**КОНКУРСНОЕ ЗАДАНИЕ**

**VII Региональный чемпионат Молодые профессионалы (WorldSkills Russia)**

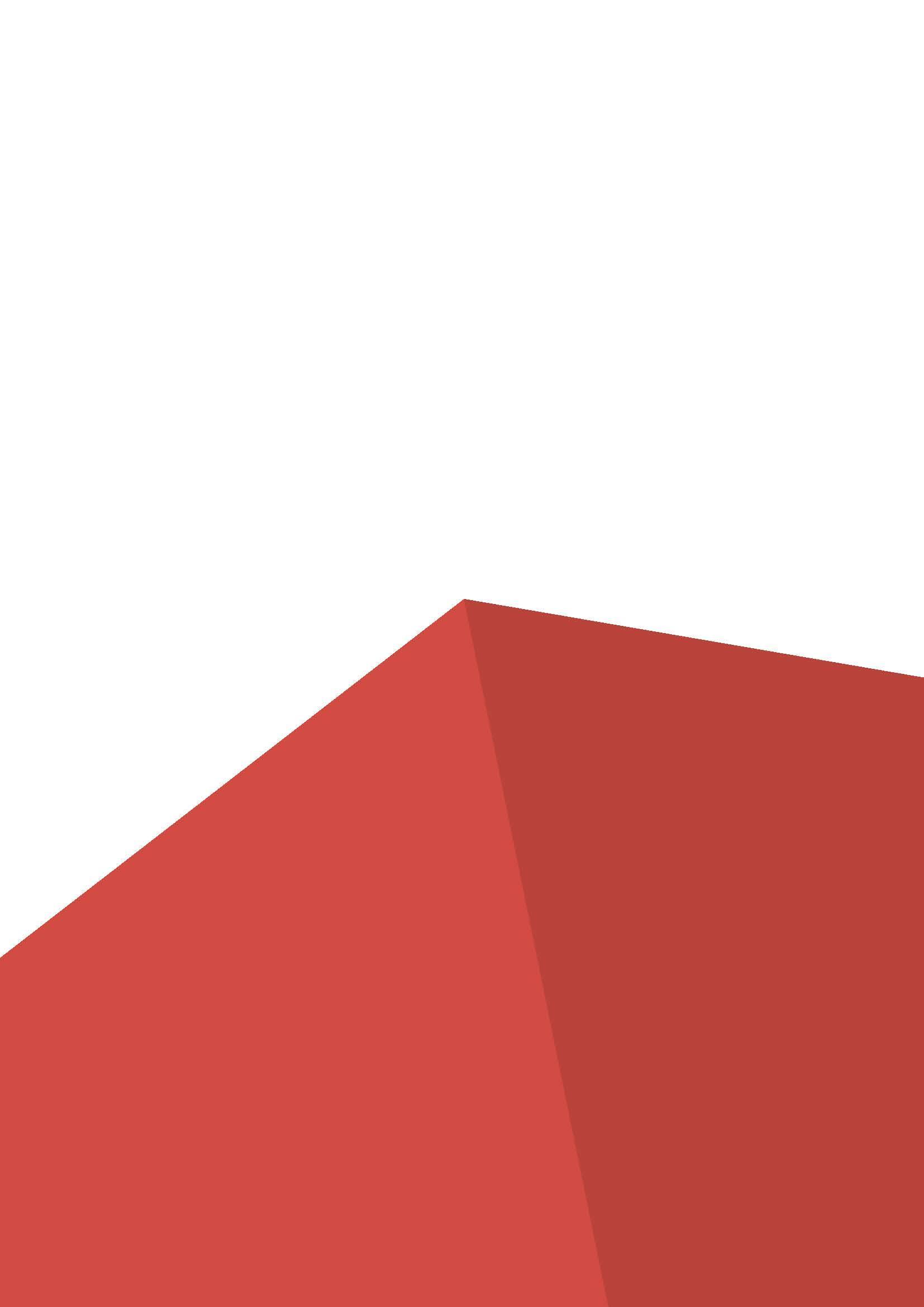
**Краснодарского края**

**компетенция**

«ВЕБ-ТЕХНОЛОГИИ»

**для основной возрастной категории**

**16-22 года**



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

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

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

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

Оценка производится в отношении работы модулей.

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

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

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

1. **Модули задания и необходимое время**

Таблица 1.

