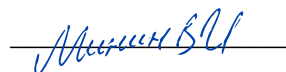


Индивидуальный предприниматель Минин Владилен Игоревич
ИНН 78171431655 ОГРНИП 317784700113455

УТВЕРЖДЕНО

Приказом ИП Минина В. И.
№ 1/ОБР
от “05” ноября 2024



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ПРОГРАММА – ДОПОЛНИТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

«Frontend-разработка»

Направленность: техническая

Адресат программы: взрослые от 18 лет

Объем программы (трудоемкость): 190 академических часов

Срок реализации программы: 1 год

Форма обучения: заочная (с применением исключительно электронного обучения, дистанционных образовательных технологий)

Ленинградская область

2024г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1. Дополнительная общеобразовательная общеразвивающая программа «Frontend-разработка» (далее – программа) разработана в соответствии со следующими нормативно-правовыми актами Российской Федерации:

- Федеральным законом Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) – Письмо Минобрнауки России от 18.11.2015 № 09-3242 О направлении информации (вместе с Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)).
- Постановлением Правительства РФ от 26.12.2017 № 1642 (ред. от 20.05.2022) "Об утверждении государственной программы Российской Федерации "Развитие образования"
- Постановлением Правительства РФ от 11.10.2023 N 1678 "Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ"
- Приказом Минпросвещения РФ от 27.07.2022 № 629 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам"
- Письмом Минобрнауки России от 18.11.2015 № 09-3242 "О направлении информации" (вместе с "Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)"))
-

1.2. Направленность программы: техническая

1.3. Уровень освоения – стартовый (ознакомительный).

1.4. Актуальность программы состоит в том, что современный период общественного развития характеризуется новыми требованиями к информационному образованию граждан, предполагающими ориентацию не только на усвоение учащимися определенной суммы знаний, но и на развитие их личности, их познавательных и созидательных способностей.

В условиях цифровизации, массовой коммуникации современного общества особую значимость приобретает получение знаний в области программирования, в частности, в области Frontend-разработки.

Frontend-разработка (фронтенд англ. frontend) — это разработка пользовательских функций и интерфейса. К ним относится всё, что пользователи видят на сайте или в приложении, и с чем можно взаимодействовать: картинки, выпадающие списки, меню, анимация, карточки товаров, кнопки, чекбоксы, интерактивные элементы. На любой странице в интернете виден результат работы фронтенд-разработчика.

Развитие информационно-коммуникационных и цифровых технологий сегодня идет стремительными темпами. Потребность в программистах и технически грамотных специалистах растет с каждым днем. Поэтому, очень важно стимулировать у обучающихся интерес к новым знаниям и пониманию базового устройства компьютерных программ для дальнейшего применения этих знаний и умений в будущей жизнедеятельности

современного человека.

Актуальность и новизна программы заключается в том, что она позволяет осуществить социальный заказ, обусловленный значимостью информатизации и цифровизации современного общества; активизировать познавательную деятельность обучающихся; реализовать их интерес к техническому направлению. Программа дает возможность реализовать обучающимся свои творческие, исследовательские способности посредством программирования в различных средах.

1.5. Адресат программы – программа предназначена для людей с возрастной категорией от 18 лет без предъявления требований к уровню образования.

1.6. Объем программы: 190 учебных часов.

1.7. Срок реализации программы: 1 год.

1.8. Режим занятий: 2 раза в неделю по 2 акад. часа (2 занятия по 45 минут с 10-минутным перерывом).

1.9. Форма обучения: заочная (с применением исключительно дистанционных образовательных технологий и электронного обучения).

1.10. Язык реализации программы – русский.

Особенности набора: программа предусматривает свободный набор взрослых. Уровень подготовки не требуется, так как программа рассчитана на стартовый (ознакомительный) уровень.

Отличительными особенностями данной программы являются:

☐ определение видов организации деятельности обучающихся, направленных на достижение личностных, метапредметных и предметных результатов;

☐ в основу реализации программы положен учет возрастных и индивидуальных особенностей обучающихся;

☐ достижения планируемых результатов отслеживаются в рамках внутренней системы оценки педагогом и обучающимися.

Ведущая идея, на которой базируется программа: каждый обучающийся есть неповторимая индивидуальность, обладающая свойственными только ей психическими, физическими и прочими особенностями. Необходимо всестороннее изучение этих особенностей и творческий, комплексный подход к формам и методам их развития.

2. ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ

Цель программы – развитие познавательных и творческих способностей обучающихся, удовлетворение их индивидуальных потребностей в интеллектуальном совершенствовании посредством освоения теоретических и практических основ Frontend-разработки.

Задачи программы:

Обучающие задачи:

- получение слушателями сведений о современных технологиях программирования; интерпретируемых языках программирования; сценарных языках программирования; компонентах программно-технических архитектур, существующих приложениях и интерфейсах взаимодействия с ними;
- формирование знаний, умений и навыков по вопросам применения языков программирования для написания программного кода и разработки SPA приложения с клиент-серверной архитектурой;
- сформировать общекультурные навыки работы с информацией (умения, связанные с поиском, пониманием, организацией, архивированием цифровой информации и ее критическим осмыслением, а также с созданием информационных объектов с использованием цифровых ресурсов (текстовых, изобразительных, аудио и видео);
- сформировать знания, позволяющие эффективно и безопасно использовать технические и программные средства для решения различных задач, в том числе использования компьютерных сетей, облачных сервисов и т.п.;
- сформировать знания, умения, мотивацию и ответственность, позволяющие решать с помощью цифровых устройств и интернета различные задачи, связанные с конкретными жизненными ситуациями, предполагающими удовлетворение различных потребностей;

Развивающие задачи:

- Развить общую культуру обучающихся;
- Учить планировать процесс познавательной деятельности, находить необходимую для выполнения работы информацию в различных источниках, анализировать предлагаемую информацию;
- Развить способности к самостоятельной работе и анализу проделанной работы.
- Развивать познавательную и творческую активность в безопасном использовании информационных и коммуникационных технологий;
- Развивать компьютерную грамотность, информационную культуру личности в использовании информационных и коммуникационных технологий;

Воспитательные задачи:

- Воспитать умение прислушиваться к мнению других, уважение к их точке зрения;
- Развить проявление технологического мышления при организации своей деятельности;
- Сформировать навыки критического мышления.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Предметные результаты:

