

ООО «АКАДЕМИЯ ЛАД»

УТВЕРЖДЕНО  
Директором ООО «Академия Лад»



А.В. Усков

(Приказ №1 от 23 мая 2023 г.)

**ДОПОЛНИТЕЛЬНОЕ ОБРАЗОВАНИЕ**  
**ДОПОЛНИТЕЛЬНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ**  
**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА -**  
**ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

«Веб-разработка»

(Наименование программы)

144 часа

**Нижний Новгород**

**2023**

## **1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**1.1.** Цель программы: совершенствование и получение новой компетенции, профессиональных знаний, которые позволят реализовать себя в сфере web - разработки.

**1.2.** Нормативные документы для разработки программы повышения квалификации:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;

- Приказ Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;

- Профессиональный стандарт 06.035 “Разработчик WEB и мультимедийных приложений” утвержденный приказом Министерством труда и социальной защиты Российской Федерации от от 18 января 2017 года N 44н

**1.3.** Категории слушателей на обучение которых рассчитана программа повышения квалификации (далее – Программа): лица, имеющие среднее профессиональное или высшее образование.

**1.4.** Входные требования к обучающимся:

Наличие базовых навыков работы с персональным компьютером.

**1.5.** Сфера применения слушателями полученных профессиональных компетенций, умений и знаний. Знания, полученные в ходе программы могут быть использованы на предприятиях малого и среднего бизнеса, в частной практике при реализации деятельности по web - разработке.

**1.6.** Программа реализуется исключительно с применением электронного обучения и дистанционных образовательных технологий.

## **2. ХАРАКТЕРИСТИКА ПОДГОТОВКИ ПО ПРОГРАММЕ**

**2.1.** Нормативный срок освоения программы 144 часа.

**2.2.** Срок обучения 20 недель

**2.3.** Общая трудоемкость 4 ЗЕ.

**2.4.** Режим обучения 7,2 часа в неделю.

### **3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ**

Слушатель, освоивший программу, должен:

**3.1.** обладать профессиональными компетенциями, включающими в себя способность:

ПК-1. Техническая поддержка процессов, создание (модификация) и анализ и сопровождение информационных ресурсов

ПК-2. Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов

ПК-3. управлять процессами и проектами по созданию (модификации) информационных ресурсов

**3.2.** знать:

- основы веб-программирования на языке JavaScript;
- основы разработки веб-приложений на React;
- основные технологии разработки сайтов (flexbox, css grid)
- инструменты разработки веб-приложений;
- методы и приемы отладки программного кода.

**3.3.** уметь:

- создавать сайты с фиксированной шириной, гибкой вёрсткой и адаптировать сайты под различные размеры экрана;
- выявлять ошибки в программном коде;
- разрабатывать сайты и приложения на языке JavaScript, в том числе с использованием библиотек, инструментов и фреймворков (React, React Redux);
- работать в команде.

**3.4.** владеть:

- современными информационными технологиями и навыками работы с программным обеспечением для разработки веб-приложений;

- технологиями разработки веб-приложений с использованием языка JavaScript и фреймворка React.

**3.5.** Сфера применения слушателями полученных профессиональных компетенций, умений и знаний.

Знания, полученные в ходе программы могут быть использованы на предприятиях малого и среднего бизнеса, в частной практике при реализации деятельности по веб-разработке.

#### **4. ТРЕБОВАНИЯ К СТРУКТУРЕ ПРОГРАММЫ**

Программа предусматривает изучение следующих дисциплин:

Дисциплина «Вёрстка сайтов»

Дисциплина «Программирование на JavaScript»

Дисциплина «Frontend-разработка на React»

#### **5. ТРЕБОВАНИЯ К ОЦЕНКЕ КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

«Веб-разработка»

Процедура **промежуточной аттестации** предусматривает выполнение практических заданий по темам дисциплин.

Процедура **итоговой аттестации** предусматривает выполнение итогового проекта в команде. При подготовке итогового проекта слушатель должен продемонстрировать полученные умения и навыки в области веб-разработки. Текущий контроль проверки качества освоения дисциплин программы повышения квалификации осуществляется в форме устного собеседования.

