

КОНКУРСНОЕ ЗАДАНИЕ КОМПЕТЕНЦИИ

«Веб-технологии»

Региональный этап Чемпионата по профессиональному мастерству «Профессионалы» Курганской области в 2025 г

2025 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.

**Модуль А. Разработка интерфейса пользователя**

*Время на выполнение модуля:* 4 часа

**Задания:**

2024 год объявлен в России Годом семьи. В связи с этим в городе Кургане решили организовать празднование Года семьи в виде городского фестиваля «Молодые семьи молодого Кургана».

С целью привлечения на семейный праздник жителей города Кургана, а также жителей Курганской области (семьи с детьми) возникла необходимость разработать сайт, посвященный фестивалю «Молодые семьи молодого Кургана».

Вам необходимо разработать дизайн и сверстать страницы веб-сайта, который будет привлекать потенциальных посетителей и мотивировать их зарегистрироваться на праздник Года семьи. Ниже приведены основные требования к внешнему виду страниц.

Потенциальные пользователи сайта предпочитают использовать мобильные устройства для доступа к интернету и поиска информации. Поэтому страницы должны корректно отображаться как на мобильных устройствах, так и на десктопных - в том числе на странице регистрации, просмотра расписания конкурсов и т.д.

В дизайне сверстанных страниц следует использовать и применять элементы, концепции и приемы, позволяющие привлечь внимание, создать положительный образ ресурса, реализовать современные концепции дизайна веб-страниц. Заказчик просит, чтобы на сайте использовался цвет, указанный в файле media/img/UIkit.jpg.

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 или созданные медиаданные. Текстовый и графический контент должны быть согласованы между собой.

Используйте анимацию, чтобы пользователи могли понять, что элементы являются интерактивным. Используемая анимация должна быть гармоничной, чтобы создать приятный визуальный эффект и не отвлекать пользователей.

**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 4 коммитов с осмысленными комментариями!** |

## ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – разработать дизайн и сверстать следующие страницы веб-сайта:

1. Главная - index.html, index.png (темная тема); index-light.html, index-light.png (светлая тема)
2. Страница регистрации на сайте – registration.html, registration.png
3. Страница с расписанием конкурсов – schedule.html, schedule.png
4. Личный кабинет – account.html, account.png
5. Страница с описанием конкурса – competition.html, competition.png
6. Страница не найдена - 404.html, 404.png

**Внимание!**

Все страницы должны быть сверстаны в темной теме. Однако для главной страницы также должен быть разработан дизайн и сверстана версия со светлой темой.

Сверстанный веб-сайт должен быть доступен по адресу [http://xxxxxx-m1.wsr.ru](http://xxxxxx-m1.wsr.ru/), адрес указан в личном кабинете конкурсанта соревнований.

Страницы должны быть адаптивны. При переходе от десктопной версии главной страницы (ширина экрана 1920px) к мобильной версии (ширина экрана 796px) должно быть реализовано не менее четырех структурных изменений контента (например, скрываются/добавляются, увеличиваются/уменьшаются элементы и др.).

На остальных разрешениях контент должен отображаться корректно.

## Главная - Landing Page

Главная страница, так же, как и остальные страницы, разрабатывается в темной теме. Однако вам необходимо также создать дизайн-макет и сверстать вариант главной страницы в светлой теме. Имена соответствующих html и png файлов указаны в задании.

Главная страница должна содержать следующие блоки (вы также можете добавить дополнительные блоки/секции, если считаете, что это усилит работу) – порядок блоков/секций может быть произвольным:

1. Шапка сайта.
   1. Логотип.

Вам необходимо создать логотип, с учетом следующих пожеланий заказчика:

Логотип должен быть уникальным и легко узнаваемым. Он должен быть достаточно простым, чтобы визуально ассоциироваться с тематикой сайта. Логотип должен быть согласован с общей концепцией брендинга сайта "Молодые семьи молодого Кургана" и выполнен в высоком качестве. Логотип должен быть представлен на сайте в виде векторного или растрового графического файла.

