**ТИПОВОЕ КОНКУРСНОЕ ЗАДАНИЕ**

***ДЛЯ РЕГИОНАЛЬНЫХ ЧЕМПИОНАТОВ***

***чемпионатного цикла 2021/2022***

**компетенции**

«ВЕБ-ТЕХНОЛОГИИ»

**для основной возрастной категории**

**16-22 года**

*Конкурсное задание включает в себя следующие разделы:*

[1.](#_heading=h.gjdgxs) Форма участия в конкурсе: 2

[2.](#_heading=h.30j0zll) Общее время на выполнение задания: 2

[3.](#_heading=h.1fob9te) Задание для конкурса 2

[4.](#_heading=h.3znysh7) Модули задания и необходимое время 2

[5.](#_heading=h.tyjcwt) Критерии оценки. 3

[6.](#_heading=h.3dy6vkm) Приложения к заданию. 4



1. **Форма участия в конкурсе**: Индивидуальный конкурс.
2. **Общее время на выполнение задания:** 15 ч.
3. **Задание для конкурса**

Конкурсное задание содержит 4 модулей: Дизайн и верстка веб-приложения; Серверное программирование REST API; Клиентское программирование Single Page Application; CMS WordPress.

В Конкурсном задании могут использоваться различные PHP и JavaScript фреймворки которые предусмотрены для каждого модуля. Также могут быть включены дополнительные библиотеки, которые могут быть включены в модули для реализации отдельного функционала.

Продолжительность Конкурсного задания должна быть 15 часов. Возрастной ценз участников для выполнения Конкурсного задания от 16 до 22 лет. Вне зависимости от количества модулей, Конкурсного задания включает оценку по каждому из разделов WSSS и не должно выходить за пределы WSSS. Оценка знаний участника проводится исключительно через практическое выполнение Конкурсного задания. При выполнении Конкурсного задания не оценивается знание правил и норм WSR.

Оценка производится в отношении работы модулей.

Если участник конкурса не выполняет требования техники безопасности, подвергает опасности себя или других конкурсантов, такой участник может быть отстранен от конкурса.

Детали конкурсного задания в зависимости от конкурсных условий могут быть изменены членами жюри.

Конкурсное задание должно выполняться помодульно. Оценка также происходит от модуля к модулю.

1. **Модули задания и необходимое время**

Таблица 1.

| **Наименование модуля** | **Соревновательный день (С1, С2, С3)** | **Время на задание** |
| --- | --- | --- |
| **A** | Дизайн и верстка веб-приложения | С1 | 6 часа |
| **B** | Клиентское программирование Single Page Application | С2 | 3 часа |
| **C** | Серверное программирование REST API | С2 | 3 часа |
| n | CMS WordPress  | С3 | 3 часа |

**Модули с описанием работ**

**Модуль A. Дизайн и верстка веб-приложения**

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

**Время на выполнение:** 6 часа

Вам необходимо использовать все имеющиеся навыки в дизайне и верстке чтобы создать Landing Page, а также все остальные страницы.

Ваша задача – создать следующие страницы веб-сайта:

* Главная страница - Landing Page
* Страница входа в личный кабинет
* Страница регистрации в личном кабинете
* Страница личного кабинета
* Страница с результатами поиска
* Страница бронирования
* Страница управления бронированием
* Страница выбора мест в салоне воздушного судна

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

Сохраните дизайны и верстку страниц со следующими именами:

* Главная страница - Landing Page – index.png, index.html
* Страница входа в личный кабинет – login.png, login.html
* Страница регистрации в личном кабинете – register.png, register.html
* Страница личного кабинета – profile.png, profile.html
* Страница с результатами поиска – search.png, search.html
* Страница бронирования – booking.png, booking.html
* Страница управления бронированием – booking\_management.png, booking\_management.html
* Страница выбора мест в салоне воздушного судна – seat.png, seat.html

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

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

Оценка будет производиться в браузере Google Chrome.

Если вы плохо знакомы с графическими редакторами, то вы можете сверстать страницы и отобразить свои знания дизайна в верстке, но в итоге страницы должны быть сохранены в png формате. Знания верстки оцениваться не будут.

**Для работы вам будут доступны следующие фреймворки: Bootstrap 5, TailwindCSS**

**Модуль B. Клиентское программирование Single Page Application**

**Технологии этого модуля:** Клиентское программирование

**Время на выполнение**: 3 часа

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.

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

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://server-m3.wsr.ru
Ваше SPA должно состоять из следующих экранов:

* Главная
* Вход в личный кабинет
* Регистрация в личном кабинете
* Личный кабинет
* Результаты поиска
* Бронирование
* Управление бронированием
* Выбор места в салоне воздушного судна

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

**Модуль C. Серверное программирование REST API**

**Технологии этого модуля:** REST API

**Время на выполнение**: 3 часа

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 – 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.

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.

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.

**Модуль D. CMS WordPress**

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

**Время на выполнение**: 3 часа

Вам предоставляют готовый HTML-шаблон, который нужно немного модернизировать и разработать новый сайт с использованием системы управления контентом Wordpress.

**Плагины**

Вам необходимо разработать два плагина.

Random Line
Плагин, который при загрузке любой страницы на сайте выводит случайную строку из поста. Строка должна выводиться в случайной части страницы во время прокрутки поверх всех элементов, но не перекрывая контент. Появление должно сопровождаться анимацией и привлекать внимание, но не сильно отвлекать. Пост-источник должен указываться в настройках данного плагина в панели управления. Данный функционал планируется использовать для демонстрации малоизвестных фактах об экспонатах музея и объектов связи.

QRPage
Плагин, который реализует шорткод [qrpage]. Шорткод должен выводить изображение с QR-кодом, который содержит ссылку на текущую страницу. QR-код должен генерироваться с помощью предоставленной JavaScript библиотеки QRCode.js.

Разработанный веб-сайт должен быть доступен по адресу http://xxxxxx-m2.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.

1. **Критерии оценки.**

Таблица 2.

| **Критерий** | **Баллы** |
| --- | --- |
| **Судейские аспекты** | **Объективная оценка** | **Общая оценка** |
| **A** | **Дизайн и верстка веб приложения** | **19** | **20** | **39** |
| **B** | **Программирование на стороне клиента** | **2** | **19** | **21** |
| **C** | **Программирование на стороне сервера** | **1** | **19** | **20** |
| **D** | **Системы управления контентом** | **7** | **13** | **20** |
| **Итого** | **29** | **71** | **100** |