**КОНКУРСНОЕ ЗАДАНИЕ**

***V вузовского чемпионата Национального исследовательского Мордовского государственного университета им. Н.П. Огарёва по стандартам Ворлдскиллс***

**ПО КОМПЕТЕНЦИИ**

«Веб-технологии»

ДЛЯ ОСНОВНОЙ ВОЗРАСТНОЙ КАТЕГОРИИ

**17-35 лет**

*Конкурсное задание включает в себя следующие разделы:*

1. Формы участия в конкурсе.
2. Общее время на выполнение задания.
3. Задание для конкурса.
4. Модули задания и необходимое время.
5. Критерии оценки.
6. Приложения к Конкурсному заданию.
7. **ФОРМЫ УЧАСТИЯ В КОНКУРСЕ:** Индивидуальный конкурс.

2. **ОБЩЕЕ ВРЕМЯ НА ВЫПОЛНЕНИЕ ЗАДАНИЯ:** 16 ч.

3. **ЗАДАНИЕ ДЛЯ КОНКУРСА.**

Конкурсное задание состоит из четырёх модулей. Результат выполнения каждого модуля не влияет на выполнение задания других модулей.

4. **МОДУЛИ ЗАДАНИЯ И НЕОБХОДИМОЕ ВРЕМЯ** (Таблица 1)**.**

Таблица 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование модуля** | | **Соревновательный день (С1, С2, С3)** | **Время на задание** |
| **A** | Проектирование | С1 | 4 часа |
| **B** | Вёрстка и дизайн | С1 | 4 часа |
| **C** | Программирование на стороне клиента | С2 | 4 часа |
| **D** | Программирование на стороне сервера | С2 | 4 часа |

*Модуль A: Проектирование*

## ВВЕДЕНИЕ

К вам обратился заказчик, который знает, чего он хочет, но в силу своей профессии он не совсем компетентен в веб-технологиях на данный момент. Поэтому он не знает, как лучше сделать тот или иной функционал.

Ваша задача разработать ТЗ на основе требований заказчика. При разработке ТЗ вам нужно опираться на несколько факторов:

1. В дальнейшем разработчик должен успеть разработать проект на основе вашего ТЗ за 12 часов, 4 из которых у него будет на верстку и дизайн проекта и 8 на серверное и клиентское программирование.
2. Требования, описанные в ТЗ должны быть современными и удобными для пользователей.
3. ТЗ должно содержать: пояснительную часть, схему базы данных, модель предметной области, сценарии работы пользователя в информационной системе, макеты интерфейсов (UX-дизайн, от 5 штук), требования к:
   1. Фреймворкам и языкам программирования.
   2. Графическому оформлению.
   3. Срокам выполнения.
   4. Аппаратному-техническому обеспечению.
   5. Производительности системы (frontend, backend).
   6. Взаимодействию с внешними системами (в случае их наличия).
   7. Описание взаимодействия подсистем (компонентов).
   8. Информационной безопасности.
   9. Доступности системы для людей с ограниченными возможностями.
4. Оформление текста ТЗ:
   1. Шрифт: Обычный текст: Times New Roman. Обычный текст: 12 пт, Заголовки: 16 пт, 14 пт.
   2. Межстрочный интервал: 1.5
   3. Абзацный отступ: 1 см.
   4. Поля: слева 3 см, справа 1.5 см, сверху и снизу 2см.
5. Решения, описанные в тексте ТЗ, должны быть обоснованы.

## ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

«Центр социальной поддержки населения» (далее – Центр) является некоммерческой организацией социальной сферы. Центр создан для оказания услуг в сфере социальной поддержки, оказываемой:

* детям (различные пособия семьям с детьми; услуги, предоставляемые семьям и детям);
* многодетным семьям (выдача удостоверений многодетной семьи, различные пособия, денежные выплаты, оплата услуг ЖКХ и др.);
* ветеранам и инвалидам боевых действий (различные денежные выплаты, комплексы услуг);
* малоимущим семьям;
* гражданам за особые заслуги перед Отечеством;
* гражданам, имеющим право на льготы (учащимся, пенсионерам, работникам спецгосучреждений и др.);
* молодым семьям (программы улучшения жилищных условий и др.);

Деятельность Центра занятости осуществляется в соответствии с действующим законодательством Российской Федерации, Административными регламентами Федеральной службы по труду и занятости и иными нормативными правовыми актами Российской Федерации.

Сотрудники Центра обратились к вам с просьбой создать веб-сервис для организации событий Центра, направленных на обеспечение социальных гарантий в области занятости населения.

**Веб-сервис должен:**

- предоставлять полную информацию о видах социальной поддержки: категория граждан (несовершеннолетний, многодетная семья, малоимущая семья, гражданин, имеющий право на льготы, молодая семья), виды социальной поддержки (пособия, услуги (уход, временное проживание, трудоустройство), выплаты, государственные соц.программы), необходимая документация;

- позволять пользователям находить необходимую информацию по социальной поддержке;

- предоставлять администраторам возможность управлять пользователями;

**Основные ролевые группы и их состав:**

1. Пользователи сервиса:

- Незарегистрированный Посетитель. Пользователь, который не имеет учетную запись и не авторизовался в системе. Функциональные возможности: просмотр информационных страниц, взаимодействие с формами регистрации и авторизации.

