



ВСЕРОССИЙСКОЕ
ЧЕМПИОНАТНОЕ
ДВИЖЕНИЕ
ПО ПРОФЕССИОНАЛЬНОМУ
МАСТЕРСТВУ

ЧЕМПИОНАТ
ПРОФЕССИОНАЛЬНОГО МАСТЕРСТВА
ГОРОДА МОСКВЫ

МОСКОВСКИЕ
МАСТЕРА
2023



Разработано главным экспертом по компетенции

«Веб-технологии»

(подпись)

/ Зимин Артем Евгеньевич

(ФИО главного эксперта)



СОГЛАСОВАНО

Индустриальный эксперт

/ Шведов Андрей Вячеславович

(подпись)

(ФИО)

Менеджер компетенции

(подпись)

/ Агарков Олег Владимирович

(ФИО)

КОНКУРСНОЕ ЗАДАНИЕ КОМПЕТЕНЦИИ «Веб-технологии» для возрастной категории Основная

2023 г.

*Чемпионат профессионального мастерства города Москвы «Московские мастера» 2023 в рамках
Регионального этапа Чемпионата по профессиональному мастерству «Профессионалы» и
Чемпионата высоких технологий - 2023 в Москве*

Конкурсное задание разработано экспертным сообществом и утверждено
индустриальным партнером, 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. Спецификация оценки компетенции	2
1.1. Конкурсное задание.....	2
1.2. Структура модулей конкурсного задания	3

1. СПЕЦИФИКАЦИЯ ОЦЕНКИ КОМПЕТЕНЦИИ

Оценка Конкурсного задания будет основываться на критериях, указанных в таблице №1:

Таблица №1

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

1.1. КОНКУРСНОЕ ЗАДАНИЕ

Формат участия: индивидуальный/командный (с указанием кол-ва человек в команде).

Возрастная категория: Основная

Общая продолжительность Конкурсного задания¹: 15 ч.

Количество конкурсных дней: 3 дней.

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

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

Конкурсное задание состоит из 4 модулей, общее количество баллов конкурсного задания составляет 100.

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

1.2. Структура модулей конкурсного задания

Модуль А. Разработка интерфейса пользователя

Время на выполнение модуля: 6 часов

ВВЕДЕНИЕ

Технологии этого модуля: HTML5, CSS3

Время на выполнение: 6 часа

К вам обратилась компания «ВсемПоВелосипедуРу» с просьбой сверстать для них пару веб-страниц их будущего сайта. ВсемПоВелосипедуРу – это компания, которая торгует велосипедами. Из-за технической ошибки исходные файлы с дизайном были удалены, и всё, что они могут вам предоставить – это небольшое техническое задание для разработки.

Вам необходимо использовать все имеющиеся навыки в верстке, чтобы сверстать страницы согласно техническому заданию.

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – сверстать главную и страницу с товаром.

В медиафайлах вы можете найти все необходимые изображения, текстовую информацию и шрифты. Предоставленной информации достаточно для верстки требуемых страниц. Рекомендуются не менять предоставленный контент.

На каждой странице необходимо отобразить meta-теги для поисковой оптимизации, которые так же предоставлены в медиафайлах. Не забудь позаботиться о значке, который будет отображаться браузером во вкладке перед названием страницы.

Главная страница

Вы должны сверстать данную страницу продумать дизайн, интерфейс и сохранить с названием main.html

Страница должна содержать следующие секции:

1. Первый экран
2. Модели
3. Мастерская
4. Запчасти и аксессуары
5. Подписка
6. Подвал

1. Секция первый экран

Данная секция должна включаться в себя следующие элементы:

- Фон
- Логотип
- Меню навигации из следующих пунктов:
 - Велосипеды
 - Аксессуары
 - Запчасти
 - Мастерская
 - Доставка и оплата
 - Контакты
- Иконки:
 - Пользователь
 - Поиск
 - Корзина
- Заголовок «Велосипеды с твоим характером»
- Элемент для перехода «В магазин»
- Ссылки-иконки на социальные сети:
 - VK
 - Одноклассники
 - RuTube

2. Секция моделей

Данная секция должна включать в себя следующие элементы: Заголовок, Подзаголовок, Три модели велосипеда, где каждый велосипед должен содержать обязательный контент (Изображение, Название, Характеристики, Цену, Элемент с иконкой для добавления в корзину), элемент с текстом «Все велосипеды» для перехода на страницу.

