

ООО "ЭЛЬБУС БУТКЕМП"

УТВЕРЖДАЮ
Генеральный директор

Бабаян Г.Г.
(подпись)

"6" марта 2023 г.



**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
ПЕРЕПОДГОТОВКИ**

Разработчик JavaScript part time

(ТРУДОЕМКОСТЬ 600 АКАДЕМИЧЕСКИХ ЧАСОВ)

Москва – 2023

2. ОПИСЬ КОМПЛЕКТА ДОКУМЕНТОВ

№ п/п	Наименование документа
1	Титульный лист
2	Опись комплекта документов
3	Общие положения
4	Цель и задачи программы
5	Планируемые результаты обучения
6	Учебный план
7	Календарный учебный график
8	Рабочая программа
9	Организационно-педагогические условия
9.1	Организация образовательной деятельности по освоению программы
9.2	Продолжительность одного занятия и трудоемкость недельной нагрузки
9.3	Материально-техническая база реализации программы
9.4	Квалификация научно-педагогических работников
10	Программа итоговой аттестации обучающихся
11	Оценочные средства итоговой аттестации
11.1.	Примерная тематика контрольных вопросов, выявляющих теоретическую подготовку обучающегося
11.2.	Примеры заданий, выявляющих практическую подготовку обучающегося
11.3.	Примеры ситуационных задач
11.4	Критерии оценки обучающегося на недифференцированном зачете
12	Иные компоненты программы
12.1.	Кадровое обеспечение образовательного процесса

3. ОБЩИЕ ПОЛОЖЕНИЯ

3.1. Характеристика программы:

Дополнительная профессиональная образовательная программа переподготовки по специальности/теме "Разработчик JavaScript parttime" трудоемкостью 600 академических часов (далее – Программа) сформирована в соответствии с:

- Федеральным законом от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»,
- приказом Министерства образования и науки РФ от 01.07.2013 № 499 «Порядок организации и осуществления образовательной деятельности по дополнительным профессиональным программам»,
- письмом Министерства образования и науки РФ от 22.04.2015 № ВК-1032/06 «О направлении методических рекомендаций»
- письмом Министерства образования и науки РФ от 21.04.2015 № ВК-1013/06 «О направлении методических рекомендаций по реализации дополнительных профессиональных программ»;
- приказом Министерства образования и науки РФ от 23.08.2017 № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- требованиями профессионального стандарта "Разработчик Web и мультимедийных приложений" (Приказ Министерства труда и социальной защиты Российской Федерации от 18.01.2017 № 44н).
- Федеральным государственным образовательным стандартом Высшего образования по направлению подготовки 09.03.04 "Программная инженерия" (Приказ Министерства науки и высшего образования Российской Федерации от 12 марта 2015 г. № 229),

Программа реализуется в дополнительном профессиональном образовании.

3.2. Характеристика профессиональной деятельности выпускников:

– **область профессиональной деятельности** в соответствии с ФГОС в сферах: индустриальное производство программного обеспечения для информационно-вычислительных систем различного назначения.

– **основная цель вида профессиональной деятельности:** Создание, модификация и сопровождение web-сайтов, корпоративных порталов организаций, мультимедиа и интерактивных приложений, информационных ресурсов (далее - ИР)

– **трудовые функции:**

Обобщенные трудовые функции			Трудовые функции		
код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
А	Техническая поддержка процессов создания (модификации) и сопровождения информационных ресурсов	4	Верстка страниц ИР	А/03.4	4
			Кодирование на языках web-программирования	А/04.4	4
			Сопровождение дизайна, разработки и выпуска обновлений ИТ продукта	А/03.4	4
			Тестирование ИР с точки зрения логической целостности (корректность ссылок, работа элементов форм)	А/05.4	4
			Тестирование интеграции ИР с внешними сервисами и учетными системами	А/06.4	4
В	Выполнение работ по созданию (модификации) и сопровождению информационных ресурсов	5	Проектирование разделов ИР	В/04.5	5
			Тестирование интеграции ИР с внешними сервисами и учетными системами с использованием взаимодействия компонентов распределенной системы	В/06.5	5
			Обеспечение безопасной и бесперебойной работы сайта	В/09.5	5
С	Управление работами по созданию (модификации) и	6	Анализ и формализация требований к ИР	С/01.6	6
			Разработка технических спецификаций на ИР	С/02.6	6

Обобщенные трудовые функции			Трудовые функции		
код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
	сопровождению информационных ресурсов		Проектирование ИР	C/03.6	6
			Тестирование ИР с точки зрения пользовательского удобства на основании данных о поведении пользователей	C/04.6	6
			Организация работ по обеспечению безопасной работы ИР	C/05.6	6
			Организация работ по интеграционному тестированию ИР с внешними сервисами и учетными системами	C/06.6	6
D	Управление процессами и проектами по созданию (модификации) информационных ресурсов	7	Управление процессом разработки программного обеспечения	D/01.6	7
			Руководство проверкой работоспособности ИР	D/05.7	7

– **вид программы:** практико-ориентированная.

3.3. Категория обучающихся по:

- основной специальности: JavaScript-разработчик;
- смежным специальностям: инженеры, маркетологи, бизнес-стратеги.

К освоению Программы допускаются лица, имеющие или получающие среднее профессиональное и (или) высшее образование, а также желательно имеющие базовые знания синтаксиса JavaScript.

3.4. Актуальность программы: Изучение JavaScript сегодня является одним из самых востребованных навыков в IT-индустрии. Этот язык программирования является основой для разработки веб-приложений и сайтов, обеспечивая интерактивность и

динамическое взаимодействие с пользователями. JavaScript используется как на стороне клиента, так и на стороне сервера (с помощью Node.js), что делает его универсальным инструментом для создания полнофункциональных веб-систем. Для слушателей, стремящихся к развитию в IT-сфере, владение JavaScript открывает широкие карьерные возможности, от создания сайтов и мобильных приложений до работы с большими данными и облачными сервисами.

Актуальность Программы также обусловлена стремительным развитием технологий и растущей потребностью в специалистах, способных работать с передовыми инструментами и фреймворками, такими как React, Angular и Vue.js. Компании всё чаще ищут разработчиков, которые не только умеют писать код, но и способны адаптировать его к современным требованиям, обеспечивая высокую производительность и пользовательский опыт. Для профессионалов, желающих оставаться конкурентоспособными на рынке труда, освоение JavaScript является ключевым шагом к успешной карьере в сфере веб-разработки.

3.5. Объем программы (трудоемкость)

Общая трудоемкость 600 академических часов, из них 0 аудиторных часов.

3.6. Документ, выдаваемый после завершения обучения:

диплом о профессиональной переподготовке

4. ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ

4.1. Цель программы заключается в получении компетенций, необходимых для профессиональной деятельности в сфере проектирования, разработки и интеграции информационных ресурсов в локальной сети и информационно-телекоммуникационной сети «Интернет».