В результате обучения по данной программе учащиеся:

- научатся использовать возможности программной архитектуры информационных ресурсов;
- научатся использовать синтаксис выбранного языка программирования, стандартные библиотеки языка программирования;
- научатся применять языки программирования для написания программного кода и разработки SPA приложения с клиент-серверной архитектурой;
- научатся работать с информацией, анализировать и обобщать полученную информацию;
- научатся безопасно использовать средства коммуникации;
- овладеют приемами безопасной организации своего личного пространства данных с использованием индивидуальных накопителей данных, интернет-сервисов и т.п.;

Метапредметные результаты:

- планирование процесса познавательной деятельности;
- умение находить необходимую для выполнения работы информацию в различных источниках;
 - выбирать из предложенных вариантов и самостоятельно искать средства/ресурсы для решения задачи/достижения цели;
 - составлять план решения проблемы (выполнения проекта, проведения исследования);
 - описывать свой опыт, оформляя его для передачи другим людям в виде технологии решения практических задач определенного класса;
 - работая по своему плану, вносить коррективы в текущую деятельность на основе анализа изменений ситуации для получения запланированных характеристик продукта/результата;
 - принимать решение в учебной ситуации и нести за него ответственность;
 - целенаправленно искать и использовать информационные ресурсы, необходимые для решения учебных и практических задач с помощью средств ИКТ;
 - выбирать, строить и использовать адекватную информационную модель для передачи своих мыслей средствами естественных и формальных языков в соответствии с условиями коммуникации;
 - использовать компьютерные технологии (включая выбор адекватных задаче инструментальных программно-аппаратных средств и сервисов) для решения информационных и коммуникационных задач;
 - использовать информацию с учетом этических и правовых норм;
 - создавать информационные ресурсы разного типа и для разных аудиторий, соблюдать правила информационной безопасности.

Личностные результаты:

- осознанное, уважительное и доброжелательное отношение к окружающим людям в реальном и виртуальном мире, их позициям, взглядам, готовность вести диалог с другими людьми;

- готовность и способность к осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в мире профессий и профессиональных предпочтений, с учетом устойчивых познавательных интересов;
- освоенность социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах;
- сформированность понимания ценности безопасного образа жизни; интериоризация правил индивидуального и коллективного безопасного поведения в информационно-телекоммуникационной среде.

В результате освоения Программы обучающиеся

будут знать:

Устройство и функционирование современных информационных ресурсов;
 Архитектуру, устройство и принцип функционирования вычислительных систем;
 Принципы работы коммуникационного оборудования;
 Сетевые протоколы и основы web-технологий;
 Основы современных систем управления базами данных;
 Теорию баз данных;
 Системы хранения и анализа баз данных;
 Современные объектно-ориентированные языки программирования;
 Программные средства и платформы для разработки web-ресурсов;
 Основы информационной безопасности web-ресурсов;
 Стандартные библиотеки выбранного языка программирования;
 Методологии разработки программного обеспечения;
 Технологии программирования;
 Современные интерпретируемые языки программирования;
 Современные сценарные языки программирования;
 Современные методики тестирования эргономики пользовательских интерфейсов;

будут уметь:

Применять выбранные языки программирования для написания программного кода;
 Использовать выбранную среду программирования и средства системы управления базами данных;
 Кодировать на скриптовых языках программирования;
 Выбирать и комбинировать техники тестирования информационных ресурсов;
 Применять инструменты подготовки тестовых данных.

4. УЧЕБНЫЙ ПЛАН ПРОГРАММЫ «FRONTEND-РАЗРАБОТКА»

№ п/п	Наименование компонентов программы	Трудоемкость, ак. ч. ¹				Форма контроля
		Всего	Теория	Практика	Кон- троль	
1	Вводный модуль	10	4	5	1	
2	Модуль 1. База JavaScript	74	27	27	20	Зачет
3	Модуль 2. Frontend	72	38	17	17	Зачет
4	Модуль 3. Backend	32	14	9	9	
5.	Промежуточная аттестация (зачет)	2			2	Защита курсового проекта
6.	Итого	190	83	58	49	

5. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Календарный учебный график является примерным и утверждается отдельно для каждой учебной группы.

Режим занятий: 4 академических часа в неделю (2 раза в неделю по 2 акад. часа).

Дата начала занятий	Дата окончания занятий	Кол-во учебных недель	Кол-во учебных часов	Режим занятий
05.09.2022	09.08.2023	48	190	2 раза в неделю по 2 акад. часа ²

¹ Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут.

² Для всех видов занятий академический час устанавливается продолжительностью 45 минут.

6. РАБОЧАЯ ПРОГРАММА

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН ПРОГРАММЫ «FRONTEND-РАЗРАБОТКА»

№ п/п	Наименование компонентов программы	Трудоемкость, ак. ч. ³				Форма контроля
		Всего	Теория	Практика	Кон- троль	
1	Вводный модуль	10	4	5	1	
1.1	Приветствие	1	1	-		
1.2	Правила обучения	1	1	-		
1.3	Команда проекта	1	-	1		
1.4	Анкета участника	1	-	1		
1.5	Начала пути. Точка «А». Входное тестирование	1	-	1		Тестирование
1.6	Точка «Б». Ставим цель на курс	1	-	1		
1.7	Введение в веб-разработку	4	2	1	1	Контрольное задание
2	Модуль 1. База JavaScript	74	27	27	20	Зачет
2.1	Добро пожаловать в мир JavaScript!	1	1	-	-	
2.2	Чек-лист по JavaScript	1	1	-	-	
2.3	Hello World	3	1	1	1	Тестирование
2.4	Синтаксис	3	1	1	1	Тестирование
2.5	Типы данных	3	1	1	1	Тестирование
2.6	Приведение типов в JavaScript	3	1	1	1	Тестирование
2.7	Работа со строками	3	1	1	1	Тестирование
2.8	Введение в основы JavaScript	4	2	1	1	Тестирование
2.9	Дополнительные задания	1	-	1	-	
2.10	Debug	3	1	1	1	Тестирование
2.11	Функции	3	1	1	1	Тестирование

³ Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут.