3. Секция мастерская

Данная секция должна включать в себя следующие элементы: Изображение, Фон, Заголовок, Описание, Элемент для перехода «Подробнее».

4. Секция запчастей и аксессуаров

Данная секция должна включать в себя следующие элементы: Заголовок, Подзаголовок, Четыре аксессуара, где каждый аксессуар должен содержать обязательный контент (Изображение, Название, Характеристики, Цену, Элемент с иконкой для добавления в корзину), элемент с текстом «Перейти в каталог» для перехода в каталог.

5. Секция для вступления в клуб

Данная секция должна включаться в себя следующие элементы:

- Заголовок,
- Подзаголовок,
- Форму из трех полей для ввода текста, трех полей для выбора, флажок, элемент для отправки формы с текстом «Подать заявку».
- Изображение

6. Секция с подвалом сайта

Данная секция должна содержать:

- Навигацию из следующих элементов
 - Велосипеды
 - Аксессуары
 - Запчасти
 - Мастерская
 - Тест драйв
 - Доставка и оплата
 - Рассрочка
 - Вступление в клуб
- Иконки-ссылки на социальные сети
- Логотип
- Копирайт и политику конфиденциальности

Страница товара

Вы должны сверстать данную страницу максимально похоже по стили с первой страницей и сохранить ее под названием product.html

Страница должна содержать следующее:

- Шапка
 - Логотип
 - Навигация
 - Иконки:
 - Пользователь
 - Поиск
 - Корзина
- Хлебные крошки
- Название товара
- Рейтинг товара и количество поставивших оценку
- Цена товара
- Описание товара
- Гарантия
- Элемент с иконкой для добавления товара в корзину
- Изображение товара
- Элемент для увеличения изображения товара
- Вкладки (не кликабельные)
 - Подробное описание
 - Условия гарантии
 - Отзывы
 - Имя пользователя
 - Дата отзыва
 - Рейтинг отзыва
 - Заголовок отзыва
 - Текст отзыва
 - Элемент для возможности оставить отзыв
 - Обзоры
- Секция «С этим товаром смотрят:»
 - Заголовок секции
 - Товары, где каждый содержит:
 - Изображение
 - Иконка для добавления товара в корзину
 - Название товара
 - Характеристики
 - Цена товара

- Навигация в подвале
 - Велосипеды
 - Аксессуары
 - Запчасти
 - Мастерская
 - Тест драйв
 - Доставка и оплата
 - Рассрочка
 - Вступление в клуб
- Иконки социальных сетей
- Логотип
- Копирайт и политика конфиденциальности

Страница регистрации в личном кабинете

На этой странице вам необходимо сделать форму со следующими полями:

- Имя
- Фамилия
- Отчество
- Паспортные данные (серия и номер)
- Телефон
- Дата рождения
- Пароль
- Повтор пароля
- Согласие с правилами регистрации
- Кнопка для регистрации

Страница входа в личный кабинет

На этой странице вам необходимо сделать форму со следующими полями:

- Телефон
- Пароль
- Кнопка для входа

Каталог товаров

Шапка:

- Логотип
- Меню навигации из следующих пунктов:
 - Велосипеды
 - Аксессуары
 - Запчасти
 - Мастерская
 - Доставка и оплата
 - Контакты
- Иконки:
 - Пользователь
 - Поиск

- Корзина

Основная часть страницы:

Фильтры:

