



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ
Директор ИЭиАС
В.Р. Храмшин

26.01.2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ FRONT-END В WEB РАЗРАБОТКЕ

Направление подготовки (специальность)
09.03.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы
Логика и дизайн пользовательских интерфейсов

Уровень высшего образования - бакалавриат

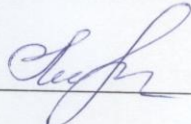
Форма обучения
очная

| | |
|---------------------|---|
| Институт/ факультет | Институт энергетики и автоматизированных систем |
| Кафедра | Вычислительной техники и программирования |
| Курс | 3 |
| Семестр | 5 |

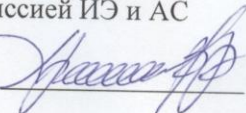
Магнитогорск
2022 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)


Рабочая программа рассмотрена и одобрена на заседании кафедры Вычислительной техники и программирования
19.01.2022 г. протокол № 4

Зав. кафедрой  О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭ и АС
26.01.2022 г. протокол № 5

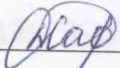
Председатель  В.Р. Храшин

Рабочая программа составлена:
доцент кафедры ВТиП,

 Е.А. Гарбар

Рецензент:

Начальник отдела технологических платформ ООО «Компас Плюс», канд. техн. наук

 Д.С. Сафонов

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2025 - 2026 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2026 - 2027 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины (модуля) «Программированное обеспечение Front-End в Web разработке» является овладение современными методами и средствами разработки интерактивных распределенных многопользовательских Web-приложений с применением современных технологий.

Для достижения поставленной цели в курсе «Программированное обеспечение Front-End в Web разработке» решаются задачи:

- изучение возможностей языка разметки HTML и языка оформления документов CSS для создания Web-страниц;
- изучение языка JavaScript для придания динамики поведению Web-приложения на клиентской стороне;
- изучение фреймворка Vue.js;
- приобретение навыков интеграции клиентской и серверной частей Web-приложений.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Программное обеспечение Front-End в Web разработке входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

- Программирование
- Средства программирования Web-приложений
- Психология визуального восприятия графической информации
- Шрифты и шрифтовые композиции
- Информатика

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

- Выполнение и защита выпускной квалификационной работы
- Управление контентом для Web-приложений
- Шаблонное программирование
- Программное обеспечение Back-End в Web разработке
- Программные решения для бизнеса
- Человеко-машинное взаимодействие
- Юзабилити-исследование Web-приложений

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Программное обеспечение Front-End в Web разработке» обучающийся должен обладать следующими компетенциями:

| Код индикатора | Индикатор достижения компетенции |
|----------------|---|
| ПК-1 | Способность к анализу проблемной ситуации, разработке требований к системе, постановке целей создания, разработке концепции и технического задания на создание приложения, представления концепции, технического задания на приложение и изменений в них заинтересованным лицам |
| ПК-1.1 | Анализирует требования к разработке приложений и базам данных |
| ПК-1.2 | Оценивает качество разработки технических спецификаций на разрабатываемое приложение |
| ПК-1.3 | Оценивает качество проекта на разработку приложения и баз данных |

| | |
|--|---|
| ПК-5 Способность к формализации и алгоритмизации поставленных задач, к написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями | |
| ПК-5.1 | Оценивает качество математической модели при формализации задачи предметной области |
| ПК-5.2 | Оценивает качество разработанных алгоритмов для последующего кодирования |
| ПК-5.3 | Оценивает выбор программных средств для программирования и манипулирования данными в соответствии установленными требованиями |

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 4 зачетных единиц 144 акад. часов, в том числе:

- контактная работа – 54,15 акад. часов;
- аудиторная – 51 акад. часов;
- внеаудиторная – 3,15 акад. часов;
- самостоятельная работа – 54,15 акад. часов;
- в форме практической подготовки – 0 акад. час;
- подготовка к экзамену – 35,7 акад. час