- Зарегистрированный Посетитель. Пользователь, который имеет учетную запись в системе и авторизовался в системе. Функциональные возможности: просмотр текстовых информационных страниц, поиск видов социальной поддержки, подача заявки, редактирование данных профиля.

1. Пользователи системы управления:

- Сотрудник Центра. Функциональные возможности: регистрация на сервисе, авторизация в системе, добавление и редактирование видов социальной поддержки, настройка, добавление фотографий, просмотр списка зарегистрированных посетителей сервиса. Сотрудник Центра может добавлять информацию только после одобрения администратором. Информация о видах социальной поддержки, добавленная или отредактированная Сотрудником Центра, должна пройти согласование администратором.

- Администратор системы. Функциональные возможности: изменение роли Посетитель/Сотрудник, формирование списка заявок на конкретный вид социальной поддержки для рассылки информационных сообщений на электронную почту пользователей, просмотр и одобрение информации о видах социальной поддержки, получение статистики о Посетителе/Сотруднике (количество, оценка работы сотрудника, список отзывов о Центре).

Ваш проект должен корректно открываться со следующих устройствах:

- мобильный телефон 375 x 820px;

- компьютеры с шириной экрана от 1200px.

Следует обратить внимание, что дизайн приложения должен быть удобен и понятен для использования, соответствовать современным тенденциям, а также учитывать особенности платформ, для которых создается сайт. При разработке дизайна необходимо использовать брендбук Центра. Также необходимо позаботиться о защите проекта: от взлома, от несанкционированного доступа к административным функциям и т.д.

## ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Разработанное ТЗ должно быть сохранено на сервере в папке xxxx-m1, где xxxx – ваш логин, все дополнительные материалы, используемые в ТЗ должны быть так же сохранены в виде файлов в папке xxxx-m1/files.

*Модуль B: Вёрстка и дизайн*

## ВВЕДЕНИЕ

Вам предоставляет готовое ТЗ на основе которого вам необходимо сверстать все необходимые веб-страницы. На каждой странице должен присутствовать логотип Центра Занятости. Дизайн проекта должен быть в цветовой гамме Центра Занятости (см. брендбук)

Примените все свои навыки верстки чтобы сверстать страницы в соответствии с современными требованиями. Уделите внимание юзабилити, анимации и интерактивности, чтобы использование сервиса было удобными и простым.

В дальнейшем программисты будут реализовывать требуемый по ТЗ функционал опираясь на сверстанные вами веб-страницы.

## ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ниже приведен список веб-страниц и их структуры, которые вы должны сверстать:

* Страница регистрации – главная страница для неавторизованного пользователя
* Страница авторизации
* Главная страница
* Личный кабинет Посетителя
  + - Личные данные
    - Моя социальная поддержка
* Личный кабинет Сотрудника
  + - Личные данные
* Личный кабинет Администратора
  + - Меры социальной поддержки
    - Управление пользователями (Посетитель/Сотрудник)
* Страница создания Сотрудника
* Страница Виды социальной поддержки

Ваша верстка должна быть валидной.

## ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Сверстанные веб-страницы должны быть размещены на сервере в папке xxxx-m2, и доступны по адресу http://xxxx-m2.wsr.ru, где xxxx – ваш логин.

В процессе работы вы можете пользоваться следующими библиотеками и фреймворками:

* Bootstrap 4
* Bootstrap 5
* Tailwind CSS

*Модуль C: Программирование на стороне клиента*

## ВВЕДЕНИЕ

К вам обратился руководитель гостиничного стартапа. В ближайших планах – открытие нового отеля, и для бронирования номеров в этом отеле с официального сайта требуется система бронирования. Заказчик пожелал, чтобы обработка запросов на бронирование велась вручную менеджерами отеля. Пользователями системы бронирования являются менеджеры отеля, администратор, а также путешественники – гости отеля. Основная задача системы состоит в учёте и управлении заявками на бронирование номеров отеля.

Заказчик предоставляет вам полностью готовую верстку со всеми страницами и рабочее API. Вам необходимо использовать все имеющиеся навыки в клиентской разработке для создания Single Page Application, далее SPA.

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

## ОПИСАНИЕ ПРОЕКТА И ЗАДАЧИ

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

Для вашего удобства, во всех URL будет использоваться переменная {host} которая обозначает хост адрес API: http://wsr.ru/server

Ваше SPA должно состоять из следующих экранов:

* Домашний
* Бронирование
* Результат бронирования (результат поиска заявки по коду)
* Подробное описание категории номера
* Авторизации
* Управление
* Редактирование заявки (управление статусом)
* Добавление менеджера
* Добавление прайса

Приложение должно обладать следующим функционалом:

1. В шапке сайта располагается форма поиска заявки на бронирование по коду заявки. При отправке формы пользователю должен отобразиться экран с данными заявки. Получить данные заявки по введенному запросу можно с помощью следующего запроса:

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/booking/{code}  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "code": "eDkj4L7",  "status": "Ожидание",  "arr\_date": "2022-06-05",  "dep\_date": "2022-06-20",  "room\_category": "Одноместный номер",  "guests": [  {  "name": "TestCase1\_first\_name",  "surname": "TestCase1\_last\_name",  "patronymic": "TestCase1\_patronymic",  "birthday": "1990-02-20",  "gender": "м",  "document\_type": "Паспорт гражданина РФ",  "document\_number": "9922335577"  },  {  "name": "TestCase1\_first\_name",  "surname": "TestCase1\_last\_name",  "patronymic": "TestCase1\_patronymic",  "birthday": "1989-02-11",  "gender": "ж",  "document\_type": "Паспорт гражданина РФ",  "document\_number": "123456789"  }  ]  }  }  Попытка просмотреть заявку по несуществующему коду  **Status:** 403  **Content-Type:** application/json  **Body:**  {  "error": {  "code": 403,  "message": " The code does not exist "  }  } |

1. Домашний экран – на данном экране располагается предварительная форма бронирования со следующими полями:
   1. Дата заезда
   2. Дата отъезда
   3. Количество гостей (от 1 до 3)

При отправке формы пользователю должен перейти на экран основного бронирования.