- Цена (ползунок для регулирования цены от до)
 - Электровелосипед (Да/Нет)
 - Производитель (Веломоторс, Велобалт, STELS, FORWARD)
 - Материал рамы (Сталь, Карбон)
 - Цвет (Черный, Красный, Голубой, Хаки)
 - Страна производитель (Китай, Россия, Беларусь, Венгрия)
 - Колеса (20, 22, 24, 26)
- Кнопка «Поиск»

Результаты поиска:

Необходимо отобразить 12 карточек товара

Карточка товара содержит:

- Изображение
- Кнопку добавить в корзину
- Название
- Цена

На странице необходимо отобразить пагинацию

Подвал:

- Навигацию из следующих элементов
 - Велосипеды
 - Аксессуары
 - Запчасти
 - Мастерская
 - Тест драйв
 - Доставка и оплата
 - Рассрочка
 - Вступление в клуб
- Иконки-ссылки на социальные сети
- Логотип
- Копирайт и политику конфиденциальности

Страницы с контактами:

Карта (предоставлена в медиафайлах)

- 8(800)0000-000
- velosiped@admin.ru
- г. Москва, ул. Пресненский Вал, 15, стр. 1

Форма обратной связи:

- Имя
- E-mail
- Сообщение
- Капча (код с картинки/арифметическое выражение)
- Поле ввода капчи

- Кнопка «Отправить»

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Сверстанные веб-страницы должны быть доступны по адресу <http://xxxxxx-m1.wsr.ru/>, где xxxxxx - логин участника.

Сохраните сверстанные страницы со следующими именами:

- Главная страница – main.html
- Страница с товаром – product.html
- Каталог товаров – catalog.html
- Регистрация в личном кабинете – cabinet.html
- Вход в личный кабинет – singup.html
- Контакты – contacts.html
- Политика конфиденциальности – policy.html

Все страницы указанные выше должны быть доступны к просмотру по соответствующим адресам:

- <http://xxxxxx-m1.wsr.ru/main.html>
- <http://xxxxxx-m1.wsr.ru/product.html>
- <http://xxxxxx-m1.wsr.ru/catalog.html>
- <http://xxxxxx-m1.wsr.ru/cabinet.html>
- <http://xxxxxx-m1.wsr.ru/singup.html>
- <http://xxxxxx-m1.wsr.ru/contacts.html>
- <http://xxxxxx-m1.wsr.ru/policy.html>

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

Стили, используемые в обеих страницах, должны подключаться в виде одного css файла с названием: style.min.css

Ваш HTML/CSS должен быть валидным.

Оценка будет производиться в браузере Google Chrome и Mozilla Firefox

Для выполнения модуля вы можете использовать: Bootstrap, Tailwind

Модуль Б. Разработка Веб-приложения на стороне клиента

Время на выполнение модуля: 3 часа

Введение

Технологии этого модуля: Клиентское программирование

Время на выполнение: 3 часа

К Вам обратилась компания «Just buy» для создания Single Page Application (далее SPA) для их сервиса заказа товаров. Заказчик предоставляет вам полностью готовую верстку со всеми страницами и рабочее API.

Описание проекта и задач

Общее описание проекта

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

В примерах будет использоваться переменная `{{host}}` которая обозначает адрес API:
`http://xxxxxx-m2.wsr.ru/api-shop`.

Ваше SPA должно состоять из следующих экранов:

- Каталог товаров;
- Регистрация;
- Вход в систему;
- Корзина;
- Оформленные заказы.

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

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

Функциональные требования

Каталог товаров

На данном домашнем экране находится список всех товаров из каталога. В зависимости от роли пользователя, отображаются те или иные дополнительные элементы интерфейса:

- Если пользователь не авторизован, то присутствуют ссылки на регистрацию и авторизацию в системе;
- Если пользователь авторизован, то присутствует ссылка на выход из аккаунта;

- Если пользователь авторизован как Клиент, то возле каждого товара отображается кнопка для добавления его в корзину. Также присутствует ссылка для просмотра ранее оформленных заказов;

Интерактивные элементы экрана

- По щелчку на ссылки регистрации и авторизации осуществляется переход на соответствующие экраны;
- По щелчку на ссылку выхода из аккаунта пользователь перестает быть авторизованным;
- По щелчку на ссылку для оформленных заказов осуществляется переход на экран ранее оформленных заказов;
- По щелчку на кнопку добавления товара, соответствующий товар добавляется в корзину.