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

**Модули с описанием работ**

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

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

**Время на выполнение:** 6 часа

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

К вам обратилась компания «Индекс Драйв» - новая и энергичная компания по аренде автомобилей. Главная цель компании – сделать любой автомобиль более доступным и практичным.

Основные принципы работы сервиса:

1. Выбрать любой удобный автомобиль
2. Возможность выбора места, времени, даты
3. Выбор тарифа
4. Удобная система оплаты

Призыв: на любой случай, любой автомобиль.

Вам необходимо использовать все имеющиеся навыки в дизайне и верстке чтобы сверстать Landing Page, а также все остальные страницы. Используйте анимацию для привлечения внимания посетителя к акцентам и основным объектам сервиса.

Заказчик хочет, чтобы сайт был современный и энергичный, а также удобный, простой и не менял свои качества при различных разрешениях экрана.

Заказчик отметил, что услугами аренды автомобиля, в основном, пользуются люди, от 18 лет и имеющие стаж вождения минимум 5 лет.

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

## ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – сверстать следующие страницы веб-сайта:

* Главная страница - Landing Page
* Страница входа в личный кабинет
* Страница регистрации в личном кабинете
* Страница личного кабинета
* Страница бронирования
* Страница история бронирования
* Страница филиалов

**Главная страница (landing page)**

Главная страница должна содержать следующие блоки:

* Шапка сайта
  + Логотип компании
  + Меню навигации
    - Акции
    - Бронирование
    - Личный кабинет
* Секция о доверии компании.
* Форма поиска. Должная содержать следующие поля:
  + Класс автомобиля (эконом, премиум)
  + Марка
  + Модель
  + Кнопка поиска автомобилей
* Акции (список акций доступен в медиафайлах). Каждая акция должна содержать следующую информацию:
  + Изображение
  + Название акции
  + Краткое описание акции
  + Кнопка для просмотра акции
* Подвал сайта
  + Телефон “8 (800) 110-10-10”

Навигация по сайту (список ссылок для навигации будет предоставлен в медиафайлах)  
**Страница с результатами поиска**

На этой странице вам необходимо отобразить результаты поиска со следующей информацией:

* + Класс автомобиля (эконом, премиум)
  + Марка
  + Модель
  + Стоимость бронирования

На странице должна быть кнопка для перехода к бронированию.

**Страница бронирования**

На этой странице необходимо отобразить форму для сбора данных клиентов и его авто, а именно:

* Данные о авто
  + Идентификатор машины
  + Марка авто
  + Название авто
  + Номер машины
  + Дата бронирования
  + Дата возврата
  + Стоимость бронирования
  + Филиал
* Данные о клиенте
  + Имя
  + Фамилия
  + Отчество
  + Номер телефона
  + Паспортные данные
  + Дата рождения
* Кнопка для бронирования машины и возможность для бронирования нескольких машин.
* Финальная стоимость

Формируется исходя из данных. Если бронируется машина срок больше 3-х дней, то клиенту формируется скидка в 5% и так за каждые 3 дня скидка прибавляется по 5% и может быть максимум 25%. Это необходимо визуализировать.

* Возможность выбора машин
* Кнопка для оформления бронирования

**Страница регистрации в личном кабинете**

На этой странице вам необходимо сделать форму со следующими полями:

* Имя
* Фамилия
* Отчество
* Паспортные данные
* Телефон
* Дата рождения
* Пароль
* Повтор пароля
* Кнопка для регистрации

**Страница входа в личный кабинет**

На этой странице вам необходимо сделать форму со следующими полями:

* Телефон
* Пароль
* Кнопка для входа

**Страница истории бронирований**

На этой странице необходимо отобразить всю информацию о истории бронирований, а именно:

* Информация о бронировании
  + Код бронирования
  + Стоимость бронирования
* Информация об аренде (для каждого заказа)
  + Номер заказа
  + Марка автомобиля
  + Модель автомобиля
  + Фото автомобиля
  + Дата начала аренды
  + Дата окончания аренды
  + Итоговое количество дней аренды
  + Наименование филиала, где производилась аренда
  + Итоговая стоимость аренды
* Кнопка для повторного бронирования

В случае если у пользователя нет бронирований, то будет сообщение: «Здесь пока ничего нет»

**Страница личного кабинета**

На этой странице необходимо отобразить информацию о пользователе, а именно:

* Имя
* Фамилия
* Отчество
* Баллы
* Количество бронирований
* Кнопка выхода из личного кабинета
* Бронирования. Каждое бронирование должно отражать следующую информацию:
  + Идентификатор авто
  + Марка
  + Название
  + Картинка машины
  + Описание машины
  + Дата бронирования
  + Дата возврата
  + Выбранный филиал

**Страница филиалов**

На этой странице необходимо отобразить карту со всеми местонахождениями филиалов. Всего в городе 5 филиалов. При нажатии на филиал отображается список из 3-х машин, доступных к бронированию.

**ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА**

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

Сохраните сверстанные страницы со следующими именами:

* Главная страница - Landing Page – index.html
* Страница входа в личный кабинет – login.html
* Страница регистрации в личном кабинете – register.html
* Страница личного кабинета – profile.html
* Страница с результатами поиска – search.html
* Страница бронирования – booking.html
* Страница история бронирования – history.html
* Страница филиалов – branches.html

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

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

Оценка будет производиться в браузере Google Chrome.

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

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

**СИСТЕМА ОЦЕНКИ**

|  |  |  |
| --- | --- | --- |
| **Секция** | **Критерий** | **Сумма** |
| A | Организация работы и управление | 2,50 |
| B | Коммуникационные и межличностные навыки | 2,50 |
| C | Графический дизайн | 16,00 |
| D | Верстка страниц | 18,00 |
| E | Программирование на стороне клиента | 0 |
| F | Программирование на стороне сервера | 0 |
| G | Системы управления контентом | 0 |
| **Всего** |  | 39,00 |

Модуль B. Клиентское программирование Single Page Application

# Введение

**Технологии этого модуля:** SPA

**Время на выполнение:** 3 часа

К вам обратилась автосервис «АвтоКлуб». В ее ближайших планах автоматизировать управление сервисом и им требуется информационная система. Пользователями системы являются сотрудники автосервиса. Основная задача системы состоит в учёте и управлении обслуживания клиентов автосервиса, а также формировании отчётной документации.

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

# Описание проекта и задач

## Общее описание проекта

Все пользователи системы подразделяются на три группы:

1. Администраторы;
2. Приемщики машин;
3. Автослесари.

Пользователи получают доступ к функциям ИС только после успешной авторизации.

Функционал администратора:

* Регистрация новых пользователей в системе;
* Создание/открытие/закрытие смен;
* Назначение/снятие работников со смены;
* Просмотр всех заказов определенной смены;
* Увольнение работников.

Функционал автослесаря:

* Просмотр заказов, принятых от клиентов;
* Изменение статуса заказа на «выполняется» и «выполнен».

Функционал приемщика машин:

* Создание нового заказа;
* Добавление/удаление работ из заказа;
* Просмотр всех принятых им заказов за смену;
* Изменение статуса заказа на «оплачен».

Ваша задача – реализовать клиентскую часть REST API (SPA).

## Общие требования к приложению

Все функции, кроме аутентификации доступны только авторизованным пользователям. Идентификацию пользователя организована посредством Bearer Token.

При попытке доступа к защищенным авторизацией функциям приложения должно переводить на домашнюю страницу

При выходе пользователя его также перенаправляет на главную страницу.

Все ошибки при действиях пользователя должны сопровождаться сообщениями об ошибках, которые приходят с сервера.

## Специфические требования к приложению

### Аутентификация

Пользователь может авторизоваться по двум обязательным полям:

* login;
* password.

### Выход

Функция предназначена для очистки значение токена пользователя и перенаправление его на главную страницу.

### Функционал администратора

Администратор может регистрировать новых пользователей в системе, с назначением им роли:

* name;
* login;
* password;
* photo\_file – не обязательное поле;
* role\_id – (1 – Администратор, 2 – Приемщик машин, 3 – Автослесарь).

Администратор имеет возможность просмотра списка всех сотрудников или одного.

Есть возможность уволить сотрудника (запись становиться не активной).

Создание смен – обязанность администратора. Указывается таймстамп начала и конца смены.

Администратор открывает/закрывает смены. Закрытую смену нельзя открыть.

Ему доступна функция добавления и удаления сотрудников со смены.

Дополнительно можно просмотреть список всех заказов.

### Функционал приемщика машин

Приемщик открывает заказ, добавляет в него работы.

Изменяет статусы заказов (1 – Принят, 4 – Оплачен).

Статус оплачен он может выставить после того, как автослесарь выполнит все работы по заказу (статус: Готов).