4.2. Задачи

В процессе обучения решаются следующие задачи:

1. Изучение основ HTML и CSS для создания семантической и адаптивной верстки. Слушатели овладеют базовыми принципами верстки веб-страниц с использованием HTML и CSS, научатся работать с Flexbox и адаптировать сайты для различных устройств.
2. Освоение базового синтаксиса и структуры JavaScript. Слушатели изучат основные элементы JavaScript, такие как, функции, массивы, объекты и рекурсия. Будут рассматриваться основы ООП проектирования, а также базовые принципы тестирования JavaScript кода.
3. Изучение расширенных возможностей JavaScript. В рамках курса будут рассмотрены более сложные аспекты JavaScript, такие как, работа с асинхронностью (callbacks, promises, async/await), объектно-ориентированное программирование и методы работы с массивами и строками. Важной частью курса станет освоение системы контроля версий Git.
4. Разработка серверных приложений с использованием Node.js. Слушатели изучат основы серверной разработки на Node.js и Express, научатся работать с базами данных на основе SQL (PostgreSQL) и ORM (Sequelize). Важной частью станет понимание процессов аутентификации и авторизации, использование cookies, JWT токенов. Формирование серверных решений с архитектурным стилем REST API.
5. Разработка клиентских веб-приложений с использованием React и статической типизации в виде TypeScript. Слушатели изучат компонентный подход к разработке веб-приложений на React, познакомятся с основными и дополнительными хуками, маршрутизацией, работой с состоянием через Redux Toolkit, и аутентификацией в приложениях. Развертывание приложений (deployment) используя технологию контейнеризации Docker.

5. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

5.1. Программа направлена совершенствование:

– общекультурных компетенций:

- способен к коммуникации в устной и письменной формах на русском и иностранном языке для решения задач межличностного и межкультурного взаимодействия;
- способен работать в коллективе, толерантно воспринимать социальные, этнические, конфессиональные и культурные различия;
- способен к самоорганизации и самообразованию, постоянно обновляя свои знания и навыки в JavaScript и веб-разработки.

– общепрофессиональных компетенций:

- владеет основными концепциями, принципами, теориями и фактами, связанными с программированием на JavaScript;
- владеет архитектурой современных веб-приложений и сервисов;
- способен применять основы JavaScript и современных технологий разработки к

- проектированию, конструированию и тестированию веб-приложений;
- способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием JavaScript и современных инструментов разработки

– профессиональных компетенций:

производственно-технологическая деятельность:

- способен применять основные методы и инструменты разработки программного обеспечения на JavaScript (ПК-1);
- владеет навыками использования различных технологий разработки веб-приложений (ПК-3);
- владеет концепциями и атрибутами качества программного обеспечения, включая надежность, безопасность и удобство использования (ПК-4)

организационно-управленческая деятельность:

- владеет современными концепциями и моделями управления проектами в сфере FullStack разработчик (ПК-6);
- владеет методами контроля версий и готовностью осуществлять управление кодовой базой

сервисно-эксплуатационная деятельность:

- владеет основными концепциями и моделями эволюции и сопровождения JavaScript-приложений (ПК-10);
- способен обеспечивать масштабируемость и производительность веб-приложений;

аналитическая деятельность:

- способен формализовать предметную область веб-проекта и разработать спецификации для компонентов программного продукта;
- владеет навыками анализа требований пользователей и бизнеса для разработки эффективных JavaScript-решений;

проектная деятельность:

- владеет навыками моделирования, анализа и использования формальных методов конструирования веб-приложений (ПК-19);
- способен создавать RESTful API и интегрировать различные сервисы в JavaScript-приложения

5.2. Программа направлена на приобретение профессиональных компетенций, необходимых для выполнения трудовых функций:

Трудовая функция с кодом ¹	Профессиональные компетенции, обеспечивающие выполнение трудовой функции	
	индекс	содержание компетенции
Верстка страниц ИР (А/03.4)	ОК-5	Способен к коммуникации в устной и письменной формах на русском и иностранном языках для решения задач межличностного и межкультурного взаимодействия.

¹ Указываются все трудовые функции из раздела 4.2.

Трудовая функция с кодом ¹	Профессиональные компетенции, обеспечивающие выполнение трудовой функции	
	индекс	содержание компетенции
	ОПК-1	Владеет основными концепциями, принципами, теориями и фактами, связанными с информатикой.
	ОПК-2	Владеет архитектурой электронных вычислительных машин и систем.
	ОПК-3	Способен применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов
	ПК-1	Способен применять основные методы и инструменты разработки программного обеспечения
Кодирование на языках web-программирования (А/04.4)	ОК-7	Способен к самоорганизации и самообразованию.
	ОПК-4	Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий
	ПК-3	Владеет навыками использования различных технологий разработки программного обеспечения
Сопровождение дизайна, разработки и выпуска обновлений ИТ продукта (А/03.4)	ОК-6	Способен работать в коллективе, толерантно воспринимать социальные, этнические, конфессиональные и культурные различия
	ПК-4	Владеет концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества.
	ПК-10	Владеет основными концепциями и моделями эволюции и сопровождения программного обеспечения.

Трудовая функция с кодом ¹	Профессиональные компетенции, обеспечивающие выполнение трудовой функции	
	индекс	содержание компетенции
Тестирование ИР с точки зрения логической целостности (корректность ссылок, работа элементов форм) (А/05.4)	ПК-4	Владеет концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества.
	ПК-10	Владеет основными концепциями и моделями эволюции и сопровождения программного обеспечения.
Тестирование интеграции ИР с внешними сервисами и учетными системами (А/06.4)	ПК-4	Владеет концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества.
	ПК-9	Владеет методами контроля проекта и готовностью осуществлять контроль версий
Проектирование разделов ИР (В/04.5)	ПК-16	Способен формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта.
	ПК-19	Владеет навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения.
Тестирование интеграции ИР с внешними сервисами и учетными системами с использованием взаимодействия компонентов распределенной системы (В/06.5)	ПК-4	Владеет концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества.
	ПК-9	Владеет методами контроля проекта и готовностью осуществлять контроль версий

Трудовая функция с кодом ¹	Профессиональные компетенции, обеспечивающие выполнение трудовой функции	
	индекс	содержание компетенции
Обеспечение безопасной и бесперебойной работы сайта (В/09.5)	ПК-4	Владеет концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества.
Анализ и формализация требований к ИР (С/01.6)	ОПК-1	Владеет основными концепциями, принципами, теориями и фактами, связанными с информатикой.
	ОПК-2	Владеет архитектурой электронных вычислительных машин и систем.
	ПК-16	Способен формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта.
	ПК-19	Владеет навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения.
Разработка технических спецификаций на ИР (С/02.6)	ПК-16	Способен формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта.
	ПК-22	Способен создавать программные интерфейсы.
Проектирование ИР (С/03.6)	ПК-19	Владеет навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения.
Тестирование ИР с точки зрения пользовательского удобства на основании данных о поведении пользователей (С/04.6)	ПК-4	Владеет концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения

Трудовая функция с кодом ¹	Профессиональные компетенции, обеспечивающие выполнение трудовой функции	
	индекс	содержание компетенции
		качества.
Организация работ по обеспечению безопасной работы ИР (С/05.6)	ПК-4	Владеет концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества.
Организация работ по интеграционному тестированию ИР с внешними сервисами и учетными системами (С/06.6)	ПК-4	Владеет концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества.
	ПК-9	Владеет методами контроля проекта и готовностью осуществлять контроль версий
Управление процессом разработки программного обеспечения (D/01.6)	ПК-6	Владеет классическими концепциями и моделями менеджмента в управлении проектами.
Руководство проверкой работоспособности ИР (D/05.7)	ПК-6	Владеет классическими концепциями и моделями менеджмента в управлении проектами.
	ПК-9	Владеет методами контроля проекта и готовностью осуществлять контроль версий.

ООО "ЭЛЬБРУС БУТКЕМП"

УТВЕРЖДАЮ
Генеральный директор

Бабаян Г.Г.
(подпись)

"6" марта 2023 г.



6. УЧЕБНЫЙ ПЛАН

дополнительной профессиональной образовательной программы
переподготовки по специальности/теме «Разработчик JavaScript parttime»
(трудоемкость 600 академических часов)

Цель: в получении компетенций, необходимых для профессиональной деятельности в сфере проектирования, разработки и интеграции информационных ресурсов в локальной сети и информационно-телекоммуникационной сети «Интернет».

Категория обучающихся по²:

- основной специальности: JavaScript разработчик;
 - смежным специальностям: инженеры, маркетологи, бизнес-стратеги.
- К освоению Программы допускаются лица, имеющие или получающие среднее профессиональное и (или) высшее образование, а также желательно имеющие базовые знания синтаксиса JavaScript.

Общая трудоемкость: 600 акад. час.

Форма обучения: очно-заочная (с вечерними занятиями)

² Повторяется на основе позиции 4.3.

№ п/ п	Наименование разделов (модулей) и тем	Аудиторные учебные занятия, учебные работы (с использованием ЭО или ДОТ ³)			Внеаудиторная работа	Формируемые компетенции ⁴	Форма контроля	Трудоемкость
		Всего ауд., час	Лекции ⁵	Практические занятия ⁶	Самостоятельная работа			
1	Раздел 1. HTML, CSS и базовый JavaScript	80	20	60	40	ОК-5, ОК-6, ОПК-3, ПК-1	Практическая работа	150
1.1	HTML, Семантическая верстка	12	3	9	6	-	-	18
1.2	CSS селекторы, Flexbox	12	3	9	6	-	Т/К	18
1.3	Адаптивная верстка	8	2	6	4	-	Т/К	12
1.4	Условные конструкции в JavaScript	8	2	6	4	-	Т/К	12
1.5	Базовый синтаксис функций в JavaScript и замыкание	8	2	6	4	-	Т/К	12
1.6	Базовый синтаксис массивов и объектов в JavaScript	8	2	6	4	-	Т/К	12

³ Дистанционное обучение.

⁴ Указываются индексы компетенций.

⁵ Объем лекционных занятий определяется целями и содержанием Программы. Общее рекомендуемое соотношение лекционных занятий составляет примерно 1/3 от общего количества учебных часов.

⁶ Объем семинарских и практических занятий составляет примерно 2/3 от общего количества учебных часов.

№ п/ п	Наименование разделов (модулей) и тем	Аудиторные учебные занятия, учебные работы (с использованием ЭО или ДОТ ³)			Внеаудиторная работа	Формируемые компетенции ⁴	Форма контроля	Трудоемкость
		Всего ауд., час	Лекции ⁵	Практические занятия ⁶	Самостоятельная работа			
1.7	Базовое понимание основ тестирования JavaScript кода с использованием дополнительных пакетов и методов	8	2	6	4	-	Т/К	12
1.8	Объектная модель документа HTML и методы DOM	8	2	6	4	-	Т/К	12
1.9	DOM Events	8	2	6	4	-	Т/К	12
2.	Раздел 2. Расширенный JavaScript	80	20	60	40	ОК-7, ОПК-1, ПК-4, ПК-9	Практическая работа	150
2.1	Node.js, NPM, NPX, CommonJS, нативный модуль fs в Node.js	6	1	5	3	-	Т/К	9
2.2	Массивы и их методы	6	1	5	3	-	Т/К	9
2.3	Отладка кода,	6	1	5	3	-	Т/К	9

№ п/ п	Наименование разделов (модулей) и тем	Аудиторные учебные занятия, учебные работы (с использованием ЭО или ДОТ ³)			Внеауд иторна я работа	Формир уемые компет енции ⁴	Форма контрол я	Трудоемк ость
		Всег о ауд., час	Лек ции ⁵	Прак тиче ские заня тия ⁶	Самосто ятельна я работа			
	рекурсия, базовые алгоритмы							
2.4	Строки и их методы, регулярные выражения, даты в JavaScript	6	1	5	3	-	Т/К	9
2.5	Git (система контроля версий), GitHub	6	1	5	3	-	Т/К	9
2.6	Объекты. Классы в JavaScript (ООП)	6	1	5	3	-	Т/К	9
2.7	Геттеры, сеттеры, приватные поля и методы в классах (ООП)	6	1	5	3	-	Т/К	9
2.8	Асинхронност ь в JavaScript (Callback Functions, Event Loop)	6	1	5	3	-	Т/К	9
2.9	Асинхронност ь в JavaScript (Promise,	6	1	5	3	-	Т/К	9

№ п/ п	Наименование разделов (модулей) и тем	Аудиторные учебные занятия, учебные работы (с использованием ЭО или ДОТ ³)			Внеауд иторна я работа	Формир уемые компет енции ⁴	Форма контрол я	Трудоемк ость
		Всег о ауд., час	Лек ции ⁵	Прак тиче ские заня тия ⁶	Самосто ятельна я работа			
	Async, Await)							
2.1 0	Базовый SQL, PCYБД PostgreSQL	14	9	5	7	-	T/K	21
2.1 1	ORM Sequelize, CRUD	6	1	5	3	-	T/K	9
2.1 2	Sequelize Methods, Sequelize CLI	6	1	5	3	-	T/K	9
3.	Раздел 3. Разработка клиент- серверных приложений на Node.js	80	20	60	40	ОПК-2, ОПК-4, ПК-6, ПК-16	Практиче ская работа	150
3.1 .	Node.js / Express, создание серверного приложения	10	4	6	5	-	T/K	15
3.2 .	Middleware, маршрутизаци я запросов REST API	10	2	8	5	-	T/K	15
3.3 .	React, компонентная разработка, состояние клиента,	10	2	8	5	-	T/K	15