Регистрация

На данном экране представлена форма для регистрации нового пользователя. При вводе некорректных значений у соответствующих полей формы отображаются тексты ошибок, а сами поля подсвечиваются красным. При успешной регистрации происходит переход на экран входа в систему.

Интерактивные элементы экрана

- По щелчку на кнопку регистрации, происходит попытка зарегистрировать нового пользователя;
- По щелчку на кнопку назад, происходит переход на домашний экран.

Вход в систему

На данном экране представлена форма для входа пользователя в систему. При вводе некорректных значений у соответствующих полей формы отображаются тексты ошибок, а сами поля подсвечиваются красным. При успешной аутентификации происходит переход на домашний экран, иначе выводится соответствующее сообщение.

Интерактивные элементы экрана

- По щелчку на кнопку входа, происходит попытка аутентифицировать пользователя;
- По щелчку на кнопку назад, происходит переход на домашний экран.

Корзина

На данном экране отображаются список добавленных в корзину товаров. Одинаковые товары в корзине сгруппированы с указанием количества. Возле каждого из товаров есть кнопки увеличения и уменьшения его количества, а также удаления его из корзины. Если в корзине есть товары, то также присутствует кнопка для оформления заказа.

Интерактивные элементы экрана

- По щелчку на кнопки увеличения/уменьшения количества товара, соответственно количество товара увеличивается/уменьшается;
- По щелчку на кнопку удаления товара из корзины, соответствующий товар удаляется из корзины;

- По щелчку на кнопку оформления заказа, происходит оформление заказа и переход на экран с заказами;
- По щелчку на кнопку назад, происходит переход на домашний экран.

Оформленные заказы

На данном экране отображается список оформленных заказов пользователя.

Интерактивные элементы экрана

- По щелчку на кнопку назад, происходит переход на домашний экран.

Описание предоставленного API

Документация API идентична описанной в Модуле 1.

Инструкция для конкурсанта

Разработанное приложение должно быть доступно по адресу <http://xxxxxx-m2.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
- Angular CLI
- Next js

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

Модуль В. Разработка Веб-приложения на стороне сервера

Время на выполнение модуля: 3 часа

К Вам обратилась компания «BreezeResort», которая имеет свою сеть отелей в разных странах мира. Так как компания хочется расширяться, им необходимо разработать свою информационную систему для администрирования своих отелей. Для начала им необходимо разработать REST API. К сожалению, предыдущая компании не исполнила свой контракт на разработку и отказалась предоставлять резервную копию базы данных, единственное, что осталось, - список HTTP-запросов с описанием работы API. Вам необходимо создать базу данных и создать REST API.

Ваша задача – создать базу данных и реализовать REST API заданной структуры. В примерах будет использоваться переменная `{{host}}`, которая обозначает адрес `http://xxxxxx-m1.wsr.ru/resortapi`, где `xxxxxx` - логин участника.

Общие требования к API

Идентификацию пользователя организуйте посредством Bearer Token.

При написании API вам необходимо разрешить кросс-доменные запросы (CORS) для обращения с другого домена.

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

Status: 401

Content-Type: application/json

Body:

```
{
  "error": {
    "message": "Unauthorized"
  }
}
```

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

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

В свойстве 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:

username – обязательно поле

password – обязательное поле

то тело ответа будет следующим:

```
{
  "message": "The given data was invalid.",
  "errors": {
    "username": [
      "The username field is required."
    ],
    "password": [
      "The password field is required."
    ]
  }
}
```

В значениях свойств errors вы можете использовать любые сообщения об ошибках (если не указана конкретная ошибка), но они должны описывать возникшую проблему.

Разработанное API должно быть доступно по адресам, указанным в требованиях к API.

