ТИПОВОЕ КОНКУРСНОЕ ЗАДАНИЕ

ДЛЯ РЕГИОНАЛЬНЫХ ЧЕМПИОНАТОВ

чемпионатного цикла 2021/2022

компетенции

«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.

4. 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 А. 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 «BusWorld» – новая и энергичная компания-перевозчик, предоставляющая услуги пассажирских перевозок. Главная цель компании – развить связность между городами путем организации маршрутов на непостоянной основе по мере накопления запросов на маршрут.

Основные принципы работы сервиса:

1. Выбрать удобную дату поездки.
2. Накопить больше 50% бронирований на рейс.
3. Получить снижение стоимости.

Призыв: Мы сближаем города, вы живете там, где хотите.

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 компании, в основном, пользуются люди в возрасте от 18 до 65 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.

Ваша задача – создать следующие страницы веб-сайта:

- Главная страница - Landing Page
- Страница входа в личный кабинет
- Страница регистрации в личном кабинете
- Страница личного кабинета
- Страница с результатами поиска
- Страница бронирования
- Страница управления бронированием
- Страница выбора мест в салоне воздушного судна

Главная страница (Landing Page)

Главная страница должна содержать следующие блоки:

- Шапка сайта
    - Логотип компании
    - Меню навигации
        - Поиск рейса
        - Акции
        - Личный кабинет
- Форма поиска. Должна содержать следующие поля ввода:
    - Откуда – город отправления

- Куда – город назначения
- Туда – дата выезда туда
- Обратно – дата выезда обратно
- Количество пассажиров (от 1 до 25 включительно)
- Кнопка для поиска билетов

● Секция(и) описания принципа накопления бронирований для осуществления перезда с целью снижения его стоимости. Три принципа необходимо визуализировать при помощи анимированной инфографики на основе вводимых демо данных посетителем. Например, выбор условной даты, ввод числа бронирований и демонстрация изменения стоимости и вероятности выезда.

● Акции (список акций доступен в медиафайлах). Каждая акция должна содержать следующую информацию:

- Изображение
- Название акции
- Краткое описание акции
- Кнопка для просмотра акции
- Кнопка для участия в акции

● Форма для подписки на закрытые акции. Должна содержать следующие поля:

- Поле для ввода телефона
- Кнопка для подписки
    - Секция о доверии компании.
    - Подвал сайта
- Навигация (дублируется)
- Телефон 8 (800) 100-10-10

Страница с результатами поиска

Попасть на эту страницу можно с главной (из формы поиска рейсов). На этой странице необходимо отобразить все найденные рейсы (туда и обратно), а именно:

- Номер рейса
- Марка автобуса
- Дата отправления
- Время отправления
- Дата прибытия
- Время прибытия
- Время в пути
- Стоимость
- Вероятность выезда

Вероятность выезда - это визуальное отображение зависящие от заполненности рейса в указанную дату. Если автобус заполнен на половину (22 из 44 пассажиров), то вероятность выезда - 100%. Если автобус заполнен на четверть (11 из 44 пассажиров), то вероятность выезда - 50% и т.д.

Предусмотрите способ для выбора понравившихся рейсов (один рейс “туда” и один рейс “обратно”) и кнопку для перехода к бронированию.

Страница бронирования

Данная страница отображается после нажатия кнопки бронирования. На этой странице необходимо отобразить:

- Данные о рейсах (для рейса «туда» и рейса «обратно»):

- Номер рейса
- Название города выезда
- Адрес места отправления
- Дата выезда
- Время выезда
- Название города назначения
- Адрес места назначения
- Время прибытия
- Стоимость

- Форма для сбора данных о пассажирах с такими полями для каждого пассажира:
  - Фамилия
  - Имя
  - Отчество
  - Дата рождения
  - Серия и номер паспорта
- Кнопка для добавления еще одного пассажира в бронирование
- Кнопка для удаления пассажира
- Финальная стоимость с учётом стоимости рейсов и количества пассажиров
- Кнопка для оформления бронирования

Страница регистрации в личном кабинете

На этой странице вам необходимо сделать форму со следующими полями:

- Фамилия

- Имя
- Отчество
- Дата рождения
- Серия и номер паспорта
- Телефон
- Email
- 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 личного кабинета

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

- Фамилия
- Имя
- Отчество
- Кнопка выхода из личного кабинета

- Количество выполненных маршрутов
- Предстоящие бронирования. Каждое бронирование должно отражать следующую информацию:
    - Код бронирования
    - Название города выезда и места отправления
    - Дата и время выезда
    - Название города назначения и места прибытия
    - Дата и время прибытия
    - Время в пути
    - Вероятность выезда
    - Стоимость
    - Кнопка управления бронированием

Страница управления бронированием

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

- Информация о бронировании:
    - Код бронирования (пример AFADS)
    - Стоимость бронирования
- Информация о маршруте (для каждого рейса):
    - Номер рейса
    - Марка автобуса
    - Название города выезда и места отправления
    - Дата и время выезда

