**Конкурсное задание**

Компетенция

Веб-дизайн и разработка

Конкурсное задание включает в себя следующие разделы:

1. Формы участия в конкурсе
2. Задание для конкурса
3. Модули задания и необходимое время
4. Критерии оценки
5. Необходимые приложения

Количество часов на выполнение задания:21ч.

## 1. ФОРМЫ УЧАСТИЯ В КОНКУРСЕ

Индивидуальный конкурс.

## 2. ЗАДАНИЕ ДЛЯ КОНКУРСА

Содержанием конкурсного задания являются разработка веб-приложений. Участники соревнований получают инструкцию, необходимые медиа данные. Конкурсное задание имеет несколько модулей, выполняемых последовательно.

Конкурс включает в себя разработку приложения на стороне сервера, разработку приложения на стороне клиента, проведение редизайна макета а также верстку.

Окончательные аспекты критериев оценки уточняются членами жюри. Оценка производится в отношении работы модулейВремя и 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.

## 3. МОДУЛИ ЗАДАНИЯ И НЕОБХОДИМОЕ ВРЕМЯ

Модули и время сведены в таблице 1

Таблица 1.

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование модуля | Рабочее время | Время на задание |
| 1 | Модуль 1: RESTful API | С1 09.00-12.00 | 3 часа |
| 2 | Модуль 2: Разработка на стороне клиента | С1 13.00-16.00 | 3 часа |
| 3 | Модуль 3: Разработка на стороне клиента  Модуль 3: Разработка на стороне клиента | С2 09.00-12.00  С2 13.00-16.00 | 3 часа  3 часа |
| 5 | Модуль 4: Разработка на CMS Wordpress | С3 09.00-12.00  С3 09.00-12.00 | 3 часа  3 часа |

## 

**Модуль 1**

## введение

К вам обратился судовладелец с просьбой вступить в команду, которая разрабатывает для него блог. Блог должен стать своеобразным информационным посредником между его компанией и клиентами – существующими или потенциальными.

От вас требуется написать RESTful API для блога, всем остальным будет заниматься другие разработчики из вашей команды.

Функционал блога будет разделен на две роли:

* администратор
* гость

Функциональные возможности гостя:

* авторизация
* просмотр записей блога и комментариев
* комментирование записей блога
* поиск записей по тегу

Функциональные возможности администратора включают в себя функциональные возможности гостя, а также:

* создание, редактирование и удаление записей блога
* удаление комментариев к записи блога

Данное задание состоит из одного модуля и рассчитано на 3 часа. Распределите свое время таким образом, чтобы успеть выполнить все поставленные задачи.

## описание проекта и задач

**API авторизации** должен быть доступен POST-запросом на адрес {API}/auth, а также принимать следующие обязательные параметры:

* login  
  логин администратора
* password  
  пароль администратора

При успешной авторизации должен возвращаться ответ в следующем виде:

* status code: 200
* status text: Successful authorization
* body:
  + status  
    true
  + token  
    bearer-токен для использования API администратора

При безуспешной авторизации или ошибке в запросе должен возвращаться ответ в следующем виде:

* status code: 401
* status text: Invalid authorization data
* body:
  + status  
    false
  + message  
    Invalid authorization data

**API создания постов** должен быть доступен только **администратору** POST-запросом на адрес {API}/posts, а также принимать следующие обязательные параметры:

* title  
  название поста, не пустой, уникальный
* anons  
  анонс поста, не пустой
* text  
  текст поста, не пустой
* tags  
  тэги поста через запятую, не обязательное поле
* image  
  изображение поста, не пустой, разрешенные форматы: jpg, png. Максимальный размер: 2 мегабайта

При успешном создании поста должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful creation
* body:
  + status  
    true
  + post\_id  
    уникальный идентификатор поста

После создания поста изображение должно загружаться на сервер в папку {API}/post\_images.