## Формы и методы контроля и оценки результатов освоения дисциплин

№ п/п	Наименование процедуры	Основные показатели оценки	Формы и методы контроля и оценки
1.	Дисциплина «Вёрстка сайтов»	<p><i>знает</i> основные инструменты и технологии разработки веб-приложений (flexbox, css grid), методы и приемы отладки программного кода</p> <p><i>умеет</i> создавать сайты с фиксированной шириной, гибкой вёрсткой и адаптировать сайты под различные размеры экрана, выявлять ошибки в программном коде, работать в команде</p> <p><i>владеет</i> современными информационными технологиями и навыками работы со специальным программным обеспечением для разработки веб-приложений; технологиями разработки веб-приложений.</p>	Текущий контроль - устный опрос. Промежуточная аттестация - выполнение практических заданий.
2.	Дисциплина «Программирование на JavaScript»	<p><i>знает</i> основы веб-программирования на языке JavaScript, методы и приемы отладки программного кода</p> <p><i>умеет</i> решать задачи программирования на JavaScript.</p> <p><i>владеет</i> современными информационными технологиями и навыками программирования на языке JavaScript</p>	Текущий контроль - устный опрос. Промежуточная аттестация - выполнение практических заданий.
3.	Дисциплина «Frontend-разработка на React»	<p><i>знает</i> основы разработки веб-приложений на React, инструменты разработки веб-приложений, методы и приемы отладки программного кода</p> <p><i>умеет</i> разрабатывать сайты и приложения на языке JavaScript, в том числе с использованием библиотек, инструментов и фреймворков (React, React Redux), работать в команде</p> <p><i>владеет</i></p>	Текущий контроль - устный опрос. Промежуточная аттестация - выполнение практических заданий.

		современными информационными технологиями и навыками работы со специальным программным обеспечением для разработки веб-приложений; технологиями разработки веб-приложений с использованием языка JavaScript и фреймворка React	
4.	Итоговая аттестация	Работа в качестве frontend-разработчика на стажировке.	Итоговый проект

**Примеры вопросов для текущего контроля освоения учебного материала:**

1. Чем отличается тег `<div>` от тега `<span>` в HTML? Какие задачи они обычно выполняют?
2. Каким образом можно создать адаптивный (responsive) дизайн веб-страницы с использованием CSS? Опишите основные подходы.
3. Что такое селекторы в CSS? Приведите примеры различных типов селекторов и их применение.
4. Каким образом можно центрировать элементы на веб-странице с помощью CSS? Объясните различные методы.
5. Что такое блочная модель в CSS? Какие атрибуты она включает в
6. Как объявить переменную в JavaScript с использованием ключевых слов `let`, `const` и `var`? В чем различия между ними?
7. Что такое функции обратного вызова (callback functions) в JavaScript? Как они используются в асинхронном программировании?
8. Каким образом можно выполнить итерацию по элементам массива в JavaScript? Приведите примеры использования циклов и методов массивов.
9. Что такое объектно-ориентированное программирование (ООП) в контексте JavaScript? Как создать класс и экземпляр класса?
10. Каким образом можно обрабатывать ошибки (exceptions) в JavaScript с помощью конструкции `try-catch`? Приведите примеры использования.

11. Что такое JSX в React? В чем преимущества его использования по сравнению с обычным JavaScript?
12. Каким образом можно передавать данные от родительского компонента к дочернему в React? Опишите механизм использования props.
13. Что такое состояние (state) компонента в React? Как оно устанавливается и изменяется?
14. Каким образом можно организовать обработку событий (event handling) в React? Приведите примеры использования.
15. Что такое жизненный цикл компонента в React? Опишите основные методы жизненного цикла и их назначение.
16. Каким образом можно организовать маршрутизацию (routing) в React-приложении? Какие библиотеки или подходы можно использовать для этого?
17. В чем заключается концепция виртуального DOM (virtual DOM) в React? Какие преимущества она предоставляет при разработке интерфейсов?
18. Каким образом можно выполнять запросы к серверу (AJAX) в React-приложении? Опишите основные подходы и библиотеки.
19. Что такое контекст (context) в React? Как он используется для передачи данных через дерево компонентов?
20. Каким образом можно оптимизировать производительность React-приложения? Укажите основные методы оптимизации.