1. Экран бронирования – данный экран отображается после перехода с секции бронирования на главной. На этой странице необходимо отобразить форму для сбора данных:

* Дата заезда (дата по умолчанию выбрана, значение переносится с предварительной формы бронирования)
* Дата отъезда (дата по умолчанию выбрана, значение переносится с предварительной формы бронирования)
* Выбор категории номера – радиокнопка, каждый элемент которой содержит
  + изображение,
  + код,
  + название,
  + стоимость (рассчитывается исходя из выбранных дат проживания),
  + площадь номера,
  + количество комнат,
  + количество мест в номере,
  + количество свободных номеров данной категории (рассчитывается исходя из выбранных дат проживания).
* Данные о гостях (для каждого гостя)
  + Фамилия
  + Имя
  + Отчество
  + Дата рождения
  + Пол (мужской/женский)
  + Тип документа, удостоверяющего личность
  + Номер документа
* Email
* Телефон
* Город

Для получения необходимых данных экрана необходимо выполнение следующих запросов:

а) Просмотр списка всех имеющихся категорий номеров размещения в отеле:

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/category  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": [          {              "id": 1,              "code": "SGL",              "name": "Одноместный номер",              "number\_of\_apartments": 5,              "guests": 1,              "square": 20,              "number\_of\_rooms": 1,              "description": "Однокомнатный номер для одного человека с одной кроватью. Небольшой размер номера и простая обстановка",              "photo": "http://wsr.ru/server/public/storage/categories/03.jpg"          },          {              "id": 4,              "code": "DBL",              "name": "Двухместный номер с одной большой кроватью",              "number\_of\_apartments": 5,              "guests": 2,              "square": 20,              "number\_of\_rooms": 1,              "description": "Однокомнатный номер для двух человек с одной двуспальной кроватью.\r\nСтандартный размер номера и интерьер.",              "photo": "http://wsr.ru/server/public/storage/categories/02.jpg"          },          {              "id": 5,              "code": "TWIN",              "name": "Двухместный номер с двумя кроватями",              "number\_of\_apartments": 3,              "guests": 2,              "square": 18,              "number\_of\_rooms": 1,              "description": "Однокомнатный номер для двух человек с двумя отдельными односпальными кроватями. Стандартный размер номера и интерьер.",              "photo": "http://wsr.ru/server/public/storage/categories/01.jpg "          },          {              "id": 6,              "code": "TRPL",              "name": "Трехместный номер",              "number\_of\_apartments": 2,              "guests": 3,              "square": 22,              "number\_of\_rooms": 1,              "description": "Однокомнатный номер для трех человек с одной двуспальной кроватью и одной односпальной (EXB).",              "photo": "http://wsr.ru/server/public/storage/categories/05.jpg "          },          {              "id": 7,              "code": "SUITE",              "name": "Люкс",              "number\_of\_apartments": 1,              "guests": 3,              "square": 40,              "number\_of\_rooms": 2,              "description": "Двухкомнатный номер (спальня и гостиная). Интерьер номера имеет роскошную обстановку и отделку. Номер отличается живописным видом из окна и удобным расположением. .). ",              "photo": "http://wsr.ru/server/public/storage/categories/06.jpg "          }      ]  } |

б) Сведения о стоимости проживания и количестве свободных номеров по датам

Запрос выводит массив объектов, в каждом из которых дата из заданного периода, категория размещения, стоимость размещения в день (value), количество свободных номеров для данной даты (available\_rooms). Количество свободных номеров для даты рассчитываются исходя из наличия заявок на эту дату в БД со статусом «Согласовано» и количества номеров заданной категории в отеле (number\_of\_apartments).

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/calculation  **Method:** GET  **Headers**  - Content-Type: application/json  **Body:**  {  "room\_category\_id":"1",  "arr\_date": "2022-01-30",  "dep\_date": "2022-02-02"  } | Успех  **Status:** 201  **Content-Type:** application/json  **Body:**  {      "data": [          {              "date": "30-06-2022",              "room\_category": "Одноместный номер",              "code": "SGL",              "value": 2220,              "available\_rooms": 5          },          {              "date": "31-06-2022",              "room\_category": "Одноместный номер",              "code": "SGL",              "value": 2220,              "available\_rooms": 5          },          {              "date": "01-06-2022",              "room\_category": "Одноместный номер",              "code": "SGL",              "value": 2500,              "available\_rooms": 4          }      ]  }  Ошибки валидации полей  **Формат ответа из общих требований** |

После отправки основной формы бронирования выполняется запрос. Созданной заявке на бронирование присваивается 7-значный случайным образом генерируемый код, а также устанавливается статус «Ожидание».