2.12	Массивы	4	2	1	1	Тестирование
2.13	Дополнительные задания	1	-	1	-	
2.14	Объекты	4	2	1	1	Тестирование
2.15	Дата и Время в JavaScript	3	1	1	1	Тестирование
2.16	Дополнительные задания	2	-	2	-	
2.17	Ключевое слово "this"	3	1	1	1	Тестирование
2.18	Классы + ООП	3	1	1	1	Тестирование
2.19	Дополнительные задания	2	-	2	-	
2.20	Работа с DOM	5	3	1	1	Тестирование
2.21	Дополнительные задания	1	-	1	-	
2.22	Асинхронность (Promise + Fetch)	3	1	1	1	Тестирование
2.23	Асинхронность (Async Await)	3	1	1	1	Тестирование
2.24	Callback + Event Loop	3	1	1	1	Тестирование
2.25	Дополнительные задания	1	-	1	-	
2.26	Webpack, rollup	3	1	1	1	Тестирование
2.27	ES6 import export	4	2	1	1	Тестирование
2.28	Промежуточная аттестация (итоговый тест модуля)	1	-	-	1	Зачет
3	Модуль 2. Frontend	72	38	17	17	Зачет
3.1	Добро пожаловать в React	1	1	-		
3.2	Введение в React	2	1	-	1	Тестирование
3.3	Компоненты	2	1	-	1	Тестирование
3.4	Курсовой проект	2	1	1		
3.5	Композиция компонентов	2	1	1		
3.6	Пагинация	3	1	1	1	Тестирование
3.7	Фильтрация	3	1	1	1	Тестирование

3.8	Сортировка	4	2	1	1	Тестирование
3.9	Маршрутизация	4	2	1	1	Тестирование
3.10	Курсовой проект	2	-	2		
3.11	Формы. Часть 1	4	2	1	1	Тестирование
3.12	Формы. Часть 2	4	2	1	1	Тестирование
3.13	Расширенные хуки и базовая оптимизация	4	2	1	1	Тестирование
3.14	Структуры данных	3	2	-	1	Тестирование
3.15	Из HTML верстки в React UI	1		1		
3.16	Вызов серверных служб (API), кастомные хуки и управление состоянием	7	5	1	1	Тестирование
3.17.	Firebase и трансформеры данных	3	2	-	1	Тестирование
3.18	Авторизация и аутентификация. Часть 1	4	2	1	1	Тестирование
3.19	Авторизация и аутентификация. Часть 2	5	3	1	1	Тестирование
3.20	Redux. Часть 1	6	4	1	1	Тестирование
3.21	Redux. Часть 2	5	3	1	1	Тестирование
3.22.	Промежуточная аттестация (итоговый тест модуля)	1	-	-	1	Зачет
4	Модуль 3. Backend	32	14	9	9	
4.1.	Введение	2	1	-	1	Тестирование
4.2.	Основы	4	2	1	1	Тестирование
4.3.	Стандартные модули	2	1	-	1	Тестирование
4.4.	Web-сервер	4	2	1	1	Тестирование
4.5.	Дополнительные задания	2	-	2		
4.6.	Создаем Backend. Часть 1	4	2	1	1	Тестирование
4.7.	Создаем Backend. Часть 2	4	2	1	1	Тестирование
4.8.	Создаем Backend. Часть 3	3	1	1	1	Тестирование
4.9.	Интеграция Backend & Frontend	3	1	1	1	Тестирование
4.10.	Deployment	4	2	1	1	Тестирование
5.	Итоговая аттестация (зачет)	2			2	Защита курсового

6.	Итого	190	83	58	49	проекта
----	-------	-----	----	----	----	---------

СОДЕРЖАНИЕ ТЕМ РАБОЧЕЙ ПРОГРАММЫ

«FRONTEND-РАЗРАБОТКА»

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Вводный модуль (10 ч.)		
Теория	1	Тема 1.1. Приветствие
Теория	1	Тема 1.2. Правила обучения
Практика	1	Тема 1.3. Команда проекта Автор модуля «Backend». Автор модуля "База JavaScript". Автор модуля "Frontend"
Практика	1	Тема 1.4. Анкета участника
Практика	1	Тема 1.5 Начала пути. Точка «А». Входное тестирование
Практика	1	Тема 1.6. Точка «Б». Ставим цель на курс
Теория	2	Тема 1.7. Введение в веб-разработку JavaScript и HTML/CSS
Практика	1	Тема 1.7. Введение в веб-разработку
Текущий контроль	1	Контрольное задание
Модуль 1. База JavaScript (74 ч.)		
Теория	1	Тема 2.1. Добро пожаловать в мир JavaScript!
Теория	1	Тема 2.2. Чек-лист по JavaScript
Теория	1	Тема 2.3. Hello World Установка редактора кода Visual Studio Code на компьютер. Базовый интерфейс программы. Подключение JavaScript-файл в HTML-файл и вывод на экран строки "Hello World"
Практика	1	Практическая работа
Текущий контроль	1	Тестирование

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Теория	1	Тема 2.4. Синтаксис Написание базового JavaScript-кода без синтаксических ошибок. Создание комментариев в коде, объявление переменных
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	1	Тема 2.5. Типы данных Типы данных в JavaScript, для чего они нужны и как их применять на практике
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	1	Тема 2.6. Приведение типов в JavaScript Принцип работы нестрого типизированных языков программирования (как JavaScript). Конвертация одного типа данных в другой (например, строку в число)
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	1	Тема 2.7. Работа со строками Базовые операции над строками. Конкатенация (объединение) строк, обрезка строки, поиск нужных символов и фраз
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	2	Тема 2.8. Введение в основы JavaScript Математика и операторы сравнения. Логические конструкции. Логические операторы. Циклы.
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Практика	1	Тема 2.9. Дополнительные задания

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Теория	1	Тема 2.10. Debug Ошибки (баги) в коде, способы их исправления. Методы console (console.warn, console.error, console.debug, и т.д.)
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	1	Тема 2.11. Функции Значимость функций в JavaScript. Типы функций. Написание функции, правильное её наименование и использование в коде
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	2	Тема 2.12. Массивы Массивы. Добавление, удаление, изменение элементов. Перебор массивов с помощью for, for...of, forEach. Работа с методами массивов: map, filter, find, findIndex, some, every. Метод массивов reduce. Метод массивов sort. Методы массивов: splice, slice, indexOf, includes, join, reverse. Объединение нескольких массивов в один
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Практика	1	Тема 2.13. Дополнительные задания
Теория	2	Тема 2.14. Объекты Создание объектов. Получение значений объектов по ключу. Объект - ссылочный тип данных. Перебор объектов. Создание массивов из объектов (keys, values, entries). Работа с ключами объекта. Объединение нескольких объектов в один. Опциональная цепочка. Spread и Rest. Деструктуризация.
Практика	1	Практическая работа
Текущий контроль	1	Тестирование