№ п/ п	Наименование разделов (модулей) и тем	Аудиторные учебные занятия, учебные работы (с использованием ЭО или ДОТ ³)			Внеаудиторная работа	Формируемые компетенции ⁴	Форма контроля	Трудоемкость
		Всего ауд., час	Лекции ⁵	Практические занятия ⁶	Самостоятельная работа			
	маршрутизация							
3.4.	AJAX, Fetch API, AXIOS	10	2	8	5	-	Т/К	15
3.6.	Cookies, JWT, аутентификация и авторизация пользователя	10	4	6	5	-	Т/К	15
3.7.	CORS, IDOR	10	2	8	5	-	Т/К	15
3.8.	Event Emitters, WebSocket	10	2	8	5	-	Т/К	15
3.9.	TypeScript, базовая типизация, типизация событий React	10	2	8	5	-	Т/К	15
4.	Раздел 4. Разработка веб-приложений с использованием React	60	10	50	15	ОК-5, ОК-6, ОК-7, ПК-3, ПК-19, ПК-22	Практическая работа	75
4.1	Расширенные хуки React, компонентный подход, функциональные компоненты	10	2	8	5	-	Т/К	15
4.2	Валидация	10	2	8	5	-	Т/К	12

№ п/ п	Наименование разделов (модулей) и тем	Аудиторные учебные занятия, учебные работы (с использованием ЭО или ДОТ ³)			Внеаудиторная работа	Формируемые компетенции ⁴	Форма контроля	Трудоемкость
		Всего ауд., час	Лекции ⁵	Практические занятия ⁶	Самостоятельная работа			
	клиентского ввода, сторонние библиотеки компонентов							
4.3	Redux Toolkit, Slices, Thunk	10	2	8	5	-	Т/К	12
4.4	Хуки оптимизации клиентского UI	10	2	8	5	-	Т/К	12
4.5	Исправление ошибок сборки	10	1	9	5	-	Т/К	12
4.6	Docker, контейнеризация приложения, развертывание на сервере	10	1	9	5	-	Т/К	12
	Итоговая аттестация⁷	1	1	-	74	-	Защита дипломного проекта	75
	Итого:	301	71	230	179	-	-	600

⁷ Итоговая аттестация проводится в том случае, если вся программа осваивалась дистанционно. Если дистанционное обучение представляет собой реализацию части программы, то по окончании проводится промежуточная аттестация.

ООО "ЭЛЬБУС БУТКЕМП"

УТВЕРЖДАЮ
Генеральный директор

Бабаян Г.Г.
(подпись)

"6" марта 2023 г.



7. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК⁸

дополнительной профессиональной образовательной программы переподготовки по специальности/теме
«Разработчик JavaScript part time»
(трудоемкость 600 академических часов)

Наименование модуля		Плановый период реализации программы 600 час., режим занятий до 10 (десяти) часов в день, перерыв на обед 90 минут.																							
		1 нед еля	2 нед еля	3 нед еля	4 нед еля	5 нед еля	6 нед еля	7 нед еля	8 нед еля	9 нед еля	10 нед еля	11 нед еля	12 нед еля	13 нед еля	14 нед еля	15 нед еля	16 нед еля	17 нед еля	18 нед еля	19 нед еля	20 нед еля	21 нед еля	22 нед еля	23 нед еля	24 нед еля
Раздел 1. HTML, CSS и базовый JavaScript	Форма	Д	Д	Д	Д	Д	Д																		
	Вид УД	Л/П/ СР	Л/П/ СР	Л/П/ СР	Л/П/ СР	Л/П/ СР	Л/П/ СР																		
	Час ов	25	25	25	25	25	25																		

⁸ Календарный учебный график – это документ, определяющий календарные периоды освоения Программы в соответствии с учебным планом Программы и является основополагающим документом для проведения учебных циклов. Календарный учебный график создается и утверждается ежегодно. График утверждается УС и обязателен для выполнения всеми участниками образовательного процесса (Календарный учебный график регламентирован п.10 ст.2 Федерального закона РФ от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»).

[illegible]

Условные обозначения:

Д – внеаудиторные, с использованием дистанционных образовательных технологий.

Л – лекционное занятие.

П – семинарское занятие.

СР – самостоятельная работа.

ПА - промежуточная аттестация

ИА - итоговая аттестация

8. РАБОЧАЯ ПРОГРАММА

8.1. Трудоемкость освоения: 600 часов

8.2. Задачи:

1. Изучение основ HTML и CSS для создания семантической и адаптивной верстки. Слушатели овладеют базовыми принципами верстки веб-страниц с использованием HTML и CSS, научатся работать с Flexbox и адаптировать сайты для различных устройств.
2. Освоение базового синтаксиса и структуры JavaScript. Слушатели изучат основные элементы JavaScript, такие как, функции, массивы, объекты и рекурсия. Будут рассматриваться основы ООП проектирования, а также базовые принципы тестирования JavaScript кода.
3. Изучение расширенных возможностей JavaScript. В рамках курса будут рассмотрены более сложные аспекты JavaScript, такие как, работа с асинхронностью (callbacks, promises, async/await), объектно-ориентированное программирование и методы работы с массивами и строками. Важной частью курса станет освоение системы контроля версий Git.
4. Разработка серверных приложений с использованием Node.js. Слушатели изучат основы серверной разработки на Node.js и Express, научатся работать с базами данных на основе SQL (PostgreSQL) и ORM (Sequelize). Важной частью станет понимание процессов аутентификации и авторизации, использование cookies, JWT токенов. Формирование серверных решений с архитектурным стилем REST API.
5. Разработка клиентских веб-приложений с использованием React и статической типизации в виде TypeScript. Слушатели изучат компонентный подход к разработке веб-приложений на React, познакомятся с основными и дополнительными хуками, маршрутизацией, работой с состоянием через Redux Toolkit, и аутентификацией в приложениях. Развертывание приложений (deployment) используя технологию контейнеризации Docker.

8.3. Учебная (рабочая) программа:

№ п/п, название темы	Виды учебных занятий, учебных работ	Содержание	Планируемые результаты обучения (Знать/Уметь/Владеть)
Раздел 1. HTML, CSS и базовый JavaScript			
1.1. Введение и базовые концепции и основы HTML	Лекции (5 ч.)	Настройки редактора кода и работы в терминале, включая базовые Unix команды и SSH подключение. HTML, где обсуждаются основные теги, строчные и блочные элементы, списки и поля. HTML-валидация по стандартам W3C и семантика HTML.	Знать: Виды и цели семантической верстки. Основные CSS-селекторы и свойства Flexbox. Условия и синтаксис JavaScript.
	Практическая работа (13 ч.)	Настройка рабочего окружения для выполнения последующих задач. Применение в работе CLI инструментов.	