* 1. Меню навигации
     1. Регистрация – ссылка на отдельную страницу
     2. Авторизация – ссылка к секции/блоку на странице
     3. Личный кабинет – ссылка на отдельную страницу
     4. О нас – ссылка к секции/блоку на странице
     5. Конкурсы - ссылка к секции/блоку на странице
     6. Поиск - ссылка к секции/блоку на странице
     7. Контакты - ссылка к секции/блоку на странице – футеру/подвалу
     8. Элемент переключения между темной и светлой темами.

1. Вывод наиболее популярных конкурсов предстоящего мероприятия в виде слайдера с индикацией активного слайда (работа слайдера не проверяется, только дизайн и верстка!). Слайд должен содержать следующую информацию:
   1. Изображение.
   2. Краткое описание.
   3. Элементы, позволяющие пользователю перейти к следующему слайду.
   4. Рейтинг конкурса (с обязательным использованием графики)
   5. Кнопка “Подробнее” для перехода на страницу с описанием конкурса.
2. Секция/блок для авторизации

Форма входа должна предоставлять поле для ввода адреса электронной почты и пароля для аутентификации, кнопка/ссылка для входа, а также кнопка/ссылка для восстановления забытого пароля.

1. Секция/блок «О нас»

Описание фестиваля – предоставлено в media/text/about.txt – необходимо использовать всю предоставленную текстовую информацию! Также в этой секции необходимо разместить хотя бы 1 изображение.

1. Секция/блок с приветствием от организаторов

В секции представлен видеоролик с приветствием посетителей мероприятия (media/video/hello.mp4)

1. Секция/блок «Конкурсы»

Перечень с карточками конкурсов (названия предоставлены в файле media/text/competitions.txt). В каждой карточке отображается название конкурса, фото, описание, количество записей на конкурс, рейтинг, кнопка/ссылка, позволяющая перейти к записи на конкурс. Общее количество карточек с конкурсами – 5. Карточки должны быть представлены в табличной форме с тремя столбцами. В мобильной версии страницы вид перечня карточек - на ваше усмотрение.

1. Поле быстрого поиска конкурсов по названию или описанию конкурса, а также кнопка «Поиск». Поле поиска содержит подсказки.
2. Секция/блок «Отзывы»

Представлены отзывы клиентов (media/text/feedback.txt), ранее участвовавших в схожих мероприятиях – в виде слайдсета (должны отображаться фрагменты других слайдов). Реализация JS не требуется, но визуально должно быть понятно, что подразумевается поочередная смена отзывов.

* 1. Имя автора отзыва
  2. Фотография автора
  3. Ссылка/кнопка на страницу с описанием конкурса
  4. Текст отзыва
  5. Дата отзыва

Порядок полей может быть любым.

1. Форма для подписки на новости мероприятия

Дизайн этой формы вам уже предоставлен - вам необходимо сверстать его максимально соответствующим предоставленному дизайн-макету (media/design/form.png) – с сохранением элементов оформления, стилей, размеров, цветов, используемых шрифтов и т.д. Для упрощения работы некоторые инструкции представлены в файле media/design/form-description.txt.

Обязательные поля формы:

* 1. Email
  2. Элемент, подтверждающий согласие на обработку персональных данных
  3. Кнопка/ссылка для подписки

1. Подвал сайта
   1. Телефон “+7 (3522) 25-45-48” (кликабельная ссылка)
   2. e-mail: [family@kurgan.ru](mailto:family@kurgan.ru) (кликабельная ссылка)
   3. Навигация по сайту:
      1. Главная
      2. Главная-светлая
      3. Регистрация
      4. Конкурс
      5. Личный кабинет
      6. Страница не найдена

Порядок и вид секций/блоков может быть произвольный (кроме вида формы подписки на новости).

## (2) Страница регистрации на сайте

На этой странице необходимо представить форму со следующими полями, обязательными для заполнения (поля должны иметь соответствующий тип):

* e-mail
* Имя
* Фамилия
* Возраст – с возможностью указания значения не менее, чем двумя способами
* Пол – элемент с возможностью выбора одного из двух значений – «мужской» или «женский»
* Пароль
* Повтор пароля
* Оформленная область для загрузки семейной фотографии
* Элемент, подтверждающий согласие на обработку персональных данных
* Кнопка/ссылка для регистрации

Все поля обязательны для заполнения. Отобразите варианты сообщений, если пользователь указал корректные данные, некорректные данные, не заполнил обязательные поля.