- Название города назначения и места прибытия
- Дата и время прибытия
- Время в пути
- Вероятность выезда

- Информация о пассажирах (для каждого пассажира):
  - Фамилия Имя Отчество
  - Дата рождения
  - Серия и номер паспорта
  - Место в салоне туда
  - Место в салоне обратно
- Кнопка для выбора места

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

Сохраните дизайны и верстку страниц со следующими именами:

- Главная страница - Landing Page – index.png, index.html
- Страница входа в личный кабинет – login.png, login.html
- Страница регистрации в личном кабинете – register.png, register.html
- Страница личного кабинета – profile.png, profile.html
- Страница с результатами поиска – search.png, search.html
- Страница бронирования – booking.png, booking.html
- Страница управления бронированием – booking_management.png, booking_management.html
- Страница выбора мест в салоне – seat.png, seat.html

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

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

Модуль B. Серверное программирование REST API

Технологии этого модуля: REST API
Время на выполнение: 3 часа

Ваша задача – реализовать REST API, которое будет отвечать требованиям заказчика.

Для вашего удобства, во всех URL будет использоваться переменная {host} которая обозначает адрес http://xxxxxx-m2.wsr.ru/, где xxxxxx - логин участника.

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

```
{
  “error”: {
    “code”: <code>,
    “message”: <message>,
    “errors”: {
       <key>: [ <error message>]
    }
  }
}
```

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

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

- phone – обязательно поле
- password – обязательное поле

то тело ответа должно быть следующим

```
{
  “error”: {
    “code”: 422,
    “message”: “Validation error”,
    “errors”: {
       phone: [ “field phone can not be blank” ],
       password: [ “field password can not be blank” ]
    }
  }
}
```

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

Регистрация

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

- first_name – обязательное поле, строка
- last_name – обязательное поле, строка
- phone – обязательное и уникальное поле, строка

- document_number – обязательное, строка из 10 цифр (может быть с ведущим нулем)
- password – обязательное поле, строка

Request	Response

URL: {host}/api/register
Method: POST
 	
Headers
- Content-Type: application/json
 	
Body:
{
"first_name": "Иван",
"last_name": "Петров",
"phone": "89001234568",
"document_number": "1224567890",
"password": "paSSword"
}
	---------- **Successful** ----------
	Status: 204
	
	---------- **Validation error** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 422,
	"message": "Validation error",
	"errors": {
	<key>: <массив ошибок>
	}
	}
	}

Аутентификация

Запрос для аутентификации пользователя в системе. При отправке запроса необходимо передать объект с телефоном 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:
{
	---------- **Successful** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"token": <сгенерированный token>
	}

"phone": "89001234567",
"password": "paSSword"
}
	}
	
	---------- **Validation error** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 422,
	"message": "Validation error",
	"errors": {
	<key>: <массив ошибок>
	}
	}
	}
	
	---------- **Unauthorized** ----------
	Status: 401
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 401,
	"message": "Unauthorized",
	"errors": {
	"phone": ["Неверный номер телефона или пароль"]
	}
	}
	}

Список автовокзалов

Запрос на поиск автовокзала по названию города или по коду. Поиск без учета регистра.

При отправке запроса обязательно нужно передать параметр query, который может содержать одно из следующих значений:

- название города (полное название или часть названия)
- название автовокзала (полное название или часть названия)

Все следующие запросы должны вернуть в результатах поиска вокзала Иркутск, т.к. все варианты подходят:

- GET {host}/api/station?query=рку
- GET {host}/api/station?query=ИРКУТСК
- GET {host}/api/station?query=ИРкутск

Request	Response

URL: {host}/api/station
Method: GET
 	
Query string (GET parameters):
- query
	---------- Successful ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"items": [
	{
	"city": "Иркутск",
	"name": "Иркутск",
	"code": "395"
	}
	]
	}
	}
	
	---------- No results ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"items": []
	}
	}

Поиск рейсов

Запрос на поиск рейсов по указанным параметрам. Должна быть возможность передать следующие GET параметры:

- from –код вокзала выезда: обязательно, должен существовать
- to – код вокзала назначения: обязательно, должен существовать
- date1 – дата выезда туда: обязательно, в формате YYYY-MM-DD
- date2 – дата выезда обратно: не обязательно, в формате YYYY-MM-DD
- passengers – количество пассажиров (от 1 до 25 включительно): обязательно

В ответе на запрос должен быть список найденных рейсов из from в to, на которые еще остались места в заданные даты.

В поле data.trip_to должны быть рейсы из from в to.

Если указана дата возвращения (data2), то в поле data.trip_back должны быть обратные рейсы (из to в from), а иначе пустой массив.

В базе данных вам предоставлены рейсы и вокзалы. Дата рейсов не указана, это означает, что рейсы осуществляются ежедневно.