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Теория	1	Тема 2.15. Дата и Время в JavaScript Дата и Время в JavaScript. Разница между двумя датами
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Практика	2	Тема 2.16. Дополнительные задания
Теория	1	Тема 2.17. Ключевое слово "this" Методы объекта и ключевое слово "this". Что такое "this" и зачем он нужен. Методы bind, call и apply. Что такое контекст "this". Как работает call, bind, apply. Потеря контекста "this".
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	1	Тема 2.18. Классы + ООП Классы в JavaScript. Создание экземпляров объекта при помощи function и class. Принципы ООП. Наследование. Принципы ООП. Инкапсуляция. Принципы ООП. Полиморфизм. Принципы ООП. Абстракция. Статические поля и методы. Конструкторы, создание объектов через new
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Практика	2	Тема 2.19. Дополнительные задания
Теория	3	Тема 2.20. Работа с DOM DOM-дерево и получение HTML-элементов через JavaScript. Свойства и методы HTML-элементов. Создание HTML-элементов и добавление их в DOM. Остальные методы HTML-элементов. Обработчики событий. Событие click. Обработчики событий. Событие submit. Обработчики событий. События keydown и keyup. Обработчики событий. События mouseover, mouseout, mousemove. Обработчики событий. События contextmenu, change, input. Обработчики событий. Всплытие и погружение. Прекращение всплытия. Обработчики событий. Делегирование событий.

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Практика	1	Тема 2.21. Дополнительные задания
Теория	1	Тема 2.22. Асинхронность (Promise + Fetch) Отличие асинхронного и синхронного кода. Что такое Promise и как с ним работать. Работа с функцией fetch. Получение данных и отрисовка их в HTML Promise.all. Promise.race
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	1	Тема 2.23. Асинхронность (Async Await) Async await в JavaScript. Как работает Async, Await. Работа с сервером с fetch
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	1	Тема 2.24. Callback + Event Loop Что такое callback и как его использовать в коде Event Loop. Что такое Call Stack. Event Loop. Что такое Очередь Вызовов (Callback queue). Event Loop. Макрозадачи и микрозадачи. Очередь микрозадач
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Практика	1	Тема 2.25. Дополнительные задания
Теория	1	Тема 2.26. Webpack, rollup Сборщики проектов. Установка NodeJS и NPM. Установка Rollup. Базовая конфигурация проекта. Установка дополнительных пакетов для сборки Rollup. Webpack. Установка и конфигурация
Практика	1	Практическая работа
Текущий контроль	1	Тестирование

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Теория	2	Тема 2.27. ES6 import export Использование import, export, export default. Реализация структуры проекта. Разработка класса JSBlock. Работа с import и export. Разработка класса TimerBlock. Работа с import и export. Вспомогательные функции. Работа с import all. Внесение последних правок в проект. LocalStorage.
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Промежуточная аттестация	1	Зачет (итоговый тест модуля)
Модуль 2. Frontend (72 ч.)		
Теория	1	Тема 3.1. Добро пожаловать в React
Теория	1	Тема 3.2. Введение в React Что такое React.js. Create React App. Структура шаблона приложения. Что такое Babel и JSX. Bootstrap. VS Code-расширения
Текущий контроль	1	Тестирование
Теория	1	Тема 3.3. Компоненты Вступление. Первый React-компонент. Указание дочерних компонентов. Вложенные выражения. Установка атрибутов. Динамический рендер классов. Обработка событий. useState. Рендер списка из массива. Передача аргументов событий. Условный рендеринг
Текущий контроль	1	Тестирование
Теория	1	Тема 3.4. Курсовой проект Прототип курсового проекта. Настройка курсового проекта.
Практика	1	Практическая работа

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Теория	1	Тема 3.5. Композиция компонентов Вступление. Рефакторинг кода. Композиция компонентов. Передача данных в компонент. Передача контента компонента в сам компонент. Debug. Отладка React-приложений. Props vs State. Создание и обработка событий. Обновление состояния. Передача атрибутов одной сущности. Единый источник истины
Практика	1	Практическая работа
Теория	1	Тема 3.6. Пагинация Вступление. Simple React Snippets. Интерфейс компонента. Отображение разбиения на страницы. Обработка изменений страницы. Разбиение данных на страницы. Проверка типов с помощью PropTypes. Настройка ESLint
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	1	Тема 3.7. Фильтрация Вступление. Интерфейс компонента фильтрации. Расширение API. useEffect. Отображение элементов. Свойства по умолчанию. Обработка выбора. Реализация фильтрации. Добавление всех типов. Рефакторинг
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	2	Тема 3.8. Сортировка Извлечение таблицы. Реализация события сортировки. Реализация сортировки. Универсальный TableHeader. Извлечение TableHeader. Извлечение TableBody. Ячейка вложенного компонента. Реализация качества пользователей. Извлечение таблицы и рефакторинг
Практика	1	Практическая работа
Текущий контроль	1	Тестирование

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Теория	2	Тема 3.9. Маршрутизация Установка react-router-dom. React Router Dom v6. Реализация Navbar. Switch и его особенности. Link. RouteProps. Параметры и свойства Route. Опциональные параметры. Query-параметры. Переадресация (редирект). History. Программная навигация. Хуки react-router-dom. Вложенные маршруты
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Практика	2	Тема 3.10. Курсовой проект: темы и технические задания
Теория	2	Тема 3.11. Формы. Часть 1 Создание формы. Контролируемые и неконтролируемые компоненты. Обработка значений множества полей. Переиспользуемое поле ввода. Обработка отправки формы. Базовая валидация. Отображение ошибки валидации. Создание валидатора. Введение. Продвинутый валидатор. isEmail и regExp. Продвинутый валидатор. Требования к паролю. Продвинутый валидатор. Отключение кнопки отправки. Добавление стилей Bootstrap. Скрыть/показать пароль
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	2	Тема 3.12. Формы. Часть 2 Реструктуризация проекта. Форма регистрации. Select. Radio. Множественный select. Checkbox. Библиотека Yup. Последние корректировки
Практика	1	Практическая работа
Текущий контроль	1	Тестирование