№ п/п, название темы	Виды учебных занятий, учебных работ	Содержание	Планируемые результаты обучения (Знать/Уметь/Владеть)
	Самостоятель ная работа (7 ч.)	Использование HTML для семантической вёрстки на основе описания / макетов.	Основы объектной модели документа (DOM) и методы работы с ней.
1.2. Основы CSS	Лекция (5 ч.)	Знакомство с CSS, использование тега style и отдельных файлов стилей. Работа с селекторами, классами и идентификаторами. Потение каскадности, наследования и вложенности селекторов. CSS- позиционирование элементов, трансформации и переходы, работа с фоном и шрифтами.	Уметь: Определять семантические элементы HTML и применять их в верстке. Использовать CSS для адаптивной верстки. Реализовывать условные конструкции и функции в JavaScript. Работать с массивами и объектами в JavaScript. Применять методы тестирования JavaScript кода.
	Практическая работа (13 ч.)	Применение CSS в HTML тегах, использование разных вариантов подключения стилей. Работа с позиционированием и трансформацией элементов.	
	Самостоятель ная работа (7 ч.)	Вёрстка HTML страницы на основе макета. Применение позиционирование элементов, наследования свойств и подключения шрифтов.	
1.3. Расширенны й CSS	Лекция (5 ч.)	Продвинутые темы CSS, такие как Flexbox и использование библиотеки Bootstrap. Работа со шрифтами, фоном, трансформацией элементов.	Владеть: Технологиями работы с DOM и обработкой событий. Инструментами разработки и отладки JavaScript кода.
	Практическая работа (13 ч.)	Работа с библиотекой Bootstrap, использование компонентов и классов для формирования необходимых блоков.	
	Самостоятель ная работа (7 ч.)	Реализация сайта визитки с использованием пройденных тем HTML, CSS.	
1.4 Основы JavaScript	Лекция (5 ч.)	Знакомство с JavaScript и способами подключения скриптов, переменных и ESLint. Изучение типов данных: числа и операторы, строки и создание строк, логические условия и сравнения. JavaScript-функции, виды функций и способы объявления.	

№ п/п, название темы	Виды учебных занятий, учебных работ	Содержание	Планируемые результаты обучения (Знать/Уметь/Владеть)	
	Практическая работа (13 ч.)	Работа со всеми типами данных: числа, строки, булевы значения, массивы и объекты. Применение математических операторов.		
	Самостоятель ная работа (7 ч.)	Решение грок-марафона по всем пройденным темам.		
1.5 Продвинуты е концепции JavaScript, тестировани е	Лекция (5 ч.)	Изучение циклов и применение их контексте массива. Знакомство с библиотекой тестирования Jest.		
	Практическая работа (13 ч.)	Практические задачи для закрепления базовых концепций циклов, функций и переменных.		
	Самостоятель ная работа (7 ч.)	Реализация веб-приложения с логикой работы лифта. Обязательное применение HTML, CSS, JavaScript.		
1.6 Реализация самостоятел ьной работы с применение м HTML, CSS, JS	Лекция (1 ч.)	Введение в работу использование линтера ESLint, оптимизация и сокращение синтаксиса при написании кода.		
	Практическая работа (13 ч.)	Практические задачи для закрепления базовых концепций циклов, функций и переменных.		
	Самостоятель ная работа (11 ч.)	Реализовать веб-приложения с использованием HTML, CSS, JavaScript. Основной задачей является функционал транслитерации русского языка. Работа с макетом и HTML семантикой.		
Раздел 2. Расширенный JavaScript				
2.1. Введение в Node.js и JavaScript	Лекции (5 ч.)	Разбор основных принципы работы с Node.js, включая модули файловой системы и путей. Вариативная деструктуризация в JavaScript, методы работы с массивами и отладка кода.		Знать: - Основные принципы работы с Node.js, включая модули файловой системы (fs) и путей. - Деструктуризация в JavaScript и методы работы с массивами. - Алгоритмы, такие как линейный поиск,
	Практическая работа (13 ч.)	Практические задачи на применение CDUD операций с массивами и файлами используя средства fs. Практическая реализация собственных методов массивов и работа с отладчиком кода.		

№ п/п, название темы	Виды учебных занятий, учебных работ	Содержание	Планируемые результаты обучения (Знать/Уметь/Владеть)
	Самостоятель ная работа (7 ч.)	Реализация настольной игры "Боггл", с помощью HTML, CSS, JavaScript. Применение в работе инструмента контроля версий git с загрузкой проекта на сервис GitHub.	бинарный поиск и сортировка пузырьком. - Концепции объектно-ориентированного программирования (ООП) и прототипное наследование.
2.2. Работа со строками, массивами, регулярным и выражениям и, рекурсией и базовыми алгоритмам и	Лекции (5 ч.)	Разбор методов работы со строками, изучения базового синтаксиса регулярных выражений. Обзор концепции рекурсии и алгоритмов использующие её. Изучение концепции замыкания функций в JavaScript.	- Асинхронные методы в JavaScript, включая Promise, async/await и механизм EventLoop. - Основы реляционных баз данных и использование ORM Sequelize.
	Практическая работа (13 ч.)	Грок задач по применению методов строк. Практическая работа с тренажером регулярных выражений. Написание и изучение базовых алгоритмов: линейный поиск, бинарный поиск, сортировка пузырьком. Практическое применение замыкания в задачах.	- Технические условия парного программирования и методы контроля версий Git.
	Самостоятель ная работа (7 ч.)	Реализация настольной игры "Судоку", с помощью нативного JavaScript.	Уметь: - Реализовывать настольные игры (например, "Боггл" и "Судоку") с помощью HTML, CSS и JavaScript.
2.3. Работа с объектами, знакомство с методологией ООП	Лекции (5 ч.)	Обсуждение ключевых понятий объектно-ориентированного программирования (ООП), включая работу с объектами, прототипное наследование, классы и их особенности, такие как статические методы и свойства.	- Решать практические задачи с использованием методов строк и регулярных выражений.
	Практическая работа (13 ч.)	Изучение темы на практических задачах требующие реализации классов и работу с экземплярами. Применение обычных и статических методов класса, реализация базового ООП.	- Применять замыкания функций в различных задачах JavaScript. - Разрабатывать CLI приложения с вводом пользователя и обработкой данных.
	Самостоятель ная работа (7 ч.)	Реализация настольной игры "Судоку", с помощью нативного JavaScript.	- Понимать и реализовывать асинхронные записи и чтения данных, а также использовать статические методы
2.4. Работа с асинхронны	Лекции (5 ч.)	Разбор асинхронных методов в JavaScript, включая использование	