В этой форме также необходимо предусмотреть возможность добавления и удаления группы полей для каждого из членов семьи:

* Фамилия,
* Имя,
* Возраст,
* Пол – выпадающий список

Само добавление полей реализовывать не нужно, но соответствующие элементы должны отображаться.

**(3) Страница с расписанием конкурсов**

На странице представлена информация с расписанием конкурсов в табличной форме (технология создания табличной формы – на ваше усмотрение):

Столбцы таблицы – временные слоты с 9 до 21 часа включительно с шагом в 1 час (например, с 8 до 9 часов, с 9 до 10 часов и т.д.)

Строки таблицы – названия конкурсов (предоставлены в файле media/text/competitions.txt), необходимо использовать все названия конкурсов.

Требования к таблице:

* Для каждого конкурса и соответствующего временного слота в таблице должно быть указано количество свободных мест,
* Ячейки таблицы с указанием свободных мест – квадратной формы (ширина и высота одинаковые),
* В таблице визуально должны быть видны слоты, на которые регистрация завершена (нет доступных мест),
* Страница должна корректно отображаться как на мобильных устройствах, так и на десктопных.

## (4) Личный кабинет

В меню должна быть представлена кнопка/ссылка для выхода из профиля, а ссылка «Регистрация» должна отсутствовать.

На странице представлена должна быть представлена форма, с помощью которой пользователь мог бы записаться на конкурсы с указанием соответствующей информации:

* Конкурс – элемент, позволяющий выбрать один конкурс из списка/перечня
* Дата – с возможностью указания/выбора даты через специализированный тип элемента/поля. Доступна только одна дата – 8 июля 2024г.
* Время – доступный интервал – с 9 до 21 часов
* Количество человек, которые будут участвовать
* Кнопка/ссылка для записи на конкурс

**(5) Страница с описанием конкурса**

На этой странице пользователи могут ознакомиться с названием конкурса, его описанием, соответствующим изображением. В качестве контента для страницы вы можете использовать предоставленные медиаданные по любому конкурсу.

Также на странице присутствуют графические элементы, отображающие рейтинг конкурса, количество посетителей, которые записались на конкурс.

При печати страницы средствами браузера, например, по сочетанию клавиш Ctrl+P, остается только информация: название конкурса, фотография в черно-белом исполнении (градации серого), появляется блок с произвольным отзывом к этому конкурсу, без шапки и подвала. Цветные элементы отсутствуют.

**(6) Страница не найдена – 404.html**

Отображается сообщение, что встреча не найдена.

Оформление страницы соответствует дизайн-решениям других страниц.

**ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА**

Ваш HTML-код должен быть валидным. Используйте лучшие практики верстки в соответствии с заданием.

Страницы должны быть согласованы между собой. Ссылки максимально должны быть кликабельными с навигацией по страницам.

Вам нужно использовать не менее одного из представленных шрифтов (media/fonts/). Рекомендуется выбирать шрифты, которые отражают стиль сайта.

Вы можете использовать и изменять предоставленные медиаданные для реализации поставленных задач, а также создавать новые.

Оценка будет производиться в браузере Google Chrome.

Сверстанный веб-сайт должен быть доступен по адресу [http://xxxxxx-m1.wsr.ru](http://xxxxxx-m1.wsr.ru/). Адрес указан в личном кабинете конкурсанта соревнований.

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

1. Главная - index.html, index.png (темная тема); index-light.html, index-light.png (светлая тема)
2. Страница регистрации на сайте – registration.html, registration.png
3. Страница с расписанием конкурсов – schedule.html, schedule.png
4. Личный кабинет – account.html, account.png
5. Страница с описанием конкурса – competition.html, competition.png
6. Страница не найдена - 404.html, 404.png

Макеты должны быть расположены в отдельной папке (/creative).

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

Процесс работы нужно организовать с использованием системы контроля версий. Не забудьте создать репозиторий и фиксировать в нем изменения по мере выполнения работы (не менее 4 коммитов, описание коммитов отражает вносимые изменения).

