



ВСЕРОССИЙСКОЕ
ЧЕМПИОНАТНОЕ
ДВИЖЕНИЕ
ПО ПРОФЕССИОНАЛЬНОМУ
МАСТЕРСТВУ

КОНКУРСНОЕ ЗАДАНИЕ
КОМПЕТЕНЦИИ
«Веб-технологии»

г. Тула 2024

Конкурсное задание разработано экспертным сообществом и утверждено Менеджером компетенции, 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:

Оглавление

1. ОСНОВНЫЕ ТРЕБОВАНИЯ КОМПЕТЕНЦИИ	3
1.1. ОБЩИЕ СВЕДЕНИЯ О ТРЕБОВАНИЯХ КОМПЕТЕНЦИИ	3
1.2. ПЕРЕЧЕНЬ ПРОФЕССИОНАЛЬНЫХ ЗАДАЧ СПЕЦИАЛИСТА ПО КОМПЕТЕНЦИИ «ВЕБ-ТЕХНОЛОГИИ»	3
1.3. ТРЕБОВАНИЯ К СХЕМЕ ОЦЕНКИ	8
1.4. СПЕЦИФИКАЦИЯ ОЦЕНКИ КОМПЕТЕНЦИИ	9
1.5. КОНКУРСНОЕ ЗАДАНИЕ	9
1.5.1. Разработка/выбор конкурсного задания	9
1.5.2. Структура модулей конкурсного задания (инвариант/ вариатив)	10
Модуль А. Разработка интерфейса пользователя	10
Модуль Б. Разработка Веб-приложения на стороне клиента	18
Модуль В. Разработка Веб-приложения на стороне сервера	22
Модуль Г. Разработка ИР с использованием готовых решений	33
2. СПЕЦИАЛЬНЫЕ ПРАВИЛА КОМПЕТЕНЦИИ	37
2.1. Личный инструмент конкурсанта	37
2.2. Материалы, оборудование и инструменты, запрещенные на площадке	37

ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ

1. *IP – Информационный ресурс*
2. *SSH – SSH (от англ. secure shell – безопасная оболочка) – это защищённый сетевой протокол для удалённого управления сервером через интернет.*
3. *FTP – File Transfer Protocol», или «протокол передачи файлов» – это набор процедур или правил, позволяющих электронным устройствам взаимодействовать между собой.*
4. *CMS – Content Management System) – это система создания и управления сайтом. Это визуально удобный интерфейс, с помощью которого можно добавлять и редактировать содержимое сайта.*

1. ОСНОВНЫЕ ТРЕБОВАНИЯ КОМПЕТЕНЦИИ

1.1. ОБЩИЕ СВЕДЕНИЯ О ТРЕБОВАНИЯХ КОМПЕТЕНЦИИ

Требования компетенции (ТК) «Веб-технологии» определяют знания, умения, навыки и трудовые функции, которые лежат в основе наиболее актуальных требований работодателей отрасли.

Целью соревнований по компетенции является демонстрация лучших практик и высокого уровня выполнения работы по соответствующей рабочей специальности или профессии.

Требования компетенции являются руководством для подготовки конкурентоспособных, высококвалифицированных специалистов / рабочих и участия их в конкурсах профессионального мастерства.

В соревнованиях по компетенции проверка знаний, умений, навыков и трудовых функций осуществляется посредством оценки выполнения практической работы.

Требования компетенции разделены на четкие разделы с номерами и заголовками, каждому разделу назначен процент относительной важности, сумма которых составляет 100.

1.2. ПЕРЕЧЕНЬ ПРОФЕССИОНАЛЬНЫХ ЗАДАЧ СПЕЦИАЛИСТА ПО КОМПЕТЕНЦИИ «ВЕБ-ТЕХНОЛОГИИ»

Таблица №1

Перечень профессиональных задач специалиста

№ п/п	Раздел	Важность в %
1	Тестирование информационных ресурсов Специалист должен знать и понимать: – способы решения возникающих проблем, анализ проблемной ситуации, возникшей в ходе решения профессиональных задач, пути их решения с учетом этических норм и правил, опираясь на профессиональную этику; – принципы, лежащие в основе сбора и представления информации; – основные приемы и методы визуального представления информации (черновое макетирование страниц, объектно-событийное моделирование, создание блок-схем и др.);	7