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Теория	2	Тема 3.13. Расширенные хуки и базовая оптимизация Вступление. useRef. useMemo. useCallback. Компоненты высшего порядка. React.memo. React.cloneElement. Children. Создание переиспользуемого компонента формы. Оптимизация ререндеринга полей формы. Автофокус поля при открытии формы. Переход на следующее поле при нажатии Enter. Code review. Заключительные слова
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	2	Тема 3.14. Структуры данных Вступление. Нормализация / Денормализация. Тип связи один к одному (one to one). Тип связи один ко многим (one to many). Тип связи многие ко многим (many to many). Заключительные слова
Текущий контроль	1	Тестирование
Практика	1	Тема 3.15. Из HTML верстки в React UI
Теория	5	Тема 3.16. Вызов серверных служб (API), кастомные хуки и управление состоянием Введение. Установка MongoDB на Mac. Установка MongoDB на Windows. Установка Backend-приложения. Установка проекта “Качества”. HTTP-Клиенты. Получение данных. Создание новых данных. Ожидаемые и неожиданные ошибки. Глобальное отлавливание неожиданных ошибок. Переиспользуемый паттерн HTTP – service. Логирование ошибок. Переиспользуемый паттерн логирования ошибок. Отображение ошибок с помощью toastify. Паттерн переиспользуемого файла config. Извлечение переиспользуемого паттерна Qualities Service. Кастомные хуки. useForm. Context. useQualities. useQualities. Обновление данных. useQualities. Создание данных. Редирект. useQualities. Удаление данных. Пессимистические и оптимистические обновления данных. Глобальная обработка ошибок. Добавление HTTP-Сервиса. User Service. useProfessions. Заключение

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	2	Тема 3.17. Firebase и трансформеры данных Вступление. Создание базы данных в Firebase. Запросы Firebase. Axios intercepters. Request. Модификация URL Axios intercepters. Response. Трансформация данных. Инициализация Mock-данных. Заключительные слова
Текущий контроль	1	Тестирование
Теория	2	Тема 3.18. Авторизация и аутентификация. Часть 1 Введение. Определения. Процесс регистрации. Рефакторинг формы регистрации. Принципы регистрации в Firebase. Auth Provider. Метод Sign Up (Регистрация). Создание экземпляров Axios. JSON Web Token. Продолжаем создавать Sign Up. Переменные окружения. DotEnv. LocalStorage Service. Обработка ошибок регистрации. Переадресация после регистрации
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	3	Тема 3.19. Авторизация и аутентификация. Часть 2 Введение. Имя в форме регистрации. Изменяем SignUp-метод. Получение и хранение User data. Получение User data при загрузке. Refresh tokens. Firebase Rules. Авторизованный запрос. Изменяем NavBar. Добавляем пользователя в NavBar. Защищенные пути (Protected Routes). Добавление Global Loading. Переадресация после входа. LogOut. Добавление изображения пользователя. Изменяем UserList Page. Изменяем UserPage. Персонализированный интерфейс. Создаем Comment Hook. Добавляем метод создания комментария. Добавляем запрос добавления комментария Получение комментариев. Отображение комментариев. Удаление комментариев. Comments Firebase Rules и отображение кнопки удаления

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	4	Тема 3.20. Redux. Часть 1 Введение. Функциональное программирование. Установка проекта. Функции. Функции высшего порядка (higher-order functions). Композиция функций. Каррирование. Чистые функции (pure functions). Неизменяемый объект. Разработка Redux с нуля. Интро. Разработка Redux с нуля. Create Store. Разработка Redux с нуля. Dispatch. Разработка Redux с нуля. Reducer. Разработка Redux с нуля. Display Tasks. Разработка Redux с нуля. Observer Pattern. Добавление функционала. Redux Create Store. Генераторы событий. Структура проекта. Ducks pattern. Redux Toolkit. Create Action. Redux Toolkit. Create Reducer. Redux Toolkit. Create Slice. Установка инструментов разработки Redux. Redux DevTools. Middleware. Создаём Middleware. Логгер. Создаём Middleware. Thunk. Создание запросов к API. Действия для отладки запросов. Redux Toolkit. Configure Store. React Redux. Providing Store. React Redux. useSelector. React Redux. useDispatch. Добавляем загрузку и обработку ошибок. Использование нескольких редюсеров в одном Store. Создаём селекторы
Практика	1	Практическая работа
Текущий контроль	1	Тестирование

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Теория	3	Тема 3.21. Redux. Часть 2 Введение. Создаем Store. Создаем Qualities Slice. Создаем Qualities Reducer. Создаем actions и selectors для Qualities. Изменяем интерфейс. Получение актуальных данных. Создаем Users Slice. Создаем actions и selectors для Users. Переносим метод регистрации в User Slice. Переносим метод создания пользователя в User Slice. Переадресация после входа. Переносим метод входа в User Slice. Изменяем ProtectedRoutes. Загружаем данные о пользователях. НОС. Загрузчики (Loaders). Удаляем Auth и User Context. Реализуем LogOut. Обрабатываем ошибки авторизации. Создаем Comments Slice. Заключение
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Промежуточная аттестация	1	Зачет (итоговый тест модуля)
Модуль 3. Backend (32 ч.)		
Теория	1	Тема 4.1. Введение
Текущий контроль	1	Тестирование
Теория	2	Тема 4.2. Основы Начало работы. Первая программа. Как работают модули. Чтение параметров консоли. Запись в файл. Чтение из файла. Вывод данных
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	1	Тема 4.3. Стандартные модули
Текущий контроль	1	Тестирование

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Теория	2	Тема 4.4. Web-сервер Создание сервера. HTTP-модуль. Автоматизация с Nodemon. Вывод HTML-страницы. Обработка POST-запроса. Работа с Express.js. Вывод списка элементов. EJS Асинхронное удаление элементов
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Практика	2	Тема 4.5. Дополнительные задания
Теория	2	Тема 4.6. Создаем Backend. Часть 1 Что будем делать в практике. Структура проекта. Настройка сервера. Настройка рабочего окружения. Добавляем middleware для приложения. Настраиваем MongoDB. Подключаем MongoDB. Заносим первоначальные данные. Создаем модели. Создаем структуру роутов. Получаем Профессии и Качества. Что такое Postman
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	2	Тема 4.7. Создаем Backend. Часть 2 Пишем логику для регистрации. Создаем сервис для работы с токенами. Валидация полей с express-validator. Создаем роут для логина. Пишем логику для обновляемого токена
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	1	Тема 4.8. Создаем Backend. Часть 3 Пишем логику для работы с пользователями. Создаем Middleware для авторизации. Создаем функционал для комментариев
Практика	1	Практическая работа
Текущий контроль	1	Тестирование