При безуспешном добавлении поста должен возвращаться ответ в следующем виде:

* status code: 400
* status text: Creating error
* body:
  + status  
    false
  + message  
    ассоциативный массив из параметров, которые содержат ошибку. В значении должно быть описание ошибки

Пример ответа при безуспешном добавлении поста:

{

“status”: false,

“message”: {

“title”: “already exists”,

“image”: “invalid file format”

}

}

**API редактирования постов** должен быть доступен только **администратору** POST-запросом на адрес {API}/posts/<POST\_ID>, а также принимать следующие параметры:

* title  
  название поста, уникальный
* anons  
  анонс поста
* text  
  текст поста
* tags  
  тэги поста через запятую
* image   
  изображение поста, разрешенные форматы: jpg, png. Максимальный размер: 2 мегабайта

При успешном редактировании поста должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful creation
* body:
  + status  
    true
  + post
    - title  
      название поста
    - datatime  
      дата и время создания поста в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
    - anons  
      анонс поста
    - text  
      текст поста
    - tags  
      массив из тэгов поста ([“tag1”, “tag2”])
    - image  
      ссылка на изображение поста

При безуспешном добавлении поста должен возвращаться ответ в следующем виде:

* status code: 400
* status text: Editing error
* body:
  + status  
    false
  + message  
    ассоциативный массив из параметров, которые содержат ошибку. В значении должно быть описание ошибки

При попытке редактирования несуществующего поста должен возвращаться ответ в следующем виде:

* status code: 404
* status text: Post not found
* body:
  + message  
    Post not found

**API удаления поста** должен быть доступен только **администратору** DELETE-запросом на адрес {API}/posts/<POST\_ID>.

При успешном удалении поста должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful delete
* body:
  + status  
    true

При попытке удаления несуществующего поста должен возвращаться ответ в следующем виде:

* status code: 404
* status text: Post not found
* body:
  + message  
    Post not found

**API просмотра записей** блога должен быть доступен GET-запросом на адрес {API}/posts. Ответ должен содержать массив объектов постов, содержащие следующие параметры:

* title  
  название поста
* datatime  
  дата и время создания поста в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
* anons  
  анонс поста
* text  
  текст поста
* tags  
  массив из тэгов поста ([“tag1”, “tag2”])
* image  
  ссылка на изображение поста

Ответ должен иметь status code “200” и status text “List posts”.

**API просмотра одной записи** блога должен быть доступен GET-запросом на адрес {API}/posts/<POST\_ID>.

Ответ должен содержать следующие параметры:

* title  
  название поста
* datatime  
  дата и время создания поста в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
* anons  
  анонс поста
* text  
  текст поста
* tags  
  массив из тэгов поста ([“tag1”, “tag2”])
* image  
  ссылка на изображение поста
* comments  
  массив из объектов комментариев содержащие следующие параметры:
  + comment\_id  
    уникальный идентификатор комментария
  + datatime  
    дата и время в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
  + author  
    имя автора, если автором комментария является администратор, то парметр содержит текст “admin”
  + comment  
    текст комментария

Ответ должен иметь status code “200” и status text “View post”.

При попытке просмотра несуществующего поста должен возвращаться ответ в следующем виде:

* status code: 404
* status text: Post not found
* body:
  + message  
    Post not found

**API добавления комментария** к посту должен быть доступен POST-запросом на адрес {API}/posts/<POST\_ID>/comments и содержать следующие обязательные параметры:

* author  
  имя комментатора, обязательное только для гостя
* comment  
  обязательное, максимум 255 символов

При успешном добавлении комментария должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful creation
* body:
  + status  
    true

При безуспешном добавлении комментария должен возвращаться ответ в следующем виде:

* status code: 400
* status text: Creating error
* body:
  + status  
    false
  + message  
    ассоциативный массив из параметров, которые содержат ошибку. В значении должно быть описание ошибки

При попытке добавить комментарий к несуществующему посту должен возвращаться ответ в следующем виде:

* status code: 404
* status text: Post not found
* body:
  + message  
    Post not found

**API удаления комментария** к посту должен быть доступен только **администратору** DELETE-запросом на адрес {API}/posts/<POST\_ID>/comments/<COMMENT\_ID>.