Вы можете использовать любой из представленных фреймворков:

Laravel 8.5.x

Yii 2.0.x

NodeJS(express, cors, mysql2, typeorm, reflect-metadata, typescript, cors-env, dotenv, config)

Регистрация администратора системы

При успешной аутентификации возвращается сообщение.

Поля: username, password – обязательные, при этом username должен быть уникальным.

Request	Response
URL: {{host}} /signup Method: POST Headers - Content-Type: application/json	Успех Status: 200 Content-Type: application/json Body: {

Body: <pre>{ "username": "admin ", "password": "admin" }</pre>	<pre>"data": { "message": "Administrator created" }</pre> <p>Ошибки валидации полей</p> <p>Формат ответа из общих требований</p>
--	--

Аутентификация администратора системы

При успешной аутентификации возвращается сгенерированный токен зарегистрированного администратора системы.

Request	Response
URL: {{host}} /login Method: POST Headers - Content-Type: application/json Body: <pre>{ "username": "admin@admin.ru", "password": "qwerty123456" }</pre>	<pre>----- УСПЕХ -----</pre> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "data": { "token_user": <сгенерированный token> } }</pre> <p>Неправильные логин или пароль</p> <p>Status: 401</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "message": "Unauthorized", "errors": { "login": "invalid credentials" } }</pre> <p>Ошибки валидации полей</p> <p>Формат ответа из общих требований</p>

Последующий функционал доступен только авторизованному администратору. Идентификация пользователя должно быть организована посредством Bearer Token.

Номер

Добавление

Для добавления новой необходимо указать два поля, оба поля являются обязательными, название должно быть уникальным.

Request	Response
<p>URL: {{host}} /room</p> <p>Method: POST</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token} <p>Body:</p> <pre>{ "name": "Название", "desc_data": "Описание", }</pre>	<p>----- УСПЕХ -----</p> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "data": { "message": "Created" } }</pre> <p>Ошибки валидации полей</p> <p>Формат ответа из общих требований</p>

Получение списка всех

Возвращается массив list, содержащий данные, которые есть в системе

Request	Response
<p>URL: {{host}} /rooms</p> <p>Method: GET</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token}	<p>----- УСПЕХ -----</p> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "list": { { "id": 1, "name": "Название", } } }</pre>

	<pre> "desc_data": "Описание" }, { "id": 2, "name": "Название", "desc_data": "Описание" } } </pre>
--	--

Удаление

Request	Response
<p>URL: {{host}} /room/{id}</p> <p>Method: DELETE</p> <p>Headers</p> <ul style="list-style-type: none"> - Content-Type: application/json - Authorization: Bearer {token} 	<p>----- УСПЕХ -----</p> <p>Status: 204</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre> { "data": { "message": "Deleted" } } </pre> <p>Попытка удалить несуществующую</p> <p>Status: 403</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre> { "error": { "message": "Not found" } } </pre>

Клиент

Добавление нового

При отправке запроса необходимо передать объект со следующими свойствами:

fio – обязательное и уникальное поле, строка;

email – обязательное и уникальное поле, электронная почта, строка;

phone – обязательное и уникальное поле, строка;

birth_date - обязательное, формат даты (ГГГГ-ММ-ДД);

id_chiilddata – обязательное поле, id существующего номера.

Request	Response
<p>URL: {{host}} /register Method: POST</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token} <p>Body:</p> <pre>{ "fio": "Ivanov Ivan Ivanovich", "email": "ivan@wsr.ru", "phone": "89001234567", "id_chiilddata": 1, "birth_date": "2000-03-12", }</pre>	<p>----- УСПЕХ -----</p> <p>Status: 201 Content-Type: application/json Body:</p> <pre>{ "data": { "message": "Created" } }</pre> <p>Ошибки валидации полей Формат ответа из общих требований</p>

Редактирование информации

Request	Response
<p>URL: {{host}} /userdata/{id} Method: PATCH</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token} <p>Body:</p> <pre>{ "phone": "89231234567", }</pre>	<p>----- УСПЕХ -----</p> <p>Status: 200 Content-Type: application/json Body:</p> <pre>{ "data": { "id": "5", "message": "Updated" } }</pre>

Удаление

При удалении из системы необходимо производить id в системе.