**Примеры практических заданий для осуществления промежуточной аттестации:**

### **Дисциплина «Вёрстка сайтов»**

1. В файле index.html описать структуру страницы в тэгах. Ссылка на макет  
<https://www.figma.com/file/Itn0zsUrZWQXCgNuPSpxIS/%D1%81%D0%B8%D0%BD%D0%B8%D0%B9-%D0%BB%D1%83%D0%B3?node-id=0>

[%3A1&t=iaICgm9EJa7GTndx-0](#). Верстать не надо Создавать style.css не надо.

2. Для проекта в задании 1 создать style.css. Применить свойства, которые проходили на занятии (размеры и цвет шрифта, ширина и высота блоков).

3. Сверстать макет

<https://www.figma.com/file/Itn0zsUrZWQXCgNuPSpxIS/%D1%81%D0%B8%D0%BD%D0%B8%D0%B9-%D0%BB%D1%83%D0%B3?node-id=0%3A1&t=iaICgm9EJa7GTndx-0> на flexbox и CSS Grid.

4. Описать структуру страницы согласно БЭМ. Верстка страницы на flex и grid с переиспользованием классов и с учетом codeguide. Ссылка на макет

<https://www.figma.com/file/NyOokSAieFmcsBK3rW0wwN/%D0%94%D0%BE%D1%81%D0%BA%D0%B0-%D0%BE%D0%B1%D1%8A%D1%8F%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9?node-id=0-1&t=rXzoWsah9vg09rVj-0> .

### **Дисциплина «Программирование на JavaScript»**

1. Придумать наименования переменных для следующих примеров:

- Переменная для “названия нашей планеты”?
- Переменная для “текущее время пользователя”?
- Переменная которая показывает “количества статей”?
- Переменная которая показывает “это оплата наличными деньгами или нет”?
- Три переменные для хранения Ф.И.О

2. Решить задачи с использованием циклов while и for.

- Вывести в консоль заданную строку N раз.
- Ежедневно количество доступных автомобилей в салоне уменьшается в два раза. Выяснить, на какой день продаж, количество доступных к



покупке авто станет меньше  $M$ , если известно, что в первый день продаж всего было  $N$  автомобилей.

- Проанализировав временной промежуток начиная с 1800 и до 2022 года найти и вывести в консоль: Год первого полета человека в космос и количество итераций которое потребовалось для поиска. Количество високосных годов принадлежащих данному отрезку и количество итераций которое потребовалось для поиска.

### 3. Стрелочные функции

- Напишите стрелочную функцию которая будет выводить переданную строку в консоль  $n$  раз.
- Напишите стрелочную функцию, которая будет принимать в качестве параметра букву и если она гласная, функция будет возвращать `true`, в противном случае `false`.
- Напишите стрелочную функцию, которая будет возвращать `true` если строка является палиндромом и `false` в противном случае.

### 4. Функции

- Реализовать функцию которая будет принимать числовой диапазон в качестве параметров `[min, max]` и будет возвращать случайное целое число из данного диапазона.
- Реализовать функцию которая будет определять, в каком регистре записан  $n$  элемент переданной строки, если в верхнем то вернуть `true`, в противном случае вернуть `false`.
- Реализовать функцию которая заменяет в строке `str`, все вхождения подстроки `find`, на подстроку `replace`.

## Дисциплина «Frontend-разработка на React»

### 1. Создание компонента "Список задач" (To-Do List):

- Создайте компонент для отображения списка задач.

- Реализуйте функционал добавления новой задачи, удаления задачи и отметки задачи как выполненной.
- Добавьте возможность фильтрации задач по статусу (выполнено/не выполнено).

## 2. Разработка простого приложения "Калькулятор":

- Создайте компонент калькулятора, который позволяет выполнять базовые математические операции (сложение, вычитание, умножение, деление).
- Добавьте функционал для очистки экрана и выполнения вычислений.

## 3. Интеграция с API для отображения данных:

- Используйте API для получения данных (например, список пользователей, постов или другую информацию).
- Создайте компоненты для отображения этих данных в удобном формате (список, карточки и т. д.).

