



ЧЕМПИОНАТ
ПРОФЕССИОНАЛЬНОГО
МАСТЕРСТВА
«МОСКОВСКИЕ МАСТЕРА»



Разработано главным экспертом по компетенции
«Веб-технологии»

/ Зимин Артем Евгеньевич

(ФИО главного эксперта)

СОГЛАСОВАНО

Индустриальный эксперт



/ Гадасин Денис Вадимович

(ФИО)

Менеджер компетенции

/ Агарков Олег Владимирович

(ФИО)

КОНКУРСНОЕ ЗАДАНИЕ
КОМПЕТЕНЦИИ
«Веб-технологии»

для категории

Основная

2024 г.

*Чемпионат профессионального мастерства города Москвы «Московские мастера» 2024 в рамках
Регионального этапа Чемпионата по профессиональному мастерству «Профессионалы» и
Чемпионата высоких технологий - 2024 в Москве*

Конкурсное задание разработано экспертным сообществом и утверждено индустриальным партнером, в котором установлены нижеследующие правила и необходимые требования владения профессиональными навыками для участия в соревнованиях по профессиональному мастерству.

Конкурсное задание включает в себя следующие разделы:

- | | |
|--|--|
| 1. Спецификация оценки компетенции | 2 |
| 1.1. Конкурсное задание..... | Ошибка! Закладка не определена. |
| 1.2. Структура модулей конкурсного задания | 3 |

1. СПЕЦИФИКАЦИЯ ОЦЕНКИ КОМПЕТЕНЦИИ

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

Таблица №1

Оценка конкурсного задания

Критерий		Методика проверки навыков в критерии
A	Разработка интерфейса пользователя	Проверка результата выполнения модуля после его завершения
Б	Разработка на стороне клиента	Проверка результата выполнения модуля после его завершения
В	Разработка на стороне сервера	Проверка результата выполнения модуля после его завершения
Г	Спидтаски	Проверка результата выполнения модуля после его завершения

1.1. КОНКУРСНОЕ ЗАДАНИЕ

Формат участия: индивидуальный

Возрастной ценз: Студенты колледжей

Общая продолжительность Конкурсного задания¹: 12 ч.

Количество конкурсных дней: 3 дней.

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

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

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

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

1.2. Структура модулей конкурсного задания

Модуль А. Разработка интерфейса пользователя

Технологии этого модуля: граф. дизайн, HTML5, CSS3, GIT

Время на выполнение: 4 часа

Задание:

В современном мире с каждым днём появляется всё больше и больше новых технологий, которые позволяют решать повседневные проблемы.

Цель данного технического задания состоит в разработке дизайна и создании страниц веб-сайта "Московские экскурсии", который будет предоставлять информацию об экскурсиях по достопримечательностям города Москва. Ниже приведены основные требования к внешнему виду страниц.

В первую очередь сайт рассчитан на гостей Москвы (как граждан России, так и представителей ближнего зарубежья), которые впервые посетили его. Потенциальные пользователи сайта предпочитают использовать мобильные устройства для доступа к интернету и поиска информации.

В дизайне сверстанных страниц следует использовать и применять элементы, концепции и приемы, позволяющие привлечь внимание, создать положительный образ ресурса, реализовать современные концепции дизайна веб-страниц.

Также при верстке страниц вам следует учитывать существующие практики UI/UX, чтобы использование веб-ресурса было максимально удобным и продуктивным.

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

Используйте анимацию, чтобы пользователи могли понять, что элементы являются интерактивным. Используемая анимация должна быть гармоничной, чтобы создать приятный визуальный эффект и не отвлекать пользователей.

Использование любых фреймворков и библиотек (Bootstrap, Tailwind, например) запрещено!

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – сверстать следующие страницы веб-сайта:

1. Главная
2. Страница с описанием экскурсии

3. Страница регистрации посетителя сайта
4. Страница личного кабинета пользователя
5. Страница истории заказов
6. Страница подтверждения бронирования
7. Страница отображения и управления бронированиями – для администратора
8. Страница управления экскурсиями – для администратора

Сверстанный веб-сайт должен быть доступен по адресу <http://xxxxxx-m1.moscow.ru> адрес указан в личном кабинете участника соревнований.

Страницы должны быть адаптивны. При переходе от десктопной версии главной страницы (ширина экрана 1440px) к мобильной версии (ширина экрана 414px) должно быть реализовано не менее четырех структурных изменений контента (например, скрываются/добавляются, увеличиваются/уменьшаются элементы).

На остальных разрешениях контент должен отображаться корректно.

Пояснение по логике сервиса (для понимания бизнес-процесса бронирования экскурсии):

- 1) Авторизованный пользователь кликает по кнопке/ссылке бронирования, указывает необходимые данные.
- 2) Администратор связывается с пользователем в некоторых случаях отменяет экскурсию вручную.