№ п/п, название темы	Виды учебных занятий, учебных работ	Содержание	Планируемые результаты обучения (Знать/Уметь/Владеть)
м JavaScript, реализация через callback и promises		обратных вызовов, Promise и синтаксическую обертку async/await. Изучения реализации связи методов then(), catch(), finally(). Знакомство с механизмом EventLoop.	класса Promise. Владеть: - Практическими навыками реализации CDUD операций с массивами и файлами с использованием модуля fs. - Инструментами контроля версий Git для загрузки проектов на GitHub.
	Практическая работа (13 ч.)	Решение задач с синхронным и асинхронным кодом, реализация асинхронной записи и чтения средствами модуля fs. Применение на практике статических методов класса Promise.	- Навыками написания и отладки кода на JavaScript с использованием различных подходов (например, деструктуризация и методы массива).
	Самостоятельная работа (7 ч.)	Реализация CLI приложения с работой ввода пользователя и CDUD операциями.	- Работа с классами, статическими и обычными методами в JavaScript.
2.5. Знакомство с реляционно й БД PostgreSQL и ORM Sequelize	Лекции (5 ч.)	Разбор основ реляционных баз данных, связи между ними и использование ORM Sequelize. Реализация “сырых” SQL запросов, реализация запросов с использованием методов ORM.	- Умением работать с реляционными базами данных, включая создание и выполнение SQL-запросов.
	Практическая работа (13 ч.)	Установка реляционной БД на ПК, реализация прав и доступов к БД. Изучения SQL запросов на примере тренажера для реализации простых и сложных CRUD операций. Применение API ORM Sequelize на практических задачах.	
	Самостоятельная работа (7 ч.)	Реализация CLI игры “Бумеранг”, с применением записи и хранения информации об игре в БД.	
2.6. Парное и соло программи рование с практически м уклоном	Лекции (1 ч.)	Обсуждение технических условий для парного программирования, использование в работе ПО контроля версий Git.	
	Практическая работа (13 ч.)	Повторение практического материала раздела 2. Подготовка к экзамену.	
	Самостоятельная работа (11 ч.)	Решение задач раздела 2, решение экзаменационного материала в практическом и теоретическом варианте. Экзамен.	
Раздел 3. Разработка клиент-серверных приложений на Node.js			

№ п/п, название темы	Виды учебных занятий, учебных работ	Содержание	Планируемые результаты обучения (Знать/Уметь/Владеть)
3.1. Серверный JavaScript с использован ием Node.js и Express. Понятие клиент- серверного взаимодейст вия	Лекции (5 ч.)	Разбор понятий клиент и сервер на примере HTTP протокола. Изучение запроса и ответа, с помощью фреймворка Express. Первое знакомство с клиентским кодом на основе UI библиотеки React.	Знать: - HTTP протокол и его основные компоненты (запросы, ответы) - Фреймework Express и его возможности - UI библиотека React и ее применение - REST API архитектура и принципы - HTTP статус-коды и их назначение - Базовые концепции веб-разработки (сервер, клиент, взаимодействие) Уметь: - Написать HTTP сервер с использованием Express - Отобразить данные из БД на клиенте - Реализовать маршрутизацию в REST API - Работать с параметризованными маршрутами - Реализовать аутентификацию и авторизацию - Работать с базой данных (CRUD операции) - Реализовать работу с CORS - Использовать WebSocket (WS) протокол - Работать с переменными окружения (.env файлы) Владеть: - Архитектура веб-приложений
	Практическая работа (13 ч.)	Практические примеры задач по написанию HTTP сервера используя Express и React. Связь БД с сервером, отображение ресурсов БД на клиенте.	
	Самостоятель ная работа (7 ч.)	Решение пула задач для пошагового написания сервера с дальнейшим усложнением.	
3.2. Работа с данными, разработка REST API приложения	Лекции (5 ч.)	Знакомство с маршрутизацией на примере архитектурного стиля REST API, работа с параметризованными маршрутами и статус кодами согласно HTTP протоколу. Разбор реализации REST API приложения.	
	Практическая работа (13 ч.)	Практическое наполнение функционалом HTTP сервера, реализация маршрутизации, статических файлов, конфигураций. Применение REST API решений на практике.	
	Самостоятель ная работа (7 ч.)	Реализация веб-приложения на основе клиент-серверного взаимодействия. Подключение БД к проекту для сохранения информации.	
3.3. Серверная аутентифика ция, авторизация применение способов защиты данных	Лекции (5 ч.)	Разбор варианта аутентификации и авторизации с помощью Express и дополнительных пакетов. Обсуждение защиты данных для исключения небезопасных ссылок в работе приложения.	
	Практическая работа (13 ч.)	Имплементация аутентификации и авторизации с применением CRUD операций над сущностями в БД	
	Самостоятель ная работа (7 ч.)	Работа по реализации веб-приложения с регистрацией и авторизацией, с учётом хэширования паролей в БД.	

№ п/п, название темы	Виды учебных занятий, учебных работ	Содержание	Планируемые результаты обучения (Знать/Уметь/Владеть)
3.4. CORS и способами работы с переменным и окружения, знакомство с протоколом WebSocket	Лекции (5 ч.)	Изучение браузерной политики безопасности SOP, изучение механизма CORS. Разбор пакетов для реализации работы CORS на сервере.	- Безопасность веб-приложений (SOP, CORS) - Типизированное программирование (TypeScript) - Практическая реализация сложных веб-проектов - Работа с различными технологиями в рамках одного проекта - Создание полноценных веб-приложений с учетом всех аспектов разработки
	Практическая работа (13 ч.)	Применение протокола WS на практических задачах, на клиентской и серверной стороне. Использование переменных окружения и пакетов для работы с файлом .env.	
	Самостоятельная работа (7 ч.)	Самостоятельная подготовка к проекту от заказчика, написание скелета приложения для быстрой реализации базовой регистрации и аутентификации.	
3.5. Проект от заказчика, выполнение требований по ТЗ	Лекции (1 ч.)	Обсуждение плана групповой разработки с учетом совместной работы с Git и выгрузкой кода на сервис GitHub.	
	Практическая работа (13 ч.)	Выполнение требований по ТЗ к проекту от заказчика: регистрация, авторизация, хэширование паролей в БД, все CRUD операции над сущностью пользователя и вспомогательными сущностями проекта.	
	Самостоятельная работа (11 ч.)	Решение задач раздела 3, решение экзаменационного материала в практическом и теоретическом варианте. Экзамен.	
3.6. TypeScript, включение TypeScript в клиентский код, базовая типизация и дженерики	Лекции (3 ч.)	Знакомство с TypeScript и его возможностями, знакомство с типизацией и её применением в работе. Разбор клиентского кода с TypeScript.	
	Практическая работа (12 ч.)	Практическое применение TypeScript, реализация собственных типов, типизация функций, параметров и методов. Применение сторонних API в работе клиентского кода.	
	Самостоятельная работа (10 ч.)	Самостоятельная реализация личного проекта с использованием TypeScript на клиентской стороне.	