Request	Response
URL: {{host}} /userdata/{id} Method: DELETE Headers - Content-Type: application/json - Authorization: Bearer {token}	Успех Status: 204 Content-Type: application/json Body: { "data": { "message": "Deleted" } } Попытка удалить несуществующего Status: 403 Content-Type: application/json Body: { "error": { "message": "Not found" } } Примечание: {id} - идентификатор в системе

Изменение номера комнаты у клиента

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

URL: {{host}} /room/{id}/userdata/{iduser} Method: GET Headers - Content-Type: application/json - Authorization: Bearer {token} Примечание: { id } – идентификатор номера в системе { iduser } – идентификатор клиента в системе	<p>----- УСПЕХ -----</p> Status: 200 Content-Type: application/json Body: <pre>{ "data": { "message": "Changed" } }</pre> Попытка указать несуществующие данные Status: 403 Content-Type: application/json Body: <pre>{ "error": { "message": "Not found" } }</pre>
---	--

Получение списка номеров и клиентов в них

Request	Response
URL: {{host}} /usersinroom Method: GET Headers - Content-Type: application/json - Authorization: Bearer {token}	<p>----- УСПЕХ -----</p> Status: 200 Content-Type: application/json Body: <pre>{ "data": [{ "name": "Название", "userdata": [{ "fio": "Иванов Иван Иванович", "phonenumber": "79001234567" }] }] }</pre>

	<pre>] } Ошибки валидации полей Формат ответа из общих требований</pre>
--	---

Отель

Добавление

При отправке запроса необходимо передать объект со следующими свойствами:

name- обязательное и уникальное

number – обязательное и уникальное

Request	Response
URL: {{host}} /hotel Method: POST Headers - Content-Type: application/json - Authorization: Bearer {token} Body: <pre>{ "name": "Название", "number": 000, }</pre>	<pre>----- УСПЕХ ----- Status: 201 Content-Type: application/json Body: { "data": { "id": 18, "name": "Название", "number": 000 } }</pre> <p>Ошибки валидации полей Формат ответа из общих требований</p>

Получение всех

Возвращается массив list, содержащий список данных в системе

Request	Response
<p>URL: {{host}} /hotels</p> <p>Method: GET</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token}	<p>----- УСПЕХ -----</p> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "list": [{ "name": "Название", "number": 0001 }, { "name": "Название", "number": 0002 }] }</pre>

Удаление

При удалении из системы необходимо производить id в системе.

Request	Response
<p>URL: {{host}} /hotel/{id}</p> <p>Method: DELETE</p> <p>Headers</p> <ul style="list-style-type: none">- Content-Type: application/json- Authorization: Bearer {token} <p>Примечание:</p> <p>{id} - идентификатор в системе</p>	<p>Успех</p> <p>Status: 204</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "data": { "message": "Deleted" } }</pre> <p>Попытка удалить несуществующие данные</p> <p>Status: 403</p>

	Content-Type: application/json Body: <pre>{ "error": { "message": "Not found" } }</pre>
--	---

Добавление комнаты в отель

При добавлении комнаты в определенный отель, необходимо передать id комнаты и id отеля.

Request	Response
URL: {{host}} /hotel/{id}/room/{idroom} Method: GET Headers - Content-Type: application/json - Authorization: Bearer {token} Примечание: { idroom } - идентификатор комнаты в системе { id } - идентификатор отеля в системе	<pre>----- УСПЕХ -----</pre> Status: 201 Content-Type: application/json Body: <pre>{ "data": { "name": "Название", "title ": "Название" } }</pre> Попытка указать несуществующие данные Status: 403 Content-Type: application/json Body: <pre>{ "error": { "message": "Not found" } }</pre>

Получение списка всех отелей, закрепленных за ними номеров и клиентов, которые входят в этот номер