Основные роли пользователей

Гость

1. Регистрация
2. Авторизация
3. Доступ к главной странице и страницам с описанием экскурсий

Авторизованный пользователь

1. Доступ к главной странице и страницам с описанием экскурсий
2. Бронирование экскурсии (отправка заявки на экскурсию)
3. Просмотр личного кабинета
4. Просмотр списка экскурсий (посещенных, отмененных, забронированных)
5. Выход из профиля

Администратор

1. Доступ к главной странице и страницам с описанием экскурсий
2. Просмотр списка экскурсий

3. Доступ к странице с возможностью управления экскурсией (добавление/редактирование контента)
4. Выход

(1) Главная - Landing Page

На главной странице пользователи могут найти общую информацию о городе Москва и его достопримечательностях.

На странице представлен перечень доступных экскурсий с кратким описанием, изображением и другой информацией, чтобы заинтересованные пользователи могли выбрать подходящую экскурсию.

Главная страница должна содержать следующие блоки (вы также можете добавить дополнительные блоки/секции, если считаете, что это усилит работу):

1. Шапка сайта.

- 1.1. Логотип.

Вам необходимо создать логотип, с учетом следующих пожеланий заказчика.

Логотип должен быть уникальным и легко узнаваемым. Он должен быть достаточно простым, чтобы визуально ассоциироваться с темой экскурсий по достопримечательностям Москвы. Логотип должен быть согласован с общей концепцией брендинга сайта "Московские экскурсии" и выполнен в высоком качестве.

- 1.2. Меню навигации.

- 1.2.1. Регистрация – ссылка на отдельную страницу

- 1.2.2. Личный кабинет – ссылка на отдельную страницу

- 1.2.3. О нас

- 1.2.4. Контакты

- 1.2.5. Поиск

- 1.2.6. Наши экскурсии

2. Вывод четырех наиболее популярных экскурсий в виде слайдера с индикацией активного слайда (работа слайдера не проверяется, только дизайн и верстка). Слайд должен содержать следующую информацию:

- 2.1. Изображение.

- 2.2. Краткое описание.

- 2.3. Кнопка "Далее" для перехода на страницу с описанием экскурсии.

- 2.4. Рейтинг экскурсии

3. Секция/блок для авторизации

Форма входа должна предоставлять поле для ввода адреса электронной почты и пароля для аутентификации, кнопка/ссылка для входа, а также кнопка/ссылка для восстановления забытого пароля.

4. Секция/блок «О нас»

4.1. Слоган – см. в медиафайлах

4.2. Описание компании - см. в медиафайлах

5. Секция/блок «Наши экскурсии»

Перечень с карточками экскурсий. В каждой карточке отображается название экскурсии, фото, описание, продолжительность, количество бронирований, рейтинг, кнопка/ссылка, позволяющая перейти к бронированию. Общее количество карточек с экскурсиями – 7. Карточки должны быть представлены в табличной форме с тремя столбцами. В мобильной версии страницы вид перечня карточек - на ваше усмотрение.

6. Поле поиска экскурсий по ключевым словам или описанию экскурсии, а также кнопка «Поиск». Поле поиска содержит подсказки.

7. Секция/блок «Отзывы клиентов»

Представлены отзывы клиентов, ранее посетивших экскурсии – в виде слайдсета (должны отображаться фрагменты других слайдов) с прокруткой по 1 элементу. Реализация JS не требуется, но визуально должно быть понятно, что подразумевается поочередная смена отзывов.

7.1. Имя автора отзыва

7.2. Фотография автора

7.3. Кликабельная ссылка/кнопка на страницу с описанием экскурсии

7.4. Текст отзыва

7.5. Дата отзыва

Порядок полей может быть любым.

8. Форма для подписки на новости сервиса

8.1. Email

8.2. Элемент, подтверждающий согласие на обработку персональных данных

8.3. Кнопка для подписки

9. Подвал сайта

9.1. Телефон “+7 (353) 123-45-67”

9.2. e-mail: info@moscow.ru

9.3. Навигация по сайту:

9.3.1. Главная

9.3.2. Регистрация

9.3.3. Личный кабинет

- 9.3.4. О нас
- 9.3.5. Контакты
- 9.3.6. Поиск
- 9.3.7. Наши экскурсии

Порядок и вид секций/блоков может быть произвольный.

На главной странице необходимо отобразить сообщение об использовании cookie: «Наш сайт использует файлы cookie, чтобы улучшить работу сайта, повысить его эффективность и удобство. Продолжая использовать сайт, вы соглашаетесь на использование файлов cookie.»

(2) Страница с описанием экскурсии

На этой странице пользователи могут ознакомиться с названием экскурсии, описанием, изображением, датами.

Также на странице присутствует кнопка/ссылка, позволяющая перейти к бронированию.

При печати страницы средствами браузера, например, по сочетанию клавиш Ctrl+P, остается только информация: название экскурсии, фотография в черно-белом исполнении (градации серого), даты, без шапки и подвала. Цветные элементы отсутствуют.

(3) Страница регистрации посетителя сайта

На этой странице вам необходимо сделать форму со следующими полями, обязательными для заполнения (поля должны иметь соответствующий тип):

- e-mail
- Имя
- Фамилия
- Телефон
- Пароль
- Повтор пароля
- Элемент, подтверждающий согласие на обработку персональных данных
- Кнопка для регистрации

Отобразите варианты сообщений, если пользователь указал некорректные данные, не заполнил обязательные поля и т.д.