|  |  |
| --- | --- |
| **Request** | **Response** |
| URL: {{host}}/api/booking  Method: POST  Headers  - Content-Type: application/json  Body:  {     "arr\_date": "2021-12-05",     "dep\_date": "2021-12-20",     "room\_category\_id": 4,     "email": "admin@admin.ru",     "phone": "+7927175510",     "city": "Москва",     "guests": [          {              "name": "TestCase1\_first\_name",              "surname": "TestCase1\_last\_name",              "patronymic": "TestCase1\_patronymic",              "birthday": "1990-02-20",              "gender": "м",              "document\_type\_id": "1",              "document\_number": "9922335577"          },          {              "name": "TestCase2\_first\_name",              "surname ": "TestCase2\_last\_name",              "birthday": "1992-09-22",              "gender": "ж",              "document\_type\_id": "1",              "document\_number": "9922335576"          }      ]  } | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          "id": 67,          "code": "Q5ZtFI0",          "status": "Ожидание"      }  }  Ошибки валидации полей  **Формат ответа из общих требований**  Ошибка при попытке заселения количества гостей большее, чем мест в номере  {      "errors": "Guests in the booking more than there are beds in the room"  } |

1. Экран с данными о бронировании (результат поиска заявки по коду). Для вывода данных необходимо выполнение запроса:

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/booking/{code}  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "code": "eDkj4L7",  "status": "Ожидание",  "arr\_date": "2022-06-05",  "dep\_date": "2022-06-20",  "room\_category": "Одноместный номер",  "guests": [  {  "name": "TestCase1\_first\_name",  "surname": "TestCase1\_last\_name",  "patronymic": "TestCase1\_patronymic",  "birthday": "1990-02-20",  "gender": "м",  "document\_type": "Паспорт гражданина РФ",  "document\_number": "9922335577"  },  {  "name": "TestCase1\_first\_name",  "surname": "TestCase1\_last\_name",  "patronymic": "TestCase1\_patronymic",  "birthday": "1989-02-11",  "gender": "ж",  "document\_type": "Паспорт гражданина РФ",  "document\_number": "123456789"  }  ]  }  }  Попытка просмотреть заявку по несуществующему коду  **Status:** 403  **Content-Type:** application/json  **Body:**  {  "error": {  "code": 403,  "message": " The code does not exist "  }  } |

1. Экран подробного описания категории номера. Для получения данных необходимо выполнение запроса:

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/category/{{id}}**Method:** GET**Headers**- Content-Type: application/json | Успех **Status:** 200**Content-Type:** application/json**Body:** {      "data": {          "id": 1,          "name": "Одноместный номер",          "code": "SGL",          "number\_of\_apartments": 5,          "guests": 1,          "square": 20,          "number\_of\_rooms": 1,          "description": "Однокомнатный номер для одного человека с одной кроватью. Небольшой размер номера и простая обстановка",  "photo": "http://server.vippo.ru/storage/categories/03.jpg",          "service\_icons": [              {                  "id": 5,                  "name": "Free Wi-Fi",                  "code": "<svg xmlns>…</svg>"              },              {                  "id": 8,                  "name": "Набор полотенец",                  "code": "<svg xmlns>…</svg>"              },              {                  "id": 1,                  "name": "Односпальная кровать",                  "code": "<svg xmlns>…</svg>"              },              {                  "id": 23,                  "name": "Душевая кабина",                  "code": "<svg xmlns>…</svg>"              }          ]      }  } |

1. Экран авторизации. Для аутентификации пользователя предусмотрен следующий запрос:

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/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”  }  } |

1. Экран управления – неавторизованный пользователь не должен иметь возможность просмотра данного экрана. Переход на экран должен происходить при удачной аутентификации. Экран предназначен для управления заявками на бронирование номеров, добавления/увольнения менеджеров, добавления/изменения прайсов на проживание в номерах отеля. Для отображения данных на экране и выполнения необходимо выполнение следующих запросов:

а) просмотр всех заявок:

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/booking  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": [  {  "id": 63,  "code": "mqwBrk8",  "status": "Согласована",  "arr\_date": "04-06-2022",  "dep\_date": "06-06-2022",  "room\_category": "Двухместный номер с одной большой кроватью"  },  {  "id": 64,  "code": "auCn2FZ",  "status": "Ожидание",  "arr\_date": "05-06-2022",  "dep\_date": "20-06-2022",  "room\_category": "Двухместный номер с одной большой кроватью"  },  {  "id": 65,  "code": "Psofflm",  "status": "Ожидание",  "arr\_date": "05-06-2022",  "dep\_date": "20-06-2022",  "room\_category": "Двухместный номер с одной большой кроватью"  }  ]  } |

б) просмотр списка всех менеджеров

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/user  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": [  {  "id": 2,  "name": "Иван",  "surname": "Иванов",  "patronymic": "Иванович",  "login": "manager",  "photo\_file": null,  "status": "working"  },  {  "id": 3,  "name": "TestName\_01",  "surname": "TestSurName\_01",  "patronymic": "TestPatronymic\_01",  "login": "TestLogin\_01",  "photo\_file": "storage/photos/ccnwmnVKimOrlfQesOCMuwUVP0mvvQz7QR8FSCmz.jpg",  "status": "fired"  }  ]  } |

в) увольнение менеджера

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/user/{{id}}/to-dismiss  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "id": 6,  "status": "fired"  }  }  Попытка уволить уже уволенного менеджера  **Status:** 403  **Content-Type:** application/json  **Body:**  {      "error": {          "code": 403,          "message": "Forbidden. The user is already fired!"      }  } |