**Проверяются 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!**

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

*Время на выполнение модуля: 2 часа*

Вам необходимо разработать Single Page Application (далее SPA) для управления кулинарными курсами с возможностью разграничения прав. Заказчик предоставляет вам полностью готовую верстку со всеми страницами и рабочее API.

Общее

Адрес API: http://jjxhzny-m3.web.ru/api-culinar

Ваша задача – реализовать SPA приложение, которое будет взаимодействовать с уже разработанным API.

ВНИМАНИЕ! Проверке подлежат приложения в формате Single Page Application!

В документации будет использоваться переменная {{host}} которая обозначает адрес API: {{host}}-m3.wsr.ru.

Ваше SPA должно состоять из следующих экранов:

· Регистрация;

· Авторизация;

· Список курсов, мастер-классов, индивидуальных занятий;

· Скачать файл с домашнем заданием;

· Загрузить домашнее задание;

· Создание курсов, мастер-классов, индивидуальных занятий;

· Изменение курсов, мастер-классов, индивидуальных занятий;

· Удаление курсов, мастер-классов, индивидуальных занятий;

· Список пользователей;

· Поиск по курсам, мастер-классам, индивидуальным занятиям.

· Просмотр действий (логов).

· Смена ролей.

У неавторизированного пользователя присутствуют ссылки на экраны

· Регистрация;

· Авторизация;

У авторизированного пользователя присутствуют ссылки на экраны

· Список доступных курсов, мастер-классов, индивидуальных занятий;

· Скачать файл с домашнем заданием;

· Поиск по курсам, мастер-классам, индивидуальным занятиям.

Также у авторизированного пользователя есть ссылка на выход из аккаунта.

У администратора присутствуют ссылки на экраны:

· Создание курсов, мастер-классов, индивидуальных занятий;

· Изменение курсов, мастер-классов, индивидуальных занятий;

· Удаление курсов, мастер-классов, индивидуальных занятий;

· Список пользователей;

· Список курсов, мастер-классов, индивидуальных занятий;

· Просмотр действий (логов).

· Смена ролей.

ВНИМАНИЕ! Вся логика приложения должна быть организована через взаимодействие с API!

Для улучшения пользовательского опыта (UX) все взаимодействия с интерфейсом, а также переходы между экранами приложения должны сопровождаться анимацией, интерактивными сообщениями и т.д.

Функциональные требования

Регистрация

На данном экране представлена форма для регистрации нового пользователя.

При вводе некорректных значений выводится предупреждающее уведомление (отображаются текст ошибки) с красным фоном.

При успешной регистрации происходит переход на экран входа в систему.

Интерактивные элементы экрана

· По щелчку на кнопку «Зарегистрироваться» происходит попытка зарегистрировать нового пользователя.

Вход в систему

На данном экране представлена форма для входа пользователя в систему. При вводе не верной пары логин - пароль выводится предупреждающее уведомление (Не верный логин или пароль!) с красным фоном.

При успешной аутентификации происходит переход на экран «Список доступных курсов».

Интерактивные элементы экрана

· По щелчку на кнопку «Войти» происходит попытка аутентификации пользователя.

Список доступных курсов, мастер-классов, индивидуальных занятий

На данном экране авторизованный пользователь может увидеть список доступных курсов, мастер-классов, индивидуальное занятие. Для каждого курса, мастер-класса, индивидуального занятия отображаются название, тип занятия, описания, дата старта, окончания набора, и максимальное количество участников.

Интерактивные элементы экрана

По щелчку на название происходит переход к конкретному курсу/мастер-классу/индивидуальному занятию;

Переход к конкретному курсу/мастер-классу/индивидуальному занятию

На данном экране авторизованный пользователь может увидеть название, тип занятия, описания, дата старта, окончания набора, и максимальное количество участников и кнопку «скачать файл с домашним заданием».