(4) Страница личного кабинета пользователя

Общая информация о пользователе (в любом порядке):

- имя и кнопка «Изменить»,
- фамилия и кнопка «Изменить»,

- фотография пользователя и кнопка «Изменить»,
- номер телефона и кнопка «Изменить»,
- e-mail и кнопка «Изменить»,
- дата регистрации пользователя,
- количество дней, прошедших с момента регистрации на сайте,
- ссылка на страницу истории заказов

В меню должна быть представлена кнопка/ссылка для выхода из профиля, а ссылка «Регистрация» должна отсутствовать.

(5) Страница истории заказов

Экскурсии должны быть сгруппированы по статусам «Посещенные», «Отмененные», «Забронированные». Каждая из групп должна иметь собственное оформление и визуально отличаться от других. Каждая группа должна содержать минимум 2 экскурсии.

В каждой экскурсии должна быть отражена следующая информация: превью фото, Название, Ссылка/кнопка на страницу с экскурсией, статус, дата заявки, а для экскурсии со статусом «Отменено» причина отмены.

В меню должна быть представлена кнопка/ссылка для выхода из профиля, а ссылка «Регистрация» должна отсутствовать.

(6) Страница подтверждения бронирования

Представлена следующая информация:

- Название экскурсии – не редактируемое поле
- Фамилия клиента – не редактируемое поле
- Имя клиента – не редактируемое поле
- Дата, на которую происходит бронирование – редактируемое поле
- Количество мест (экскурсантов) – редактируемое поле
- Поле для пожеланий/комментариев – редактируемое поле
- Кнопка/ссылка для отправки заявки

В меню должна быть представлена кнопка/ссылка для выхода из профиля, а ссылка «Регистрация» должна отсутствовать.

(7) Страница отображения и управления бронированиями – для администратора

На странице представлена информация по каждому из бронирований:
Название экскурсии

Фамилия и имя пользователь, забронировавшего экскурсию
Контактный номер телефона пользователя
Количество экскурсантов, указанных в заявке
Дата бронирования
Дата экскурсии

Только для бронирований со статусом «Забронированные» администратор может поменять статус на «Отмененные» с указанием причины отмены.

Также на странице представлены элементы, позволяющие администратору:

- сортировать бронирования по дате, статусу
- фильтровать бронирования по пользователю, статусу

В меню должен быть представлена кнопка/ссылка для выхода из профиля, а ссылка «Регистрация» должна отсутствовать.

(8) Страница управления экскурсиями – для администратора

Представлены поля для:

- Добавления/редактирования названия экскурсии
- Добавления/редактирования описания экскурсии
- Добавления/редактирования дат экскурсии. Есть возможность указать несколько дат.
- Добавления/редактирования изображения экскурсии
- Добавления/редактирования продолжительности экскурсии
- Кнопка/ссылка для сохранения изменений

В меню должен быть представлена кнопка/ссылка для выхода из профиля, а ссылка «Регистрация» должна отсутствовать.

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Ваш HTML/CSS должен быть валидным. Используйте лучшие практики верстки в соответствии с заданием.

Страницы должны быть согласованы между собой. Ссылки максимально должны быть кликабельными с навигацией по страницам.

Вам нужно использовать не менее одного из представленных шрифтов. Рекомендуется выбирать шрифты, которые отражают стиль сайта и создают ощущение надежности и профессиональности.

Вы можете использовать и изменять предоставленные медиа-данные для реализации поставленных задач.

Оценка будет производиться в браузере Google Chrome.

Сверстанный веб-сайт должен быть доступен по адресу <http://xxxxxx-m1.moscow.ru> адрес указан в личном кабинете участника соревнований.

Сохраните сверстанные страницы и созданные дизайн-макеты со следующими именами:

1. Главная – index.html, index.png
2. Страница с описанием экскурсии – excursion.html, excursion.png
3. Страница регистрации посетителя сайта – register.html, register.png
4. Страница личного кабинета пользователя – account.html, account.png
5. Страница истории заказов – history.html, history.png
6. Страница подтверждения бронирования – confirm.html, confirm.png
7. Страница отображения и управления бронированиями – для администратора – config.html, config.png
8. Страница управления экскурсиями – для администратора – manage.html, manage.png

Макеты должны быть расположены в отдельной папке (/mockups).

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

Процесс работы нужно организовать с использованием системы контроля версий. Не забудьте создать репозиторий и фиксировать в нем изменения по мере выполнения работы (не менее 3 коммитов, описание коммитов отражает вносимые изменения).

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

Работы, выполненные с использованием CSS-фреймворков, проверяться не будут!

Модуль Б. Разработка на стороне клиента

Технологии этого модуля: JavaScript

Время на выполнение: 2 часа

Вам необходимо разработать Single Page Application (далее SPA) для организации хранения данных с возможностью разграничения прав доступа к файлам. Заказчик предоставляет вам полностью готовую верстку со всеми страницами и рабочее API.