## 4. Форма обратной связи:

- Разработайте форму обратной связи с полями для ввода имени, email, сообщения.
- Добавьте валидацию полей (например, проверку на корректность email) и отправку данных на сервер (можно использовать фейковый API).

## 5. Реализация маршрутизации в приложении:

- Создайте несколько страниц в вашем приложении (например, домашняя страница, страница о нас, контакты).
- Используйте библиотеку React Router для организации навигации между этими страницами.

## 6. Аутентификация и авторизация:

- Разработайте механизм аутентификации пользователей (например, форма входа).

- Реализуйте защиту определенных страниц доступом только для авторизованных пользователей.

### Критерии оценки текущих практических заданий по дисциплине

На основании выполненных практических заданий обучающемуся определяется оценка – «зачтено», «не зачтено».

Оценка	Уровень подготовки
Зачтено	Хорошая подготовка. Обучающийся выполнил все предложенные задания.
Не зачтено	Подготовка недостаточная и требует дополнительного изучения материала.

### Пример описания итогового зачетного проекта

Написать полноценное SPA (single page application) приложение (например блог, интернет магазин, админ панель, сайт визитка или любое другое веб-приложение), которое должно удовлетворять следующим условиям:

- Функциональные компоненты, React-hooks
- Модульные стили или styled-components
- Роутинг React-router-dom или NextJS
- Redux/ReduxToolKit. Асинхронные экшены Redux-thunk или Redux-saga
- Работа с API с помощью клиента Axios (можно использовать любое открытое [API](#))

### Критерии оценки итогового проекта

На основании выполненного итогового проекта обучающемуся определяется оценка – «зачтено», «не зачтено».

Оценка	Уровень подготовки
Зачтено	Хорошая подготовка. Обучающийся выполнил все предложенные этапы работы над проектом. Проведено исследование пользовательского опыта, представлены его результаты. Сделаны корректные выводы.
Не зачтено	Подготовка недостаточная и требует дополнительного изучения материала.

## Литература

1. "HTML и CSS. Дизайн и построение веб-сайтов" Джон Даккетт - Перевод книги, которая поможет понять основы HTML и CSS.
2. "CSS. Технология создания стилей для профессионалов" Эрик Мейерс - Книга о продвинутых техниках работы с CSS.
3. "JavaScript. Шаблоны" Стоян Стефанов - Книга о шаблонах программирования на JavaScript.
4. "JavaScript. Подробное руководство" Дэвид Флэнаган - Классическое руководство по JavaScript.
5. "React. Быстрый старт" Пол Картер - Книга, которая поможет вам начать работу с React.
6. MDN Web Docs (<https://developer.mozilla.org/>): Официальный ресурс от Mozilla, который предлагает подробные и понятные статьи по HTML, CSS и JavaScript.
7. CSS-Tricks (<https://css-tricks.com/>): Этот сайт предлагает множество полезных статей, уроков и ресурсов по CSS.
8. learn.javascript (<https://learn.javascript.ru/>): Интерактивный учебник по JavaScript.
9. React.js документация (<https://ru.legacy.reactjs.org/>): Здесь вы найдете подробные инструкции и примеры использования React.
10. W3Schools (<https://www.w3schools.com/>): Этот ресурс предлагает простые примеры и уроки по HTML, CSS, JavaScript и другим технологиям веб-разработки.
11. Stack Overflow (<https://stackoverflow.com/>): Форум для программистов, где можно найти ответы на многие вопросы по HTML, CSS, JavaScript и React.

## 6. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ ОБЕСПЕЧЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

*Руководитель программы повышения квалификации:*

Усков А.В. Директор ООО «Академия Лад»

*Разработчики программы повышения квалификации:*

Усков А.В. Директор ООО «Академия Лад»

*Составители учебно-тематического плана программы повышения квалификации:*

Усков А.В. Директор ООО «Академия Лад»

Куликов Д.И. Старший инженер программист ООО Студия Т\_Г

Сведения о педагогических (научно-педагогических) работниках, участвующих в реализации программы повышения квалификации, и лицах, привлекаемых к реализации дополнительной образовательной программы на иных условиях, представлены в таблице 2.