	<ul style="list-style-type: none"> – английский язык в рамках чтения и понимания официальной технической документации по используемым технологиями и языкам программирования. <p>Специалист должен уметь:</p> <ul style="list-style-type: none"> – собирать, анализировать и оценивать информацию; – использовать навыки грамотности для толкования стандартов и требований; – составлять тестовую документации для тестирования новых функциональностей продукта – проводить ручное тестирование новых функциональностей – проводить регрессионное ручное тестирование – вести баг-репорты – составлять отчеты по итогам тестирования – автоматизировать регрессионное тестирование – общаться с заказчиком, командой разработки и тестирования 	
2	<p>Техническая поддержка и администрирование информационных ресурсов</p> <p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> – принципы и практики, которые позволяют продуктивно работать, в том числе в команде; – аспекты систем, которые позволяют повысить продуктивность и выработать оптимальную стратегию; – основные принципы выбора технологий и инструментария для решения поставленных задач (проектов); - основные подходы к планированию и документированию проекта. <p>Специалист должен уметь:</p> <ul style="list-style-type: none"> – формировать архитектуру проекта (программного продукта) в соответствии с последними отраслевыми решениями; – выбирать технологии и инструменты для решения поставленных задач; – планировать график рабочего дня с учетом требований; – планировать задачи, учитывать временные ограничения и сроки; – решать распространенные задачи веб-дизайна и разработки кода; – формировать тестовые наборы, применять инструменты автоматического тестирования; – производить отладку кода программ и находить ошибки; – оптимально использовать компьютерное оборудование и программное обеспечение для повышения эффективности своей работы; – использовать менеджеры пакетов при разработке проекта; - использовать систему контроля версий. 	7
3	<p>Разработка интерфейса пользователя</p> <p>Специалист должен знать и понимать:</p>	26

	<ul style="list-style-type: none"> – структуру и общепринятые элементы веб-страниц различных видов и назначений; – основные принципы организации контента веб-приложения; – основные правила выбора цвета, работы с типографикой и композицией; – принципы и методы создания и адаптации графики для использования ее на веб-сайтах; – ограничения, которые накладывают мобильные устройства и разрешения экранов при использовании их для просмотра веб-сайтов; – принципы построения эстетичного и креативного дизайна; – методы обеспечения доступа к страницам веб-сайтов аудитории с ограниченными возможностями; – World Wide Web Consortium (W3C) стандарты HTML и CSS; – методы верстки веб-сайтов и их стандартную структуру; – Web accessibility initiative (WAI) стандарт доступности активных Интернет-приложений для людей с ограниченными возможностями; – основные принципы применения соответствующих CSS правил и селекторов для получения ожидаемого результата; – лучшие практики для Search Engine Optimization (SEO) и интернет-маркетинга; 	
	<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> – создавать, использовать и оптимизировать изображения для веб-сайтов; – выбирать дизайнерское решение, которое будет наиболее подходящим для целевого рынка; – принимать во внимание влияние каждого элемента, который добавляется в проект во время разработки дизайна; – использовать все требуемые элементы при разработке дизайна; – создавать «отзывчивый» дизайн, который будет отображаться корректно на различных устройствах и при разных разрешениях; – создавать html-страницы сайта на основе предоставленных графических макетов их дизайна; – корректно использовать CSS для обеспечения единого дизайна в разных браузерах; – создавать адаптивные веб-страницы, которые способны оставаться функциональными на различных устройствах при разных разрешениях; – создавать веб-сайты полностью соответствующие текущим стандартам W3C (http://www.w3.org); – создавать и модифицировать веб-интерфейсы с учетом принципов Search Engine Optimization; – использовать препроцессоры. 	
4	<p>Разработка на стороне клиента</p> <p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> – основные принципы паттерновой разработки веб-приложений; 	25

	<ul style="list-style-type: none"> – ECMAScript (JavaScript); – принципы, особенности и способы использования открытых фреймворков; – принципы разработка кода с использованием открытых библиотек; – различные интерфейсы взаимодействия с объектами браузера <p>Специалист должен уметь:</p> <ul style="list-style-type: none"> – создавать и модифицировать JavaScript код для улучшения функциональности и интерактивности сайта; – манипулировать элементами страницы веб-приложения; – разрабатывать анимацию для повышения доступности и визуальной привлекательности веб-приложения; – применять открытые библиотеки и фреймворки; – тестировать веб-приложение. 	
5	<p>Разработка веб приложения на стороне сервера</p> <p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> – процедурные и объектно-ориентированные языки PHP, Python, Node.js; – основные принципы и правила использования открытых библиотек и фреймворков; – распространенные модели организации и хранения данных; – основные принципы создания баз данных; – основные принципы обмена данными между клиентом и сервером; – методы работы с протоколами SSH/(s)FTP при подключении к серверам; – способы разработки программного кода в соответствии с паттернами проектирования; – основные принципы обеспечения безопасности веб-приложения. <p>Специалист должен уметь:</p> <ul style="list-style-type: none"> – разрабатывать процедурный и объектно-ориентированный программный код; – разрабатывать веб-сервисы с применением PHP, Python, Node.js в соответствии с техническим заданием; – создавать библиотеки и модули для выполнения повторяющихся задач; – разрабатывать веб-приложения с доступом к различным базам данных; – создавать SQL (Structured Query Language) запросы и конструкции; – обеспечивать безопасность (устойчивость веб-приложения к атакам и взломам); – интегрировать существующий и создавать новый программный код с API (Application Programming Interfaces); – использовать открытые библиотеки и фреймворки; 	25