г) просмотр списка всех прайсов

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/price  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": [  {              "id": 53,              "code": "DBL-07-02-2022",              "room\_category": "Двухместный номер с одной большой кроватью",              "value": 1300,              "start\_date": "07-02-2022"          },          {              "id": 52,              "code": "SUITE-01-02-2022",              "room\_category": "Люкс",              "value": 1000,              "start\_date": "01-02-2022"          },          {              "id": 49,              "code": "SUITE-06-02-2022",              "room\_category": "Люкс",              "value": 1200,              "start\_date": "06-02-2022"          },          {              "id": 48,              "code": "SUITE-08-02-2022",              "room\_category": "Люкс",              "value": 1300,              "start\_date": "08-02-2022"          },  ]  } |

д) изменение прайса для категории

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/price/{id}  **Method:** PATCH  **Headers**  - Content-Type: application/json  **Body:**  {  "value": "1200"  } | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "id": “25”,  "code": "single-01-03-2022" ,  "room\_category": "Одноместный",  "value": "1200",  "start\_date": "01-03-2022"  }  } |

1. Экран редактирования заявки на бронирование (изменение статуса заявки)

Запрос предназначен для изменения статуса заявки. При отправке запроса необходимо отправить объект со следующим полем:

● booking\_status\_id – обязательное поле, id статуса заявки (1 - «Ожидание», 2 - «Согласована», 3 - «Отклонена»)

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/booking/{{id}}  **Method:** PATCH  **Headers**  - Content-Type: application/json  **Body:**  {  "booking\_status\_id":"2"  } | Успех  **Status:** 201  **Content-Type:** application/json  **Body:**  {  "data": {  "id": 67,  "code": "Q5ZtFI0",  "status": "Согласована",  "arr\_date": "05-06-2022",  "dep\_date": "20-06-2022",  "room\_category": "Двухместный номер с одной большой кроватью"  }  }  Ошибки валидации полей  **Формат ответа из общих требований** |

1. Экран добавления менеджера содержит форму. При отправке запроса необходимо передать FormData со следующими полями:

* name – обязательное поле, строка;
* surname – не обязательное поле, строка;
* patronymic – не обязательное поле, строка;
* login – обязательное и уникальное поле, строка;
* password – обязательное поле, строка;
* photo\_file – файл с фото нового сотрудника (загружается в папку storage/photos), не обязательное поле, файл jpeg, png.

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/user  **Method:** POST  **Headers**  - Content-Type: multipart/form-data  **FormData:**  “name”: “Ivan”,  “login”: “Ivanov”,  “password”: “paSSword” | Успех  **Status:** 201  **Content-Type:** application/json  **Body:**  {  "data": {  "id": 6,  "status": "created"  }  }  Ошибки валидации полей  **Формат ответа из общих требований** |

1. Экран добавления нового прайса. При отправке формы с экрана выполняется запрос:

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/price  **Method:** GET  **Headers**  - Content-Type: application/json  **Body:**  {  "room\_category\_id": "1",  "value": "1000",  "start\_date": "2021-12-05"  } | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "code": "single-01-03-2022" ,  "status": "created"  }  }  Попытка создать прайс с существующим code  **Status:** 403  **Content-Type:** application/json  **Body:**  {  "error": {  "code": 403,  "message": " The code single-01-03-2022 already exist"  }  } |

1. Должна быть возможность выйти из личного кабинета.
2. Необходимо позаботиться об уведомлении пользователей о каких-либо действиях (ошибки валидации, подтверждения и т.п).
3. Навигация в SPA должна соответствовать логике приложения, а также предоставляемому Layout.

## Инструкция для конкурсанта

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

*Модуль D: Программирование на стороне сервера*

## Введение

К вам обратился руководитель гостиничного стартапа. В ближайших планах – открытие нового отеля, и для бронирования номеров в этом отеле с официального сайта требуется система бронирования. Заказчик пожелал, чтобы обработка запросов на бронирование велась вручную менеджерами отеля. Пользователями системы бронирования являются менеджеры отеля, администратор, а также путешественники – гости отеля. Основная задача системы состоит в учёте и управлении заявками на бронирование номеров отеля.

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 системы бронирования

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

В медиафайлах вам предоставляется sql дамп с готовой базой данных. База данных уже содержит набор данных, которые НЕ должны быть изменены! Структуру БД менять также запрещается. Любое изменение или удаление предоставленных данных в БД будет влиять на вашу оценку.

## Описание проекта и задач

## Общее описание проекта

Все пользователи системы подразделяются на три группы:

1. Гости;
2. Администратор (авторизованный пользователь);
3. Менеджеры (авторизованные пользователи).

Администратор и менеджеры получают доступ к функциям системы только после успешной авторизации.

Функционал гостя:

* Просмотр списка имеющихся категорий номеров размещения в отеле;
* Просмотр категории номера размещения с описанием;
* Просмотр цен размещения для заданной категории номера;
* Получение сведений о свободных номерах и стоимости размещения;
* Создание новой заявки на бронирование;
* Просмотр заявки по персональному коду;

Функционал менеджера (доступно после авторизации):

* Просмотр заявок на бронирование;
* Изменение статуса заявки на «согласована» или «отклонена»;

Функционал администратора (доступно после авторизации):

* Регистрация новых менеджеров в системе;
* Увольнение менеджеров;
* Весь функционал менеджера;
* Просмотр, добавление, изменение цен на проживание.