При успешном удалении комментария должен возвращаться ответ в следующем виде:

* status code: 201
* status text: Successful delete
* body:
  + status  
    true

При попытке удаления комментария к несуществующему посту должен возвращаться ответ в следующем виде:

* status code: 404
* status text: Post not found
* body:
  + message  
    Post not found

При попытке удаления несуществующего комментария должен возвращаться ответ в следующем виде:

* status code: 404
* status text: Comment not found
* body:
  + message  
    Comment not found

**API поиска постов** по тегу должен быть доступен GET-запросом на адрес {API}/posts/tag/<TAG\_NAME>.

В ответ должен возвращаться ответ в следующем виде:

* status code: 200
* status text: Found posts
* body  
  массив из объектов постов, содержащие данный тэг:
  + title  
    название поста
  + datatime  
    дата и время создания поста в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
  + anons  
    анонс поста
  + text  
    текст поста
  + tags  
    массив из тэгов поста ([“tag1”, “tag2”])
  + image  
    ссылка на изображение поста

Авторизация пользователя при использовании административных функциональных возможностей будет проверяться через bearer-token авторизацию. Т.е. к каждому такому запросу в headers будет дописываться bearer-token:

Authorization: Bearer xxxxxxxx

При безуспешной попытке пройти авторизацию по токену на любой из запросов, который должен быть доступен только администратору, должен возвращаться ответ в следующем виде:

* status code: 401
* status text: Unauthorized
* body:
  + message  
    Unauthorized

Под {API} подразумевается адрес <http://xxx-m1.wsr.ru/api>, где xxx - ваш логин.

Логин администратора admin, пароль - sakhalin2018.

**ИНСТРУКЦИя ДЛЯ КОНКУРСАНТА**

Готовая система должна быть доступна по адресу: <http://xxxxxx-m1.wsr.ru/>

Вам доступны фреймворки Yii2 и Laravel для работы с этим модулем.

Все ответы должны приходить в JSON формате.

Во время выполнения конкурсного задания можно использовать Postman, предложенный в медиа данных.

**Модуль 2**

## введение

В последние годы Интернет стал неотъемлемой частью нашей повседневной жизни. Использование игр приобрело заметную роль в этой вселенной, позволяя миллионам людей получать доступ к развлечениям быстро и бесплатно.

Наш заказчик сделал срочный заказ: необходимо разработать игру и воплотить желание заказчика в реальность! Все что вам необходимо – это реализовать логику игры.

Название игры: Рыцарь.

Технологии этого модуля: HTML 5, CSS3, JavaScript, jQuery, Граф. дизайн, PHP

Время модуля: 3 часа

Вам необходимо реализовать функционал игры. Готовый шаблон и все необходимые файлы предоставлены. Использование шаблона обязательно.

## описание проекта и задач

В игре используются элементы, описанные ниже:

1. Рыцарь: элемент, который контролируется игроком.
2. Монстры: элементы, которые необходимо уничтожить игроку.
3. Шкала жизней (HP): шкала, отражающая запас жизненной энергии игрока.
4. Шкала энергии (MP): шкала, отражающая запас магической энергии игрока.
5. Имя игрока: имя игрока, которое он ввел на стартовом экране.
6. Убийства: количество убитых монстров.
7. Панель умений: умения, которые может использовать игрок, для сражения с монстрами.

Игра должна начинаться со стартового экрана с инструкцией к игре, полем для ввода имени игрока и кнопкой "Начать", если поле имени пустое, то кнопка не активна. Инструкция к игре должна быть представлена анимировано.

Игровой функционал:

1. По нажатию на кнопку "Начать игру" игрок попадает на экран игры. Изначально у игрока 0 очков, 100 HP и 100 MP, таймер 00:00 (в формате mm:ss).
2. В начале игры запускается таймер, персонаж располагается в левой стороне игрового поля, что является начальной границей карты.
3. Игрок может передвигаться с помощью клавиш-стрелок: налево и направо. Когда игрок находиться в левой половине видимой части экрана фон не передвигается, а «замирает». Когда игрок доходит центральной точки он остается в центральной части экрана, происходит анимация его передвижения, а фон в свою очередь начинает «прокручиваться». При этом когда фон достигает конца, игрок должен передвигаться в правую часть экрана.
4. По мере прохождения игры на игрока нападают монстры, которые случайно генерируются в видимой части игрового поля и движутся справа налево. Игра заканчивается, как только игрок доходит до конца карты и достигает правого местоположения или происходит смерть рыцаря, а таймер останавливается.
5. За каждого убитого монстра дается 1 очко.
6. Если жизни игрока кончаются (0HP), то игра заканчивается.
7. Жизни игрока (HP) регенерируются со скоростью 2HP/с.
8. Энергия игрока (MP) регенерируется со скоростью 5MP/с.
9. Игрок может использовать умения. Каждое умение тратит какое-то количество MP и имеет время перезарядки. Вот список умений:

|  |  |
| --- | --- |
| **nc-game-media/layout/img/skill-swordУдар мечом.** Описание: Игрок делает удар мечом, нанося урон монстрам перед собой. Клавиша активации: 1. Время перезарядки: 0 сек. Количество потребляемой MP: 0. Урон: 15 ед. | **Блок.** Описание: Игрок выставляет щит перед собой. Щит блокирует весь урон. Монстры не могут пройти за щит. Клавиша активации: 2. Время перезарядки: 0 сек. Количество потребляемой MP: 5. |
| **nc-game-media/layout/img/skill-sword-3.pngТрио мечей.** Описание: Игрок выпускает три меча перед собой. Мечи летят, нанося 40 единиц урона всем монстрам на своем пути. Клавиша активации: 3. Время перезарядки: 3 сек. Количество потребляемой MP: 10. Урон: 40 ед. | **nc-game-media/layout/img/skill-sword-8.pngГрад мечей.** Описание: в радиусе нескольких метров от игрока на землю обрушивается град мечей уничтожая всех, кто был в данном радиусе. Клавиша активации: 4. Время перезарядки: 15 сек. Количество потребляемой MP: 30. Урон: 100 ед. |

1. При нажатии на кнопку ESC игра ставится на паузу, а при повторном нажатии - снимается с паузы. Во время паузы останавливаются все интерактивные действия, а также вся анимация, таймер тоже замирает, умениями пользоваться нельзя, игрок и монстры не двигаются.
2. При ударе мечом урон наносится один раз каждому монстру, который находится в зоне поражения мечом, т.е. если перед игроком находится один монстр, то он получит 15 ед. урока, а если 3 монстра, то каждый из них получит по 15 ед. урона.
3. Монстры должны следовать за игроком. Если игрок пробежал монстров, то они должны развернуться и следовать за ним.
4. Когда игрок идет обратно, он поворачивается в обратную сторону.
5. Каждый из монстров имеет свою скорость атаки и скорость движения, которая не превышает скорость движения игрока. На экране не должно находиться более 10 монстров. Вот описания монстров:

|  |  |  |
| --- | --- | --- |
| nc-game-media/enemies/dog/sprites/run/1.png  Пёс  Наносит 2ед урона  Имеет 15HP | nc-game-media/enemies/orc3/sprites/Attack2/Attack2_000.png  Эльф  Наносит 5ед урона  Имеет 30HP | nc-game-media/enemies/orc1/sprites/Attack2/Attack2_000.png  Гринч  Наносит 10ед урона  Имеет 60HP |

1. Ваша игра должна работать без отображения JavaScript ошибок или сообщений в консоли браузера.
2. Ваш HTML/CSS и JavaScript код должен быть организован и понятным. Используйте корректные наименования переменных, методов и не забывайте оставлять комментарии для дальнейшей поддержки в будущем.
3. После завершения игры результаты должны быть сохранены на сервере и показана таблица с рейтингом.
4. Вам дан PHP файл, который сохраняет данные в базу данных и возвращает массив с результатами. Используйте этот файл для сохранения результатов и получения данных для рейтинга. В этом файле можно изменять только данные доступа к базе данных (логин, пароль, имя базы данных). Этот файл принимает следующий AJAX запрос:

* метод: POST
* username - имя пользователя
* score - количество убитых монстров
* time – время игры

В ответ возвращается массив с данными:

[

{"id":"1","username":"Player 1","score":"10","time":"20"},  
{"id":"2","username":"Player 2","score":"8","time":"30"}

]

1. После завершения игры открывается экран результатов, в котором формируется таблица с лучшими 10 игроками, если игрок не вошел в таблицу, то выводятся 9 лучших игроков, а в последней строке указывается место и результат игрока только что завершившего игру.
2. Данные в таблице должны быть отсортированы по количеству убитых монстров по убыванию. Если несколько строк имеют одинаковое кол-во убитых монстров, то они сортируются между собой по времени по возрастанию. Если несколько строк имеют одинаковое количество убитых монстров и одинаковое время, то они занимают одинаковую позицию в рейтинге.
3. На экране результатов должна быть кнопка «Играть сначала», которая позволяет начать игру с начала, первый экран не отображается.

Проявите все свои навыки для создания качественной игры, с анимацией и интерактивностью.

**ИНСТРУКЦИя ДЛЯ КОНКУРСАНТА**

Игра должна быть доступна по адресу: http://xxxxxx-m2.wsr.ru

Оценка будет производиться при помощи браузера Google Chrome.