6	Разработка информационных ресурсов с использованием готовых решений	10
	<p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> – преимущества и ограничения системы управления контентом с открытым исходным кодом; – методы работы с плагинами/модулями; – способы реализации функциональных возможностей CMS; – основные принципы организации контента веб-приложения; – понимать необходимость поддержания и обновления для плагинов CMS и соответствующих модулей для безопасности системы; <p>основные принципы интеграции с внешними веб-приложениями.</p>	
	<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> – устанавливать, настраивать и модифицировать систему управления контентом; – устанавливать, настраивать и обновлять плагины/модули CMS; – создать пользовательские темы/шаблоны для системы управления контентом; – создавать пользовательские плагины/модули и шаблоны/темы; – использовать встроенные методы и средства CMS при разработке веб-приложения. 	

1.3. ТРЕБОВАНИЯ К СХЕМЕ ОЦЕНКИ

Сумма баллов, присуждаемых по каждому аспекту, должна попадать в диапазон баллов, определенных для каждого раздела компетенции, обозначенных в требованиях и указанных в таблице №2.

Таблица №2

Матрица пересчета требований компетенции в критерии оценки

Критерий/Модуль						Итого баллов за раздел ТРЕБОВАНИЙ КОМПЕТЕНЦИИ
Разделы ТРЕБОВАНИЙ КОМПЕТЕНЦИИ		А	Б	В	Г	
	1	2	1	2,5	1,5	7
	2	2	2	2	1	7
	3	20	1		5	26
	4		23		2	25
	5			22	3	25
6				10	10	
Итого баллов за критерий/модуль		24	27	26,5	22,5	22,5

1.4. СПЕЦИФИКАЦИЯ ОЦЕНКИ КОМПЕТЕНЦИИ

Оценка Конкурсного задания будет основываться на критериях, указанных в таблице №3:

Таблица №3

Оценка конкурсного задания

Критерий		Методика проверки навыков в критерии
А	Разработка интерфейса пользователя	Проверка результата выполнения модуля после его завершения
Б	Разработка Веб-приложения на стороне клиента	Проверка результата выполнения модуля после его завершения
В	Разработка Веб-приложения на стороне сервера	Проверка результата выполнения модуля после его завершения
Г	Разработка ИР с использованием готовых решений	Проверка результата выполнения модуля после его завершения

1.5. КОНКУРСНОЕ ЗАДАНИЕ

Общая продолжительность Конкурсного задания¹: 12 ч.

Количество конкурсных дней: 3 дня

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

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

1.5.1. Разработка/выбор конкурсного задания

(<https://disk.yandex.ru/i/XJfQr1jBmZ2rew>)

Конкурсное задание состоит из 4 модулей, включает обязательную к выполнению часть (инвариант) – 3 модуля, и вариативную часть – 1 модуль. Общее количество баллов конкурсного задания составляет 100.

Обязательная к выполнению часть (инвариант) выполняется всеми регионами без исключения на всех уровнях чемпионатов.

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

Количество модулей из вариативной части, выбирается регионом самостоятельно в зависимости от материальных возможностей площадки соревнований и потребностей работодателей региона в соответствующих специалистах. В случае если ни один из модулей вариативной части не подходит под запрос работодателя конкретного региона, то вариативный (е) модуль (и) формируется регионом самостоятельно под запрос работодателя. При этом, время на выполнение модуля (ей) и количество баллов в критериях оценки по аспектам не меняются.

Таблица №4

Матрица конкурсного задания

Обобщенная трудовая функция	Трудовая функция	Нормативный документ/ЗУН	Модуль	Константа/вариатив	ИЛ	КО
1	2	3	4	5	6	7

Инструкция по заполнению матрицы конкурсного задания (**Приложение № 1**)

1.5.2. Структура модулей конкурсного задания (инвариант/ вариатив)

Модуль А. Разработка интерфейса пользователя

Время на выполнение модуля: 4 часа

Задания:

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

К вам обратилась организация, оказывающая помощь в поиске потерянных домашних животных с просьбой разработать сервис “Новая жизнь”, на котором пользователи могут оставлять информацию о найденных животных, а владельцы, у которых пропал питомец, могут их искать.