Форма аттестации - экзамен

| Раздел/ тема дисциплины | Семестр | Аудиторная контактная работа (в акад. часах) | | | Самостоятельная работа студента | Вид самостоятельной работы | Форма текущего контроля успеваемости и промежуточной аттестации | Код компетенции |
|--------------------------------|---------|--|-----------|-------------|---------------------------------|--|--|--|
| | | Лек. | лаб. зан. | практ. зан. | | | | |
| 1. Основы Javascript | | | | | | | | |
| 1.1 Структура кода | 5 | 2 | 4 | | 6 | 1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к лабораторной работе | Устный опрос (собеседование) | ПК-1.1, ПК-1.2, ПК-1.3, ПК-5.1, ПК-5.2, ПК-5.3 |
| 1.2 Переменные, типы данных | | 3 | 8 | | 15 | 1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к лабораторной работе. 4. Выполнение лабораторной работы | 1. Устный опрос (собеседование). 2. Проверка лабораторной работы. | ПК-1.1, ПК-1.2, ПК-1.3, ПК-5.1, ПК-5.2, ПК-5.3 |
| 1.3 Основные конструкции языка | | 2 | | | 2,3 | 1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. | Устный опрос (собеседование). | ПК-1.1, ПК-1.2, ПК-1.3, ПК-5.1, ПК-5.2, ПК-5.3 |
| Итого по разделу | | 7 | 12 | | 23,3 | | | |

| | | | | | | | | |
|--|---|----|----|--|-------|--|--|--|
| 2. Объектно-ориентированное программирование Javascript | | | | | | | | |
| 2.1 Циклы и функции | 5 | 2 | 4 | | 6 | 1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к лабораторной работе | Устный опрос (собеседование). | ПК-1.1, ПК-1.2, ПК-1.3, ПК-5.1, ПК-5.2, ПК-5.3 |
| 2.2 Классы | | 2 | 4 | | 4 | 1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. | 1. Устный опрос (собеседование). 2. Проверка лабораторной работы. | ПК-1.1, ПК-1.2, ПК-1.3, ПК-5.1, ПК-5.2, ПК-5.3 |
| 2.3 Свойства объектов, их конфигурация. Прототипы и наследование | | 2 | 4 | | 4 | 1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. | 1. Устный опрос (собеседование). 2. Выполнение лабораторной работы. | ПК-1.1, ПК-1.2, ПК-1.3, ПК-5.1, ПК-5.2, ПК-5.3 |
| Итого по разделу | | 6 | 12 | | 14 | | | |
| 3. Реактивный фреймворк Vue.js | | | | | | | | |
| 3.1 Основы работы с Vue.js | 5 | 2 | 4 | | 12 | 1. Выполнение лабораторной работы 2. Самостоятельное изучение учебной и научной литературы | Тестирование Проверка лабораторной работы | ПК-1.1, ПК-1.2, ПК-1.3, ПК-5.1, ПК-5.2, ПК-5.3 |
| 3.2 Шаблоны и компоненты Vue.js | | 2 | 6 | | 4,85 | 1. Выполнение лабораторной работы 2. Самостоятельное изучение учебной и научной литературы | Коллоквиум Проверка лабораторной работы | ПК-1.1, ПК-1.2, ПК-1.3, ПК-5.1, ПК-5.2, ПК-5.3 |
| Итого по разделу | | 4 | 10 | | 16,85 | | | |
| Итого за семестр | | 17 | 34 | | 54,15 | | экзамен | |
| Итого по дисциплине | | 17 | 34 | | 54,15 | | экзамен | |

5 Образовательные технологии

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающие прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

Формы учебных занятий с использованием технологий проблемного обучения:

Практическое занятие в форме практикума – организация учебной работы, направленная на решение комплексной учебно-познавательной задачи, требующей от студента применения как научно-теоретических знаний, так и практических навыков.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

а) Основная литература:

1. Заяц А.М. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js : учебное пособие / А.М. Заяц, Н.П. Васильев. — Санкт-Петербург : Лань, 2019. 120 с. ISBN 978-5-8114-3527-2. Текст : электронный // Лань : электронно-библиотечная система. URL: <https://e.lanbook.com/book/115516>.

2. Сафронов М. Разработка веб-приложений в Yii 2 : руководство / М. Сафронов. — Москва : ДМК Пресс, 2015. 392 с. ISBN 978-5-97060-252-2. Текст : электронный // Лань : электронно-библиотечная система. URL: <https://e.lanbook.com/book/82821>.

