив ошибок>       }     }  } |

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://xxxxxx-m2.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 медиа файлах 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

**Задания:** перед вами стоит задача разработать сайт-визитку для презентации информационной системы «Счастливая семья» в глобальной сети Интернет.

Информационная система «Счастливая семья» предоставляет родителям возможность получения бесплатной психолого-педагогической, методической и консультационной помощи в рамках федерального проекта «Современная школа» национального проекта «Образование».

Ваша цель – разработать сайт в стиле landing page, позволяющий привлечь внимание к проекту и познакомить пользователей с его возможностями. Вам необходимо продемонстрировать навыки работы с CMS, проектирования структуры сайта и страниц согласно заданию, наполнения сайта контентом и его оформления.

**ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ**

Ваша задача – разработать рекламный сайт портала «Счастливая семья» с указанной структурой и функционалом. В ходе разработки дизайна сайта необходимо учитывать особенности целевой аудитории. Текстовая и графическая информация для наполнения разделов сайта формируется вами самостоятельно.

**Структура главной страницы**

Главная страница должна содержать следующие секции/блоки:

* *«Шапка сайта»*, содержащая логотип, наименование сайта, навигационное меню, форму поиска по сайту.

Меню, позволяет перемещатьсяпо секциям на странице или переходить на отдельные страницы и содержит следующие пункты:

* О проекте (секция);
* Новости (страница);
* Личный кабинет (секция);
* Стать клиентом (страница);
* Отзывы (секция);
* Контакты (страница).

- Слайдер, содержащий не менее пяти слайдов и созданный с использованием стороннего плагина. Изображение для каждого слайда обрабатывается заранее: на изображении размещается подходящий по смыслу мотивирующий рекламный текст.

- О проекте. В секции размещаются:

* краткая текстовая информация о проекте, изображение;
* три карточки с описанием основных возможностей портала, содержащие изображения и текстовое описание;
* кнопка «Стать клиентом» (обеспечивает переход на отдельную страницу).
* Возможности для вас. Секция содержит перечисление функциональных возможностей системы для разных категорий пользователей: Администратор, Консультант, Родитель.
* Партнеры проекта. Секция содержит карусель с пятью видимыми изображениями и кнопками управления каруселью.
* Личный кабинет. Секция содержит текстовую и графическую информацию, иллюстрирующую особенности личных кабинетов для различных категорий пользователей, а также кнопку «Подробнее» для перехода на страницу «Личный кабинет».
* Наш проект в цифрах. Секция содержит три блока, в каждом из которых указан цифровой показатель и текстовый комментарий. При прокрутке страницы до момента видимости данной секции происходит анимированное увеличение цифровых показателей в каждом из блоков, сопровождающееся появлением текстовых комментариев. Способ и порядок анимации определяется вами самостоятельно.
* Запись на консультацию. Процесс записи пользователя на консультацию специалиста в системе осуществляется в несколько шагов:

Шаг 1 - Внесение данных о пользователе

Шаг 2 - Подбор консультации

Шаг 3 - Назначение консультации

В данной секции вам необходимо наглядно представить последовательность шагов подачи заявки на консультацию.

* Отзывы. Секция содержит информацию о последних отзывах (не менее трех) и кнопку «Все отзывы» (обеспечивает переход на отдельную страницу отзывов). Блок отзыва содержит: фото, имя пользователя, текст отзыва, дату отзыва, оценку пользователя (указанное количество звезд).
* Наши клиенты на карте. В секции расположено изображение-заглушка. При щелчке в области изображения карты происходит переход на отдельную страницу с центрами, использующими продукт.
* Подвал сайта содержит:
* логотип;
* наименование сайта;
* навигационное меню (дублирует верхнее меню сайта);
* панель ссылок на социальные сети (VK, OK, Telegram), созданное с использованием стороннего плагина;
* номер телефона;
* политика конфиденциальности;
* копирайт.

Переход к отдельным секциям по нажатию кнопок навигационных меню (верхнего и нижнего) происходит с эффектом плавного прокручивания страницы.

Все кнопки, расположенные на страницах, а также иные кликабельные элементы снабжены эффектами наведения. Кнопка «Стать клиентом» в секции «О проекте» дополнительно выделена на странице с помощью анимации.