Подобных ресурсов достаточное количество, а заказчик хочет в кратчайшие сроки занять лидирующие позиции в этой нише (максимально увеличить количество пользователей, оставляющих информацию о найденных животных и находящихся в процессе их поиска). Для этого вам необходимо использовать в дизайне сверстанных страниц 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.

Подразумевается, что пользователями сайта являются жители г. Оренбург, которые обращаются к ресурсу с помощью смартфонов. Также заказчик хотел бы, чтобы сайтом активно пользовались бы представители молодежных волонтерских организаций.

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

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – сверстать следующие страницы веб-сайта:

1. Главная
2. Страница входа в личный кабинет
3. Страница регистрации в личном кабинете
4. Страница личного кабинета пользователя
5. Страница добавления информации о найденном животном
6. Страница личного кабинета модератора (управление объявлениями)
7. Страница поиска животных
8. Страница с карточкой найденного животного

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

Страницы должны быть адаптивны. При переходе от десктопной версии главной страницы (ширина экрана 1920px) к мобильной версии (ширина экрана 768px) должно быть реализовано не менее четырех структурных изменений контента.

На остальных разрешениях контент должен отображаться корректно.

Основные роли пользователей

Гость

1. Регистрация
2. Авторизация
3. Добавление объявлений о найденном животном
4. Поиск животного по базе существующих объявлений
- 5.

Авторизованный пользователь

1. Добавление объявления о найденном животном
2. Поиск животного по базе существующих объявлений
3. Просмотр своих объявлений
4. Просмотр информации о себе с возможностью редактирования некоторых данных
5. Выход из профиля

Модератор

1. Подтверждения объявлений и их модерация
2. Выход

Главная - Landing Page

Главная страница должна содержать следующие блоки:

1. Шапка сайта.
 - 1.1. Логотип.

Вам предоставлен существующий логотип компании logo.png, однако заказчик просит разработать и использовать на сайте новый логотип. Учтите, что новый логотип должен содержать основные цвета сайта, сохранять преемственность с существующим, но, тем не менее, иметь отличия от предоставленного.
 - 1.2. Меню навигации.
 - 1.2.1. Поиск.
 - 1.2.2. Регистрация.
 - 1.2.3. Личный кабинет.
 - 1.2.4. Добавить.
 - 1.2.5. Отзывы.
2. Вывод фотографий недавно добавленных объявлений с найденными животными 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. Вид животного (например, кот, собака, суслик, хорек, вомбат и т.д.).
 - 2.3. Краткое описание обстоятельств, при которых животное было найдено.
 - 2.4. Кнопка “подробнее” для перехода на страницу с описанием найденного животного.
3. Секция с быстрым поиском животного, например, по виду животного. Предусматривает наличие одного поля ввода и кнопки для поиска. В поле ввода должны выводиться подсказки по поиску.
4. Факты о компании и сайте
 - 4.1. Помогли найти более 500 животных.
 - 4.2. Более трех лет способствуем возвращению питомцев к хозяевам.
 - 4.3. Все услуги оказываются бесплатно.

Около каждого из пунктов заказчик просит разработать и разместить соответствующее графическое изображение.

5. Карточки с найденными животными. В каждой карточке отображается фото, район, где животное было найдено, контактный номер телефона нашедшего, дата размещения карточки, индикатор, кем было добавлено объявление (зарегистрированным или незарегистрированным пользователем)
Общее количество карточек с найденными животными - 6. Верстка не должна нарушаться, если объявлений будет меньше.
6. Форма поиска. Должна содержать следующие поля ввода:
 - 6.1. Район.
 - 6.2. Вид животного (например, кошка, собака, суслик, хорек и т.д.)
 - 6.3. Поле ввода текста для поиска по виду животного
 - 6.4. Кнопка поиска питомца.
7. Отзывы владельцев питомцев, которые уже нашли потерянное животное, в виде слайдсета 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. Фото животного
 - 7.3. Отзыв
 - 7.4. Дата отзываПорядок 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 “8 (800)123-45-67”
 - 9.2. e-mail: mail@newlife.ru
 - 9.3. Навигация по сайту:
 - 9.3.1. Главная
 - 9.3.2. Регистрация
 - 9.3.3. Авторизация
 - 9.3.4. Личный кабинет
 - 9.3.5. Найдено животное
 - 9.3.6. Поиск

Страница входа в личный кабинет

На этой странице вам необходимо сделать форму со следующими полями:

- e-mail
- Пароль
- Кнопка для входа
- Кнопка восстановления пароля

Страница регистрации в личном кабинете

На этой странице вам необходимо сделать форму со следующими полями, обязательными для заполнения:

- 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.