Request	Response

URL: {host}/api/trip
Method: GET
 	
Query string (GET parameters):
- from *(395)*
- to *(3953)*
- date1 *(2021-10-01)*
- date2 *(2021-10-13)*
- passengers *(2)*
	---------- **Successful** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"trip_to": [
	{
	"trip_id": 2,
	"trip_code": "FP1200",
	"from": {
	"city": "Братск",
	"station": " Братск ",
	"code": "3952",
	"date": "2021-10-01",
	"time": "12:00"
	},
	"to": {
	"city": "Иркутск",
	"station": " Иркутск ",
	"code": "395",
	"date": "2021-10-01",
	"time": "13:35"
	},
	"cost": 9500,
	"availability": 156
	},
	{
	"trip_id": 14,
	"trip_code": "FP 1201",
	"from": {
	"city": "Иркутск",
	"station": "Иркутск",

```
            "code": "395",
            "date": "2021-10-01",
            "time": "08:35"
          },
          "to": {
            "city": "Братск",
            "station": "Братск",
            "code": "3952",
            "date": "2021-10-01",
            "time": "10:05"
          },
          "cost": 10500,
          "availability": 156
        }
      ],
      "trips_back": [
        {
          "trip_id": 1,
          "trip_code": "FP 2100",
          "from": {
            "city": "Братск",
            "station": " Братск ",
            "code": "3952",
            "date": "2021-10-10",
            "time": "08:35"
          },
          "to": {
            "city": "Иркутск",
            "station": "Иркутск",
            "code": "395",
            "date": "2021-10-10",
            "time": "10:05"
          },
          "cost": 10500,
          "availability": 156
        },
        {
          "trip_id": 13,
          "trip_code": "FP 2101",
          "from": {
            "city": "Братск",
            "station": " Братск ",
            "code": "3952",
            "date": "2021-10-10",
            "time": "12:00"
          },
          "to": {
            "city": "Иркутск",
            "station": "Иркутск",
```


	"code": "395",
	"date": "2021-10-10",
	"time": "13:35"
	},
	"cost": 12500,
	"availability": 156
	}
	]
	}
	}
	
	---------- **Validation error** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 422,
	"message": "Validation error",
	"errors": {
	<key>: <массив ошибок>
	}
	}
	}

Оформление бронирования

При оформлении бронирования клиент должен передать на сервер идентификаторы рейсов из базы данных, даты рейсов (в формате YYYY-MM-DD), а также список пассажиров. Каждый пассажир должен содержать следующие поля:

- first_name – обязательно поле, строка
- last_name – обязательно поле, строка
- birth_date – обязательно поле, дата в формате YYYY-MM-DD
- document_number – обязательное поле, строка из цифр длиною в 10 символов

При создании бронирования необходимо также проверить, что на выбранных рейсах есть свободные места. Если на каком-то из рейсов недостаточно мест, то всё бронирование не может быть оформлено.

В случае успешного создания бронирования, с сервера должен вернуться уникальный код бронирования, который может состоять из пяти латинских символов.

Request	Response

URL: {host}/api/booking
Method: POST
 	
Body:
{
"trip_from": {
"id": 1,
"date": "2021-09-20"
},
"trip_back": {
"id": 2,
"date": "2021-09-30"
},
"passengers": [
{
"first_name": "Иван",
"last_name": "Иванов",
"birth_date": "1990-02-20",
"document_number": "1234567890"
},
{
"first_name": "Иван",
"last_name": "Петров",
"birth_date": "1990-03-20",
"document_number": "1224567890"
}
]
}
	---------- **Successful** ----------
	Status: 201
	Content-Type: application/json
	Body:
	{
	"data": {
	"code": "QSASE"
	}
	}
	
	---------- **Validation error** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 422,
	"message": "Validation error",
	"errors": {
	<key>: <массив ошибок>
	}
	}
	}

Информация о бронировании

Получить информацию о бронировании можно по коду бронирования.

Request	Response
URL: {host}/api/booking/{code} **Method:** GET	**Status:** 200 **Content-Type:** application/json **Body:**

```
{
  "data": {
    "code": "AKIJF",
    "cost": 40000,
    "trips": [
      {
        "trip_id": 1,
        "trip_code": "FP2100",
        "from": {
          "city": "Братск",
          "station": " Братск ",
          "code": "3952",
          "date": "2021-10-01",
          "time": "08:35"
        },
        "to": {
          "city": "Иркутск",
          "station": "Иркутск",
          "code": "395",
          "date": "2021-10-01",
          "time": "10:05"
        },
        "cost": 10500,
        "availability": 56
      },
      {
        "trip_id": 2,
        "trip_code": "FP1200",
        "from": {
          "city": "Иркутск",
          "station": "Иркутск",
          "code": "395",
          "date": "2021-10-12",
          "time": "12:00"
        },
        "to": {
          "city": "Братск",
          "station": " Братск ",
          "code": "3952",
          "date": "2021-10-12",
          "time": "13:35"
        },
        "cost": 9500,
        "availability": 56
      }
    ],
    "passengers": [
      {
        "id": 1,
```