Кроме главной страницы сайт содержит шесть отдельных страниц. Ниже расположено описание содержания данных страниц.

**Страница «Новости»**

На странице сверху вниз расположены блоки с новостями. Блок с новостью содержит: изображение, заголовок, текст новости, дату публикации, имя автора. На странице выводятся пять последних новостей в порядке новизны. Страница снабжена пагинацией.

**Страница «Личный кабинет»**

Страница содержит текстовую и графическую информацию о структуре, внешнем виде и преимуществах личных кабинетов для различных категорий пользователей: Администратор, Родитель, Консультант.

**Страница «Наши клиенты»**

На странице сверху вниз расположены в две колонки расположены наименования и изображения районов, названия консультационных центров, расположенных в них.

**Страница «Отзывы»**

На странице выводятся все отзывы пользователей, в порядке убывания их оценок.

**Страница «Стать клиентом»**

На странице представлен текст приветствия и форма подачи предварительной заявки на использование продукта:

* наименование организаци;
* ФИО контактного лица;
* телефон;
* e-mail;
* кнопка «Отправить».

При полном заполнении формы и нажатии на кнопку «Отправить заявку», на пять секунд появляется диалоговое окно с надписью «Ваша заявка принята. Спасибо за ваше обращение!». Появление окна сопровождается простой анимацией. Закрытие окна происходит автоматически через пять секунд или при щелчке мышью вне основного содержимого окна.

Управление заявками пользователей должно осуществляться из панели администратора.

**Страница «Контакты»**

На странице расположены два блока данных:

* наименование, адрес и контактные данные компании-разработчика;
* фото, ФИО, должность и контактные данные ведущих сотрудников компании-разработчика.

На всех вложенных страницах щелчок мышью на логотипе в шапке сайта происходит переход на главную страницу сайта.

Формирование контента и дизайна сайта необходимо выполнять с учетом целевой аудитории. Выбранные цветовые и стилистические решения сайта должны быть ориентированы на привлечение большего числа клиентов. Отдельное внимание необходимо уделять обработке изображений, используемых на страницах сайта.

На главной странице необходимо предусмотреть вывод блока с кнопкой «Стать клиентом». Кнопка размещается в левой нижней части окна браузера и фиксируется там при прокрутке страницы. Кнопка позволяет перейти на страницу «Стать клиентом». На всех страницах, кроме главной, кнопка не отображается. Вставка блока с кнопкой производится с помощью собственного плагина с именем **BecomeClient**.

Для расширения функционала сайта вам необходимо разработать плагин **AddReview**, позволяющий выводить на странице форму добавления отзыва. Работа плагина осуществляется с использованием шорткода. При формировании отзыва пользователь вносит следующие данные: фото, имя пользователя, текст отзыва, дату отзыва, оценку пользователя (выбор количества звезд). При добавлении отзыва поля формы должны проходить проверку на валидность: заполненность полей, типы и формат данных, формат файла фото - png, jpg. При включении плагина в режиме администрирования сайта появляется панель управления отзывами с возможностью их удаления.

**ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА**

HTML и CSS-код должны быть удовлетворять стандартам W3C (HTML 5 и CSS 3).

Разработанный веб-сайт должен быть доступен по адресу http://xxxxxx-m1.wsr.ru/, где xxxxxx - логин участника (указан на индивидуальной карточке).

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.

В папке media расположены: установочный пакет CMS, набор необходимых плагинов, изображения и тексты для размещения на страницах сайта, «нулевой» шаблон для CMS.

Доступ к административной части сайта должен осуществляться с помощью следующих данных: логин: **admin,** пароль: **admin.**

## 

## 2. СПЕЦИАЛЬНЫЕ ПРАВИЛА КОМПЕТЕНЦИИ*[[2]](#footnote-2)*

### 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 C-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.

1. *Указывается суммарное время на выполнение всех модулей КЗ одним конкурсантом.* [↑](#footnote-ref-1)
2. *Указываются особенности компетенции, которые относятся ко всем возрастным категориям и чемпионатным линейкам без исключения.* [↑](#footnote-ref-2)