Страница личного кабинета пользователя

Общая информация о пользователе (в любом порядке):

- номер телефона и кнопка “Изменить”,
- e-mail и кнопка “Изменить”,
- количество добавленных объявлений,
- количество животных, которые вернулись к хозяевам,
- дата регистрации пользователя,
- количество дней, прошедших с момента регистрации на сайте,
- информация по объявлениям, созданным пользователем ранее.

Эти объявления должны быть сгруппированы по статусам “Активно”, “На модерации”, “Найдено”, “В архиве”. Каждая из групп должна иметь собственное оформление и визуально отличаться от других. Каждая группа должна содержать минимум 2 объявления.

В каждом объявлении должна быть отражена следующая информация: район, в котором было найдено животное; дата, когда объявление было добавлено; дополнительная информация (например, при каких обстоятельствах, порода); клеймо (если есть), фото. Для объявлений со статусами “Активно” и “На модерации” должны быть кнопки “Удалить” и “Редактировать”.

Страница добавления информации о найденном животном

Информацию о найденном животном может добавлять, как авторизованный пользователь, так и гость. Для гостя предусмотрена возможность создать профиль, для этого необходимо выбрать соответствующую опцию.

Ниже представлены поля, которые необходимо заполнить пользователю при добавлении объявления о найденном животном.

- Контактный номер телефона нашедшего питомца

- e-mail человека, нашедшего животное
- Элемент выбора опции автоматической регистрации пользователя при добавлении объявления о найденном животном. Если пользователь выбрал вариант с регистрацией, то появляется блок с полями для ввода пароля и его подтверждения (по умолчанию он скрыт). Отображение блока происходит без использования js.
Если пользователь решил после этого отказаться от регистрации, блок с полями для ввода паролей скрывается (также без использования js).
- вид животного
- Фото найденного животного. Отобразите в верстке возможность добавления нескольких фотографий, но не более 3 фотографий.
- Дополнительная информация, например, описание обстоятельств, при которых было найдено животное, порода, кличка-и т.д.
- Клеймо (если есть)
- Район
- Дата, когда было найдено животное
- Элемент, подтверждающий согласие на обработку персональных данных

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

Страница личного кабинета модератора

На странице должны отображаться объявления пользователей о найденных животных. Для каждого объявления должна отображаться вся информация, указанная пользователем, нашедшим животное.

Модератор может сортировать объявления по дате, статусу - вам необходимо предусмотреть соответствующие элементы управления.

По умолчанию у новых объявлений установлен статус “На модерации”. Модератору должны быть доступны элементы управления, позволяющие изменить статус объявления на “Активно”, “Найдено”, “В архиве”.

Страница с результатами поиска животных

Попасть на эту страницу можно с главной (из формы поиска животных). Вам необходимо продублировать форму поиска с главной страницы. На этой странице необходимо отобразить в виде таблицы информацию обо всех найденных питомцах, а именно:

- вид животного
- Фото найденного животного (одно)
- Дополнительная информация, например, описание обстоятельств, при которых было найдено животное

- Клеймо (если есть)
- Район
- Дата, когда было найдено животное
- Контактный номер телефона нашедшего питомца

Вам следует использовать пагинацию для отображения результатов поиска - не более 10 животных на одной странице.

Страница с карточкой найденного животного

На странице отображается следующая информация:

- 3 фотографии животного
- вид животного
- Имя нашедшего
- Контактный номер телефона нашедшего питомца
- e-mail человека, нашедшего животное
- Дополнительная информация, например, описание обстоятельств, при которых было найдено животное
- Клеймо (если есть)
- Район
- Дата, когда было найдено животное

Порядок и вид отображения полей может быть произвольный

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

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

Страницы должны быть согласованы между собой.

Вам нужно использовать не менее одного из представленных шрифтов.

Оценка будет производиться в браузере Google Chrome.

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

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

- Главная - index.html, index.png
- Страница входа в личный кабинет - login.html, login.png
- Страница регистрации в личном кабинете - register.html, register.png
- Страница личного кабинета пользователя - profile.html, profile.png
- Страница добавления информации о найденном животном - add.html, add.png

- Страница личного кабинета модератора (управление объявлениями) - orders-management.html, orders-management.png
 - Страница поиска животных - search.html, search.png
 - Страница с карточкой найденного животного - pet.html, pet.png
- Макеты должны быть расположены в отдельной папке (/design).

Все страницы указанные выше должны быть доступны к просмотру по соответствующим адресам: <http://xxxxxx-m1.domain.ru/index.html>, <http://xxxxxx-m1.domain.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!