Можно просмотреть конкретный заказ или все заказы.

### Функционал автослесаря

Автослесарь может видеть список всех заказов и те, что уже имеют статус «Принят», может отправлять на выполнение.

Изменяет статусы заказа (2 – Выполняется, 3 – Готов).

Никаких функций ему больше не доступно.

## Инструкция для конкурсанта

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

Полную postman коллекцию смотрите в папке media.

## Система оценки

|  |  |  |
| --- | --- | --- |
| **Секция** | **Критерий** | **Сумма** |
| 1 | Организация работы и управление | 1,00 |
| 2 | Коммуникационные и межличностные навыки | 1,00 |
| 3 | Графический дизайн | 1,00 |
| 4 | Верстка страниц | 0,00 |
| 5 | Программирование на стороне клиента | 18,00 |
| 6 | Программирование на стороне сервера | 0,00 |
| 7 | Системы управления контентом | 0,00 |
| **Всего** |  | 21,00 |

Модуль C. Серверное программирование REST API

Введение

**Технологии этого модуля:** REST API

**Время на выполнение:** 3 часа

К вам обратилась автосервис «АвтоКлуб». В ее ближайших планах автоматизировать управление сервисом и им требуется информационная система. Пользователями системы являются сотрудники автосервиса. Основная задача системы состоит в учёте и управлении обслуживания клиентов автосервиса, а также формировании отчётной документации.

Так как заказчик ограничен в деньгах, была достигнута договоренность о разработке MVP (minimum viable product). На данном этапе ваша задача состоит в разработке минимального функционала REST API для этой информационной системы.

Заказчик планирует дальнейшее развитие проекта, поэтому использование фреймворков будет плюсом.

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

Описание проекта и задач

Общее описание проекта

Все пользователи системы подразделяются на три группы:

1. Администраторы;
2. Приемщики работ;
3. Автослесари.

Пользователи получают доступ к функциям ИС только после успешной авторизации.

Функционал администратора:

* Регистрация новых пользователей в системе;
* Назначение работников на смены;
* Просмотр всех заказов определенной смены.

Функционал автослесаря:

* Просмотр заказов, принятых от клиентов;
* Изменение статуса заказа на «выполняется» и «выполнен».

Функционал приемщика машин:

* Создание нового заказа;
* Добавление/удаление работ из заказа;
* Просмотр всех принятых им заказов за смену;
* Изменение статуса заказа на «оплачен».

Ваша задача – реализовать REST API заданной структуры.

В примерах будет использоваться переменная {{host}} которая обозначает адрес http://xxxxxx-m2.wsr.ru/, где xxxxxx - логин участника.

Общие требования к API

Api должно быть доступно с других доменов. Позаботьтесь о настройках правил CORS.

Все функции, кроме аутентификации доступны только авторизованным пользователям. Идентификацию пользователя организуйте посредством Bearer Token.

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

**Status:** 403

**Content-Type:** application/json

**Body:**

{

“error”: {

“code”: 403,

“message”: “Login failed”

}

}

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

**Status:** 403

**Content-Type:** application/json

**Body:**

{

“error”: {

“code”: 403,

“message”: “Forbidden for you”

}

}

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

{

“error”: {

“code”: <code>,

“message”: <message>,

“errors”: {

<key>: [ <error message>]

}

}

}

Специфические требования к API

Аутентификация

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

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/api-avtoclub/login  **Method:** POST  **Headers**  - Content-Type: application/json  **Body:**  {  “login”: “admin“,  “password“: “admin“  } | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  “data”: {  “user\_token”: <сгенерированный token>  }  }  Ошибки валидации полей  **Формат ответа из общих требований**  Неправильные логин или пароль  **Status:** 401  **Content-Type:** application/json  **Body:**  {  “error”: {  “code”: 401,  “message”: “Authentication failed”  }  } |

Выход

Запрос предназначен для очистки значения токена пользователя. Выход доступен только для авторизированного пользователя.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/api- avtoclub /logout  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  “data”: {  “message”: “logout”  }  } |

Функционал администратора

Просмотр списка всех сотрудников

Запрос предназначен для получения списка всех сотрудников автосервиса.