б) Дополнительная литература:

1. Джош Л. Современный PHP. Новые возможности и передовой опыт / Л. Джош ; перевод с английского Р.Н. Рагимов. — Москва : ДМК Пресс, 2016. 304 с. ISBN 978-5-97060-184-6. Текст : электронный // Лань : электронно-библиотечная система. URL: <https://e.lanbook.com/book/93269>.

2. Самарев Р.С. Создание простейших веб-приложений с помощью Ruby on Rails и AJAX : учебное пособие / Р.С. Самарев. — Москва : МГТУ им. Н.Э. Баумана, 2015. 50 с. ISBN 978-5-7038-4218-8. Текст : электронный // Лань : электронно-библиотечная система. URL: <https://e.lanbook.com/book/103539>.

3. Снетков В.М. Программирование на ASP.NET : учебное пособие / В.М. Снетков. — 2-е изд. — Москва : ИНТУИТ, 2016. 901 с. Текст : электронный // Лань : электронно-библиотечная система. URL: <https://e.lanbook.com/book/100402>.

4. Дронов В.А. Практика создания веб-сайтов на Python / В.А. Дронов. – Санкт-Петербург: Профессиональное программирование, БХВ-Петербург, 2019. 672 с.

ISBN 978-5-9775-4058-2.

в) Методические указания:

Торчинский В. Е. Разработка интерактивных WEB-страниц [Текст] : учебное пособие / В. Е. Торчинский, Л. Л. Демиденко, Ю. А. Демиденко ; МГТУ. - Магнитогорск, 2011. - 95 с.

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

| Наименование ПО | № договора | Срок действия лицензии |
|---|------------------------------|------------------------|
| NotePad++ | свободно распространяемое ПО | бессрочно |
| JetBrains IDEA Community Edition | свободно распространяемое ПО | бессрочно |
| MS Visual Studio Code | свободно распространяемое ПО | бессрочно |
| MariaDB | свободно распространяемое ПО | бессрочно |
| PostgreSQL | свободно распространяемое ПО | бессрочно |
| Git | свободно распространяемое ПО | бессрочно |
| Браузер Yandex | свободно распространяемое ПО | бессрочно |
| Браузер Mozilla Firefox | свободно распространяемое ПО | бессрочно |
| SCO OpenServer | свободно распространяемое ПО | бессрочно |
| MS Visual Studio 2017 Community Edition | свободно распространяемое ПО | бессрочно |
| GIMP | свободно распространяемое ПО | бессрочно |

Профессиональные базы данных и информационные справочные системы

| Название курса | Ссылка |
|--|--|
| Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности» | URL: http://www1.fips.ru/ |
| Информационная система - Единое окно доступа к информационным ресурсам | URL: http://window.edu.ru/ |

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

1. Лекционная аудитория ауд. 282. Мультимедийные средства хранения, передачи и представления информации.

2. Компьютерные классы Центра информационных технологий ФГБОУ ВО «МГТУ». Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники.

3. Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки. Все классы УИТ и АСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.

4. Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Ауд. 282 и классы УИТ и АСУ.

5. Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации. Классы УИТ и АСУ.

6. Помещения для хранения и профилактического обслуживания учебного оборудования Центра информационных технологий – ауд. 372.

