****

КОНКУРСНОЕ ЗАДАНИЕ КОМПЕТЕНЦИИ

«Веб-технологии»

*Основная*

Финал Чемпионата по профессиональному мастерству «Профессионалы» в 2025 г

*г. Нижний Новгород, Федеральный технопарк профессионального образования*

2025 г

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:**

[1. ОСНОВНЫЕ ТРЕБОВАНИЯ КОМПЕТЕНЦИИ 4](#_Toc196911952)

[1.1. ОБЩИЕ СВЕДЕНИЯ О ТРЕБОВАНИЯХ КОМПЕТЕНЦИИ 4](#_Toc196911953)

[1.2. ПЕРЕЧЕНЬ ПРОФЕССИОНАЛЬНЫХ ЗАДАЧ СПЕЦИАЛИСТА ПО КОМПЕТЕНЦИИ «ВЕБ-ТЕХНОЛОГИИ» 4](#_Toc196911954)

[1.3. ТРЕБОВАНИЯ К СХЕМЕ ОЦЕНКИ 9](#_Toc196911955)

[1.4. СПЕЦИФИКАЦИЯ ОЦЕНКИ КОМПЕТЕНЦИИ 10](#_Toc196911956)

[1.5. КОНКУРСНОЕ ЗАДАНИЕ 10](#_Toc196911957)

[1.5.1. Разработка/выбор конкурсного задания 11](#_Toc196911958)

[1.5.2. Структура модулей конкурсного задания (инвариант/вариатив) 11](#_Toc196911959)

[2. СПЕЦИАЛЬНЫЕ ПРАВИЛА КОМПЕТЕНЦИИ 30](#_Toc196911960)

[2.1. Личный инструмент конкурсанта 30](#_Toc196911961)

[2.2. Материалы, оборудование и инструменты, запрещенные на площадке 30](#_Toc196911962)

[3. ПРИЛОЖЕНИЯ 31](#_Toc196911963)

**ИСПОЛЬЗУЕМЫЕ СОКРАЩЕНИЯ**

*1. ФГОС – Федеральный государственный образовательный стандарт*

*2. ПС – Профессиональный стандарт*

*3. КЗ – Конкурсное задание*

*4. ИЛ – Инфраструктурный лист*

# 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** | 7 |
| 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.);  − английский язык в рамках чтения и понимания официальной технической документации по используемым технологиями и языкам программирования. |
| Специалист должен уметь:  − 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** | **Техническая поддержка и администрирование информационных ресурсов** | 7 |
| Специалист должен знать и понимать:  − 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. |
| Специалист должен уметь:  − 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. |
| **3** | **Разработка интерфейса пользователя** | 26 |
| Специалист должен знать и понимать:  − 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; |
| Специалист должен уметь:  − 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** | **Разработка на стороне клиента** | 25 |
| Специалист должен знать и понимать:  − основные принципы паттерновой разработки веб-приложений;  − ECMAScript (JavaScript);  − принципы, особенности и способы использования открытых фреймворков;  − принципы разработка кода с использованием открытых библиотек;  − различные интерфейсы взаимодействия с объектами браузера |
| Специалист должен уметь:  − 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** | **Разработка веб приложения на стороне сервера** | 25 |
| Специалист должен знать и понимать:  − 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. |
| Специалист должен уметь:  − 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; |
| **6** | **Разработка информационных ресурсов с использованием готовых решений** | 10 |
| Специалист должен знать и понимать:  − 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;  основные принципы интеграции с внешними веб-приложениями. |
| Специалист должен уметь:  − 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 в индивидуальном формате**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Критерий/Модуль** | | | | | **Итого баллов за раздел ТРЕБОВАНИЙ КОМПЕТЕНЦИИ** |
| **Разделы ТРЕБОВАНИЙ КОМПЕТЕНЦИИ** |  | **A** | **Б** | **В** |  |
| **1** | 2 | 3 | 2 | **7** |
| **2** | 2 | 2 | 3 | **7** |
| **3** | 24 |  | 2 | **26** |
| **4** |  |  | 25 | **25** |
| **5** |  | 25 |  | **25** |
| **6** | 5 | 3 | 2 | **10** |
| **Итого баллов за критерий/модуль** | | **33** | **33** | **34** | **100** |