Request	Response
<p>URL: {{host}} /roomsinhotels</p> <p>Method: GET</p> <p>Headers</p> <ul style="list-style-type: none"> - Content-Type: application/json - Authorization: Bearer {token} 	<p>----- УСПЕХ -----</p> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "data": [{ "title": "Название", "number": 3, "data_children": [{ "name": "Название", "userdata": [{ "fio": "Иванов Иван Иванович", "phonenumber": "79001234567" }] }], }, { "name": "Название", "userdata": [] }] }, { "title": "Название", "number": 3, "data_children": [] }]</pre> <p>----- Validation error -----</p> <p>Ошибки валидации полей</p> <p>Формат ответа из общих требований</p>

Модуль Г. Разработка ИР с использованием готовых решений

Время на выполнение модуля: 3 часа

ВВЕДЕНИЕ

К вам обратилась компания «Gallery», которая занимается продажей частных художественных картин. Компании необходим сайт для презентации картин. Вам предоставляют готовый HTML-шаблон, который нашёл заказчик, его нужно немного модернизировать и разработать новый сайт с использованием системы управления контентом Wordpress.

ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!

ВНИМАНИЕ! Тема должна быть разработана с использованием предоставленного HTML-шаблона.

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

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

Структура главной страницы

Главная страница должна содержать следующие секции/блоки последовательно:

- **Меню** (навигация по странице или к отдельной странице):
 - О компании (секция);
 - Экспозиции (отдельная страница);
 - Контакты (секция);
- **Слайдер** (не менее трех слайдов с использованием стороннего плагина).
- **О компании** (краткая информация о компании из медиа).
- **Интересные экспозиции** (от 6 штук, с возможностью перехода на отдельную страницу экспозиции). Решение о публикации экспозиции принимает администратор сайта, путем изменения признака необходимости вывода экспоната на главную страницу. Секция должна иметь ссылку на страницу со всеми экспозициями. Элементы карточки экспозиции:
 - Главное изображение;
 - Название;
 - Цена;
- **Наши контакты** должна содержать:
 - Номер телефона;
 - Адрес компании;
 - Время работы;

- *Форма заявки* на обратный звонок (с использованием стороннего плагина):
 - Имя;
 - Номер телефона;
 - Кнопка отправки формы;
- *Подвал сайта* (должен быть реализован с помощью области для виджетов):
 - Продублированное меню;
 - Адрес компании;
 - Ссылки на социальные сети (должно быть реализовано с использованием стороннего плагина);
 - Номер телефона;

Страница со всеми экспозициями

На странице со всеми экспозициями отображается не более 6 элементов экспозиции с пагинацией и фильтрацией по категориям:

- Портрет;
- Пейзаж;
- Исторический;
- Анималистика;

Карточка экспозиции содержит:

- Главное изображение;
- Название;
- Цена;

Страница экспозиции

На странице экспозиции отображаются следующие свойства:

- Название;
- Категория;
- Главное изображение;
- Описание;
- Дополнительные изображения;
- Цена;
- Автор картины;
- Год создания;

В административной панели на странице редактирования отдельной экспозиции должны быть представлены отдельные поля для редактирования всех перечисленных свойств. Для некоторых из свойств, вы можете использовать стандартные поля.

Плагины

Вам необходимо разработать плагин.

Random Image Widget

Вам необходимо разработать плагин, который позволяет добавить виджет вывода изображения рандомной экспозиции заданной категории. Т.е. администратору необходимо выбрать категорию, из которой будет выводиться главное изображение рандомной экспозиции. Изображение должно являться ссылкой на страницу экспозиции. Виджет должен быть добавлен в область для виджетов в подвале. Изображение должно сменяться каждые 3 секунды, анимировано с использованием JavaScript.

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Разработанный веб-сайт должен быть доступен по адресу <http://xxxxxx-m5.wsr.ru/>, где xxxxxx - логин участника (указан на индивидуальной карточке).

Панель администратора должна быть доступна по логину – **admin** и паролю – **PROF2022MOSCOW**

Название темы - **Gallery**

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.