Интерактивные элементы экрана

По щелчку на кнопку «Скачать файл с домашним заданием» происходит скачивание файла с домашним заданием;

Список курсов, мастер-классов, индивидуальных занятий

На данном экране администратор видит список курсов, мастер-классов и индивидуальных занятий. Для каждого занятия отображается название, описание, дата старта, окончание набор и максимальное количество участников и действия.

Интерактивные элементы экрана

По щелчку на кнопку "Редактировать" происходит перенаправление на экран "Редактирование занятия";

По щелчку на кнопку "Удалить" - курс удаляется, и список обновляется;

По щелчку на кнопку "Создать курс" происходит перенаправление на экран "Создание занятия".

Загрузить домашнее задание

На данном экране администратор может отправить домашнее задание.

Интерактивные элементы экрана

По щелчку на кнопку "Отправить" происходит загрузка файла и его проверка;

По щелчку на кнопку "Отправить" файл отправляется, и пользователь получает уведомление об успешной отправке.

Создание курсов, мастер-классов, индивидуальных занятий

На данном экране только администратор может создать новое занятие.

Интерактивные элементы экрана

По щелчку на кнопку "Сохранить" появляется уведомление об успешном создании курса, и пользователь перенаправляется на экран "Список курсов, мастер-классов, индивидуальных занятий";

При вводе некорректных данных отображаются ошибки и подсвечиваются соответствующие поля.

Изменение курсов, мастер-классов, индивидуальных занятий

На данном экране только администратор может редактировать выбранное занятие.

Интерактивные элементы экрана

По щелчку на кнопку "Сохранить" изменения сохраняются, и пользователь перенаправляется на экран "Список курсов, мастер-классов, индивидуальных занятий";

При вводе некорректных данных отображаются ошибки и подсвечиваются соответствующие поля.

Удаление курсов, мастер-классов, индивидуальных занятий

На данном экране только администратор может удалить выбранное занятие.

Интерактивные элементы экрана

По щелчку на кнопку "Удалить" появляется подтверждение удаления. При подтверждении курс удаляется, и пользователь возвращается на экран "Список курсов, мастер-классов, индивидуальных занятий".

Список пользователей

На данном экране администратор видит список всех пользователей системы. Для каждого пользователя отображаются имя, email и роль.

Интерактивные элементы экрана

По щелчку на иконку "Удаление" пользователь удаляется из системы;

По щелчку на кнопку "Добавить пользователя" происходит перенаправление на экран "Регистрация".

Поиск по курсам, мастер-классам, индивидуальным занятиям

На данном экране пользователь может выполнить поиск по названию курса, мастер-класса или индивидуального занятия.

Интерактивные элементы экрана

По щелчку на кнопку "Найти" отображаются результаты поиска в виде списка курсов, мастер-классов, индивидуальных занятий с возможностью перейти к подробной информации о них.

### Лог действий

Позволяет отслеживать действия пользователей в системе.

Пример действий, которые фиксируются:

* Авторизация пользователя;
* Создание, редактирование и удаление курсов, мастер-классов и индивидуальных занятий;
* Загрузка и отправка файлов;
* Изменение данных пользователя или роли.

Интерактивные элементы экрана

* По щелчку на кнопку "Показать логи" администратор видит список действий с фильтрами по пользователям, датам и типам действий.

### Получение списка ролей

**Назначение**: Отображение доступных ролей в системе для управления доступом пользователей.

**Интерактивные элементы экрана**

* По щелчку на кнопку "Список ролей" администратор видит перечень доступных ролей и их описание.

Описание предоставленного API

Документация API идентична описанной в Модуле В.

Инструкция для конкурсанта

Разработанное приложение должно быть доступно по адресу {{host}}-m3.wsr.ru.

где xxxxxx - логин участника.

Вам предоставляются следующие конфигурации фреймворков и библиотек:

· jQuery 3.x

· jQuery UI 1.x

· VueJS 2.x

· VueJS 3.x

· Vue Router 3.x

· Vue Router 4.x

· Vue CLI

· React

· React Router

· React CLI

· Angular CLI

Вы можете использовать любой из представленных фреймворков. Для оценки качества кода необходимо выгружать на сервер также не скомпилированный вариант проекта в папку source.