**Матрица пересчета требований компетенции в критерии оценки в командном формате**

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий/Модуль** | | | **Итого баллов за раздел ТРЕБОВАНИЙ КОМПЕТЕНЦИИ** |
| **Разделы ТРЕБОВАНИЙ КОМПЕТЕНЦИИ** |  | **Г** |  |
| **1** | 5 | **5** |
| **2** | 5 | **5** |
| **3** | 5 | **5** |
| **4** | 5 | **5** |
| **5** | 5 | **5** |
| **6** | 5 | **5** |
| **Итого баллов за критерий/модуль** | | **30** | **30** |

## 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 сервера** | Проверка результата выполнения модуля после его завершения |
| **В** | **Разработка Веб-приложения на стороне клиента** | Проверка результата выполнения модуля после его завершения |
| **Г** | **Система управления мероприятиями (EventHub)** | Проверка результата выполнения модуля после его завершения |

## 1.5. КОНКУРСНОЕ ЗАДАНИЕ

Количество конкурсных дней: 4 дня (Д1-Д3 в индивидуальном формате, Д4 в командном формате).

Общая продолжительность Конкурсного задания: 22 часа, в том числе выполнение заданий:

Д1-Д3 в индивидуальном формате (модули А, Б, В) – 18 часов.

Д4 в командном формате (модуль Г) – 4 часа.

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

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

### **1.5.1. Разработка/выбор конкурсного задания**

Конкурсное задание состоит из 4-х модулей, включает обязательную к выполнению часть (инвариант) –3 модуля. Из них Модуль Г. направлен на командообразование внутри одной компетенции с распределением функциональных задач для реализации производственной задачи с соблюдением цикла производства (проекта).

Общее количество баллов конкурсного задания в индивидуальном формате составляет 100, в командном формате – 30 баллов.

### **1.5.2. Структура модулей конкурсного задания (инвариант/вариатив)**

<https://disk.yandex.ru/d/3EdhAY5WOpJSiw>

**Модуль А. Разработка интерфейса пользователя (Инвариант)**

*Время на выполнение модуля: 6 часов*

Стек: HTML, CSS

**Задания:**

Концепция сервиса

“Зеленый помощник” — это онлайн-платформа для поиска и бронирования экологически чистых продуктов и услуг. Сервис объединяет:

* Фермерские хозяйства
* Эко-отели
* Эко-туристические маршруты
* Органические продукты питания
* Экологические услуги (например, переработка отходов)

Целевая аудитория

* Частные пользователи:
  + Экологически сознательные горожане
  + Семьи с детьми
  + Люди, следящие за здоровьем
  + Туристы
* Корпоративные клиенты:
  + Компании, организующие эко-командировки
  + Туристические агентства
  + Эко-организации
  + Образовательные учреждения

Требования к проекту

1. Адаптивность
   1. Сайт должен корректно отображаться на всех устройствах
   2. Минимум 4 структурных изменения при переходе от десктопной версии (1920px) к мобильной (768px)
2. Запрет на фреймворки
   1. Запрещено использование Bootstrap, Tailwind и других CSS-фреймворков
3. Система контроля версий
   1. Обязательное использование Git
   2. Минимум 3 осмысленных коммита

Структура проекта

Ваша задача – сверстать следующие страницы веб-сайта:

1. Главная страница
2. Личный кабинет
3. Страница входа
4. Страница регистрации
5. Пользовательский профиль
6. История заказов
7. Избранное
8. Административная панель
9. Управление предложениями
10. Модерация заявок
11. Статистика
12. Каталог услуг
13. Страница поиска
14. Карточка предложения
15. Форма добавления нового предложения