**Модуль 3**

## введение

В наше время автоматизация и информатизация плотно вошли в жизнь каждого. Появляется все больше и больше гаджетов, в том числе для дома. И чем их больше, тем сложнее их контролировать.

Вам предстоит решить данную проблему, разработав интерфейс системы умного дома.

Система должна быть адаптивная, открываться на телефоне, планшете и компьютере без проблем. Мы повседневно пользуемся множеством гаджетов в своих домах. Поэтому приложение должно быть удобным и продуманным, чтобы пользователь хотел им пользоваться.

Вам предстоит разработать дизайн, сверстать его и подключить к готовой серверной части. Задача сложная, но вы один из лучших разработчиков, поэтому вы справитесь!

Технологии этого задания: HTML5, CSS3, JavaScript, VueJS, React, Граф. Дизайн

Данное задание рассчитано на 6 часов и состоит из 2ух модулей по 3ч. Распределите свое время таким образом, чтобы успеть выполнить все поставленные задачи.

Вам необходимо реализовать функционал сервиса. Серверная часть интерфейса в формате REST уже представлена. Также вы получите Postman-коллекцию с запросами к серверной части для простоты работы.

## описание проекта и задач

Ваша задача – разработать дизайн приложения, которым удобно пользоваться на смартфонах с разрешением 375х812px.

Несмотря на то, что основная целевая группа будет пользоваться вашим приложением с мобильных устройств, вы должны предусмотреть возможность удобного пользования с планшетов и компьютеров. Поэтому ваша верстка должна быть адаптивна и корректно открываться со следующих устройств:

* Мобильные телефоны с разрешением 375x812px (IPhone X);
* Планшеты с разрешением 1366х1024px (IPad Pro);
* Компьютеры с шириной экрана от 1600px.

В приложении должны быть реализованы следующие экраны:

* Экран авторизации
  + Поле для ввода логина;
  + Поле для ввода пароля;
  + Кнопка входа.
* Домашний экран
  + Доступные комнаты, каждая комната должна иметь название и изображение;
  + Избранные устройства и датчики.
* Экран комнаты
  + Название комнаты;
  + Устройства и датчики комнаты.
* Экран устройств и датчиков
  + Все устройства и датчики. Каждое устройство должно отражать свое состояние, а датчик показывать свое значение. Все устройства и датчики должны иметь название.
* Экран макросов
  + Уже имеющиеся макросы, с названиями и возможностью удаления;
  + Кнопка для создания нового макроса.

*Под макросом понимается набор команд (инструкция, сценарий), которые должны быть выполнены при его запуске.*

**ПЕРВЫЕ 3 ЧАСА**

За первые 3ч вам необходимо разработать дизайн и сверстать нарисованные экраны.

Сохраните разработанные дизайны в директорию design под следующими именами:

* Экран авторизации – mobile\_login.png
* Домашний экран – mobile\_home.png
* Экран комнаты – mobile\_room.png
* Экран устройств и датчиков – mobile\_devices.png
* Экран макросов – mobile\_macros.png

Для общей демонстрации всех экранов вам предоставляется Mockup с 5 экранами Iphone X. Используйте данный мокап и сохраните результат в папке design с именем mobile\_mockup.png

Сохраните все исходные файлы в директорию sources в папке design. Под исходными файлами понимаются файлы с разрешениями .ai, .psd, .eps и т.п.

Для демонстрации дополнительных состояний, например, интерактивности или анимации сохраните дополнительные файлы под следующими именами:

* Экран авторизации – mobile\_login\_2.png, mobile\_login\_x.png …
* Домашний экран – mobile\_home\_2.png, mobile\_home\_x.png …
* Экран комнаты – mobile\_room\_2.png, mobile\_room\_x.png …
* Экран устройств и датчиков – mobile\_devices\_2.png, mobile\_devices\_x.png …
* Экран макросов – mobile\_macros\_2.png, mobile\_macros\_x.png …

Для общей демонстрации всех экранов вам предоставляется Mockup с 5 экранами Iphone X. Используйте данный мокап и сохраните результат в папке design с именем mobile\_mockup.png

Сохраните все исходные файлы в директорию sources в папке design. Под исходными файлами понимаются файлы с разрешениями .ai, .psd, .eps и т.п.