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {{host}}/api- avtoclub /users  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": [  {  "id": 1,  "name": "Bell",  "login": "admin",  "status": "working",  "group": "Администратор"  },  {  "id": 2,  "name": "Shana",  "login": "inspector",  "status": "working",  "group": "Приемщик"  },  {  "id": 3,  "name": "Estrella",  "login": "mechanic",  "status": "working",  "group": "Автослесарь"  }  ]  } |

Добавление сотрудника

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

* 1. Имя (name)
  2. Фамилия (login)
  3. Пароль (password)
  4. Роль (1 – Администратор, 2 – Приемщик машин, 3 – Автослесарь) (id)

Для добавления сотрудника используйте следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/ api- avtoclub / new\_user  **Method:** POST | **--------------------- Successful ----------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {  “data”: {  “code”: “QSAVE”  }  }  **----------------- Validation error -------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {  “error”: {  “code”: 422,  “message”: “Validation error”,  “errors”: {  <key>: <массив ошибок>  }  }  } |

Создание смены

Добавить смену можно нажав на кнопку “Добавить смену”. Для каждой смены необходимо заполнить следующие поля:

* 1. Название (work\_shift)
  2. Дата (date)

Для добавления смены используйте следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/ api- avtoclub / new\_shift  **Method:** POST | **--------------------- Successful ----------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {  “data”: {  “code”: “QSAVE”  }  }  **----------------- Validation error -------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {  “error”: {  “code”: 422,  “message”: “Validation error”,  “errors”: {  <key>: <массив ошибок>  }  }  } |

Добавление сотрудников на смену

Экран с выбором смены – на данном экране должна быть список смен и список сотрудников. Должна быть возможность выбрать сотрудника, а после для него смену. При нажатии на кнопку “Back” экран должен смениться на предыдущий без сохранения выбранных смен. При нажатии на кнопку “Confirm” информация о выбранных сменах должна быть сохранена, и пользователь должен быть перенаправлен на предыдущую страницу.

Для добавления сотрудника на смену нужно использовать следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/ api- avtoclub / work /shift  **Method:** PATCH  **Headers**  **- Content-Type:** application/json | **------------------------ Success --------------------------**  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "id": 2,  "name": "Shana",  "login": "inspector",  "group": "Приемщик"  "date": "2021-11-20",  " work\_shift": "1",  }  }  **---------------------- Validation error -----------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {  “error”: {  “code”: 422,  “message”: “Validation error”,  “errors”: {  <key>: <массив ошибок>  }  }  } |

Просмотр заказов конкретной смены

Для просмотра заказов конкретной смены нужно использовать следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/ api-avtoclub/work/{code}/  **Method:** GET | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "code": "1\_2021-11-20",  "item": 1,  "customer": Иванов,  "operations": {  "id\_ work": 1,  "name\_work": «Покраска»  },  "name": "Estrella"  “status\_work”: “Выполняется”  }, |

Функционал приемщика работ

Создание заказа

Добавить заказ можно нажав на кнопку “Добавить заказ”. Для добавления заказа необходимо заполнить следующие поля:

Заказчик (customer)

Работы (operations - Номер (id\_ work), Наименование работы (name\_work)

Имя автослесаря (name), выполняющего заказ

Статус заказа при создании автоматически присваивается “Выполняется”

Для добавления заказа используйте следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/ api- avtoclub / new\_work  **Method:** POST | **--------------------- Successful ----------------------**  **Status:** 201  **Content-Type:** application/json  **Body:**  {  “data”: {  “code”: “QSAVE”  }  }  **----------------- Validation error -------------------**  **Status:** 422  **Content-Type:** application/json  **Body:**  {  “error”: {  “code”: 422,  “message”: “Validation error”,  “errors”: {  <key>: <массив ошибок>  }  }  } |

Просмотр конкретного принятого заказа

Для просмотра конкретного заказа нужно использовать следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/ api-avtoclub/work/{code\_work}  **Method:** GET | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "code\_work": "Иванов\_2021-11-20",  "customer": Иванов,  "operations": {  "id\_ work ": 1,  "name\_work": «Покраска»  },  "name": "Estrella"  “status\_work”: “Выполняется”  }, |

Просмотр принятых заказов конкретной смены

Для просмотра заказов конкретной смены нужно использовать следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/ api-avtoclub/work/{code}/  **Method:** GET | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "code": "1\_2021-11-20",  "item": 1,  "customer": Иванов,  "operations": {  "id": 1,  "name\_work": «Покраска»  },  "name": "Estrella"  “status\_work”: “Выполняется”  }, |