Вид занятий	Кол-во часов	Наименование модуля, темы и содержание
Теория	1	Тема 4.9. Интеграция Backend & Frontend Старт интеграции. Рефакторинг. Исправляем ошибку CORS. Исправляем ошибки при регистрации. Чиним комментарии. Завершаем интеграцию
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Теория	2	Тема 4.10. Deployment Что такое Deployment и зачем нужен Docker. Реализовываем Production mode для сервера. Настраиваем Docker. Запускаем приложение в Docker контейнере. Настраиваем VPS. Переносим проект на сервер с Git. Исправляем ошибку с MongoDB. Обновляем приложение на сервере и запускаем его
Практика	1	Практическая работа
Текущий контроль	1	Тестирование
Промежуточная аттестация (зачет) – 2 часа. Защита курсового проекта		
Всего часов	190	

7. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Формы аттестации и контроля

В процессе реализации программы используются следующие **виды контроля**: входной, текущий, промежуточный (по итогам отдельных модулей) и итоговый.

1) входной контроль осуществляется в форме ознакомительной беседы с обучающимися с целью введения их в учебный курс, знакомства с правилами организации образовательного процесса, правилами безопасной работы;

2) текущий контроль включает в себя тесты, выполнение практических работ (заданий);

3) промежуточный (по итогам отдельных модулей) в форме зачета посредством тестирования;

4) итоговый контроль осуществляется в форме итогового зачета.

Оценка качества освоения программы осуществляется преподавателем на основе промежуточного и итогового контроля знаний.

Промежуточный контроль знаний проводится с использованием тестовой методики, содержащей вопросы с вариантами ответов к разделам курса.

Оценка за выполнение тестового задания выставляется в виде двухбалльной системы «зачет/незачет». Для получения оценки «зачет» слушателю необходимо безошибочно ответить на 80% предложенных вопросов.

Итоговый контроль обучающихся осуществляется в форме зачета посредством защиты курсового проекта (разработка собственного SPA приложения с клиент-серверной архитектурой).

Слушатель допускается к итоговой аттестации после изучения модулей и тем программы в объеме, предусмотренном учебным планом.

Оценка по результатам промежуточной аттестации выставляется по двухбалльной системе: «зачтено/не зачтено».

Оценка «зачтено» ставится при выполнении требований, предъявляемых к данной форме контроля (соответствие разработки техническим требованиям; использование библиотеки React; Node.js, Express, MongoDB; реализация настройки Express (с помощью Express нужно добавить все роуты, необходимые приложению для работы по REST API); реализация всех видов запросов (GET, POST, PUT и/или PATCH, DELETE); использование промежуточных обработчиков (минимум 1 кастомный middleware в проекте); реализация модели mongoose (для всех сущностей, которые будут храниться в MongoDB, необходимо создать модель mongoose); авторизация и регистрация).

Для аттестации обучающихся на соответствие их персональных достижений требованиям соответствующей образовательной программы созданы фонды оценочных средств, включающие типовые задания, тесты и методы контроля, позволяющие оценить знания и умения.

8. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Обучение по программе осуществляется в форме теоретических и практических занятий.

Принципы отбора содержания программы

При отборе теоретического материала и установлении его последовательности соблюдаются следующие принципы:

- ☐ структурирование учебного материала с учетом объективно существующих связей между его темами;
- ☐ актуальность, практическая значимость учебного материала для обучающегося.

Основные формы и методы

Формы занятий делятся на следующие категории:

- ☐ микрогрупповые;
- ☐ индивидуальные.

В процессе реализации программы используются следующие методы обучения:

Методы обучения. Методическое обеспечение программы предусматривает наличие следующих видов методического материала: электронные учебники, видео-уроки, видеоролики, информационные материалы на сайте, посвященном данной программе.

В процессе обучения по данной программе используются следующие методы:

- 1) исследовательский, применяемый при самостоятельной работе обучающихся;
- 2) репродуктивный, используемый в процессе применения полученных в процессе обучения знаний.
- 3) наглядный, в процессе которого осуществляется демонстрация мультимедийных иллюстраций и рисунков, видеороликов и видео-уроков;
- 4) практический, при помощи которого обучающиеся выполняют творческие работы (задания), отвечают на вопросы по изученному материалу и проч.
- 5) объяснительно-иллюстративный (применение демонстрационного материал (изображений, видеороликов) для формирования знаний и образа действий);
- 6) стимулирующий (развитие познавательного интереса у обучающегося, эмоциональное стимулирование и т.д.).

Педагогические технологии и методики. В обучении применяются особые технологии, выбор которых будет зависеть от выбранной модели обучения индивидуально с каждым обучающимся. Личностно – ориентированные технологии ставят в центр всей образовательной системы личность обучаемого. Обеспечение комфортных, бесконфликтных условий ее развития, реализацию ее природных потенциалов. Именно на такие технологии опирается программа с индивидуальным форматом обучения.

1) Технология индивидуального образовательного маршрута

Данная технология имеет целью реализовать следующие права и возможности обучающегося:

- право на выбор или выявление индивидуального смысла и целей в обучении;
- право выбора индивидуального темпа обучения, форм и методов решения образовательных задач, способов контроля, рефлексии и самооценки своей деятельности;
- превышение (опережение или углубление) осваиваемого содержания учебного плана.

Основные элементы индивидуальной образовательной деятельности обучающегося – это смысл деятельности (зачем я это делаю); постановка личной цели (предвосхищающий результат); план деятельности; реализация плана; рефлексия (осознание собственной деятельности); оценка; корректировка или переопределение целей.

Условием достижения целей и задач личностно-ориентированного обучения является сохранение индивидуальных особенностей обучающегося, его уникальности и разноплановости. Для этого применяются следующие способы; индивидуальные задания; формулировка обучающимся открытых заданий, которые предполагают их выполнение индивидуально каждым обучающимся; предложение обучающемуся составить план занятия для себя, выбрать содержание своего задания для самостоятельной работы.