№ п/п, название темы	Виды учебных занятий, учебных работ	Содержание	Планируемые результаты обучения (Знать/Уметь/Владеть)
Раздел 4. Разработка веб-приложений с использованием TypeScript и React			
4.1. React, базовые хуки и типизация основных событий	Лекции (5 ч.)	Изучение хуков: useState, useEffect, useContext, useReducer и их типизация. Работа со сборщиком клиентского приложения Vite.	Знать: - Принципы работы с расширенными хуками React и функциональными компонентами. - Основы валидации клиентского ввода и работы со сторонними библиотеками компонентов. - Реализацию клиентской маршрутизации. - Концепции Redux Toolkit, Slices и Thunk. Уметь: - Оптимизировать клиентский UI с помощью хуков. - Исправлять ошибки сборки и проводить контейнеризацию приложения с помощью Docker. Владеть: - Методами развертывания приложений на сервере.
	Практическая работа (13 ч.)	Применение в практической работе нативных хуков React. Реализация приложения прогноза погоды со сторонним API. Работа с собственным API, формирование списков ресурсов из БД на стороне клиента.	
	Самостоятельная работа (7 ч.)	Самостоятельная реализация игры "Своя игра", с использованием типизированных React событий	
4.2 Хуки оптимизации, библиотека клиентского роутинга React Router	Лекции (5 ч.)	Изучение хуков: useCallback, useMemo. Обзор библиотеки React Router, для клиентской маршрутизации, обсуждение способов параметризованных запросов на клиенте.	
	Практическая работа (13 ч.)	Практическое применение библиотеки React Router в сборщике Vite, реализация многостраничного веб-приложения с CRUD операциями клиентского состояния.	
	Самостоятельная работа (7 ч.)	Самостоятельная реализация игры "Своя игра", с использованием типизированных React событий и HTTP сервера с БД.	
4.3. Знакомство с менеджером состояний Redux/RTK	Лекции (5 ч.)	Изучение хуков: useSelector, useDispatch и их типизация. Перенос состояния из React приложения в менеджер состояний RTK.	
	Практическая работа (13 ч.)	Использование в работе библиотеки RTK. Реализация асинхронных операций с помощью RTK Thunk.	

№ п/п, название темы	Виды учебных занятий, учебных работ	Содержание	Планируемые результаты обучения (Знать/Уметь/Владеть)
		Сборка клиентского кода с исправлением ошибок типизации.	
	Самостоятельная работа (7 ч.)	Решение задач раздела 4, решение экзаменационного материала в практическом и теоретическом варианте. Экзамен.	
4.4 Финальный проект / Итоговая аттестация	Лекция (3 ч.)	Обсуждение проектировки проекта, версионирование ПО, объяснение этапов разработки и сборки проекта.	
	Практическая работа (31 ч.)	Разработка БД под нужды финального проекта, архитектурное проектирование клиентского и серверного кода. Оптимизация клиентского и серверного кода. Презентация технических возможностей проекта.	
	Самостоятельная работа (41 ч.)	Реализация технических требований согласно ТЗ финального проекта. Развёртывание проекта на реальном сервере с помощью технологии Docker. Ведение базовой документации проекта.	

8.4. Текущая аттестация.

Текущий контроль успеваемости и качества подготовки осуществляется в пределах времени, отведенного на учебные занятия, и выполняет одновременно обучающую функцию. Текущий контроль успеваемости проводится в процессе изучения каждого раздела (темы, подтемы) внутри модуля данной Программы и проводится как в форме устного опроса преподавателем, так и практического написания программного кода.

Промежуточная и итоговая аттестации проводятся в форме написания программного кода на ноутбуке или персональном компьютере слушателя, который использовался во время обучения, под дистанционным наблюдением преподавателя.

По окончании каждого модуля рабочей программы проводится промежуточная аттестация в виде практической работы (экзамена), по теме каждого модуля данной Программы.

8.5. Оценочные средства промежуточной аттестации обучающихся по освоению рабочей программы

Раздел 1

Необходимый минимум для повтора (иначе незачет):

- студент понимает структуру HTML кода
- студент знает базовые свойства CSS
- студент знает базовый синтаксис JavaScript: переменные, типы данных, циклы

Зачёт:

- выполнение итогового проекта

Раздел 2

Необходимый минимум для повтора (иначе незачет):

- студент знает как установить необходимые npm пакеты для экзамена
- студент умеет экспортировать и импортировать сущности
- 50% от любой задачи сделаны (либо в первой либо во второй части проходят половина тестов, которые изначально не проходили, либо в третьей выделен один базовый класс)
- уметь создавать миграцию и объяснить ее назначение

Зачёт:

- part-1 полностью решён, студент может аргументировать любую часть
- part-2 полностью решён или есть незначительные ошибки в реализации
- part-3 не решён, но студент может объяснить смысл задачи и при необходимости напишет способ решения
- part-4 не решён, но студент понимает как создается БД, что такое таблицы, что такое модели и миграции

Раздел 3

Необходимый минимум для повтора (иначе незачет):

- есть базовая маршрутизация в отдельном файле (хотя бы 1 router)
- есть сиды, миграции и модели для ДБ согласно ТЗ
- есть минимум 2 CRUD операции (READ + любая на выбор) над основной сущностью
- студент может пояснить свой код

Зачёт:

- реализованы все CRUD операции над основной сущностью (не User) по теме экзамена, как минимум две из них без перезагрузки страницы
- есть проверка на IDOR (на уровне серверных роутов проверяется, что пользователь может удалить / изменить только свои сущности)
- авторизация / регистрация

Раздел 4

Необходимый минимум для повтора (иначе незачет):

- самостоятельно реализовать вывод сущностей на страницу на основе библиотеки Redux/Redux Toolkit
- объяснить назначение 5 любых нативных хуков React на выбор студента — пояснить, что эти хуки принимают, что возвращают
- объяснить назначение 4 любых сторонних хука React (из программы обучения) на выбор студента — пояснить, что эти хуки принимают, что возвращают
- изучить серверный код экзамена и самостоятельно реализовать одну из CRUD операций (кроме вывода) на основе библиотеки Redux на выбор преподавателя

Примеры заданий, выявляющих уровень сформированности компетенций:

- 0 фаза: продемонстрировать реализацию метода `document.createElement()`
- 1 фаза: продемонстрировать реализацию методов массива `map()` и `filter()`
- 2 фаза: продемонстрировать реализацию `fetch API`
- 3 фаза: продемонстрировать формирование списка элементов в UI библиотеке React с использованием TypeScript

8.6. Литература

Основная⁹:

1. Дэвид Флэнаган, JavaScript. Полное руководство: Вильямс, 2021 - 720 с. - ISBN - 9785907203792
2. Марейн Хавербеке, Выразительный JavaScript. Современное веб-программирование.: Питер, 2019. - 480 с. - ISBN - 978-5-4461-1226-5
3. Адитья Бхаргава, Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих.: Питер, 2024. - 288 с. - ISBN - 978-5-4461-0923-4

Дополнительная¹⁰:

1. Роберт Мартин, Чистый код. Создание, анализ и рефакторинг.: Питер, 2024. - 464 с. - ISBN - 978-5-4461-0960-9
2. Кайл Симпсон, Вы не знаете JavaScript.: Питер, 2022. - 192 с. - ISBN - 9785446118755

Интернет-ресурсы:

1. Современный учебник JavaScript. Ресурсы: <https://learn.javascript.ru>
2. Ресурсы для разработчиков. Ресурсы: <https://developer.mozilla.org/ru/>
3. JavaScript Tutorial. Ресурсы: <https://www.w3schools.com/>

9. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

9.1. Организация образовательной деятельности по освоению программы: При реализации программы используются дистанционные образовательные технологии.