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

Модуль Б. Разработка Веб-приложения на стороне клиента

Время на выполнение модуля: 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.

В примерах будет использоваться переменная `{{host}}` которая обозначает адрес API: `http://xxxxxx-m2.domain.ru/`.

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 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 авторизованный 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.wsr.ru>, где xxxxxx - логин участника.

Для оценки качества кода необходимо выгрузить на сервер также не скомпилированный вариант проекта.

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 выгрузить на сервер также не скомпилированный вариант проекта.

Модуль В. Разработка Веб-приложения на стороне сервера

Время на выполнение модуля: 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.

Общее

Вам необходимо реализовать REST API заданной структуры.

В примерах будет использоваться переменная `{{host}}` которая обозначает адрес `http://xxxxxx-m3.wsr.ru/`, 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"
}
```

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

Status: 422

Content-Type: application/json

Body:

```
{
  "success": false,
  "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,
  "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.

Вы должны завести тестовый аккаунт с email test@gmail.com и паролем test.

Описание функционала гостя

Аутентификация:

При успешной аутентификации возвращается сгенерированный токен пользователя.

Request	Response
URL: {{host}}/authorization Method: POST Headers - Content-Type: application/json Body: { "email": "admin@admin.ru",	<p style="text-align: right;"><u>Успех</u></p> Status: 200 Content-Type: application/json Body: { "success": true, "message": "Success", "token": <сгенерированный token> }

<pre>"password": "Qa1" }</pre>	<p style="text-align: center;"><u>Неправильные логин или пароль</u></p> <p>Status: 401 Content-Type: application/json Body: <pre>{ "success": false, "message": "Login failed" }</pre></p>
--------------------------------	--

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,
обязательное
- last_name - фамилия
обязательное

Request	Response
<p>URL: {{host}}/registration Method: POST</p> <p>Headers - Content-Type: application/json</p> <p>Body: <pre>{ "email": "admin@admin.ru", "password": "Qa1", "first_name": "name", "last_name": "last_name" }</pre></p>	<p style="text-align: center;"><u>Успех</u></p> <p>Status: 200 Content-Type: application/json Body: <pre>{ "success": true, "message": "Success", "token": <сгенерированный token> }</pre></p> <hr/> <p style="text-align: center;"><u>Ошибки валидации</u></p> <p>Формат ответа из общих требований</p>

Описание функционала авторизованного пользователя

Сброс авторизации:

Запрос предназначен для очистки значение токена пользователя.

Request	Response
URL: {{host}}/logout Method: GET	<p style="text-align: center;"><u>Успех</u></p> <p>Status: 200 Content-Type: application/json Body: { "success": true, "message": "Logout" }</p> <hr/> <p style="text-align: center;"><u>Попытка обратиться не авторизованным пользователем</u></p> <p>Формат ответа из общих требований</p>

Загрузка файлов:

Функция должна принимать следующие параметры:

- files - файлы, разрешены файлы размером не более 2 мб, 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
---------	----------

<p>URL: {{host}}/files</p> <p>Method: POST</p> <p>Headers</p> <p>- Content-Type: multipart/form-data</p> <p>FormData:</p> <p>“files”:<массив с файлами></p>	<p style="text-align: right;"><u>Успех</u></p> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>[{ "success": true, "message": "Success", "name": "Имя загруженного файла", "url": "{{host}}/files/qweasd1234", "file_id": "qweasd1234" }, { "success": false, "message": "File not loaded", "name": "Имя НЕ загруженного файла" }]</pre> <hr/> <p style="text-align: center;"><u>Ошибки валидации</u></p> <p style="text-align: center;">Вывод в общем списке загруженных файлов (см. пример выше)</p> <hr/> <p style="text-align: center;"><u>Попытка обратиться не авторизованным ПОЛЬЗОВАТЕЛЕМ</u></p> <p style="text-align: center;">Формат ответа из общих требований</p>
---	--

Редактирование файла:

Функция должна принимать следующие параметры:

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

Request	Response
<p>URL: {{host}}/files/<file_id></p> <p>Method: PATCH</p> <p>Headers</p> <p>- Content-Type: application/json</p>	<p style="text-align: right;"><u>Успех</u></p> <p>Status: 200</p> <p>Content-Type: application/json</p> <p>Body:</p> <pre>{ "success": true, "message": "Renamed" }</pre>