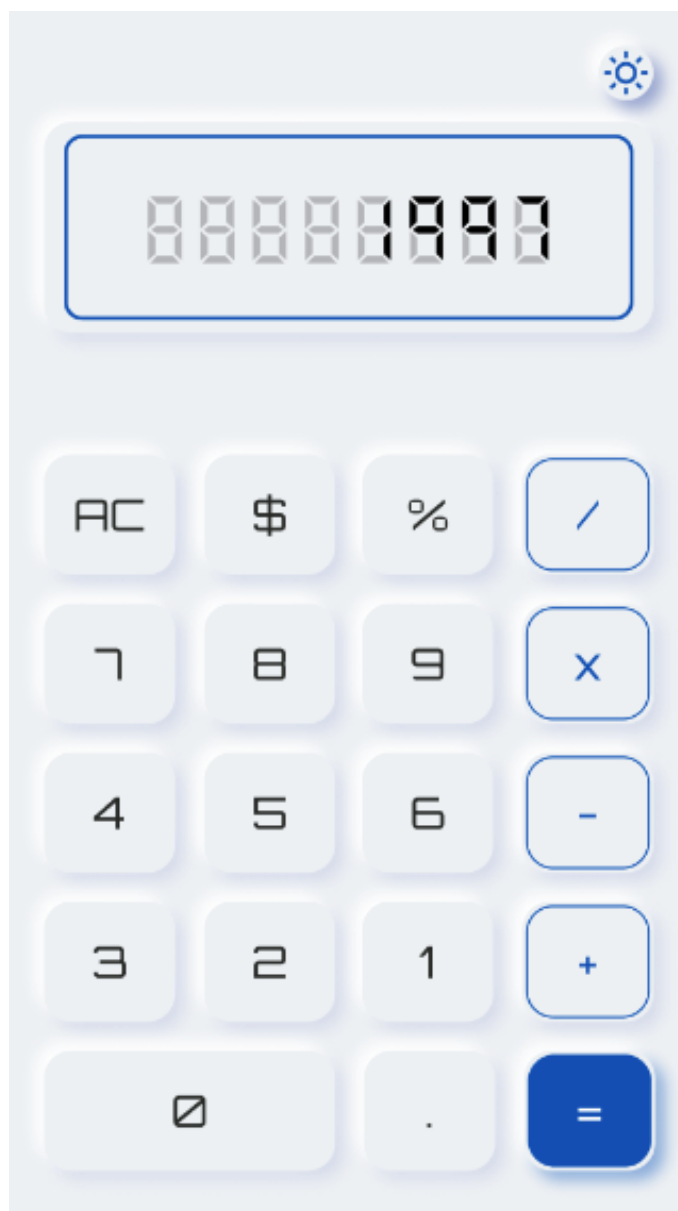
Приложение 1

Учебно-методическое обеспечение самостоятельной работы обучающихся

В течение семестра каждый студент выполняет лабораторные работы.

Лабораторная работа №1.

1. JPG файл с макетом.
2. Текстовое описание задания и требования.



Описание задания и требования:


1. Необходимо создать верстку макета, при создании можно использовать готовые UI библиотеки, например bootstrap.
2. Реализовать логику работы калькулятора с помощью Javascript без использования фреймворков
3. Реализовать логику работы значка «лампа» в правом верхнем углу так, чтобы при нажатии задний фон экрана менялся на светло-желтый
4. Реализовать работу значка \$ так, чтобы при нажатии число указанное на экране калькулятора переводилось в долларový эквивалент
5. Опционально. Получать актуальный курс доллара из любого доступного API,

например <https://www.cbr-xml-daily.ru/>

Лабораторная работа №2.

1. JPG файл с макетом.
2. Текстовое описание задания и требования.

Информация о студентах

| | | | | |
|---------------|---|--------------|---|--|
| ФИЛЬТР | | | |  Добавить студента |
| Фамилия | <input type="text" value="Введите фамилию"/> | Группа | <input type="text" value="Выберите группу"/> | |
| Имя | <input type="text" value="Введите имя"/> | Средний бал | От <input type="text"/> До <input type="text"/> | |
| Отчество | <input type="text" value="Введите отчество"/> | Год обучения | <input type="text" value="Выберите год"/> | |

| Фамилия ▲ | Имя | Отчество | Группа | Средний бал | Год обучения | ⚙ |
|-----------|-------|-----------|----------|-------------|--------------|-----|
| Иванов | Иван | Иванович | АВа-19-1 | 4,5 | 3 | ✎ ⚙ |
| Петров | Петр | Петрович | АВа-19-2 | 4,2 | 4 | ✎ ⚙ |
| Сидоров | Сидор | Сидорович | АВа-19-1 | 3,1 | 2 | ✎ ⚙ |
| Иванов | Иван | Иванович | АВа-19-1 | 4,5 | 3 | ✎ ⚙ |
| Петров | Петр | Петрович | АВа-19-2 | 4,2 | 4 | ✎ ⚙ |
| Сидоров | Сидор | Сидорович | АВа-19-1 | 3,1 | 2 | ✎ ⚙ |
| Иванов | Иван | Иванович | АВа-19-1 | 4,5 | 3 | ✎ ⚙ |
| Петров | Петр | Петрович | АВа-19-2 | 4,2 | 4 | ✎ ⚙ |
| Сидоров | Сидор | Сидорович | АВа-19-1 | 3,1 | 2 | ✎ ⚙ |
| Иванов | Иван | Иванович | АВа-19-1 | 4,5 | 3 | ✎ ⚙ |