Общее

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

ВНИМАНИЕ! Проверке подлежат приложения в формате Single Page Application!

В документации будет использоваться переменная {{host}} которая обозначает адрес API: [http://rest -m1.moscow.ru/api-file](http://rest-m1.moscow.ru/api-file).

Ваше SPA должно состоять из следующих экранов:

- Регистрация;
- Вход в систему;
- Загрузка файлов;
- Редактирование файла;
- Права доступа для файла;
- Список файлов пользователя;
- Список файлов, к которым пользователь имеет доступ.

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

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

Функциональные требования

Регистрация

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

Интерактивные элементы экрана

- По щелчку на кнопку регистрации происходит попытка зарегистрировать нового пользователя.

Вход в систему

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

Интерактивные элементы экрана

- По щелчку на кнопку входа происходит попытка аутентифицировать пользователя.

Список файлов пользователя

На данном экране отображаются список файлов авторизованного пользователя. Для каждого файла отображается имя и идентификатор. Возле каждого файла есть кнопки удаления и редактирования, а также изменения прав доступа и скачивания файла.

Интерактивные элементы экрана

- По щелчку на кнопку удаления файла файл удаляется;
- По щелчку на кнопку редактирования файла происходит переход на экран “Редактирование файла”;
- По щелчку на кнопку изменения прав файла происходит переход на экран “Права доступа для файла”;
- По щелчку на кнопку скачивания файла соответствующий файл скачивается.

Список файлов, к которым пользователь имеет доступ
На данном экране отображаются список файлов к которым имеет доступ
авторизированный пользователь. Для каждого файла отображается имя и
идентификатор. Возле каждого файла есть кнопка для скачивания.

Интерактивные элементы экрана

- По щелчку на кнопку скачивания файла соответствующий файл
скачивается.

Редактирование файла

На данном экране пользователь может изменить имя файла. При вводе
некорректного значения у соответствующего поля формы отображаются тексты
ошибок, а поле подсвечивается красным.

Интерактивные элементы экрана

- По щелчку на кнопку сохранения происходит попытка переименовать
файл;
- По щелчку на кнопку назад происходит перенаправление на экран
“Список файлов пользователя”.

Права доступа для файла

На данном экране отображается список пользователей, для которых открыт
доступ для скачивания текущего файла. Каждый из этих пользователей
представлен полным именем и email. Возле каждого из них есть кнопка для
удаления доступа. Кроме того на экране присутствует поле для ввода email
нового пользователя, которому необходимо добавить права на файл и кнопка
добавления этого пользователя. При вводе некорректных значений у
соответствующего поля формы отображаются тексты ошибок, а само поле
подсвечивается красным.

Интерактивные элементы экрана

- По щелчку на кнопку добавления пользователя происходит попытка
предоставления ему права на скачивание текущего файла;
- По щелчку на кнопку удаления доступа к файлу, пользователь удаляется
из списка и доступ к файлу теряет;
- По щелчку на кнопку назад происходит перенаправление на экран
“Список файлов пользователя”.

Загрузка файлов

На данном экране отображаются поле для выбора нескольких файлов для загрузки на сервер. Кроме того на экране присутствует область для Drag-and-drop. После попытки загрузить файлы на экране отображается список файлов. Для каждого из них отображается имя, пометка загружен ли он, а также кнопка для скачивания если он успешно загружен.

Интерактивные элементы экрана

- По щелчку на поле выбора нескольких файлов для загрузки и последующего выбора этих файлов, происходит попытка их загрузить на сервер;
- При перетаскивании нескольких файлов на специальную область экрана происходит попытка их загрузить на сервер;
- По щелчку на кнопку назад происходит перенаправление на экран “Список файлов пользователя”.

Описание предоставленного API

Документация API идентична описанной в Модуле В.

Инструкция для конкурсанта

Разработанное приложение должно быть доступно по адресу <http://xxxxxx-m2.moscow.ru>, где xxxxxx - логин участника.

Вам предоставляются следующие конфигурации фреймворков и библиотек:

- jQuery 3.x
- jQuery UI 1.x
- VueJS 2.x
- VueJS 3.x
- Vue Router 3.x
- Vue Router 4.x
- Vue CLI
- React
- React Router
- React CLI
- Angular CLI

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

Модуль В. Разработка на стороне клиента

Технологии этого модуля: PHP, Python

Время на выполнение: 2 часа

Ваша задача - создать облако хранения данных с возможностью разграничения прав доступа к файлам.

ОСНОВНОЙ ФУНКЦИОНАЛ

Для неавторизованного пользователя:

- авторизация
- регистрация

Для авторизованного пользователя:

- возможность сброса авторизации
- работа с файлами
 - загрузка
 - редактирование
 - удаление
- разграничение прав доступа к файлам

Функционал авторизованного пользователя не должен быть доступен гостю.

ЗАКАЗЧИК НЕ ПРЕДОСТАВЛЯЕТ ГОТОВУЮ БАЗУ ДАННЫХ И ВАМ НЕОБХОДИМО РАЗРАБОТАТЬ ЕЕ САМОСТОЯТЕЛЬНО.