**Модуль В. Разработка Веб-приложения на стороне сервера**

*Время на выполнение модуля: 2 часа*

**Введение**

В рамках этого задания вам предстоит разработать API, который предоставляет информацию о курсах, мастер-классов, индивидуальных заданий, связанных с образованием, а также позволяет пользователям загружать, и получать информацию о домашнем задании.

Идентификацию пользователя организуйте посредством Bearer Token.

При попытке доступа гостя к защищенным авторизацией функциям системы во всех запросах необходимо возвращать ответ следующего вида:

**Status:** 403

**Content-Type:** application/json

**Body:**

{

   "message": "Login failed"

}

 При попытке доступа авторизованным пользователем к функциям недоступным для него во всех запросах необходимо возвращать ответ следующего вида:

**Status:** 403

**Content-Type**: application/json

**Body:**

{

   "message": "Forbidden for you"

}

 При попытке получить не существующий ресурс необходимо возвращать ответ следующего вида:

**Status:** 404

**Content-Type:** application/json

**Body:**

{

   "message": "Not found",

   "code": 404

}

 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”: {  
           patronymic: [ “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.

**Регистрация**

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

* first\_name – обязательное поле, строка, первая буква в верхнем регистре
* last\_name – обязательное поле, строка, первая буква в верхнем регистре
* patronymic – обязательное поле, строка, первая буква в верхнем регистре
* email – обязательное и уникальное поле, строка
* password – пароль пользователя, обязательное поле, состоит минимум из 3 символов, из которых как минимум одна строчная, одна прописная и одна цифра
* birth\_date – обязательное поле, строка

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/registration  **Method:** POST    **Headers**  - Content-Type: application/json    **Body:**  {     "first\_name": "Alexey",     "last\_name": "Smirnov"     "patronymic": "Ivanovich",     “email”: "user@prof.ru",     "password": "paSSword1",     "birth\_date": "2001-02-15"  } | Успех  **Status:** 201  **Content-Type:** application/json  **Body:**  {      "data": {          "user": {              "name": "Smirnov Alexey Ivanovich",              "email": "user@prof.ru"          },          "code": 201,          "message": "Пользователь создан"      }  }      Ошибки валидации  **Формат ответа из общих требований** |

**Аутентификация**

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

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/authorization  **Method:** POST    **Headers**  - Content-Type: application/json    **Body:**  {     "email": "user@prof.ru",     "password": "paSSword1"  } | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          "user": {              "id": 1,              "name": "Alexey Ivanovich Smirnov",              "birth\_date": "2001-02-15",              "email": "user@prof.ru"          },          "token": <сгенерированный токен>      }  }  Ошибки валидации  **Формат ответа из общих требований**    Неправильные логин или пароль  **Формат ответа из общих требований** |

**Описание функционала авторизированного пользователя**

**Сброс авторизации:**

Запрос предназначен для очистки значение токена пользователя.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/logout  **Method:** GET | Успех  **Status:** 204  **Content-Type:** application/json    **Body (не отдается)** |

Список доступных курсов, мастер-классов, индивидуальных занятий

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/activities/available  **Method:** GET | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": [  {  "id": 1,  "title": "Основы программирования",  "description": "Курс для начинающих",  "homework\_file": "homework1.pdf"  },  {  "id": 2,  "title": "Мастер-класс по веб-дизайну",  "description": "Создание современных веб-дизайнов",  "homework\_file": "homework2.pdf"  }  ]  } |
|  |  |

Скачать файл с домашним заданием

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/activities/{id}/homework  **Method:** GET    **Headers**  - Content-Type: application/json | Успех  Браузеру отдается файл |

Отправка файла с домашним заданием

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/ activities/{id}/homework/upload  **Method:** POST  **Headers**  - Content-Type: application/json  **FormData:**  “file”: <Файл>, | Успех  **Status:** 201  **Content-Type:** application/json  **Body:**  {  "data": {  "message": "Файл успешно отправлен"  }  } |