<< 1 2 3 >>

10 ▼

Описание задания и требования:

Задание выполняется с помощью фронтенд на Javascript или Typescript без использования JS-фреймворков

1. Сделать верстку макета
2. Реализовать логику работы фильтра, после изменения данных в фильтре, таблица должна перестраиваться с учетом этих данных
3. Реализовать сортировку данных при нажатии на заголовок столбца таблицы
4. Сделать возможность настройки отображаемых столбцов при клике на значек «гайка» в правом верхнем углу таблицы
5. Реализовать возможность удаления и редактирования записи таблицы при нажатии на иконки справа от каждой записи
6. Реализовать возможность добавления записи при нажатии на иконку «+»
7. Опционально. Хранить данные в облачной базе данных, имеющую готовое API напрямую или через бекенд часть, например, Google Cloud databases

Лабораторная работа №2.

Задание аналогично лабораторной работе №2, но реализовать с помощью реактивного фреймворка Vue.js (также можно использовать Angular).

Приложение 2

Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|---|---|---|
| ПК-5: Способность к формализации и алгоритмизации поставленных задач, к написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями | | |
| ПК-5.1 | Оценивает качество математической модели при формализации задачи предметной области | <p><i>Перечень теоретических вопросов</i></p> <p>1. История компьютерных сетей. Основные вехи и ступени развития. Появление глобальных сетей</p> <p>2. Многоуровневая архитектура сетевого взаимодействия. Модель ISO OSI</p> |
| ПК-5.2 | Оценивает качество разработанных алгоритмов для последующего кодирования | <p>3. Основы архитектуры локальных сетей, технология ethernet.</p> <p>4. Многоуровневая структура стека протоколов TCP/IP. Общая характеристика стека.</p> <p>5. Протокол FTP. Назначение, основные команды.</p> |
| ПК-5.3 | Оценивает выбор программных средств для программирования и манипулирования данными в соответствии установленными требованиями | <p>Протокол HTTP. Назначение. Структура. Основные методы. Структура URL.</p> <p><i>Перечень теоретических вопросов</i></p> <p>1. Web-приложения – определение, основные элементы, достоинства и недостатки использования.</p> <p>2. Адресация ресурсов в глобальных сетях. URI, URL, URN адреса. Абсолютная и относительная адресация в Web-приложениях</p> <p>3. Протокол HTTP: порядок взаимодействия, формат запроса и ответа.</p> <p>4. Язык гипертекстовой разметки страниц HTML: назначение, история развития, стандарты языка.</p> <p>5. Язык гипертекстовой разметки страниц HTML: общая структура документа, теги и их атрибуты.</p> <p>6. Теги заголовка HTML-документа: назначение, виды, примеры использования.</p> <p>7. Блочные и строчные html-элементы: назначение, примеры использования, отличия,</p> <p>8. HTML5: обзор возможностей, достоинства в сравнении с предыдущими версиями.</p> <p>9. Оформление HTML-документов с использованием каскадных таблиц стилей. Способы записи стилей для элементов.</p> <p>10. CSS. Блоковая модель элемента.</p> <p>11. CSS. Основной поток элементов и способы извлечения элемента из потока (всплывающие элементы, позиционирование).</p> <p>12. CSS. Приоритеты стилей в объявлении, расчет специфичности.</p> <p>13. CSS3. Новые возможности оформления</p> |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|---|
| | | <p>документов.</p> <p>14. Общие подходы к дизайну сайта. Разработка макета страницы. Блочный и табличный макеты.</p> <p>15. Адаптивная верстка сайта: базовые принципы и инструментарий.</p> <p>16. Валидность HTML-документов</p> <p>17. Front-end Web-приложения: назначение, ограничения. Язык JavaScript: основы синтаксиса.</p> <p>18. Объектная модель HTML страницы.</p> <p>22. Web-сервер: назначение, порядок обработки клиентских запросов, способы конфигурирования.</p> <p>23. Виды серверных скриптов, отличия в принципах их функционирования</p> <p>24. Динамическое формирование html-страниц на стороне сервера: инструменты, преимущества, примеры реализации.</p> <p>25. Сохранение состояния Web-приложения: механизм cookie.</p> <p>26. Сохранение состояния Web-приложения: сессии.</p> <p>27. Средства обработки запроса клиента на стороне сервера. Средства разбора параметров запроса.