Детализация страниц

* Главная страница
  + Логотип и навигация
  + Слайдер с эко-предложениями
  + Быстрый поиск по категориям
  + Инфографика о преимуществах сервиса
  + Популярные эко-продукты
  + Форма подписки
* Личный кабинет
  + Информация о пользователе
  + История заказов
  + Избранное
  + Настройки профиля
* Страница предложения
  + Фото и описание
  + Характеристики
  + Отзывы
  + Форма бронирования
* Административная панель
  + Управление предложениями
  + Модерация заявок
  + Статистика использования
  + Уведомления

Технические требования

* Общие требования
  + Адаптивность (4+ точки брейкпоинта)
  + Валидный HTML/CSS
  + Кроссбраузерная совместимость
  + Оптимизированные изображения
  + Быстрая загрузка
* Дизайн-принципы
  + Экологическая цветовая гамма
  + Минималистичный стиль
  + Интуитивно понятный интерфейс
  + Четкая структура контента
  + Анимации и переходы
* Требования к коду
  + Семантическая вёрстка
  + БЭМ методология
  + Комментарии к коду
  + Оптимизация производительности
  + Доступность (A11y)

Требования к названию файлов

* index.html (главная)
* login.html (вход)
* register.html (регистрация)
* profile.html (личный кабинет)
* add.html (добавление предложения)
* admin.html (админка)
* search.html (поиск)
* offer.html (карточка предложения)

Сверстанный веб-сайт должен быть доступен по адресу http://xxxxxx-m1.web.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.

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

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

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

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

**Модуль Б. Разработка Веб-приложения на стороне сервера (Инвариант)**

*Время на выполнение модуля: 6 часов*

Стек: PHP(Python), MySQL

**Задание**

Вам необходимо разработать серверную часть интернет-магазина. Задание включает в себя:

* разработку административной панели по предоставленному шаблону;
* реализацию REST API по предоставленной спецификации.

Админ-панель

Панель должна быть реализована на основе предоставленного HTML-шаблона.

Функционал доступен только авторизованному пользователю, в объеме, предусмотренным его ролью.

При работе с формами необходимо реализовать валидацию:

* + Поля с ошибками должны быть визуально выделены.
  + Под каждым некорректным полем должно отображаться текстовое сообщение об ошибке.

Категории

Экран авторизации

Экран авторизации содержит поля для ввода email и пароля, а также кнопку отправки формы.

При вводе некорректных данных должно отображаться соответствующее сообщение об ошибке.

Оба поля являются обязательными для заполнения. *Работает правило валидации форм в админ панели.*

После успешной авторизации администратора перенаправляет на список категорий.

Категории

Администратор должен иметь возможность управлять категориями, а именно:

* создавать категории
* редактировать категории
* удалять категории
* просматривать категории

При создании и редактировании категории администратор должен заполнять следующие поля:

* название категории  
  обязательно, 15 символов
* описание категории  
  не обязательно, 50 символов

Просмотр списка категорий должен включать в категории с идентификатором, названием, описанием и количеством товаров в данной категории.

Список должен иметь пагинацию по 5 на страницу.

В списке должны быть кнопки удаления, редактирования и просмотра. Кнопка удаления должна отсутствовать, если в категории есть товары.

Просмотр категории включает в себя список товаров. Каждый товар содержит идентификатор, название и цену, кнопки/ссылки просмотра, редактирования и удаления.

Товары

В админ панели администратор должен иметь возможность создавать, удалять, редактировать и просматривать товары.

Создание и редактирования товаров

Форма содержит следующие поля:

* Название товара  
  обязательное, максимум 20 символов
* Описание товара  
  не обязательное, максимум 50 символов
* Цена товара  
  обязательное, число больше 10
* Изображение товара  
  обязательное при создании, при редактировании, если не указано, то остается старое изображение, jpg, максимум 2 мегабайта

При создании или редактировании товара, если было загружено изображение, для него необходимо автоматически создавать миниатюру с максимальными размерами 300×300 пикселей. Миниатюра должна сохранять пропорции изображения и полностью его отображать, как при использовании свойства contain в CSS.