Создание курсов, мастер-классов, индивидуальных занятий

**Все поля обязательны для заполнения**

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/activities  **Method:** POST    **Headers**  - Content-Type: application/json    **Body:**  {  "title": "Основы программирования",  "description": "Курс для начинающих",  "homework\_file": "homework1.pdf"  } | Успех  **Status:** 201  **Content-Type:** application/json  **Body:**  {  "data": {  "id": 1,  "message": "Занятие успешно создано"  }  }  Ошибки валидации  **Формат ответа из общих требований** |

Удаление курсов, мастер-классов, индивидуальных занятий

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/activities/{id}  **Method:** DELETE | Успех  **Status:** 204  **Content-Type:** application/json    **Body:**  {  "data": {  "message": "Занятие успешно удалено"  }  }    Ошибки валидации  **Формат ответа из общих требований**    Попытка доступа к несуществующему объекту  **Формат ответа из общих требований** |

Изменение курсов, мастер-классов, индивидуальных занятий

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/activities/{id}  **Method:** PUT    **Body:**  {  "title": "Основы веб-разработки",  "description": "Обновленный курс для начинающих",  "homework\_file": "homework\_updated.pdf"  } | Успех  **Status:** 200  **Content-Type:** application/json    **Body:**  {  "data": {  "message": "Занятие успешно обновлено"  }  }  Ошибки валидации  **Формат ответа из общих требований**    Попытка доступа к несуществующему объекту  **Формат ответа из общих требований** |

**Список пользователей**

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/users  **Method:** POST    **Headers**  - Content-Type: multipart/form-data | Успех  **Status:** 200  **Content-Type:** application/json    **Body:**  {  "data": [  {  "id": 1,  "first\_name": "Alexey",  "last\_name": "Smirnov",  "email": "user1@prof.ru",  "role": "user",  "birth\_date": "2001-03-19",  },  {  "id": 2,  "first\_name": "Ivan",  "last\_name": "Petrov",  "email": "admin@prof.ru",  "role": "admin",  "birth\_date": "2001-02-15",  }  ]  }  Ошибки валидации  **Формат ответа из общих требований** |

Роли и права доступа

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/users/{id}/role  **Method:** PATCH  **Body:**  {  "role": "admin"  } | Успех  **Status:** 200  **Content-Type:** application/json    **Body:**  {  "data": {  "user": {  "id": 1,  "name": "Alexey Smirnov",  "role": "admin"  },  "code": 200,  "message": "Роль пользователя успешно обновлена"  }  } |

Логи действий

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/logs  **Method:** GET    **Query string (GET Parameters):**  - user\_id  - action  user\_id=2&action=create\_course | Успех  **Status:** 200  **Content-Type:** application/json    **Body:**  {  "data": [  {  "id": 1,  "user": {  "id": 2,  "name": "Alexey Smirnov",  "email": "admin@prof.ru"  },  "action": "create\_course",  "details": "Создал курс 'Основы программирования'",  "timestamp": "2025-01-21T10:00:00Z"  },  {  "id": 2,  "user": {  "id": 3,  "name": "Ivan Ivanov",  "email": "user@prof.ru"  },  "action": "upload\_homework",  "details": "Загрузил файл 'homework1.pdf'",  "timestamp": "2025-01-21T10:05:00Z"  }  ]  } |

Поиск по курсам, мастер-классам, индивидуальным занятиям

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/activities/search  **Method:** GET    **Query string (GET Parameters):**  - query    **Query=**программирование | Успех  **Status:** 200  **Content-Type:** application/json    **Body:**  {  "data": [  {  "id": 1,  "title": "Основы программирования",  "description": "Курс для начинающих",  "homework\_file": "homework1.pdf"  },  {  "id": 2,  "title": "Продвинутое программирование",  "description": "Курс для опытных разработчиков",  "homework\_file": "homework2.pdf"  }  ]  } |

## Инструкция для конкурсанта

Вам необходимо создать две тестовые учетные записи со следующими данными:

* **email**: user1@eduplatform.ru, **password**: Educ@tion123
* **email**: admin@eduplatform.ru, **password**: Adm1nC0urse

Вам предоставляются следующие конфигурации фреймворков и библиотек:

* Laravel 10.x
* Laravel 11.x
* Yii Basic 2.0.x
* Django 4.0.x (включая пакеты: djangorestframework, django-cors-headers, pillow, django-filter, mysqlclient)

Вы можете использовать любой из представленных фреймворков.