	"first_name": "Иван",
	"last_name": "Иванов",
	"birth_date": "1990-02-20",
	"document_number": "1234567890",
	"place_from": "7B",
	"place_back": null
	},
	{
	"id": 2,
	"first_name": "Иван",
	"last_name": "Петров",
	"birth_date": "1990-03-20",
	"document_number": "1224567890",
	"place_from": null,
	"place_back": null
	}
	]
	}
	}

place_from и place_back должны быть null пока не выбрано место.

Получение занятых мест в автобусе

Данный запрос должен возвращать список занятых мест в самолете. Если обратного рейса нет, то occupied_back должен содержать пустой массив.

Request	Response

URL: {host}/api/booking/{code}/seat
Method: GET
	---------- **Success** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"occupied_from": [
	{
	"passenger_id": 1,
	"place": "7B"
	}
	],
	"occupied_back": []
	}

	}

Выбор места в салоне

Данный запрос должен позволять изменить место в салоне автобуса на выбранный рейс для определенного пассажира.

При отправке запроса клиент должен обязательно указать ID пассажира, выбранное место и тип рейса (from/back).

Request	Response

URL: {host}/api/booking/{code}/seat
Method: PATCH
 	
Headers
- Content-Type: application/json
 	
Body:
{
"passenger": 1,
"seat": "7B",
"type": "from/back"
}
	---------- **Success** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"id": 1,
	"first_name": "Иван",
	"last_name": "Иванов",
	"birth_date": "1990-02-20",
	"document_number": "1234567890",
	"place_from": "7B",
	"place_back": null
	}
	}
	---------- **Seat is occupied** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 422,
	"message": "Место занято"
	}
	}
	
	---------- **Forbidden** ----------
	Status: 403
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 403,

	"message": "Forbidden"
	}
	}
	---------- **Validation error** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 422,
	"message": "Validation error",
	"errors": {
	<key>: <массив ошибок>
	}
	}
	}

Получение своих бронирований

Данный запрос должен возвращать все бронирования пользователя. Соотнести бронирования с аутентифицированным пользователем можно по номеру документа.

Request	Response

URL: {host}/api/user/booking
Method: GET
 	
Headers
- **Content-Type:** application/json
- **Authorization:** Bearer {token}
	---------- **Success** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"items": [
	{
	"code": "MGERS",
	"cost": 40000,
	"trips": [
	{
	"trip_id": 1,
	"trip_code": "FP2100",
	"from": {
	"city": "Братск",
	"station": " Братск ",
	"code": "3952",
	"date": "2021-10-01",
	"time": "08:35"

```
        },
        "to": {
          "city": "Иркутск",
           "station": "Иркутск",
           "code": "395",
          "date": "2021-10-01",
          "time": "10:05"
        },
        "cost": 10500,
        "availability": 58
      },
      {
        "trip_id": 2,
        "trip_code": "FP1200",
        "from": {
          "city": "Иркутск",
          "station": "Иркутск",
          "code": "395",
          "date": "2021-10-12",
          "time": "12:00"
        },
        "to": {
          "city": "Братск",
          "station": " Братск ",
          "code": "3952",
          "date": "2021-10-12",
          "time": "13:35"
        },
        "cost": 9500,
        "availability": 58
      }
    ],
    "passengers": [
      {
        "id": 1,
        "first_name": "Иван",
        "last_name": "Иванов",
        "birth_date": "1990-02-20",
        "document_number": "1234567890",
        "place_from": null,
        "place_back": null
      },
      {
        "id": 2,
        "first_name": "Иван",
        "last_name": "Петров",
        "birth_date": "1990-03-20",
        "document_number": "1224567890",
        "place_from": null,
```

	"place_back": null
	}
	]
	}
	]
	}
	}
	
	---------- **Unauthorized** ----------
	Status: 401
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 401,
	"message": "Unauthorized"
	}
	}

Получение информации о пользователе

Request	Response

URL: {host}/api/user
Method: GET
 	
Headers
- **Content-Type:** application/json
- **Authorization:** Bearer {token}
	---------- **Success** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"first_name": "Иван",
	"last_name": "Иванов",
	"phone": "89001234567",
	"document_number": "1234567890"
	}
	---------- **Unauthorized** ----------
	Status: 401
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 401,
	"message": "Unauthorized"
	}
	}

Заказчик допускает возможность изменения базы данных в будущем, поэтому вам необходимо подготовить свой вариант схемы базы данных и сохранить его в корне с модулем. Сохраните файл с названием DB.png.