2) Технология сотрудничества

Главная идея обучения в сотрудничестве — педагог и обучающийся вместе проходят весь образовательный процесс, находятся на равных позициях, что помогают обучающемуся чувствовать себя более раскованно и быстрее адаптироваться к образовательному процессу. Такая технология предполагает общность цели и задач, индивидуальную ответственность и равные возможности успеха.

Учебно-методический комплекс программы состоит из трех компонентов:

1. учебные и методические материалы для педагогов и обучающихся;
2. система средств обучения;
3. система средств контроля результативности обучения.

Первый компонент включает в себя списки литературы и интернет-источники, необходимые для работы педагогов и обучающихся.

Второй компонент – система средств обучения.

Дидактические средства:

1. Иллюстративный материал к темам программы.
2. Видеотека.

Третий компонент – система средств контроля результативности реализации программы:

- диагностические и контрольные материалы.

Реализация программы осуществляется с применением электронного обучения и дистанционных образовательных технологий.

Обучающимся предоставляется доступ (удаленный доступ) к современным профессиональным базам данных и информационным справочным системам, электронной библиотеке, состав которых определяется настоящей программой.

При осуществлении дистанционного обучения слушателям выдаются логин и пароль для вхождения на образовательную платформу, с помощью которой необходимо будет реализовывать требования программы.

Кадровое обеспечение

Реализация программы обеспечивается педагогическими кадрами, имеющими среднее профессиональное или высшее образование (по направлению, соответствующему направлению программы, реализуемой организацией, осуществляющей образовательную

деятельность) и отвечающими квалификационным требованиям, указанным в квалификационных справочниках, и (или) профессиональным стандартам.

При заключении договора с педагогом дополнительного образования ИП Минин В.И. проводит собеседование в целях определения уровня знаний педагогического работника по теме образовательной программы, а также на определение уровня знаний и навыков работы с техническими устройствами, необходимыми для реализации образовательной программы. Каждые 3 (три) года в отношении педагогических работников ИП Минин В.И. проводит повторную проверку знаний и навыков работы с техническими устройствами и программами, необходимыми для обучения, в форме собеседования.

Все педагогические работники, задействованные в образовательном процессе, регулярно проходят повышение квалификации по теме программы и (или) по направлению педагогической деятельности, а также не имеют ограничений на занятия педагогической деятельностью, установленных законодательством Российской Федерации.

Соотношение объема занятий, проводимых путем непосредственного взаимодействия педагогического работника с обучающимися, подбирается персонально применительно к каждому обучающемуся в зависимости от уровня понимания обучающимся теоретического материала, выполнения практических заданий и успешности прохождения тестов. Педагог проводит проверку выполнения практических заданий и отвечает на вопросы обучающихся в течение всего периода обучения с 09.00 до 18.00 часов по московскому времени с понедельника по субботу включительно, контроль прохождения тестов автоматизирован на платформе ИП Минина В.И. Обучающиеся имеют возможность задавать вопросы и оставлять мнения на платформе в ходе прохождения программы. С учетом того, что адресатами программы являются люди, достигшие 18 лет (взрослые) дистанционное обучение и занятия в виде видеолекций с последующим закреплением пройденного материала при выполнении практических заданий представляются оптимально подобранным соотношением.

После освоения обучающимся образовательной программы выдается документ, удостоверяющий прохождение курса обучения, образец которого устанавливается образовательной организацией самостоятельно (Сертификат).

9. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Примерные вопросы теста для промежуточного контроля:

1. С помощью какого тега можно подключить JavaScript-файл к HTML-файлу?

- a) Src
- b) Script
- c) Connect

2. Какая функция в JavaScript выводит информацию в консоль?

- a) console.log
- b) log.console
- c) log

3. Какие существуют способы для объявления комментариев в JS?

- a) // и /* */
- b) /// и /* */

с) `///` и `/*` /

4. Сколько всего типов данных в JavaScript?

- a) 10
- b) 9
- c) 8

5. Какой тип данных является не примитивом?

- a) Object
- b) Number
- c) Symbol

6. Какие типы преобразований типов данных существуют в JavaScript?

- a) К булевому значению, к null, к строке
- b) К булевому значению, к объекту, к строке
- c) К булевому значению, к числу, к строке

7. Какие существуют популярные способы для конкатенации (объединения) строк?

- a) с помощью знака + и с помощью следующей конструкции ``${name}``.
- b) с помощью знака - и с помощью следующей конструкции ``${name}``.
- c) с помощью знака + и с помощью следующей конструкции `"${name}"`.

8. Как сокращенно можно записать данную строчку кода: `counter = counter + 1`?

- a) `counter += 1`
- b) `counter ==+ 1`
- c) `counter +== 1`

9. Чем отличается нестрогое сравнение `"=="` от строгого `"==="`?

- a) строгое `"==="` сравнивает по типам данных, нестрогое `"=="` - по значениям
- b) нестрогое `"=="` сравнивает по типам данных, строгое `"==="` - по значениям
- c) Никакой разницы между ними нет

10. Какими символами в JavaScript обозначается логическое И, ИЛИ и НЕ?

- a) И - `||`, ИЛИ - `&&`, НЕ - `!`
- b) И - `$$`, ИЛИ - `||`, НЕ - `!`
- c) И - `&&`, ИЛИ - `||`, НЕ - `!`

11. Какие 3 параметра передаются в цикл `for`?

- a) Начальное значение, условие, шаг
- b) Шаг, начальное значение, условие
- c) Условие, начальное значение, шаг

12. Какое ключевое слово в JavaScript используется для добавления breakpoint в определенную строчку кода?

- a) breakpoint;
- b) debugger;
- c) debug;

14. Какие типы функций существуют?

- a) function declaration, function expression, стрелочная функция
- b) function abbreviation, function expression, стрелочная функция
- c) function declaration, function mutation, стрелочная функция

15. Какие методы используются для добавления элементов в массив?