Body: <pre>{ "name": "new Name" }</pre>	<pre>}</pre> <hr/> <p style="text-align: center;"><u>Ошибки валидации</u> Формат ответа из общих требований</p> <hr/> <p style="text-align: center;"><u>Попытка обратиться не авторизованным пользователем</u> Формат ответа из общих требований</p> <hr/> <p style="text-align: center;"><u>Попытка изменить файл не владельцем</u> Формат ответа о запрете доступа из общих требований</p> <hr/> <p style="text-align: center;"><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</p> <p>- Content-Type: application/json</p>	<p style="text-align: center;"><u>Успех</u></p> <p>Status: 200 Content-Type: application/json Body:</p> <pre>{ "success": true, "message": "File already deleted" }</pre> <hr/> <p style="text-align: center;"><u>Попытка обратиться не авторизованным пользователем</u> Формат ответа из общих требований</p> <hr/> <p style="text-align: center;"><u>Попытка удалить файл не владельцем</u> Формат ответа о запрете доступа из общих требований</p> <hr/> <p style="text-align: center;"><u>Попытка доступа к несуществующему объекту</u></p>

	Формат ответа из общих требований
--	--

Скачивание файла:

Request	Response
URL: {{host}}/files/<file_id> Method: GET	<p style="text-align: center;"><u>Успех</u></p> <p>Браузеру отдается файл для скачивания</p> <hr/> <p style="text-align: center;"><u>Попытка обратиться не авторизованным пользователем</u></p> <p style="text-align: center;">Формат ответа из общих требований</p> <hr/> <p style="text-align: center;"><u>Попытка скачать файл к которому нет доступа</u></p> <p style="text-align: center;">Формат ответа о запрете доступа из общих требований</p> <hr/> <p style="text-align: center;"><u>Попытка доступа к несуществующему объекту</u></p> <p style="text-align: center;">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
URL: {{host}}/files/<file_id>/access es Method: POST Headers - Content-Type: application/json Body: { "email": "user@user.ru"	<p style="text-align: center;"><u>Успех</u></p> <p>Status: 200 Content-Type: application/json Body: [{ "fullname": "name last_name", "email": "admin@admin.ru", "type": "author" }, { "fullname": "user last_name",</p>

<pre> } </pre>	<pre> "email": "user@user.ru", "type": "co-author" }] </pre> <hr/> <p style="text-align: center;"><u>Попытка обратиться не авторизованным пользователем</u> Формат ответа из общих требований</p> <hr/> <p style="text-align: center;"><u>Попытка обратиться не владельцем</u> Формат ответа о запрете доступа из общих требований</p> <hr/> <p style="text-align: center;"><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>/access es</p> <p>Method: DELETE</p> <p>Headers</p> <p>- Content-Type: application/json</p> <p>Body: { "email": "user@user.ru" }</p>	<p style="text-align: center;"><u>Успех</u></p> <p>Status: 200 Content-Type: application/json Body: [{ "fullname": "name last_name", "email": "admin@admin.ru", "type": "author" }]</p> <hr/> <p style="text-align: center;"><u>Попытка удаления пользователя которого нет в списке соавторов</u> Формат ответа о ненайденном ресурсе из</p>

	общих требований
	<u>Попытка удаления самого себя</u> Формат ответа о запрете доступа из общих требований
	<u>Попытка обратиться не авторизованным пользователем</u> Формат ответа из общих требований
	<u>Попытка обратиться не владельцем</u> Формат ответа о запрете доступа из общих требований
	<u>Попытка доступа к несуществующему объекту</u> Формат ответа из общих требований

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: <pre>[{ "file_id": "qweasd1234", "name": "Имя файла", "url": "{{host}}/files/qweasd1234", "accesses": [{ "fullname": "name last_name", "email": "admin@admin.ru", "type": "author" }, { "fullname": "user last_name", "email": "user@user.ru",</pre>

	<pre> "type": "co-author" }] }, { "file_id": "aaaaaaaaaa", "name": "Имя файла 1", "url": "{{host}}/files/aaaaaaaaaa", "accesses": [{ "fullname": "name last_name", "email": "admin@admin.ru", "type": "author" }] }] </pre> <hr/> <p style="text-align: center;"> <u>Попытка обратиться не авторизованным</u> <u>пользователем</u> Формат ответа из общих требований </p>
--	---

Просмотр файлов, к которым имеет доступ пользователь:

В списке не должны присутствовать файлы самого пользователя

Request	Response
<p>URL: {{host}}/shared Method: GET</p> <p>Headers - Content-Type: application/json</p>	<p style="text-align: center;"><u>Успех</u></p> <p>Status: 200 Content-Type: application/json Body: [{ "file_id": "qweasd1234", "name": "Имя файла", "url": "{{host}}/files/qweasd1234",</p>

	<pre>}, { "file_id": "aaaaaaaaaa", "name": "Имя файла 2", "url": "{{host}}/files/aaaaaaaaaa", }]</pre> <hr/> <p><u>Попытка обратиться не авторизованным пользователем</u> Формат ответа из общих требований</p>
--	---

Инструкция для конкурсанта

Вам предоставляются следующие конфигурации фреймворков и библиотек:

- Laravel 10.x
- Yii Basic 2.0.x
- Django 4.0.x (включая пакеты: djangorestframework, django-cors-headers, pillow, django-filter, mysqlclient)

Вы можете использовать любой из представленных фреймворков.