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

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

Модуль С. Клиентское программирование Single Page Application

Технологии этого модуля: Клиентское программирование
Время на выполнение: 3 часа

К вам обратилась компания «BusWorld» – новая и энергичная компания-перевозчик, предоставляющая услуги пассажирских перевозок.

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.

ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!

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://server-m3.wsr.ru
Ваше SPA должно состоять из следующих экранов:

- Главная
- Вход в личный кабинет
- Регистрация в личном кабинете
- Личный кабинет
- Результаты поиска
- Бронирование
- Управление бронированием
- Выбор места в салоне автобуса

Приложение должно обладать следующим функционалом:

1. Домашний экран – на данном экране располагается статическая информация и форма для поиска подходящих рейсов со следующими полями:
    a. Откуда (From where) – город или вокзал отправления
    b. Куда (To where) – город или вокзал прибытия
    c. Туда (Departing) – дата выезда
    d. Обратно (Returning) – дата возвращения обратно
    e. Количество пассажиров (Passengers) – от 1 до 25

При отправке формы пользователю должен отобразиться экран с найденными рейсами.

Получить список вокзалов по введенном запросу можно с помощью следующего запроса:

Request	Response

URL: {host}/api/station
Method: GET
 	
Query string (GET parameters):
- query
	---------- **Successful** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	“data”: {
	“items”: [
	{
	“name”: “Иркутск”,
	“code”: “395”
	}
	]
	}
	}
	
	---------- **No results** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{

	“data”: {
	“items”: []
	}
	}

2. Экран с найденными рейсами – на данном экране пользователю должен предоставляться список найденных рейсов в зависимости от указанных данных, а также город отправления и прибытия. Каждый рейс должен содержать:
    a. Номер рейса (Trip)
    b. Марка автобуса (Bus)
    c. Дата и время отправления (Date and time of departure)
    d. Время прибытия (Bus time)
    e. Время в полете (Trip time)
    f. Общую цену, учитывая количество пассажиров (Cost)

Если пользователь не указал дату возвращения обратно (Returning), то список рейсов должен включать в себя только маршрут из точки А в точку В. Если пользователь указал дату возвращения обратно (Returning), то список рейсов должен включать в себя и маршруты из точки А в В и из В в А на указанные даты.

Пользователь должен выбрать 1 рейс туда и 1 обратно (если была указана обратная дата), которые ему подходят и нажать на кнопку “Go to booking” для перехода на экран с оформлением бронирования.

На данном экране должна быть предусмотрена фильтрация рейсов следующими способами:

- Фильтрация по времени отправления – должны отображаться рейсы, время отправления которых находится внутри выбранного диапазона.
- Также должна работать сортировка по следующим параметрам:

- Самый быстрый — рейсы должны сортироваться по времени в пути (наиболее быстрые в начале)
- Самый дешевый – рейсы должны сортироваться по стоимости (наиболее дешевые в начале)

Пользователь не должен иметь возможность выбрать рейс, на который не осталось мест.

Получить список рейсов можно используя следующий запрос:

Request	Response

URL: {host}/api/trip
Method: GET
 	
Query string (GET parameters):
- from *(395)*
- to *(3953)*
- date1 *(2020-10-01)*
- date2 *(2020-10-13)*
- passengers *(1)*
	---------- **Successful** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"trips_to": [
	{
	"trip_id": 2,
	"trip_code": "FP1200",
	"from": {
	"city": "Братск",
	"station": "Братск",
	"code": "3953",
	"date": "2020-10-01",
	"time": "12:00"
	},
	"to": {
	"city": "Иркутск",
	"station": "Иркутск",
	"code": "395",
	"date": "2020-10-01",
	"time": "13:35"
	},
	"cost": 9500,
	"availability": 156
	},
	{
	"trip_id": 14,
	"trip_code": "FP 1201",
	"from": {
	"city": "Братск",
	"station": "Братск",
	"code": "3953",

```
            "date": "2020-10-01",
            "time": "08:35"
        },
        "to": {
            "city": "Иркутск",
            "station": "Иркутск",
            "code": "395",
            "date": "2020-10-01",
            "time": "10:05"
        },
        "cost": 10500,
        "availability": 156
    }
],
"trips_back": [
    {
        "trip_id": 1,
        "trip_code": "FP 2100",
        "from": {
            "city": "Иркутск",
            "station": "Иркутск",
            "code": "395",
            "date": "2020-10-10",
            "time": "08:35"
        },
        "to": {
            "city": "Братск",
            "station": "Братск",
            "code": "3953",
            "date": "2020-10-10",
            "time": "10:05"
        },
        "cost": 10500,
        "availability": 156
    },
    {
        "trip_id": 13,
        "trip_code": "FP 2101",
        "from": {
            "city": "Иркутск",
            "station": "Иркутск",
            "code": "395",
            "date": "2020-10-10",
            "time": "12:00"
        },
        "to": {
            "city": "Братск",
            "station": "Братск",
            "code": "3953",
```

	"date": "2020-10-10",
	"time": "13:35"
	},
	"cost": 12500,
	"availability": 156
	}
	]
	}
	}
	
	---------- **Validation error** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	“error”: {
	“code”: 422,
	“message”: “Validation error”,
	“errors”: {
	<key>: <массив ошибок>
	}
	}
	}

3. Экран оформления бронирования – на данном экране пользователь должен видеть информацию о выбранных рейсах, а также иметь возможность добавить пассажиров.