Ваша задача – реализовать REST API заданной структуры.

В примерах будет использоваться переменная {{ host }} которая обозначает адрес http://xxxxxx-m4.wsr.ru/hotel-api, где xxxxxx - логин участника.

## Общие требования к API

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

Идентификацию пользователей 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>]

}

}

}

Обратите внимание, что вместо <code> и <message> необходимо указывать соответствующее значение, определенное в описании ответа на соответствующий запрос. В свойстве error.errors необходимо перечислить те свойства, которые не прошли валидацию, а в их значениях указать массив с ошибками валидации.

Например если отправить пустой запрос на сервер, где проверяется следующая валидация:

* phone – обязательно поле
* password – обязательное поле

то тело ответа будет следующим:

{

“error”: {

“code”: 422,

“message”: “Validation error”,

“errors”: {

phone: [ “field phone can not be blank” ],

password: [ “field password can not be blank” ]

}

}

}

Примите во внимание, что code и message могут быть определены иначе, если в запросе указано иное. В значениях свойств errors вы можете использовать любые сообщения об ошибках (если не указана конкретная ошибка), но они должны описывать возникшую проблему.

## 

## 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/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 значение токена пользователя.

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

### 

### **Функционал гостя**

#### Просмотр списка всех имеющихся категорий номеров размещения в отеле

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/booking/category  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": [          {              "id": 1,              "code": "SGL",              "name": "Одноместный номер",              "number\_of\_apartments": 5,              "guests": 1,              "square": 20,              "number\_of\_rooms": 1,              "description": "Однокомнатный номер для одного человека с одной кроватью. Небольшой размер номера и простая обстановка",              "photo": "http://wsr.ru/server/public/storage/categories/03.jpg"          },          {              "id": 4,              "code": "DBL",              "name": "Двухместный номер с одной большой кроватью",              "number\_of\_apartments": 5,              "guests": 2,              "square": 20,              "number\_of\_rooms": 1,              "description": "Однокомнатный номер для двух человек с одной двуспальной кроватью.\r\nСтандартный размер номера и интерьер.",              "photo": "http://wsr.ru/server/public/storage/categories/02.jpg"          },          {              "id": 5,              "code": "TWIN",              "name": "Двухместный номер с двумя кроватями",              "number\_of\_apartments": 3,              "guests": 2,              "square": 18,              "number\_of\_rooms": 1,              "description": "Однокомнатный номер для двух человек с двумя отдельными односпальными кроватями. Стандартный размер номера и интерьер.",              "photo": "http://wsr.ru/server/public/storage/categories/01.jpg"          },          {              "id": 6,              "code": "TRPL",              "name": "Трехместный номер",              "number\_of\_apartments": 2,              "guests": 3,              "square": 22,              "number\_of\_rooms": 1,              "description": "Однокомнатный номер для трех человек с одной двуспальной кроватью и одной односпальной (EXB).",              "photo": "http://wsr.ru/server/public/storage/categories/05.jpg"          },          {              "id": 7,              "code": "SUITE",              "name": "Люкс",              "number\_of\_apartments": 1,              "guests": 3,              "square": 40,              "number\_of\_rooms": 2,              "description": "Двухкомнатный номер (спальня и гостиная). Интерьер номера имеет роскошную обстановку и отделку. Номер отличается живописным видом из окна и удобным расположением. .). ",              "photo": "http://wsr.ru/server/public/storage/categories/06.jpg"          }      ]  } |

### *Просмотр одной категории номера*

Запрос предназначен для просмотра описания заданной категории номера с перечислением сервисных иконок, характеризующих категорию номера в формате SVG.

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/category/{{id}}**Method:** GET**Headers**- Content-Type: application/json | Успех **Status:** 200**Content-Type:** application/json**Body:** {      "data": {          "id": 1,          "name": "Одноместный номер",          "code": "SGL",          "number\_of\_apartments": 5,          "guests": 1,          "square": 20,          "number\_of\_rooms": 1,          "description": "Однокомнатный номер для одного человека с одной кроватью. Небольшой размер номера и простая обстановка",  "photo": "http://server.vippo.ru/storage/categories/03.jpg",          "service\_icons": [              {                  "id": 5,                  "name": "Free Wi-Fi",                  "code": "<svg xmlns>…</svg>"              },              {                  "id": 8,                  "name": "Набор полотенец",                  "code": "<svg xmlns>…</svg>"              },              {                  "id": 1,                  "name": "Односпальная кровать",                  "code": "<svg xmlns>…</svg>"              },              {                  "id": 23,                  "name": "Душевая кабина",                  "code": "<svg xmlns>…</svg>"              }          ]      }  } |

#### Сведения о стоимости проживания и количестве свободных номеров по датам

Запрос выводит массив объектов, в каждом из которых дата из заданного периода, категория размещения, стоимость размещения в день (value), количество свободных номеров для данной даты (available\_rooms). Количество свободных номеров для даты рассчитываются исходя из наличия заявок на эту дату в БД со статусом «Согласовано» и количества номеров заданной категории в отеле (number\_of\_apartments). Последняя дата периода в массив не включается. При отправке запроса необходимо отправить объект со следующими полями:

* room\_category\_id – обязательное поле, id категории номера размещения;
* arr\_date – обязательное поле, дата заезда, не должна быть раньше, чем текущая дата, формат Y-m-d;
* dep\_date – обязательное поле, дата отъезда, должна быть больше, чем дата заезда , формат Y-m-d.