Модуль Г. Разработка ИР с использованием готовых решений

Время на выполнение модуля: 2 часа

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

Задание. К вам обратилась администрация Тульской области. Сайт краеведческого музея давно устарел и нуждается в обновлении. Вам предоставляют готовый HTML-шаблон, который нужно немного модернизировать и разработать новый сайт с использованием системы управления контентом WordPress.

ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!

ОПИСАНИЕ ПРОЕКТА И ЗАДАЧ

Ваша задача – разработать сайт музея с использованием предоставленного HTML-шаблона, указанной структурой и требуемого функционала.

Структура главной страницы

Главная страница должна содержать следующие секции/блоки последовательно:

- Меню (навигация по странице или к отдельной странице):
 - О музее (секция);
 - Экспонаты (отдельная страница);
 - Контакты (секция);
 - Ростелеком (отдельная страница);
 - Минсвязи (отдельная страница);
 - ГПОУ ТО «ТЭК» (отдельная страница)

- Слайдер (не менее трех слайдов с использованием стороннего плагина).
 - Описание событий;

- О музее (краткая информация о музее из медиа).

- Интересные экспонаты (от 6 штук, с возможностью перехода на отдельную страницу экспоната). Решение о публикации экспоната принимает администратор сайта, путем изменения признака необходимости вывода экспоната на главную страницу. Секция должна иметь ссылку на страницу со всеми экспонатами. Элементы карточки экспоната:
 - Главное изображение;
 - Краткое название;
 - Год выпуска;

- Наши контакты должна содержать:
 - Номер телефона;
 - Адрес музея;
 - Правила посещения;
 - Время работы;
 - Социальные сети;

- Форма заявки на посещение музея:
 - Дата посещения;
 - Время посещения;
 - Имя;
 - Номер WhatsApp;

- Кнопка отправки формы;
- Карта (заглушка в виде изображения из медиа).
- Подвал сайта:
 - Контакты;
 - Адрес музея;
 - Навигация по сайту;

Страница со всеми экспонатами

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

- Кости, кожа, ткани;
- Керамика, стекло;
- Оружие;
- Археологическое собрание;
- Изобразительные источники;
- Восточная коллекция;

Карточка экспоната содержит:

- Главное изображение;
- Краткое название;
- Год выпуска;

Страница экспоната

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

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

- Описание;

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

Плагины

Вам необходимо разработать два плагина.

Random Line

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

QRPage

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

ИНСТРУКЦИЯ ДЛЯ КОНКУРСАНТА

Разработанный веб-сайт должен быть доступен по адресу <http://xxxxxx-m4.wsr.ru/>, где xxxxxx - логин участника (указан на индивидуальной карточке).

Вся информация (например, заголовки, текст, меню и т.д.) должна редактироваться в панели управления администратором сайта без знаний программирования, верстки или доступа к файловой системе сервера.

2. СПЕЦИАЛЬНЫЕ ПРАВИЛА КОМПЕТЕНЦИИ

2.1. Личный инструмент конкурсанта

Конкурсанту разрешается использовать собственные:

- клавиатуру на любом языке. Если конкурсант пользуется своей клавиатурой, и она выходит из строя, организатор предоставляет ему замену;

- языковые файлы для клавиатуры;
- мышь;
- графический планшет;
- наушники;
- аудиофайлы с музыкальными композициями (не более 30 файлов в формате mp3). Файлы предоставляются на флеш-носителях в день С-1 техническому эксперту на проверку.

Все оборудование не должно содержать встроенной памяти.

2.2. Материалы, оборудование и инструменты, запрещенные на площадке

Всё оборудование, принесенное конкурсантами, может быть проверено экспертами на наличие внутренних запоминающих устройств. В случае обнаружения материалы будут изыматься.

Экспертам допускается использовать персональные компьютеры, но в специальной зоне. В помещениях для проведения оценки использование любых электронных устройств запрещено, кроме специально организованных для оценки.

Также запрещено приносить:

- дополнительные программы и библиотеки, не предусмотренные инфраструктурным листом;
- мобильные телефоны;
- фото/видео устройства;
- карты памяти и другие носители информации;
- внутренние устройства памяти в собственном оборудовании.