На миниатюру нужно наложить водяной знак — текстовую надпись “Shop” в правом нижнем углу. Оригинал изображения можно не сохранять и не отображать.

Просмотр товара включает в себя отображение всех его полей включая изображение с водяным знаком.

Удаление товара происходит без каких-либо ограничений по логике, после удаления товар больше не отображается в админ-панели и в API.

Просмотр заказов

В админ-панели должен быть разработан просмотр заказов пользователей. В списке заказов должны быть отражены следующие поля:

* название товара
* email пользователя, заказавшего товар
* цена товара
* статус: оплачено / не оплачено

API

API должно предоставлять пользователю следующий функционал:

* регистрация
* авторизация
* просмотр категорий и товаров
* покупку товара
* просмотр списка покупок

### 

Общие требования

Тело запросов должно быть формате json как при отправке, так и при получении.

Авторизация должна работать по заголовку Authorization с использованием Bearer-токена.

Вам необходимо придерживаться одного шаблона, если пользователь попытался отправить невалидные данные:

Status code: 422

Body:

{

“message”: “Invalid fields”,

“errors”: {

“{имя поля}”: [  
 “{текст ошибки}”

]

}

}

Попытка получения доступа к функционалу авторизованного пользователя гостем:

Status code: 403

Body:

{

“message”: “Forbidden for you”,

}

Регистрация

|  |  |
| --- | --- |
| Запрос | Пример ответа |
| Method: POST  Link: {{api-host}}/registration Body:  {  “email”: “[name@example.com](mailto:name@example.com)”,  “password”: “password”  } | Успешный Status code: 201 Body: {  “success”: true } |

Валидация полей:

email - обязательное, валидный email, уникальный

password - обязательное, минимум 3 символа

Авторизация

|  |  |
| --- | --- |
| Запрос | Пример ответа |
| Method: POST  Link: {{api-host}}/auth Body:  {  “email”: “[name@example.com](mailto:name@example.com)”,  “password”: “password”  } | Успешный  Status code: 200 Body: {  “token”: “{токен авторизации}” }  Некорректные данные:  Status code: 422  {  “message”: “Invalid data”,  “errors”: {  “email”: [  “Invalid data”  ]  }  } |

Валидация полей:

email - обязательное

password - обязательное

Получения списка категорий

|  |  |
| --- | --- |
| Запрос | Пример ответа |
| Method: GET  Link: {{api-host}}/categories | Успешный  Status code: 200 Body: {  “data”: [  {  “id”: {идентификатор},  “name”: “{название}”,  “description”: “{описание}”  },  … {остальные категории}  ] } |

Получения списка товаров категории

|  |  |
| --- | --- |
| Запрос | Пример ответа |
| Method: GET  Link: {{api-host}}/categories/{category\_id}/products | Успешный  Status code: 200 Body: {  “data”: [  {  “id”: {идентификатор},  “name”: “{название}”,  “description”: “{описание}”,  “price”: {цена},  “image\_url”: “{ссылка на изображение}”  },  … {остальные товары}  ] } |

Просмотр товара

|  |  |
| --- | --- |
| Запрос | Пример ответа |
| Method: GET  Link: {{api-host}}/products/{product\_id} | Успешный  Status code: 200 Body: {  “data”: {  “id”: {идентификатор},  “name”: “{название}”,  “description”: “{описание}”,  “price”: {цена},  “image\_url”: “{ссылка на изображение}”  }, } |

Покупка товара

|  |  |
| --- | --- |
| Запрос | Пример ответа |
| Method: POST  Headers:  Authorization: Bearer {token}  Link: {{api-host}}/products/{product\_id}/buy | Успешный  Status code: 200 Body: {  “pay\_url”: “{ссылка на оплату товара}” } |

При покупке товара вы должны отправить запрос на сторонний сервер для получения ссылки на покупку товара. В данном запросе помимо цены вы должны отправить ссылку на маршрут подтверждения или отклонения покупки.