- a) addToEnd, addToStart
- b) push, unshift
- c) pop, shift

Тематика курсовых проектов для итогового контроля:

Тема 1. Блог

- Страница с перечнем всех статей и кнопкой перехода на отдельную статью
- Страница отдельной статьи
- Страница администратора для управления статьями
- Страница администратора для добавления новой статьи
- Страницы входа / регистрации в админ панель

Тема 2. Бронирование отелей

- Страница с доступными номерами
- Страница отдельного номера с возможностью забронировать
- Страница с забронированными номерами пользователя и возможностью удалить бронь
- Страница администратора со статусом всех номеров
- Страницы входа/регистрации

Тема 3. Интернет-магазин

- Поиск товара по его названию
- Категории товаров
- Сортировка товаров (по стоимости)
- Отдельная страница товара
- Корзина
 - Итоговая стоимость
 - Возможность удаления товара из корзины
 - Возможность изменения количества товаров

- Админ панель с возможностью добавления товаров на сайт (форма добавления / редактирования товара; таблица всех товаров)
- Страница регистрации и входа

Тема 4. Отслеживание времени

- Авторизация / регистрация
- Страница всех проектов
 - Возможность удалить проект
 - Возможность редактировать проект
 - Возможность добавить новый проект
- Страница создания / редактирования проекта
- Страница аналитики
 - Фильтры
 - Сортировка
 - Пагинация
 - Графики
- Страница настроек аккаунта
- Главная
 - Возможность засекать время для какого-то проекта

Тема 5. Учет расходов и доходов (приложение для учета расходов и доходов пользователя)

- Авторизация / регистрация
- Главная страница
 - Аналитика (графики)
 - Отображение счетов
 - Отображение категорий расходов / доходов
 - Возможность открыть страницу добавления / редактирования счета или категории
 - Возможность открыть страницу добавления операции
- Новая операция и редактирование операции
 - Выбор счета, с которого будет производиться оплата / пополнение
 - Выбор категории траты / дохода
 - Ввод суммы расхода / дохода
 - Ввод комментария для операции
- Новая категория и редактирование категории
 - Название категории
 - Тип категории (расход / доход)
 - Иконка на выбор пользователя (по желанию)
- Новый счет и редактирование счета
 - Название счета
 - Тип счета (вклад, дебетовая карта и тд)
 - Иконка на выбор пользователя (по желанию)
- История операций
 - Просмотр всех операций
 - Возможность открыть страницу добавления / изменения операции
 - Возможность удалить операцию
 - Фильтрация, пагинация
 - Фильтрация по датам, по счетам, по категориям и тд
 - Пагинация (чтобы не выводилось сразу очень большое количество операций на странице)
- Настройки

- Возможность сменить данные о пользователе

Тема 6. Блокнот заметок

- Страница со всеми заметками и возможностью добавления новой/удалением старых
- Страница отдельной заметки с возможностью редактирования и удаления
- Страницы входа / регистрации
- Страница с детальным отображением одной заметки
- В админке статистика по количеству заметок в системе

Тема 7. Прогнозы погоды

- Страница со строкой поиска города
- Страница с погодой запрашиваемого города, кнопкой добавления города в избранное и поисковой строкой
- Страница с избранными городами
- Страницы входа / регистрации
- Избранные города можно просматривать только после авторизации в админке

Тема 8. Трекер задач

- Страница со всеми задачами, разделёнными по статусу и возможность добавления новой задачи
- Страница добавления / редактирования отдельной задачи
- Страницы входа / регистрации
- В админ панели статистика по задачам (количество, статусы)

10. МАТЕРИАЛЬНО-ТЕХНИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

12.1. Материально-техническая база образовательной организации оснащена необходимым оборудованием для доступа в интернет по выделенному каналу.

12.2. Образовательная организация имеет необходимое серверное оборудование, обеспечивающее функционирование электронной информационно-образовательной среды, и высокоскоростной канал доступа к электронной информационно-образовательной среде.

12.3. Обучение проводится посредством электронной образовательной среды (платформы) <https://getcourse.ru/>.

Платформа позволяет:

12.3.1. размещать обучающие материалы и задания;

12.3.2. загружать обучающимся выполненные письменные, фото, видео задания, а также вопросы в адрес преподавателя;

12.3.3. проводить вебинары преподавателями, а также в процессе их проведения задавать вопросы в форме текстовых сообщений;

12.3.4. осуществлять контроль прогресса изучения учебных материалов, количество выполненных обучающимися заданий, а также проверять выполненные ими задания.

12.4. Для освоения образовательной программы обучающийся должен иметь доступ в сеть интернет, а также персональный компьютер или смартфон. Используемое для обучения программное обеспечение и техника обучающегося должны соответствовать следующим техническим требованиям:

12.4.1. для персонального компьютера: процессор с частотой работы от 1.5ГГц, Память ОЗУ объемом не менее 4 Гб, Жесткий диск объемом не менее 128 Гб, Монитор от 10 дюймов с разрешением от 1440*900 точек (пикселей), ОС Windows 7+ или Mac OS X от 10.7+, Браузер Google Chrome последней версии.

12.4.2. для смартфона: операционная система Android версии 5.0 и выше, а также ОС iOS версии 10.0 и выше. оперативная память от 1 гб и выше, экран от 720×1280 и выше, Браузер Google Chrome последней версии.

11. СПИСОК ЛИТЕРАТУРЫ

Методическая литература:

1. Симпсон Кайл. Асинхронная обработка и оптимизация. – СПб.: Питер, 2022. – 352 с.
2. Мартин Роберт. Идеальный программист. Как стать профессионалом разработки ПО. - СПб.: Питер, 2022. – 224 с.
3. Черный Б. Профессиональный TypeScript. Разработка масштабируемых JavaScript-приложений. – СПб.: Питер, 2022. - 352 с
4. Семакин И. Г., Рагулина М. И. Методика обучения информатике: Учебное пособие. – Лань, 2020. – 392 с.

Литература для обучающихся

1. Бастрон А.А. Современные подходы в изучении и применении Frontend разработки // Конкурс научно-исследовательских работ студентов Волгоградского государственного технического университета. – 2022. – С. 438-439. <https://www.elibrary.ru/item.asp?id=49233330>
2. Заика Д.В., Кирина В.С. HTMX: НОВЫЙ ВЗГЛЯД НА FRONTEND РАЗРАБОТКУ // Технологическое предпринимательство. – 2023. - № 1. – С. 229-232. <https://www.elibrary.ru/item.asp?id=68607789>.

Интернет-ресурсы:

Современный учебник JavaScript <https://learn.javascript.ru/>