Для демонстрации дополнительных состояний, например, интерактивности или анимации сохраните дополнительные файлы под следующими именами:

* Экран авторизации – mobile\_login\_2.png, mobile\_login\_x.png …
* Домашний экран – mobile\_home\_2.png, mobile\_home\_x.png …
* Экран комнаты – mobile\_room\_2.png, mobile\_room\_x.png …
* Экран устройств и датчиков – mobile\_devices\_2.png, mobile\_devices\_x.png …
* Экран макросов – mobile\_macros\_2.png, mobile\_macros\_x.png …

Во второй части вам необходимо разработать веб-интерфейс приложения «умный дом», которое отражает состояние системы.

**ПОСЛЕДНИЕ 3 ЧАСА**

Во второй части вам необходимо разработать веб-интерфейс приложения «умный дом», которое отражает состояние системы.

В умном доме поддерживаются следующие виды устройств:

|  |  |
| --- | --- |
| **Наименование устройства** | **Состояния** |
| Электронный замок | open/close |
| Люстра | on/off |
| Светильник | on/off |
| Электрокарниз | open/close |
| Датчик температуры | от -100 до +100 |
| Термостат | от 10 до +30 |

Для взаимодействия с системой вам предоставлено следующее API:

**Авторизация**

Method: POST

URL: http://wsr.ru/smart-home/api/login

Body:

* login
* password

Response (OK):

* token

Response (NOT OK):

* errors

**Получение списка комнат**

Method: GET

URL: <http://wsr.ru/smart-home/api/rooms>

Response - массив объектов, каждый объект содержит следую поля:

* id
* name
* photo

**Получение информации о комнате**

Method: GET

URL: <http://wsr.ru/smart-home/api/rooms/><id>

Response - объект, который содержит следую поля:

* id
* name
* photo

**Получение информации об устройствах и датчиках комнаты**

Method: GET

URL: <http://wsr.ru/smart-home/api/rooms/><id>/devices

Response - массив объектов, каждый объект содержит следую поля:

* id
* room\_id
* type\_id
* name
* value
* type\_name

**Получение информации об устройстве или датчике**

Method: GET

URL: <http://wsr.ru/smart-home/api/devices/><id>

Response - объект, который содержит следую поля:

* id
* room\_id
* type\_id
* name
* value
* type\_name

**Изменение состояния устройства**

Method: PATCH

URL: http://wsr.ru/smart-home/api/devices/<id>

Body:

* value – on | off | close | open | [10 – 30] | [-100 – 100]

Response (OK):

* id – device id
* value – current value

Response (NOT OK):

* Errors

**Список макросов**

Method: GET

URL: http://wsr.ru/smart-home/api/macros

Response - массив объектов, каждый объект содержит следую поля:

* Id
* name
* devices
  + id
  + macro\_id
  + device\_id
  + value

**Создание макроса**

Method: POST

URL: http://wsr.ru/smart-home/api/macros

Body:

* name – name of the macro.
* devices – array with devices.
* id
* value

Response (OK):

* id

Response (NOT OK):

* Errors

**Удаление макроса**

Method: DELETE

URL: http://wsr.ru/smart-home/api/macros/<id>

Response (OK):

* success - true

Response (NOT OK):

* success – false

**Активация (запуск) макроса**

Method: GET

URL: http://wsr.ru/smart-home/api/macros/<id>

Response (OK):

* success - true

Response (NOT OK):

* success - false

*С каждым запросом, кроме авторизации, должен отправляться заголовок авторизации (Authorization: Bearer token), содержащий токен, который получается при авторизации.*

Форма на экране авторизации должна отображать ошибки валидации.

На экране макросов пользователь должен видеть существующие макросы и иметь возможность воспользоваться ими. Также на данном экране пользователь должен иметь возможность создать новый макрос. При создании нового макроса пользователь должен указать его название, выбрать устройства и задать им значения, которые должны быть установлены при активации макроса.

Система должна получать актуальные данные, поэтому информация о датчиках должна обновляться каждую секунду.

Ваш HTML/CSS и JavaScript код должен быть организован и понятным. Используйте корректные наименования переменных, методов и не забывайте оставлять комментарии для дальнейшей поддержки в будущем.

Проявите все свои навыки для создания качественного веб-интерфейса, с анимацией, интерактивностью и любыми другими вещами, которые могут улучшить привлекательность работы с системой.