</p> <p>28. Организация загрузки файлов на сервер.</p> <p>29. Взаимодействие серверных скриптов с базами данных. Обзор расширений для работы с базами данных.</p> <p>30. Подготовленные запросы к базам данных назначение, средства реализации, преимущества использования.</p> <p>31. Асинхронная передача данных в Web-приложениях. Технология AJAX. Объект XMLHttpRequest.</p> <p>32. Синхронные и асинхронные AJAX-запросы. События асинхронного обмена данными.</p> <p>33. Формат данных JSON: назначение, примеры использования.</p> <p>34. XML, его роль в современных Web-приложениях. XML DOM. Схема и пространство имен XML-документа.</p> <p>35. Построение серверной части Web-приложения с использованием шаблона MVC.</p> <p>36. Web-сервисы: назначение, принципы функционирования, технологии реализации (SOA, SOFEA).</p> <p>37. Разработка RESTful Web-приложений.</p> <p>38. Безопасность работы Web-приложений: обзор угроз и методов их предотвращения.</p> <p><i>Практические задания</i></p> <p>1. Создать верстку макета калькулятора, при создании можно использовать готовые UI библиотеки, например bootstrap.</p> |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|---|--|---|
| | | 2. Реализовать логику работы калькулятора с помощью Javascript без использования фреймворков 3. Реализовать логику работы значка «лампа» в правом верхнем углу так, чтобы при нажатии задний фон экрана менялся на светло-желтый 4. Реализовать работу значка \$ так, чтобы при нажатии число указанное на экране калькулятора переводилось в долларовой эквивалент 5. Опционально. Получать актуальный курс доллара из любого доступного API, например https://www.cbr-xml-daily.ru/ <i>Задания на решение задач из профессиональной области, комплексные задания</i> Разработать интерактивный калькулятор |
| ПК-1: Способность к анализу проблемной ситуации, разработке требований к системе, постановке целей создания, разработке концепции и технического задания на создание Web-приложения, представления концепции, технического задания на Web-приложение и изменений в них заинтересованным лицам | | |
| ПК-1.1 | Анализирует требования к разработке Web-приложений и базам данных | <i>Перечень теоретических вопросов</i> 1. В чем разница между null и undefined? 2. Для чего используется оператор "&&"? 3. Для чего используется оператор " "? 4. Является ли использование унарного плюса (оператор "+") самым быстрым способом преобразования строки в число? |
| ПК-1.2 | Оценивает качество разработки технических спецификаций на Web-приложения | 5. Что такое DOM? 6. Что такое распространение события (Event Propagation)? |
| ПК-1.3 | Оценивает качество проекта на разработку Web-приложения и баз данных | 7. Что такое всплытие события (Event Bubbling)? 8. Что такое погружение события (Event Capturing)? 9. В чем разница между методами event.preventDefault() и event.stopPropagation()? 10. Как узнать об использовании метода event.preventDefault()? 11. Почему obj.someprop.x приводит к ошибке? 12. Что такое цель события или целевой элемент (event.target)? 13. Что такое текущая цель события (event.currentTarget)? 14. В чем разница между операторами "==" и "==="? 15. Почему результатом сравнения двух похожих объектов является false? 16. Для чего используется оператор "!"? 17. Как записать несколько выражений в одну строку? 18. Что такое поднятие (Hoisting)? 19. Что такое область видимости (Scope)? 20. Что такое замыкание (Closures)? 21. Какие значения в JS являются ложными? 22. Как проверить, является ли значение ложным? |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|---|
| | | <p>23. Для чего используется директива «use strict»?</p> <p>24. Какое значение имеет this?</p> <p>25. Что такое прототип объекта?</p> <p>26. Что такое ПФЕ?</p> <p>27. Для чего используется метод Function.prototype.apply?</p> <p>28. Для чего используется метод Function.prototype.call?</p> <p>29. В чем разница между методами call и apply?</p> <p>30. Для чего используется метод Function.prototype.bind?</p> <p>31. Что такое функциональное программирование и какие особенности JS позволяют говорить о нем как о функциональном языке программирования?</p> <p>32. Что такое функции высшего порядка (Higher Order Functions)?