ОБЩЕЕ

Вам необходимо реализовать REST API заданной структуры.

В примерах будет использоваться переменная {{host}} которая обозначает адрес http://xxxxxx-m3.moscow.ru/api-file, где xxxxxx - логин участника.

Идентификацию пользователя организуйте посредством Bearer Token.

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

Status: 403

Content-Type: application/json

Body:

```
{  
    "message": "Login failed"  
}
```

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

Status: 403

Content-Type: application/json

Body:

```
{  
    "message": "Forbidden for you"  
}
```

При попытке получить не существующий ресурс необходимо возвращать ответ следующего вида:

```

Status: 404
Content-Type: application/json
Body:
{
  "message": "Not found",
  "code": 404
}

```

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

```

Status: 422
Content-Type: application/json
Body:
{
  "success": false,
  "code": <code>,
  "message": {
    <key>: [<error message>]
  }
}

```

В свойстве `message.<key>` 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:

```

{
  "success": false,
  "code": 422,
  "message": {
    "phone": [ "field phone can not be blank"],
    "password": [ "field password can not be blank"]
  }
}

```

В поле `message` могут быть любые сообщения об ошибках (если не указана конкретная ошибка), но они должны описывать возникшую проблему.

Файлы должны загружаться на сервер в папку `uploads` (находящуюся в корне веб-сервера) и иметь уникальные сгенерированные имена.

ОПИСАНИЕ ФУНКЦИОНАЛА ГОСТЯ

Аутентификация:

При успешной аутентификации возвращается сгенерированный токен пользователя.

- `email` — обязательное
- `password` - обязательное

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

<p>- Content-Type: application/json</p> <p>Body:</p> <pre>{ "email": "admin@admin.ru", "password": "Qa1" }</pre>	<p>data: { "success": true, "code": 200, "message": "Success", "token": <сгенерированный token> }</p> <p style="text-align: center;"><u>Ошибки валидации</u> Формат ответа из общих требований</p> <p style="text-align: center;"><u>Неправильные логин или пароль</u></p> <p>Status: 401 Content-Type: application/json Body:</p> <pre>data: { "success": false, "code": 401, "message": "Authorization failed" }</pre>
---	---

Регистрация:

При успешной регистрации возвращается сгенерированный токен добавленного пользователя.

Функция должна принимать следующие параметры:

- email - email пользователя обязательный, валидный e-mail адрес, уникальный
- password - пароль пользователя обязательный, состоит минимум из 3 символов, из которых как минимум одна строчная, одна прописная и одна цифра
- first_name — имя, состоит минимум из 2 символов обязательное
- last_name - фамилия обязательное

Request	Response
<p>URL: {{host}}/registration Method: POST</p> <p>Headers - Content-Type: application/json</p> <p>Body:</p> <pre>{ "email": "admin@admin.ru", "password": "Qa1", "first_name": "name", "last_name": "last_name" }</pre>	<p style="text-align: center;"><u>Успех</u></p> <p>Status: 201 Content-Type: application/json Body:</p> <pre>data: { "success": true, "code": 201, "message": "Success", "token": <сгенерированный token> }</pre> <p style="text-align: center;"><u>Ошибки валидации</u> Формат ответа из общих требований</p>

ОПИСАНИЕ ФУНКЦИОНАЛА АВТОРИЗИРОВАННОГО ПОЛЬЗОВАТЕЛЯ

Сброс авторизации:

Запрос предназначен для очистки значение токена пользователя.

Request	Response
URL: {{host}}/logout	<u>Успех</u>

Method: GET	Status: 204 Content-Type: application/json Body (не отдается) <div style="text-align: center; font-size: small; margin-top: 20px;"> <u>Попытка обратиться не авторизованным</u> <u>пользователем</u> <u>Формат ответа из общих требований</u> </div>
--------------------	--

Загрузка файлов:

Функция должна принимать следующие параметры:

- files - файлы, разрешены файлы размером не более 2 мб,
- files - типы файлов: doc, pdf, docx, zip, jpeg, jpg, png;

Если какие-либо из файлов по какой-либо причине не загружены, остальные должны загрузиться и занестись в БД.

Если имя загружаемого файла совпадает с именем файла, который уже загружен у данного пользователя, то загружаемый файл переименовывается в "{NAME} ({I}).{EXT}". Например: загружаем файл "Juicy chicken.doc", и если такой файл уже есть, то загружаемый файл переименовывается в "Juicy chicken (1).doc", или "Juicy chicken (2).doc", если такой файл уже там есть и т.д. Также с каждым файлом ассоциирован его уникальный идентификатор file_id - случайная строка из 10 символов.