Учебный процесс происходит с использованием разнообразных методов организации и осуществления учебно-познавательной деятельности (словесные, наглядные и практические методы передачи информации, проблемные лекции и др.); стимулирования и мотивации учебно-познавательной деятельности (дискуссии и др.); контроля и самоконтроля. Используются активные формы проведения занятий: ситуационный анализ, ролевые игры, эвристические технологии, тренинги.

9.2. Продолжительность одного занятия и трудоемкость недельной нагрузки не более 10 академических часов в день, перерыв на обед 90 минут, не более 25 академических часов в неделю.

9.3 Материально-техническая база реализации программы

соответствует действующим противопожарным правилам и нормам и обеспечивает проведение всех видов дисциплинарной и междисциплинарной подготовки, практической работы обучающихся, предусмотренной учебным планом.

Все ноутбуки, персональные компьютеры слушателей на период проведения обучения должны соответствовать техническим характеристикам:

- процессор 4 ядра 2,7 ГГц
- оперативная память - 8 Гб
- жесткий диск - 1 Тб
- монитор 21,5 ", разрешение 1920x1080
- установка ОС Ubuntu, Windows, MacOS
- поддержка архитектуры x64
- использовать SSD и процессоры с оценкой от 2500 и выше.

9.4. Квалификация научно-педагогических работников соответствует квалификационным характеристикам, установленным квалификационными требованиями к работникам, утвержденными Министерством образования Российской Федерации, и квалификационным характеристикам, установленным в Едином квалификационном справочнике должностей руководителей, специалистов и служащих, разделе «Квалификационные характеристики должностей руководителей и специалистов высшего профессионального и дополнительного профессионального образования» и профессиональным стандартам *(при наличии)*.

10. ПРОГРАММА ИТОГОВОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ

Обучающиеся допускаются к итоговой аттестации после изучения дисциплин в объеме, предусмотренном учебным планом программы.

Рекомендуемый порядок проведения итоговой аттестации включает в себя последовательность этапов:

- Тестирование, направленное на контроль и оценку знаний, умений, составляющих содержание профессиональных компетенций.
- Решение ситуационных задач, направленное на контроль и оценку умений, составляющих содержание профессиональных компетенций.
- Выполнение практических заданий, проверяющих уровень сформированности профессиональной компетенции.

Обучающиеся, освоившие дополнительную профессиональную программу переподготовки по специальности/теме «Разработчик JavaScript» и успешно прошедшие итоговую аттестацию, получают документ установленного образца о дополнительном профессиональном образовании – диплом о профессиональной переподготовке.

10.1 Итоговая аттестация

Подготовка презентации выпускного проекта согласно к дате общих показов (защиты дипломных работ).

11. ОЦЕНОЧНЫЕ СРЕДСТВА ИТОГОВОЙ АТТЕСТАЦИИ

11.1. Примерная тематика контрольных вопросов, выявляющих теоретическую подготовку обучающегося:

1. Какие основные типы данных в JavaScript?
2. Что такое переменные и как их объявлять?
3. Что такое объекты и массивы в JavaScript?
4. Как работает асинхронное программирование в JavaScript?
5. Как создавать пользовательский интерфейс на JavaScript?
6. Что такое Node.js и как он используется в FullStack разработке?
7. Что такое REST API и как его реализовать на Node.js?
8. Что такое ORM и как они используются в FullStack проектах?
9. Как использовать PostgreSQL в Node.js приложениях?
10. Какой функционал предоставляют хуки в React?
11. Объяснить понятия состояние компонента React
12. Как реализуется сборка проекта?
13. Как реализуется развертывание проекта на реальном сервере?

11.2. Примеры заданий, выявляющих практическую подготовку обучающегося:

Основные функциональные требования к проекту:

1. Реализовать систему регистрации пользователей:
 - 1.1 Создание новых аккаунтов
 - 1.2 Валидация входных данных
 - 1.3 Хранение информации о пользователях в базе данных
2. Реализовать систему авторизации:

- 2.1 Логин пользователей
- 2.2 Проверка учетных данных
- 2.3 Генерация и управление JWT токенами
- 3. Разработать RESTful API:
 - 3.1 Определить эндпоинты для регистрации, логина, получения текущего пользователя
 - 3.2 Реализовать обработку запросов и ответов в формате JSON
 - 3.3 Настроить CORS для взаимодействия между клиентом и сервером
- 4. Создать клиентское приложение на React:
 - 4.1 Реализовать интерфейс регистрации и авторизации
 - 4.2 Базовая оптимизация клиентской логики
- 5. Использование TypeScript на клиентской части проекта:
 - 5.1. Настройка типизации
 - 5.2 Реализация проверки типов во время компиляции
- 6. Настроить Docker:
 - 6.1 Создать Dockerfile для сервера
 - 6.2 Настроить сборку и запуск приложения через Docker
 - 6.3 Настроить выгрузку итогового кода на реальный сервер (развертывание приложения)

Оценка создания слушателем веб-приложения формируется на основании следующих требований:

1. Сервер полностью соответствует архитектурному стилю REST API
2. Применено хэширование паролей пользователей с помощью алгоритма SHA-2 или с использованием специализированных NPM-пакетов
3. В качестве СУБД использована PostgreSQL в связке с ORM Sequelize
4. В качестве frontend-библиотеки используется JavaScript библиотека React
5. В качестве менеджера состояний используется Redux Toolkit
6. Frontend должен полностью соответствовать концепции SPA
7. Реализована адаптивная кроссбраузерная верстка

12. ИНЫЕ КОМПОНЕНТЫ ПРОГРАММЫ

12.1. Кадровое обеспечение образовательного процесса

№ п/ п	Фамилия, имя, отчество,	Основное место работы, должность	Место работы и должность по совместительству
1	Бабаян Георгий Георгиевич	Генеральный директор	соавтор курса, ООО "ЭЛЬБРУС БУТКЕМП"
2	Плисковский Юрий Валерьевич	Директор образовательной программы	соавтор курса, ООО "ЭЛЬБРУС БУТКЕМП"
3	Князев Александр Юрьевич	Руководитель команды преподавателей г. Москва	соавтор курса, ООО "ЭЛЬБРУС БУТКЕМП"