|  |  |
| --- | --- |
| Запрос | Пример ответа |
| Method: POST  Link: <http://localhost:8081/payments>  Body:  {  “price”: {цена},  “webhook\_url”: “{{api-host}}/payment-webhook”  } | Успешный  Status code: 200 Body: {  “pay\_url”: “{ссылка на оплату товара}”,  “order\_id”: “{идентификатор заказа, который придет на webhook url для изменения статуса}” } |

Webhook для платежной системы

|  |  |
| --- | --- |
| Запрос | Пример ответа |
| Method: POST  Link: {{api-host}}/payment-webhook  Body:  {  “order\_id”: {идентификатор заказа},  “status”: “{success | failed}”  } | Любой  Status code: 204  Body: отсутствует |

На данный webhook будет отправлен запрос от платежной системы, когда пользователь перейдет по ссылке и вводит реквизиты карты. Статус будет зависеть от того, какие реквизиты введет пользователь. order\_id будет тот, который платежная система отправила в ответ на запрос получения ссылки на оплату.

Реквизиты для успешной покупки: 8888 0000 0000 1111

Реквизиты для безуспешной покупки: 8888 0000 0000 2222

Просмотр списка заказов пользователя

|  |  |
| --- | --- |
| Запрос | Пример ответа |
| Method: GET  Headers:  Authorization: Bearer {token}  Link: {{api-host}}/orders | Успешный  Status code: 200  Body:  {  “data”: [  {  “id”: {идентификатор},  “status”: {pending(в процессе) | success(оплачен) | failed(ошибка)},  “product”: {  “id”: {идентификатор},  “name”: “{название}”,  “description”: “{описание}”,  “price”: {цена},  “image\_url”: “{ссылка на изображение}”  }  }  ]  } |

После того, как пользователь перешел по ссылке для оплаты, статус меняется в зависимости от того, какие реквизиты карты он ввел.

Админ панель должна быть доступна по адресу {{site-url}}/admin.

API должно быть доступа по адресу {{site-url}}/api

Вам необходимо завести аккаунт администратора со следующими реквизитами:

логин: [admin@shop.com](mailto:admin@shop.com)

пароль: shop2015

Если реквизиты для аккаунта администратора будут неверные, то вы рискуете потерять баллы как за разработку админ-панели, так и за разработку API, если не будет возможности проверить функционал работы с товарами.

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

Разрешены следующие фреймворки: Laravel (10, 11, 12 последних версий), Django (5.1.7), Yii 2 (basic, advanced посленийх версий).

Для работы с базой данных вы должны использовать MySQL с предоставленными вам реквизитами подключения.

**Модуль В. Разработка Веб-приложения на стороне клиента (Инвариант)**

*Время на выполнение модуля: 6 часов*

Стек: JS, Vue, React

**Задание**

Вам необходимо разработать одностраничное приложение (SPA) для платформы объявлений. Пользователь должен иметь возможность:

* зарегистрироваться и авторизоваться;
* просматривать список объявлений (гость и авторизованный);
* добавлять новые объявления;
* редактировать и удалять свои объявления;
* получать уведомления о новых объявлениях в реальном времени через WebSocket (гость и авторизованный).

Необходимо использовать предоставленный шаблон. Шаблон должен отображаться корректно.

Документация к API вам будет предоставлена в медиафайлах.

Общие требования

* Приложение должно быть реализовано как SPA.
* Все взаимодействие с сервером осуществляется через предоставленный REST API.
* Для получения уведомлений о новых объявлениях используется WebSocket-соединение.
* При всех действиях должна быть реализована валидация через запрос на сервер. Ошибки должны быть отображены в удобном виде рядом с соответствующим полем, поля должны подсвечиваться.
* Должна быть реализована базовая навигация (например, с использованием React Router или Vue Router).
* Нужно использовать предоставленный шаблон

Функциональность приложения

Авторизация и регистрация

Пользователь должен иметь возможность зарегистрироваться и авторизоваться в системе. После авторизации должен использоваться токен в заголовке Authorization.