Request	Response
URL: {{host}}/files Method: POST Headers - Content-Type: multipart/form-data FormData: "files":<массив с файлами>	<u>Успех</u> Status: 200 Content-Type: application/json Body: data: [{ "success": true, "code": 200, "message": "Success", "name": "Имя загруженного файла", "url": "{{host}}/files/qweasd1234", "file_id": "qweasd1234" }, { "success": false, "message": [<error message>], "name": "Имя НЕ загруженного файла" }] <div style="text-align: center; font-size: small; margin-top: 20px;"> <u>Ошибки валидации</u> <u>Вывод в общем списке загруженных файлов</u> <u>(см. пример выше)</u> <u>Попытка обратиться не авторизованным</u> <u>пользователем</u> <u>Формат ответа из общих требований</u> </div>

Редактирование файла:

Функция должна принимать следующие параметры:

- name - имя файла, не пустое, уникальное для пользователя

Request	Response
<p>URL: {{host}}/files/<file_id> Method: PATH</p> <p>Headers - Content-Type: application/json</p> <p>Body: { "name": "new Name" }</p>	<p><u>Успех</u> Status: 200 Content-Type: application/json Body: data: { "success": true, "code": 200, "message": "Renamed" }</p> <p><u>Ошибки валидации</u> Формат ответа из общих требований</p> <p><u>Попытка обратиться не авторизованным пользователем</u> Формат ответа из общих требований</p> <p><u>Попытка изменить файл не владельцем</u> Формат ответа о запрете доступа из общих требований</p> <p><u>Попытка доступа к несуществующему объекту</u> Формат ответа из общих требований</p>

Удаление файла:

Request	Response
<p>URL: {{host}}/files/<file_id> Method: DELETE</p> <p>Headers - Content-Type: application/json</p>	<p><u>Успех</u> Status: 200 Content-Type: application/json Body: data: { "success": true, "code": 200, "message": "File deleted" }</p> <p><u>Попытка обратиться не авторизованным пользователем</u> Формат ответа из общих требований</p> <p><u>Попытка удалить файл не владельцем</u> Формат ответа о запрете доступа из общих требований</p> <p><u>Попытка доступа к несуществующему объекту</u> Формат ответа из общих требований</p>

Скачивание файла:

Request	Response
<p>URL: {{host}}/files/<file_id> Method: GET</p>	<p><u>Успех</u> Браузеру отдается файл для скачивания</p> <p><u>Попытка обратиться не авторизованным пользователем</u> Формат ответа из общих требований</p>

	<p><u>Попытка скачать файл к которому нет доступа</u> Формат ответа о запрете доступа из общих требований</p> <p><u>Попытка доступа к несуществующему объекту</u> Формат ответа из общих требований</p>
--	---

Добавление прав доступа:

Функционал добавление прав доступа к файлу должен быть доступен только владельцу файла. В ответ должны возвращаться все пользователи имеющие доступ к объекту.

Request	Response
<p>URL: {{host}}/files/<file_id>/accesses Method: POST</p> <p>Headers - Content-Type: application/json</p> <p>Body:</p> <pre>{ "email": "user@user.ru" }</pre>	<p><u>Успех</u></p> <p>Status: 200 Content-Type: application/json Body:</p> <pre>data: [{ "fullname": "name last_name", "email": "admin@admin.ru", "type": "author", "code": 200, }, { "fullname": "user last_name", "email": "user@user.ru", "type": "co-author", "code": 200, }]</pre> <p><u>Попытка обратиться не авторизованным пользователем</u> Формат ответа из общих требований</p> <p><u>Попытка обратиться не владельцем</u> Формат ответа о запрете доступа из общих требований</p> <p><u>Попытка дать доступ к несуществующему пользователю</u> Формат ответа из общих требований</p> <p><u>Попытка доступа к несуществующему объекту</u> Формат ответа из общих требований</p>

Удаление прав доступа:

Функционал удаления прав доступа к файлу должен быть доступен только создателю. В ответ должны возвращаться все пользователи имеющие доступ к объекту.

Request	Response
<p>URL: {{host}}/files/<file_id>/accesses Method: DELETE</p> <p>Headers - Content-Type: application/json</p> <p>Body:</p> <pre>{}</pre>	<p><u>Успех</u></p> <p>Status: 200 Content-Type: application/json Body:</p> <pre>data: [{ "fullname": "name last_name", "email": "admin@admin.ru", }]</pre>

<pre> "email": "user@user.ru" } </pre>	<pre> "code": 200, "type": "author" }] </pre> <p><u>Попытка удаления пользователя которого нет в списке соавторов</u></p> <p>Формат ответа о ненайденном ресурсе из общих требований</p> <p><u>Попытка удаления самого себя</u></p> <p>Формат ответа о запрете доступа из общих требований</p> <p><u>Попытка обратиться не авторизованным пользователем</u></p> <p>Формат ответа из общих требований</p> <p><u>Попытка обратиться не владельцем</u></p> <p>Формат ответа о запрете доступа из общих требований</p> <p><u>Попытка удаления прав у несуществующего пользователя</u></p> <p>Формат ответа из общих требований</p> <p><u>Попытка доступа к несуществующему объекту</u></p> <p>Формат ответа из общих требований</p>
--	--

Просмотр файлов пользователя:

Request	Response
URL: {{host}}/files/disk Method: GET Headers - Content-Type: application/json	<u>Успех</u> Status: 200 Content-Type: application/json Body: data: [{ "file_id": "qweasd1234", "name": "Имя файла", "code": 200, "url": "{{host}}/files/qweasd1234", "accesses": [[{ "fullname": "name last_name", "email": "admin@admin.ru", "type": "author" }, { "fullname": "user last_name", "email": "user@user.ru", "type": "co-author" }]], "file_id": "aaaaaaaaaaa", "name": "Имя файла 1", "code": 200, }]