**ИНСТРУКЦИя ДЛЯ КОНКУРСАНТА**

Готовая система должна быть доступна по адресу: http://xxxxxx-m3.wsr.ru/

Оценка будет производиться при помощи браузера Google Chrome с использованием Device Toolbar.

## Модуль 4

## введение

Покупки в интернете - это неотъемлемая часть жизни современного человека, миллионы людей покупают в интернете все, что только можно придумать, и наш заказчик – ООО «Дальтехмонтаж» решил начать продажу изделий из металла через интернет. Правда один из друзей заказчика сказал ему, что делать сайт лучше на WordPress. Желание заказчика – закон!

ООО «ДАЛЬТЕХМОНТАЖ» на сегодняшний день – это одно из ведущих предприятий Сахалинской области и всего Дальнего Востока.

Назначение наших металлоконструкций – изготовление металлокаркасов для промышленных и административных зданий, мостов, мостовых переходов, опор линий электропередач, рекламных стел, ферм, ограждений, емкостей и бункеров различного назначения (для хранения топлива, сбора различного вида отходов, в том числе химических), а также всевозможных нестандартных конструкций.

## описание проекта и задач

Ваша задача разработать интернет-магазин для компании ООО «ДАЛЬТЕХМОНТАЖ» с помощью WordPress.

CMS WordPress можно получить через репозиторий – GIT. И не забудьте сделать commit выполненной работы в репозиторий, мы же не хотим потерять проект?

Весь необходимый контент (изображения и тексты) предоставляется в папке «Media». Соблюдайте следующие требования в процессе разработки:

1. Доступ к сайту происходит по адресу http://xxxxxx-m4.wsr.ru/  
   где xxxxxxxxx = код участника
2. Чтобы обеспечить последующее администрирование и управление данными веб-сайта, используйте эту информацию для доступа к области администрирования WordPress:
   1. Логин: Admin
   2. Пароль: adminAdmin
3. На сайте ООО «ДАЛЬТЕХМОНТАЖ» должны быть представлены следующие разделы и содержание:
   1. Логотип организации ООО «ДАЛЬТЕХМОНТАЖ»
   2. Главное меню должно быть зафиксировано в верхней части веб-страницы. Для показа контента будет использована прокрутка.
   3. Разделы в главном меню:
      1. Главная
         * Текст и изображения, дающие представление о деятельности компании
      2. О компании
         * Текст и изображения, дающие представление об оказываемых услугах компании и их выполненных проектах
      3. Напишите нам
         * Представлена форма для отправки сообщения с полями:
           + Имя:
           + Адрес электронной почты:
           + Номер телефона:
           + Способ получения сообщений: почта, SMS, WhatsApp сообщение;
           + Сообщение:
      4. Новости
         * Каждая новость содержит: текст, изображение, дату публикации.
         * На странице должно выводится не более 4 новостей. Если новостей больше должна выводится пагинация.
         * Дата публикации не должна совпадать для всех новостей.
      5. Каталог
         * Каталог товаров разделенный на группы: Пристройки, Бытовки, Винтовые сваи, Монтаж кровли, Сайдинг, Теплицы, Качели, Лестницы, Разное. Все категории должны содержать хотя бы один соответствующий товар.
         * Каждый товар должен содержать изображение, название, стоимость и информация о наличии.
         * Товар можно добавить в корзину и заказать. Сформированный заказ должен быть доступен через панель управления.
   4. Блок в нижней части страницы (колонтитул)
      * + ООО «ДАЛЬТЕХМОНТАЖ» © Copyright, 2018
        + Ссылка на условия использования интернет-магазина
        + Используя плагин, добавить ссылки на социальные сети:
          - Facebook - http://www.facebook.com
          - Twitter - http://www.twitter.com
          - YouTube - http://www.youtube.com
4. Разработайте плагин, который реализует указанный функционал:

* Создание формы обратной связи на выбранной странице с помощью шорткода;
* Обработку и валидацию данных, которые отправляет пользователь через эту форму.
* Пользователь может вводить номер телефона в любом формате, используя дополнительные символы, но в панели администратора номер отображается только в виде цифр.
* Способ получения сообщений предполагает множественный выбор:
* почта
* SMS
* WhatsApp сообщение
* Все поля формы должны быть заполнены.
* Отправленные сообщения с формы должны быть доступны через панель управления для просмотра и удаления;
* Плагин должен размещаться в сайдбаре;
* После отправки сообщения, должно отобразиться уведомление о статусе отправки.