Если пользователь авторизован, то в бронировании изначально должен быть указан авторизованный пассажир.

Добавить пассажира можно нажав на кнопку “Add passenger”. Для каждого пассажира необходимо заполнить следующие поля:

a. Имя (First name)
b. Фамилия (Last name)
c. Дата рождения (Date of Birth)
d. Номер документа (Document number)

Также должна быть возможность удалить пассажира, но нельзя удалить пассажира, если он единственный.

Для подтверждения бронирования необходимо нажать на кнопку “Confirm”. После этого пользователь должен быть перенаправлен на страницу управления бронированием.

Для оформления бронирования используйте следующий запрос:

Request	Response

URL: {host}/api/booking
Method: POST
Body:
{
"trip_from": {
"id": 1,
"date": "2020-09-20"
},
"trip_back": {
"id": 2,
"date": "2020-09-30"
},
"passengers": [
{
"first_name": "Ivan",
"last_name": "Ivanov",
"birth_date": "1990-02-20",
"document_number": "1234567890"
},
{
"first_name": "Ivan",
"last_name": "Gorbunov",
"birth_date": "1990-03-20",
"document_number": "1224567890"
}
]
}
	---------- **Successful** ----------
	Status: 201
	Content-Type: application/json
	Body:
	{
	“data”: {
	“code”: “QSASE”
	}
	}
	---------- **Validation error** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	“error”: {
	“code”: 422,
	“message”: “Validation error”,
	“errors”: {
	<key>: <массив ошибок>
	}
	}
	}

4. Экран управления бронированием – на данном экране необходимо отобразить всю имеющуюся информацию о бронировании:
    a. Информация о бронировании
        i. Код бронирования
        ii. Стоимость бронирования

b. Информация о перелете (для каждого рейса) – обратите внимание, что если бронирование оформлено “туда-обратно”, то рейса должно быть два (туда и обратно):
   i. Номер рейса (Trip)
   ii. Самолет (Bus)
   iii. Откуда (From where)
   iv. Дата и время отправления (Date and time of departure)
   v. Куда (To where)
   vi. Время прибытия (Bus time)
   vii. Время в полете (Trip time)
c. Информация о пассажирах
   i. Имя (First name)
   ii. Фамилия (Last name)
   iii. Дата рождения (Date of Birth)
   iv. Номер документа (Document number)
   v. Место (Seat)
d. Также на этом экране присутствует кнопка для выбора места “Select seats”, при нажатии на которую пользователь должен перейти на экран с выбором места для пассажиров в бронировании.

Получить информацию о бронировании можно используя следующий запрос:

Request	Response

URL: {host}/api/booking/{code}
Method: GET
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"code": "AKIJF",
	"cost": 40000,
	"trips": [
	{
	"trip_id": 1,
	"trip_code": "FP2100",
	"from": {
	"city": "Иркутск",
	"station": "Иркутск",
	"code": "395",
	"date": "2020-10-01",
	"time": "08:35"

```
        },
        "to": {
          "city": "Братск",
          "station": "Братск",
          "code": "3953",
          "date": "2020-10-01",
          "time": "10:05"
        },
        "cost": 10500,
        "availability": 56
      },
      {
        "trip_id": 2,
        "trip_code": "FP1200",
        "from": {
          "city": "Братск",
          "station": "Братск",
          "code": "3953",
          "date": "2020-10-12",
          "time": "12:00"
        },
        "to": {
          "city": "Иркутск",
          "station": "Иркутск",
          "code": "395",
          "date": "2020-10-12",
          "time": "13:35"
        },
        "cost": 9500,
        "availability": 56
      }
    ],
    "passengers": [
      {
        "id": 1,
        "first_name": "Ivan",
        "last_name": "Ivanov",
        "birth_date": "1990-02-20",
        "document_number": "1234567890",
        "place_from": "7B",
        "place_back": null
      },
      {
        "id": 2,
        "first_name": "Ivan",
        "last_name": "Larin",
        "birth_date": "1990-03-20",
        "document_number": "1224567890",
        "place_from": null,
```

	"place_back": null }] } }

5. Экран с выбором места – на данном экране должна быть представлена схема автобуса и список пассажиров. Должна быть возможность выбрать пассажира, а после для него место. При нажатии на кнопку “Back” экран должен смениться на предыдущий без сохранения выбранных мест. При нажатии на кнопку “Confirm” информация о выбранных местах должна быть сохранена, и пользователь должен быть перенаправлен на предыдущую страницу.