Регистрация:

* email
* пароль

Авторизация:

* email;
* пароль.

После успешной авторизации пользователь перенаправляется на страницу списка объявлений.

После успешной регистрации пользователь перенаправляется на страницу авторизации.

#### 

#### Список объявлений

На главной странице отображается список всех объявлений. Для каждого объявления выводится:

* название;
* описание;
* цена;
* изображение.

Объявления должны быть отсортированы по дате создания (новые вверху). Должна быть предусмотрена сортировка по цене в обе стороны.

#### 

#### Список своих объявлений

На странице отображается список своих объявлений. Для каждого объявления выводится:

* название;
* описание;
* цена;
* изображение;
* кнопка удаления;
* кнопка редактирования.

#### Просмотр объявления

При просмотре объявления отображается:

* название;
* описание;
* цена;
* изображение;
* контакт пользователя (email) отображается только при запросе(клике).

#### Создание и редактирование объявления

Авторизованный пользователь может создать объявление. Форма содержит следующие поля:

* название
* описание
* цена
* изображение

Редактировать можно только свои объявления.

Удаление объявления

Пользователь может удалить только свои объявления. После удаления оно должно исчезнуть из общего списка.

#### Уведомления в реальном времени (WebSocket)

Когда любое новое объявление создается в системе, уведомление о нем должно появиться у всех пользователей, находящихся на странице со списком, без перезагрузки страницы. Для этого необходимо использовать WebSocket-подключение к серверу и слушать соответствующий канал.

Инструкция

* Приложение должно быть доступно по адресу {{host}}
* Приложение должно быть собрано и опубликовано на сервере без необходимости дополнительных настроек. (билд на сервере)
* Все данные хранятся на сервере. Данные не должны сохраняться в локальном хранилище браузера, кроме токена авторизации.
* Исходники проекта вам необходимо сохранить в папке source на сервере
* **ВНИМАНИЕ! Проверяться будут только работы, загруженные на сервер!**

**Модуль Г. Система управления мероприятиями (EventHub) (Командный модуль)**

*Время на выполнение модуля: 4 часа*

Стек: PHP(Django), MySQL, JS, Vue, React, Figma

**Задание**

Разработать полноценную веб-систему для организации мероприятий (например, конференций, семинаров, встреч), включающую регистрацию пользователей, расписание, бронирование мест, оплату билетов и админ-панель.

Интернет будет предоставлен.

Задание будет выполняться в командах по 3 конкурсанта.

Состав:

* Разработчик API и логики
* Разработчик авторизации/регистрации и безопасности
* Разработчик работы с базой данных и архитектуры
* Документация API (Swagger/OpenAPI или Postman)

Задачи:

● Создать REST API для:

○ Регистрации и логина пользователей

○ Просмотра и бронирования мероприятий

○ Админ-панели: CRUD для событий, пользователей, билетов

● Интеграция с MySQL (реляционная структура)

● Валидация, обработка ошибок

● Базовые тесты и документация

Frontend + UI/UX

Состав:

* Верстальщик и UI/UX дизайнер
* Разработчик клиентской логики Vue/React
* Разработчик панели администратора
* Интегратор API + стилизация

Задачи:

● Создание дизайна в Figma (если предусмотрено) или адаптивной верстки

● Реализация SPA-интерфейса:

○ Главная страница со списком мероприятий

○ Регистрация и вход

○ Просмотр деталей мероприятия и бронирование

○ Личный кабинет пользователя

○ Админ-панель (для CRUD операций)

● Интеграция с API

● Обработка ошибок, уведомления

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

# 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 Д-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.

# 3. ПРИЛОЖЕНИЯ

Приложение №1. Инструкция по заполнению матрицы конкурсного задания

Приложение №2. Матрица конкурсного задания

Приложение №3. Инструкция по охране труда по компетенции «Веб-технологии».

Приложение № 4. Медиа файлы