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/calculation  **Method:** GET  **Headers**  - Content-Type: application/json  **Body:**  {  "room\_category\_id":"1",  "arr\_date": "2022-06-30",  "dep\_date": "2022-07-02"  } | Успех  **Status:** 201  **Content-Type:** application/json  **Body:**  {      "data": [          {              "date": "30-01-2022",              "room\_category": "Одноместный номер",              "code": "SGL",              "value": 2220,              "available\_rooms": 5          },          {              "date": "31-01-2022",              "room\_category": "Одноместный номер",              "code": "SGL",              "value": 2220,              "available\_rooms": 5          },          {              "date": "01-02-2022",              "room\_category": "Одноместный номер",              "code": "SGL",              "value": 2500,              "available\_rooms": 4          }      ]  }  Ошибки валидации полей  **Формат ответа из общих требований** |

#### Создание заявки на бронирование

Запрос предназначен для создания заявки на бронирование. Созданной заявке присваивается 7-значный случайным образом генерируемый код, а также устанавливается статус «Ожидание». При отправке запроса необходимо отправить объект со следующими полями:

* room\_category\_id – обязательное поле, id категории номера размещения;
* arr\_date – обязательное поле, дата заезда, не должна быть раньше, чем текущая дата, формат Y-m-d;
* dep\_date – обязательное поле, дата отъезда, не должна быть раньше даты заезда и не равна дате заезда, формат Y-m-d;
* email – обязательное поле, адрес электронной почты, применяется правило валидации email;
* phone – обязательное поле, номер телефона;
* city – обязательное поле, город проживания.

Свойство объекта guests является массивом из объектов из личных данных гостей, которые будут проживать в номере:

* name – обязательное поле, имя;
* birthday – обязательное поле, дата рождения, формат Y-m-d;
* gender – обязательное поле, пол гостя («м» / «ж»);
* document\_type\_id – обязательное поле, id типа документа;
* document\_number – обязательное поле, номер документа.

|  |  |
| --- | --- |
| **Request** | **Response** |
| URL: {{host}}/api/booking  Method: POST  Headers  - Content-Type: application/json  Body:  {     "arr\_date": "2022-06-05",     "dep\_date": "2022-06-20",     "room\_category\_id": 4,     "email": "admin@admin.ru",     "phone": "+7927175510",     "city": "Москва",     "guests": [          {              "name": "TestCase1\_first\_name",              "surname": "TestCase1\_last\_name",              "patronymic": "TestCase1\_patronymic",              "birthday": "1990-02-20",              "gender": "м",              "document\_type\_id": "1",              "document\_number": "9922335577"          },          {              "name": "TestCase2\_first\_name",              "surname ": "TestCase2\_last\_name",              "birthday": "1992-09-22",              "gender": "ж",              "document\_type\_id": "1",              "document\_number": "9922335576"          }      ]  } | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {      "data": {          "id": 67,          "code": "Q5ZtFI0",          "status": "Ожидание"      }  }  Ошибки валидации полей  **Формат ответа из общих требований**  Ошибка при попытке заселения количества гостей большее, чем мест в номере  {      "errors": "Guests in the booking more than there are beds in the room"  } |

#### Просмотр заявки на бронирование

Запрос предназначен для просмотра заявки на бронирование по ранее полученному коду

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/booking/{code}  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "code": "eDkj4L7",  "status": "Ожидание",  "arr\_date": "2022-06-05",  "dep\_date": "2022-06-20",  "room\_category": "Одноместный номер",  "guests": [  {  "name": "TestCase1\_first\_name",  "surname": "TestCase1\_last\_name",  "patronymic": "TestCase1\_patronymic",  "birthday": "1990-02-20",  "gender": "м",  "document\_type": "Паспорт гражданина РФ",  "document\_number": "9922335577"  },  {  "name": "TestCase1\_first\_name",  "surname": "TestCase1\_last\_name",  "patronymic": "TestCase1\_patronymic",  "birthday": "1989-02-11",  "gender": "ж",  "document\_type": "Паспорт гражданина РФ",  "document\_number": "123456789"  }  ]  }  }  Попытка просмотреть заявку по несуществующему коду  **Status:** 403  **Content-Type:** application/json  **Body:**  {  "error": {  "code": 403,  "message": " The code does not exist "  }  } |

### **Функционал менеджера**

#### Просмотр всех заявок

Запрос предназначен для получения списка всех заявок.

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/booking  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": [  {  "id": 63,  "code": "mqwBrk8",  "status": "Согласована",  "arr\_date": "04-06-2022",  "dep\_date": "06-06-2022",  "room\_category": "Двухместный номер с одной большой кроватью"  },  {  "id": 64,  "code": "auCn2FZ",  "status": "Ожидание",  "arr\_date": "05-06-2022",  "dep\_date": "20-06-2022",  "room\_category": "Двухместный номер с одной большой кроватью"  },  {  "id": 65,  "code": "Psofflm",  "status": "Ожидание",  "arr\_date": "05-06-2022",  "dep\_date": "20-06-2022",  "room\_category": "Двухместный номер с одной большой кроватью"  }  ]  } |

#### Изменения статуса заявки

Запрос предназначен для изменения статуса заявки. При отправке запроса необходимо отправить объект со следующим полем:

* booking\_status\_id – обязательное поле, id статуса заявки («Ожидание», «Согласована», «Отклонена»)

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/booking/{{id}}  **Method:** PATCH  **Headers**  - Content-Type: application/json  **Body:**  {  "booking\_status\_id":"2"  } | Успех  **Status:** 201  **Content-Type:** application/json  **Body:**  {  "data": {  "id": 67,  "code": "Q5ZtFI0",  "status": "Согласована",  "arr\_date": "05-06-2022",  "dep\_date": "20-06-2022",  "room\_category": "Двухместный номер с одной большой кроватью"  }  }  Ошибки валидации полей  **Формат ответа из общих требований** |

### 

### **Функционал администратора**

#### Просмотр списка всех менеджеров

Запрос предназначен для получения списка всех менеджеров отеля.