Для сохранения выбранного места нужно использовать следующий запрос:

Request	Response

URL:
{host}/api/booking/{code}/seat
 	
Method: PATCH
 	
Headers
- Content-Type: application/json
 	
Body:
{
"passenger": 1,
"seat": "7B",
"type": "from/back"
}
	---------- **Success** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"id": 1,
	"first_name": "Ivan",
	"last_name": "Ivanov",
	"birth_date": "1990-02-20",
	"document_number": "1234567890",
	"place_from": "7B",
	"place_back": null
	}
	}
	---------- **Seat is occupied** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 422,
	"message": "Seat is occupied",
	}
	}

	---------- **Forbidden** ----------
	Status: 403
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 403,
	"message": "Passenger does not apply to booking"
	}
	}
	---------- **Validation error** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	“error”: {
	“code”: 422,
	“message”: “Validation error”,
	“errors”: {
	<key>: <массив ошибок>
	}
	}
	}

6. Регистрация – неавторизованный пользователь должен иметь возможность зарегистрироваться в системе на странице с регистрацией.

Для регистрации пользователя предусмотрен следующий запрос:

Request	Response

URL: {host}/api/register
Method: POST
 	
Headers
- Content-Type: application/json
 	
Body:
{
“first_name”: “Ivan”,
“last_name”: “Ivanov”,
“phone”: “89001234567”,
“document_number”: “7567999222”,
“password”: “paSSword”
}
	---------- **Successful** ----------
	Status: 204
	---------- **Validation error** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	“error”: {
	“code”: 422,
	“message”: “Validation error”,
	“errors”: {
	<key>: <массив ошибок>
	}
	}
	}

7. Вход в личный кабинет – неавторизованный пользователь должен иметь возможность войти в систему на странице со входом.

Для аутентификации пользователя предусмотрен следующий запрос:

Request	Response

URL: {host}/api/login
Method: POST
 	
Headers
- Content-Type: application/json
 	
Body:
{
“phone”: “89001234567”,
“password”: “paSSword”
}
	---------- **Successful** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	“data”: {
	“token”: <сгенерированный token>
	}
	}
	
	---------- **Validation error** ----------
	Status: 422
	Content-Type: application/json
	Body:
	{
	“error”: {
	“code”: 422,
	“message”: “Validation error”,
	“errors”: {
	<key>: <массив ошибок>
	}
	}
	}
	
	---------- **Unauthorized** ----------
	Status: 401
	Content-Type: application/json
	Body:
	{
	“error”: {
	“code”: 401,
	“message”: “Unauthorized”,
	“errors”: {
	“phone”: [“phone or password incorrect”]
	}
	}
	}

8. Личный кабинет – на странице с профилем пользователь должен увидеть следующую информацию:

a. О себе
    i. Имя
    ii. Фамилия
    iii. Количество прошедших полетов
b. Избранные рейсы. Каждый рейс должен отображать следующие данные:
    i. Город отправления
    ii. Город прибытия
    iii. Время отправления
    iv. Время прибытия
    v. Кнопка “Book now” – при нажатии на эту кнопку экран должен смениться на “Экран бронирования” для оформления нового бронирования. На экране бронирования автоматически должен быть заполнен авторизованный пассажир, а в блоке с рейсами должна быть возможность выбрать дату для рейса.
    vi. Кнопка “Remove from favorites” – при нажатии на эту кнопку рейс должен быть удален из избранного.
c. Прошедшие рейсы. Каждый рейс должен отображать следующие данные:
    i. Номер рейса
    ii. Город отправления
    iii. Город прибытия
    iv. Время отправления
    v. Время прибытия
    vi. Кнопка “Add to favorites” – при нажатии на эту кнопку рейс должен быть добавлен в избранное.
d. Предстоящие бронирования. Каждое бронирование должно содержать следующую информацию:
    i. Код бронирования
    ii. Дата отправления
    iii. Время отправления
    iv. Время прибытия
    v. Откуда
    vi. Куда

При нажатии на код бронирования пользователь должен перейти на страницу управления бронированием.

Для получения информации о пользователе вы можете использовать следующий запрос:

Request	Response

URL: {host}/user
Method: GET
 	
Headers
- Content-Type: application/json
- Authorization: Bearer {token}
	---------- **Success** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"first_name": "Ivan",
	"last_name": "Ivanov",
	"phone": "89001234567",
	"document_number": "1224567890"
	}
	---------- **Unauthorized** ----------
	Status: 401
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 401,
	"message": "Unauthorized"
	}
	}

Для получения информации о бронированиях пользователя вы можете использовать следующий запрос:

Request	Response

URL: {host}/user/booking
Method: GET
 	
Headers
- Content-Type: application/json
- Authorization: Bearer {token}
	---------- **Success** ----------
	Status: 200
	Content-Type: application/json
	Body:
	{
	"data": {
	"items": [
	{
	"code": "MGERS",
	"cost": 40000,
	"trips": [
	{

```
      "trip_id": 1,
      "trip_code": "FP2100",
      "from": {
        "city": "Иркутск",
        "station": "Иркутск",
        "code": "395",
        "date": "2020-10-01",
        "time": "08:35"
      },
      "to": {
        "city": "Братск",
        "station": "Братск",
        "code": "3953",
        "date": "2020-10-01",
        "time": "10:05"
      },
      "cost": 10500,
      "availability": 58
    },
    {
      "trip_id": 2,
      "trip_code": "FP1200",
      "from": {
        "city": "Братск",
        "station": "Братск",
        "code": "3953",
        "date": "2020-10-12",
        "time": "12:00"
      },
      "to": {
        "city": "Иркутск",
        "station": "Иркутск",
        "code": "395",
        "date": "2020-10-12",
        "time": "13:35"
      },
      "cost": 9500,
      "availability": 58
    }
  ],
  "passengers": [
    {
      "id": 1,
      "first_name": "Ivan",
      "last_name": "Ivanov",
      "birth_date": "1990-02-20",
      "document_number": "1234567890",
      "place_from": null,
      "place_back": null
```


	},
	{
	"id": 2,
	"first_name": "Ivan",
	"last_name": "Sergeev",
	"birth_date": "1990-03-20",
	"document_number": "1224567890",
	"place_from": null,
	"place_back": null
	}
	]
	}
	]
	}
	}
	---------- **Unauthorized** ----------
	Status: 401
	Content-Type: application/json
	Body:
	{
	"error": {
	"code": 401,
	"message": "Unauthorized"
	}
	}

9. Должна быть возможность выйти из личного кабинета.
10. Необходимо позаботиться об уведомлении пользователей о каких-либо действиях (ошибки валидации, подтверждения и т.п).

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

Модуль D. CMS WordPress

Технологии модуля: HTML5, CSS3, CMS WordPress, JavaScript, граф. дизайн
Время на выполнение: 3 часа

Предметом разработки является разработка сайта для сети школ иностранных языков «Полиглот» на WordPress.

Цель проекта: Разработка сайта для привлечения клиентов, с возможностью расчета стоимости обучения.

Структура интернет-ресурса и навигация.

1. Карта сайта:

1. Главная
2. Наши школы (фото, контактная информация)
    2.1 Иркутск
    2.2 Тулун
    2.3 Новосибирск
    2.4 Чита
    2.5 Красноярск
3. Расписание занятий
4. Новости
5. Наши услуги
    5.1. Английский язык
    5.2 Китайский язык
    5.3. Индивидуальные занятия
    5.4. Калькулятор стоимости услуг
6. Преподаватели
    6.1. Логопеды
    6.2. Преподаватели

7. Отзывы

8. Контакты

2. Главная страница должна быть выполнена в виде Landing Page и включает в себя следующие блоки:

- **Шапка** с элементами фирменного стиля школы, иллюстрациями услуг компании (баннер), контактной информацией.

- **Выбор города**. При выборе города в шапке контактная информация меняется.

- **Блок меню**. Главное меню должно быть зафиксировано в верхней части веб-страницы. Для показа контента будет использована прокрутка.

- **Описание преимуществ компании**. С возможностью записи на пробное занятие.

- **Информация о преподавателях** (фотография, ФИО, краткая информация) должно быть реализовано в виде сладера (с автоматическим перелистыванием и возможностью самостоятельного перемещения по слайдам курсовами).

- **Новости**. Должно выводится не более 10 последних новостей с возможностью постраничного просмотра.

- **Отзывы** (фото, текстовая информация). Должно выводится не менее 3 отзывов.

- **Футер** (реквизиты организации, ссылки на социальные сети, счетчики).

3. Внутренние страницы:

- Наши школы. После выбора города на странице отображается изображение и контактная информация школы.

- Расписание занятий (для группового обучения).

- Новости. Каждая новость содержит: текст, изображение, дату публикации. На странице должно выводится не более 4 новостей. Если новостей больше должна

выводится пагинация, каждая последующая новость добавляется вниз ленты без перезагрузки страницы.

- Наши услуги. При выборе услуги дается краткое пояснение образовательной услуги и предоставляется возможность рассчитать ее стоимость с помощью калькулятора, а также возможность записаться на пробное занятие.

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.

Разработанный веб-сайт должен быть доступен по адресу http://xxxxxx-m2.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.

5. Критерии оценки.

Таблица 2.

Критерий		Баллы		
		Судейские аспекты	Объективная оценка	Общая оценка
A	Дизайн и верстка веб приложения	17	16.5	33.5
B	Программирование на стороне клиента	2	22	24
C	Программирование на стороне сервера	3	16.5	19.5
D	Системы управления контентом	10.9	12.1	23
Итого		32.9	67.1	100