Таблица 2

### Преподаватели программы повышения квалификации «Веб-разработка»

№ п/п	Наименование дисциплин (модулей)	Фамилия, имя, отчество,	Ученая степень, ученое звание	Основное место работы, должность	Место работы и должность по совместительству (если есть)
1.	Дисциплина «Вёрстка сайтов»	Усков А.В.		Директор ООО «Академия Лад»	
2.	Дисциплина «Программирование на JavaScript»	Лушкин С.Н.		TastyTeam, frontend-разработчик	
3.	Дисциплина «Frontend-разработка на React»	Куликов Д.И.		Старший инженер программист ООО Студия Т_Г	

## 7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

№ п.п.	Наименование дисциплины (модуля)	Материально-технические условия для реализации программ (наличие лабораторий, производственных участков и т.п. по профилю программы повышения квалификации)
1.	Дисциплина «Вёрстка сайтов»	- Система дистанционного обучения: <a href="https://getcourse.ru">https://getcourse.ru</a>
2.	Дисциплина «Программирование на JavaScript»	
3	Дисциплина «Frontend-разработка на React»	

**УЧЕБНЫЙ ПЛАН**  
**ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**  
**«Веб-разработка»**

№ пп	Наименование модулей	Всего, час.	В том числе			Самост оятельн ая работа
			Аудито рных	Лекции	Семинар ы, практиче ские занятия	
1.	<b>Модуль 1. Верстка сайтов</b>					
	Основы html и css	11	6	2	4	5
	Codestyle и codeguide	3	2	1	1	1
	CSS flexbox	8	5	1	4	3
	CSS Grid	7	5	1	4	2
	БЭМ — методология	5	4	2	2	1
	Адаптация сайтов под мобильные устройства	6	5	1	4	1
	Автоматизация верстки	4	3	2	1	1
	Промежуточная аттестация	2				
	<b>ИТОГО</b>	<b>46</b>	<b>30</b>	<b>10</b>	<b>20</b>	<b>14</b>
2.	<b>Модуль 2. Программирование на JavaScript</b>					
	Введение в JavaScript. Переменные и типы данных	4	3	1	2	1
	Git. Настройка среды. Преобразование типов. Операторы сравнения	5	3	1	2	2
	Условные и логические операторы, циклы	4	3	1	2	1
	Функции	5	3	1	2	2
	Массивы и их методы, объекты	5	3	1	2	2
	Продвинутая работа с функциями, ключевое слово this, контекст, замыкания	5	3	1	2	2
	Прототипы и классы	4	3	1	2	1
	Document Object Model	4	3	1	2	1
	Промисы, цепочки промисов, методы, перехват ошибок. Event Loop, микро и макрозадачи	5	3	1	2	2

	Обработка ошибок. Fetch	5	3	1	2	2
	Промежуточная аттестация	2				
	<b>ИТОГО</b>	<b>48</b>	<b>30</b>	<b>10</b>	<b>20</b>	<b>16</b>
3.	<b>Модуль 3. Frontend-разработка на React</b>					
	Инициализация react-приложения, структура приложения и работа с пакетным менеджером npm. Настройка рабочей среды.	4	3	1	2	1
	Базовые концепции React	4	3	1	2	1
	Функциональные компоненты.	4	3	1	2	1
	Знакомство с понятиями props и state	4	3	1	2	1
	Основные этапы и методы жизненного цикла компонентов.	4	3	1	2	1
	Классовые и функциональные компоненты.	4	3	1	2	1
	Знакомство с React-Hooks	4	3	1	2	1
	Роутинг в react-приложении	4	3	1	2	1
	Подключение и настройка менеджера состояний Redux	4	3	1	2	1
	ReactDevTools и ReduxDevTools	4	3	1	2	1
	http-клиент axios.	3	3	1	2	
	Рендер React приложений	3	3	1	2	
	Промежуточная аттестация	2				
	<b>ИТОГО</b>	<b>48</b>	<b>36</b>	<b>12</b>	<b>24</b>	<b>10</b>
	Итоговая аттестация	2				
	<b>ИТОГО ПО ПРОГРАММЕ</b>	<b>144</b>	<b>96</b>	<b>32</b>	<b>64</b>	<b>40</b>