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/user  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": [  {  "id": 2,  "name": "Иван",  "surname": "Иванов",  "patronymic": "Иванович",  "login": "manager",  "photo\_file": null,  "status": "working"  },  {  "id": 3,  "name": "TestName\_01",  "surname": "TestSurName\_01",  "patronymic": "TestPatronymic\_01",  "login": "TestLogin\_01",  "photo\_file": "storage/photos/ccnwmnVKimOrlfQesOCMuwUVP0mvvQz7QR8FSCmz.jpg",  "status": "fired"  }  ]  } |

#### 

#### Добавление менеджера

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

* name – обязательное поле, строка;
* surname – не обязательное поле, строка;
* patronymic – не обязательное поле, строка;
* login – обязательное и уникальное поле, строка;
* password – обязательное поле, строка;
* photo\_file – файл с фото нового сотрудника (загружается в папку storage/photos), не обязательное поле, файл jpeg, png.

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/user  **Method:** POST  **Headers**  - Content-Type: multipart/form-data  **FormData:**  “name”: “Ivan”,  “login”: “Ivanov”,  “password”: “paSSword” | Успех  **Status:** 201  **Content-Type:** application/json  **Body:**  {  "data": {  "id": 6,  "status": "created"  }  }  Ошибки валидации полей  **Формат ответа из общих требований** |

#### 

#### Увольнение менеджера

Запрос предназначен для увольнения менеджера.

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/user/{{id}}/to-dismiss  **Method:** GET | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "id": 6,  "status": "fired"  }  }  Попытка уволить уже уволенного менеджера  **Status:** 403  **Content-Type:** application/json  **Body:**  {      "error": {          "code": 403,          "message": "Forbidden. The user is already fired!"      }  } |

#### Создание прайса для категории

Запрос предназначен для создания прайса для определенной категории номера размещения. При создании генерируется code, следующего формата: "{КОД категории номера размещения} – {Дата начала действия прайса в формате d-m-Y}". При отправке запроса необходимо отправить объект со следующими полями:

* room\_category\_id – обязательное поле, id категории номера размещения;
* value – обязательное поле, стоимость проживания в номере в сутки, формат Y-m-d;
* start\_date – обязательное поле, дата начала действия прайса, формат Y-m-d.

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/price  **Method:** GET  **Headers**  - Content-Type: application/json  **Body:**  {  "room\_category\_id": "1",  "value": "1000",  "start\_date": "2021-12-05"  } | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "code": "single-01-03-2022" ,  "status": "created"  }  }  Попытка создать прайс с существующим code  **Status:** 403  **Content-Type:** application/json  **Body:**  {  "error": {  "code": 403,  "message": " The code single-01-03-2022 already exist"  }  } |

#### Изменение прайса для категории

Запрос предназначен для изменения суммы прайса с идентификатором {id}. При отправке запроса необходимо отправить объект со следующими полями:

* value – обязательное поле, стоимость проживания в номере в сутки.

|  |  |
| --- | --- |
| **Request** | **Response** |
| **URL:** {{host}}/api/price/{id}  **Method:** PATCH  **Headers**  - Content-Type: application/json  **Body:**  {  "value": "1200"  } | Успех  **Status:** 200  **Content-Type:** application/json  **Body:**  {  "data": {  "id": “25”,  "code": "single-01-03-2022" ,  "room\_category": "Одноместный",  "value": "1200",  "start\_date": "01-03-2022"  }  } |

## Инструкция для конкурсанта

Разработанное API должно быть доступно по адресу http://xxxxxx-m4.wsr.ru/api, где xxxxxx - логин участника (указан на индивидуальной карточке).

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

**5. КРИТЕРИИ ОЦЕНКИ**.

Таблица 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Критерий** | | **Баллы** | | |
| **Судейские аспекты** | **Объективная оценка** | **Общая оценка** |
| **A** | **Организация работы и управление** | 2.00 | 4.00 | 6.00 |
| **B** | **Коммуникация и навыки межличностного общения** | 4.00 | 2.00 | 6.00 |
| **C** | **Дизайн** | 16.00 | 4.00 | 20.00 |
| **D** | **Вёрстка** | 0.00 | 20.00 | 20.00 |
| **E** | **Программирование на стороне клиента** | 0.00 | 20.00 | 20.00 |
| **F** | **Программирование на стороне сервера** | 0.00 | 20.00 | 20.00 |
| **G** | **Проектирование** | 6.00 | 2.00 | 8.00 |
| **Итого** | | 28.00 | 72.00 | 100.00 |

**6. Приложения к заданию.**

* Модуль B: ***ТЗ.docx, брендбук.pdf, media\_m2.zip***
* Модуль C: ***media\_m3.zip***
* Модуль D: ***media\_m4.zip***