</p> <p>33. Почему функции в JS называют объектами первого класса (First-class Objects)?</p> <p>34. Как бы Вы реализовали метод Array.prototype.map?</p> <p>35. Как бы Вы реализовали метод Array.prototype.filter?</p> <p>36. Как бы Вы реализовали метод Array.prototype.reduce?</p> <p>37. Что такое объект arguments?</p> <p>38. Как создать объект, не имеющий прототипа?</p> <p>39. Почему в представленном коде переменная b становится глобальной при вызове функции?</p> <p>40. Что такое ECMAScript?</p> <p>41. Что нового привнес в JS стандарт ES6 или ECMAScript2015?</p> <p>42. В чем разница между ключевыми словами «var», «let» и «const»?</p> <p>43. Что такое стрелочные функции (Arrow Functions)?</p> <p>44. Что такое классы (Classes)?</p> <p>45. Что такое шаблонные литералы (Template Literals)?</p> <p>46. Что такое деструктуризация объекта (Object Destructuring)?</p> <p>47. Что такое модули (Modules)?</p> <p>48. Что такое объект Set?</p> <p>49. Что такое функция обратного вызова (Callback Function)?</p> <p>50. Что такое промисы (Promises)?</p> <p>51. Что такое async/await?</p> <p>52. В чем разница между spread-оператором и rest-оператором?</p> <p>53. Что такое параметры по умолчанию (Default Parameters)?</p> <p>54. Что такое объектная обертка (Wrapper Objects)?</p> |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|--|
| | | <p>55. В чем разница между явным и неявным преобразованием или приведением к типу (Implicit and Explicit Coercion)?</p> <p>56. Что такое NaN? Как проверить, является ли значение NaN?</p> <p>57. Как проверить, является ли значение массивом?</p> <p>58. Как проверить, что число является четным, без использования деления по модулю или деления с остатком (оператора "%")?</p> <p>59. Как определить наличие свойства в объекте?</p> <p>60. Что такое AJAX?</p> <p>61. Как в JS создать объект?</p> <p>62. В чем разница между методами Object.freeze и Object.seal?</p> <p>63. В чем разница между оператором «in» и методом hasOwnProperty?</p> <p>64. Какие приемы работы с асинхронным кодом в JS Вы знаете?</p> <p>65. В чем разница между обычной функцией и функциональным выражением?</p> <p>66. Как в JS вызвать функцию?</p> <p>67. Что такое запоминание или мемоизация (Memoization)?</p> <p>68. Как бы Вы реализовали вспомогательную функцию запоминания?</p> <p>69. Почему typeof null возвращает object? Как проверить, является ли значение null?</p> <p>70. Для чего используется ключевое слово «new»?</p> <p><i>Практические задания</i></p> <p>Настройка web-сервера Apache 2.4 на платформе ОС Windows</p> <p><i>Практические задания</i></p> <ol style="list-style-type: none"> 1. Сделать верстку макета 2. Реализовать логику работы фильтра, после изменения данных в фильтре, таблица должна перестраиваться с учетом этих данных 3. Реализовать сортировку данных при нажатии на заголовок столбца таблицы 4. Сделать возможность настройки отображаемых столбцов при клике на значек «гайка» в правом верхнем углу таблицы 5. Реализовать возможность удаления и редактирования записи таблицы при нажатии на иконки справа от каждой записи 6. Реализовать возможность добавления записи при нажатии на иконку «+» 7. Опционально. Хранить данные в облачной базе данных, имеющую готовое API напрямую или через бекенд часть, например, Google Cloud databases |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|----------------------------------|--|
| | | <p><i>Задания на решение задач из профессиональной области, комплексные задания</i></p> <p>Разработать web-приложение «Студенты» позволяющий управлять списком студентов, реализующий пользовательский интерфейс для CRUD операций</p> |

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Экзамен по дисциплине «Программное обеспечение Front-End в Web разработке» проводится по результатам отчетности на практических занятиях с опросом в устной форме по этапам выполнения и активного выступления в беседе-обсуждении на лекционных занятиях.

Показатели и критерии оценивания экзамена:

- на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.
- на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
- на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
- на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.
- на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.