**Модуль Г. Разработка ИР с использованием готовых решений**

**ВВЕДЕНИЕ**

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

**Время на выполнение**: 3 часа

К вам обратилась администрация Курганской области, с целью создать сайт для организации экскурсий по области для обучающихся учебных заведений. Вам предоставляют готовый HTML-шаблон, который нужно немного модернизировать и разработать новый сайт с использованием системы управления контентом Wordpress.

|  |
| --- |
| **ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!** |

## ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – разработать сайт с использованием предоставленного HTML-шаблона, указанной структурой и требуемого функционала.

Заказчик отметил, что пользователями сайта являются люди в возрасте от 7 до 18 лет.

**Структура главной страницы**

Главная страница должна содержать следующие секции/блоки последовательно:

* *Меню* (навигация по странице или к отдельной странице):
  + О проекте (секция);
  + Муниципальные округа (отдельная страница);
  + Контакты (секция);
  + Департамент образования и науки (отдельная страница);
  + Администрация города Курган (Отдельная страница);
  + Краеведческий музей города Курган (Отдельная страница);
* *Слайдер* (не менее трех слайдов с использованием стороннего плагина).
* *О проекте* (краткая информация о проекте из медиа).
* *Муниципальные округа* (от 6 штук, с возможностью перехода на отдельную страницу округа). Решение о публикации округа принимает администратор сайта, путем изменения признака необходимости вывода на главную страницу. Секция должна иметь ссылку на страницу со всеми округами. Элементы карточки округа:
  + Главное изображение;
  + Название округа
  + Площадь;
  + Население;
* *Контакты* должна содержать(из медиа):
  + Номер телефона;
  + Адрес администрации;
  + Адрес электронной почты;
  + Время работы;
* *Записаться на экскурсию*:
  + Дата экскурсии;
  + Время экскурсии;
  + Ответственное лицо;
  + Номер телефона;
  + Округ;
  + Кнопка отправки формы;
* *Карта (*заглушка в виде изображения из медиа).
* *Подвал сайта*:
  + Номер телефона;
  + Навигация по сайту;
  + Социальные сети (vk, telegram).

**Страница со всеми округами**

На странице со всеми округами отображаются все округи с пагинацией и фильтрацией по категориям:

* Курган;
* Шадринский;
* Куртамышский;
* Шатровский;
* Кетовский;
* Далматовский;
* Белозерский;

Карточка округа содержит:

* Главное изображение (герб);
* Название округа;
* Административный центр;
* Площадь;

**Страница округа**

На странице округа отображается следующая информация (каждый элемент должен быть представлен и настраиваться как отдельное поле или свойство округа, порядок на ваше усмотрение):

* Главное изображение(герб);
* Дополнительные изображения(флаг);
* Название округа;
* Административный центр;
* Площадь;
* Численность;
* Народности(русские, татары, казахи, белорусы);
* Ссылки на дополнительные материалы (http external links);
* QR-код с ссылкой на текущую страницу (вставляется с помощью собственного плагина QRpage);
* Маршрутный лист;

На странице округа необходимо предусмотреть возможность вывода на печать информационной таблички для дальнейшего размещения в зале музея. Примерный шаблон для печати:

|  |  |
| --- | --- |
|  | **Название округа Административный центр** |
| **Площадь** |
| **Численность** |
| **Маршрутный лист** |

Страница Департамент образования и науки (отдельная страница)

Основная информация (Главное изображение, ссылка на сайт)

Страница Администрации города Кургана (Отдельная страница)

Основная информация (Главное изображение, ссылка на сайт)

Страница Краеведческого музея города Кургана (Отдельная страница)

Основная информация (Главное изображение, ссылка на сайт)

**Плагины**

Вам необходимо разработать два плагина.

RandomOKRUG

Плагин, который при загрузке любой страницы на сайте выводит информацию о случайном округе из каталога. 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 привлечения внимания к округу.

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.

**ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА**

Разработанный веб-сайт должен быть доступен по адресу http://xxxxxx-m4.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.

Данные для админ панели находятся в текстовом документе в медиа.