1. В директории media лежат файлы для выполнения задачи, а именно:

* Плагины WordPress
* Текст контента и изображения для использования на сайте

1. Необходимо разработать собственную тему интернет-магазина и разместить ее именем Sahalin. Допускается использование стандартной темы WordPress и Bootstrap. Ваша тема должна быть адаптирована под мобильные устройства и корректно отображать все функции и модули интернет-магазина.
2. Чтобы улучшить результаты в поисковых системах, установите и настройте один плагин для SEO на своем веб-сайте.
3. Для проверки статистики доступа, установите и настройте плагин для подключения к Google Analytics. Использовать идентификатор UA-333438812-2.
4. HTML и CSS-код должны быть удовлетворять стандартам W3C (HTML 5 и CSS 3).
5. Логотип WordPress на странице входа на страницу администрирования должен быть заменён на логотип организации ООО «ДАЛЬТЕХМОНТАЖ».

## Инструкции для участника испытания

Участнику предоставляются медиафайлы в каталоге «media». Вы можете изменить предоставленные файлы, чтобы быть уверенным в том, что сайт выполняет свою задачу.

## 4. Критерии оценки

В данном разделе определены критерии оценки и количество начисляемых баллов (субъективные и объективные) таблица 2. Общее количество баллов задания/модуля по всем критериям оценки составляет 100.

Таблица 2.

Модуль 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| секция | критерий | судейская | объективная | сумма |
| A | Организация работы и управление | 0.80 | 0.00 | 0.80 |
| B | Коммуникация и навыки межличностного общения | 1.00 | 0.00 | 1.00 |
| С | Дизайн | 0.00 | 0.00 | 0.00 |
| D | Верстка | 0.00 | 0.00 | 0.00 |
| Е | Программирование на стороне клиента | 0.00 | 0.00 | 0.00 |
| F | Программирование на стороне сервера | 0.00 | 11.75 | 11.75 |
| G | CMS | 0.00 | 0.00 | 0.00 |
| **Всего** |  | **1.80** | **11.75** | **13.55** |

Модуль 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| секция | критерий | судейская | объективная | сумма |
| A | Организация работы и управление | 1.10 | 0.75 | 1.85 |
| B | Коммуникация и навыки межличностного общения | 1.50 | 0.50 | 2.00 |
| С | Дизайн | 0.00 | 0.00 | 0.00 |
| D | Верстка | 4.00 | 1.00 | 5.00 |
| Е | Программирование на стороне клиента | 0.00 | 10.00 | 10.00 |
| F | Программирование на стороне сервера | 0.00 | 0.25 | 0.25 |
| G | CMS | 0.00 | 0.00 | 0.00 |
| **Всего** |  | **6.60** | **12.50** | **19.10** |

Модуль 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| секция | критерий | судейская | объективная | сумма |
| A | Организация работы и управление | 0.75 | 0.60 | 1.35 |
| B | Коммуникация и навыки межличностного общения | 1.00 | 0.00 | 1.00 |
| С | Дизайн | 14.00 | 4.00 | 18.00 |
| D | Верстка | 4.50 | 9.00 | 13.50 |
| Е | Программирование на стороне клиента | 1.00 | 9.00 | 10.00 |
| F | Программирование на стороне сервера | 0.00 | 0.00 | 0.00 |
| G | CMS | 0.00 | 0.00 | 0.00 |
| **Всего** |  | **21.25** | **22.60** | **43.85** |

Модуль 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| секция | критерий | судейская | объективная | сумма |
| A | Организация работы и управление | 0.25 | 1.75 | 2.00 |
| B | Коммуникация и навыки межличностного общения | 2.00 | 0.00 | 2.00 |
| С | Дизайн | 2.5 | 1.5 | 4.00 |
| D | Верстка | 0.50 | 3.50 | 3.50 |
| Е | Программирование на стороне клиента | 0.00 | 2.00 | 2.00 |
| F | Программирование на стороне сервера | 0.00 | 2.00 | 2.00 |
| G | CMS | 1.00 | 7.00 | 8.00 |
| **Всего** |  | **6.25** | **17.25** | **23.50** |

**Субъективные оценки -** Не применимо.

**5. Приложения к заданию**

1. *Приложение 1. Медиа данные для выполнения конкурсного задания*