	<pre> "url": "{{host}}/files/aaaaaaaaaa", "accesses": [{ "fullname": "name last_name", "email": "admin@admin.ru", "type": "author" }] </pre> <p><u>Попытка обратиться не авторизованным пользователем</u> <u>Формат ответа из общих требований</u></p>
--	---

Просмотр файлов, к которым имеет доступ пользователь:

В списке не должны присутствовать файлы самого пользователя

Request	Response
URL: {{host}}/files/shared Method: GET Headers - Content-Type: application/json	<u>Успех</u> Status: 200 Content-Type: application/json Body: data: [{ "file_id": "qweasd1234", "code": 200, "name": "Имя файла", "url": "{{host}}/files/qweasd1234", }, { "file_id": "aaaaaaaaaaa", "code": 200, "name": "Имя файла 2", "url": "{{host}}/files/aaaaaaaaaa", }]

Попытка обратиться не авторизованным пользователем
Формат ответа из общих требований

Инструкция для конкурсанта

Вам необходимо две тестовые учетные записи со следующими данными:

- email: user1@test.ru, password: Qa1
- email: user2@test.ru, password: As2

Вам предоставляются следующие конфигурации фреймворков и библиотек:

- Laravel 10.x
- Yii Basic 2.0.x
- Django 5.0.x (включая пакеты: djangorestframework, django-cors-headers, pillow, django-filter, mysqlclient)

Вы можете использовать любой из представленных фреймворков.

Модуль Г. Спидтаски

Время на выполнение: 4 часа

Ваша задача - решить максимальное количество небольших задач.

№	Задача
Разработка интерфейса пользователя	
1	Пропорционально уменьшить изображение до 1000x700 (по любой из сторон). <i>Задачу можно выполнить как с помощью HTML+CSS так и с помощью графического редактора</i>
2	Разработать набор иконок для видеоплеера (воспроизведение, пауза, отключение звука, увеличение/уменьшение громкости, быстрая перемотка вперед, назад) <i>Задачу можно выполнить как с помощью HTML+CSS так и с помощью графического редактора</i>
3	С помощью CSS/HTML сделать 3D-куб, который медленно бесконечно вращается
4	Добавьте темно-синий фильтр на изображение и добавьте на изображение текст «Добро пожаловать в наш магазин» <i>Задачу можно выполнить как с помощью HTML+CSS так и с помощью графического редактора</i>
5	Сделать адаптивный макет (только секции) для блога: шапка, основная, боковая панель, футер. Макет должен корректно отображаться в следующих значениях ширины: $\geq 576\text{px}$, $\geq 768\text{px}$, $\geq 992\text{px}$.
6	Создайте веб-страницу с 5 различными уникальными стилями для кнопок, каждая с анимацией наведения и анимацией нажатия, анимации должны уникальными для обоих событий
Разработка на стороне клиента	
7	Создать параллакс эффект из представленных изображений. Сцена остается в неподвижном состоянии. Второй слой (шторы) - при движении мыши (влево/вправо) производят движения, а так же двигается третий слой (люстра), но с меньшим смещением.
8	В центре экрана разместить окружность с радиусом не менее 200px. От центра окружности до края окружности отрисовывается отрезок, следующий за курсором мыши. При наведении в центр окружности отрезок исчезает.
9	В центре страницы поместить квадрат размером не менее 400px. Менять его цвет в зависимости от курсора мыши с плавным переходом цветов. В правом верхнем углу у вас должен быть максимальный красный цвет. В левом нижнем углу у вас должен быть максимальный синий цвет. В правом нижнем углу у вас должен быть максимальный белый цвет. Верхний левый угол является чёрным цветом.
10	Случайное количество окружностей (от 3-х до 5-ти) разного размера (диаметром от 100px до 200px) хаотично перемещаются по экрану. Если они касаются курсора мыши, то должны оттолкнуться от него и изменить свое направление. Так же должна быть коллизия окружностей с краем области.
11	Используя JavaScript, создайте область страницы на которой пользователь может рисовать