Изменение статуса заказа

Удаление работ из заказа

Функционал автослесаря

Просмотр принятых заказов смены

Для просмотра заказов конкретной смены нужно использовать следующий запрос:

|  |  |
| --- | --- |
| Request | Response |
| **URL:** {host}/ api-avtoclub/work/{code}/  **Method:** GET | **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "code": "1\_2021-11-20",  "item": 1,  "customer": Иванов,  "operations": {  "id\_work": 1,  "name\_work": «Покраска»  },  "name": "Estrella"  “status\_work”: “Выполняется”  }, |

Изменение статуса заказа

Инструкция для конкурсанта

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

Форматы запросов и ответов, а также форматы дат и времени должен соответствовать примерам из задания.

Схема оценки

|  |  |  |
| --- | --- | --- |
| **Секция** | **Критерий** | **Сумма** |
| 1 | Организация работы и управление | 1,50 |
| 2 | Коммуникационные и межличностные навыки | 1,50 |
| 3 | Графический дизайн | 0,00 |
| 4 | Верстка страниц | 0,00 |
| 5 | Программирование на стороне клиента | 0,00 |
| 6 | Программирование на стороне сервера | 17,00 |
| 7 | Системы управления контентом | 0,00 |
| **Всего** |  | 20,00 |

**Модуль D. CMS WordPress**

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

**Время на выполнение**: 3 часа

К вам обратилась администрация Краснодарского края. Сайт краеведческого музея давно устарел и нуждается в обновлении. 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 музея 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 музее (секция);
  + Экспонаты (отдельная страница);
  + Контакты (секция);
  + Ростелеком (отдельная страница)
  + Минсвязи (Отдельная страница)
  + ОГБПОУ «ТТИТ» (Отдельная страница)

* *Слайдер* (не менее трех слайдов с использованием стороннего плагина).
* *О музее* (краткая информация о музее из медиа).
* *Интересные экспонаты* (от 6 штук, с возможностью перехода на отдельную страницу экспоната). Решение о публикации экспоната принимает администратор сайта, путем изменения признака необходимости вывода экспоната на главную страницу. Секция должна иметь ссылку на страницу со всеми экспонатами. Элементы карточки экспоната:
  + Главное изображение;
  + Краткое название;
  + Год выпуска;
* *Наши контакты* должна содержать:
  + Номер телефона;
  + Адрес музея;
  + Правила посещения;
  + Время работы;
* *Форма заявки* на посещение музея:
  + Дата посещения;
  + Время посещения;
  + Имя;
  + WhatsApp номер;
  + Кнопка отправки формы;
* *Карта (*заглушка в виде изображения из медиа).
* *Подвал сайта*:
  + Номер телефона;
  + Навигация по сайту

**Страница со всеми экспонатами**

На странице со всеми экспонатами отображаются все экспонаты с пагинацией и фильтрацией по категориям:

* Скелеты
* Шкуры, меха
* Обереги
* Орудия труда;
* Макеты домов;
* Славянская одежда;
* Макеты животных;

Карточка экспоната содержит:

* Главное изображение;
* Краткое название;
* Год выпуска;

**Страница экспоната**

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

* Главное изображение;
* Дополнительные изображения;
* Полное название;
* Категория;
* Краткое название;
* Год выпуска;
* Завод-изготовитель;
* Ссылки на дополнительные материалы (http external links);
* QR-код с ссылкой на текущую страницу (вставляется с помощью собственного плагина QRPage);
* Описание;

На странице экспоната необходимо предусмотреть возможность вывода на печать информационной таблички для дальнейшего размещения ее рядом с экспонатом в зале. Шаблон для печати:

**Краткое название выпуска**



**Полное название**

Описание



**Плагины**

Вам необходимо разработать два плагина.

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

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

**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.

**СИСТЕМА ОЦЕНКИ**

|  |  |  |
| --- | --- | --- |
| **Секция** | **Критерий** | **Сумма** |
| A | Организация работы и управление | 1,00 |
| B | Коммуникационные и межличностные навыки | 1,00 |
| C | Графический дизайн | 3,00 |
| D | Верстка страниц | 2,00 |
| E | Программирование на стороне клиента | 2,00 |
| F | Программирование на стороне сервера | 3,00 |
| G | Системы управления контентом | 8,00 |
| **Всего** |  | 20,00 |

1. **Критерии оценки.**

Таблица 2.

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