	своим курсором при зажатой левой клавиши мыши.
Разработка на стороне сервера	
12	С помощью PHP добавить водяной знак “Professional” в нижнем правом краю изображения, открыть ссылку на измененное изображение. (Использовать файл 12.html для загрузки фотографии в ваш скрипт)
13	Дана строка. Необходимо найти подстроку, которая является самым длинным палиндромом. <i>Wiki: Палиндромом - число, буквосочетание, слово или текст, одинаково читающееся в обоих направлениях. Например, число 101; слова «топот» в русском языке и фин. saippuakivikauppiaas (продавец мыльного камня; торговец стеатитом) — самое длинное слово-палиндром в мире; текст «а роза упала на лапу Азора» и прочие являются палиндромами.</i> Используйте файл 13.html
14	Создайте набор таблиц, необходимых для отношения «многие ко многим» между автомобилями (номер автомобиля(уникальный), пробег) и водителями автомобилей (ФИО, номер водительского удостоверения(уникальный), дата рождения). <i>В директории сохраните файл dump.sql</i>
15	Создайте функцию для сортировки букв заданной строки в порядке возрастания. «backdsts» => «abcdsst» (Использовать файл 15.html)
16	Даны три таблицы: пользователи, товары и заказанные товары. С помощью запроса Вам нужно получить список заказанных товаров, включая идентификатор пользователя, имя пользователя, идентификатор заказа, время заказа, идентификатор товара и название товара. <i>В директории сохраните файл select.sql</i>
Разработка информационных ресурсов с использованием готовых решений <i>Каждая задача может быть выполнена как в составе одного проекта WordPress, так и в отдельном проекте.</i>	
17	Установите wordpress. Данные для авторизации администратора: Логин: admin2023 Пароль: password2
18	Создайте плагин, который реализовывает шорткод [current-datetime]. Шорткод выводит текущую дату и время сервера в формате Д Месяц(текстом) ГГГГ ЧЧ:ММ (3 июля 2023 14:59)
19	Создайте плагин, который добавляет новый тип поста reviews. Плагин должен также реализовать два шорткода: [reviews-form] - выводит форму добавления отзыва в любом стиле. Отзыв может добавить как авторизованный пользователь так и гость. Форма состоит из двух обязательных полей: имени и сообщение. [reviews-list] - выводит список отзывов в любом оформлении. Отзыв состоит из имени, сообщения и даты публикации
20	Возьмите предоставленный шаблон и создайте тему на основе этого шаблона. Все стили должны быть сохранены как и интерактив, который реализовывается через JavaScript. Тема должна выглядеть идентично шаблону. Выводить посты и прочую информацию не нужно,

нужно вывести только admin-bar для администратора. Тема должна отображаться в списке тем, изображение темы в списке должно соответствовать виду шаблона

Инструкция для конкурсанта

Для выполнения заданий по технологии “Разработка информационных ресурсов с использованием готовых решений” вам предоставляется CMS Wordpress 6.2.x. Остальные задачи необходимо решить без использования фреймворков и сторонних библиотек.

Разработанные задания должен быть размещены по адресу <http://xxxxxx-m4.moscow.ru/y/>, где xxxxxx - логин участника (указан на индивидуальной карточке), у — номер задачи. Медиа-файлы задач должны быть скопированы в соответствующую папку задачи. Также в медиа возможны уточняющие требования, специфичные для конкретной задачи.

Ваша задача - решить максимальное количество небольших задач.

№	Задача
Разработка интерфейса пользователя	
1	Пропорционально уменьшить изображение до 1000x700 (по любой из сторон). <i>Задачу можно выполнить как с помощью HTML+CSS так и с помощью графического редактора</i>
2	Разработать набор иконок для видеоплеера (воспроизведение, пауза, отключение звука, увеличение/уменьшение громкости, быстрая перемотка вперед, назад) <i>Задачу можно выполнить как с помощью HTML+CSS так и с помощью графического редактора</i>
3	С помощью CSS/HTML сделать 3D-куб, который медленно бесконечно вращается
4	Добавьте темно-синий фильтр на изображение и добавьте на изображение текст «Добро пожаловать в наш магазин» <i>Задачу можно выполнить как с помощью HTML+CSS так и с помощью графического редактора</i>
5	Сделать адаптивный макет (только секции) для блога: шапка, основная, боковая панель, футер. Макет должен корректно отображаться в следующих значениях ширины: $\geq 576\text{px}$, $\geq 768\text{px}$, $\geq 992\text{px}$.
6	Создайте веб-страницу с 5 различными уникальными стилями для кнопок, каждая с анимацией наведения и анимацией нажатия, анимации должны уникальными для обоих событий
Разработка на стороне клиента	
7	Создать параллакс эффект из представленных изображений. Сцена остается в неподвижном состоянии. Второй слой (шторы) - при движении мыши (влево/вправо) производят движения, а так же двигается третий слой (люстра), но с меньшим смещением.
8	В центре экрана разместить окружность с радиусом не менее 200px. От центра окружности до края окружности отрисовывается отрезок, следующий за курсором мыши. При наведении в центр окружности отрезок исчезает.
9	В центре страницы поместить квадрат размером не менее 400px. Менять его цвет в зависимости от курсора мыши с плавным переходом цветов. В правом верхнем углу у вас должен быть максимальный красный цвет. В левом нижнем углу у вас должен быть максимальный синий цвет. В правом нижнем углу у вас должен быть максимальный белый цвет. Верхний левый угол является чёрным цветом.
10	Случайное количество окружностей (от 3-х до 5-ти) разного размера (диаметром от 100px до 200px) хаотично перемещаются по экрану. Если они касаются курсора мыши, то должны оттолкнуться от него и изменить свое направление. Также должна быть коллизия окружностей с краем области.
11	Используя JavaScript, создайте область страницы на которой пользователь может рисовать своим курсором при зажатой левой клавиши мыши.