ЧОУ ВО «Тольяттинская академия управления»

Кафедра прикладной информатики и высшей математики

Б1.В.ДВ.02.01

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА)

Учебная дисциплина	Fullstack-разработка web-приложений
По направлению подготовки	*09.03.03*
	«Прикладная информатика»
Профиль (программа бакалавриата)	«Разработка и управление web-контентом»
Форма обучения	*Очная*

Оценочные материалы (средства) дисциплины рассмотрены (актуализированы) и утверждены на заседании кафедры прикладной информатики и высшей математики

Протокол заседания № 10 от «25» мая 2026 г.

Заведующий кафедрой Стрекалова Наталья Борисовна

2023

ОГЛАВЛЕНИЕ

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ....3
2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ОБЕСПЕЧИВАЮЩИХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ....10
3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ....32
4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНОВАНИЯ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ....38
5. ДИАГНОСТИЧЕСКИЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ УРОВЕНЬ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ....53
6. ЛИСТ СОГЛАСОВАНИЯ....54

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Оценочные материалы (средства) сформированы в соответствии с ФГОС ВО - бакалавриат по направлению подготовки 09.03.03 «Прикладная информатика», утвержденный приказом Министерства образования и науки Российской Федерации от 19.09.2017 № 922 (с изменениями и дополнениями от 26.11.2020, 08.02.2021). В соответствии с матрицей компетенций основной профессиональной образовательной программы по направлению подготовки 09.03.03 «Прикладная информатика» (уровень бакалавриата) в процессе обучения по дисциплине «Fullstack-разработка web-приложений» происходит формирование закрепленных за дисциплиной компетенций обучающихся. Оценка сформированности компетенций на каждом этапе обучения происходит через оценку планируемых результатов обучения по дисциплине (знаний, умений, навыков).

Шифр и название компетенции	**Индикаторы компетенции**	**Планируемые результаты обучения по дисциплине**	Этапы формирования компетенции (семестры и темы)	Оценочные средства
ПК-1. Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку web-приложения	ПК-1.1. Проводит обследование организаций, выявляет информационные потребности заказчика, используя цифровые инструменты и облачные решения для сбора данных	**Знать:** - методику проведения обследования организации и выявления информационных потребностей пользователей; - виды требований, предъявляемых к web-приложения; - виды информации, необходимой для формализации требований заказчика для разработки базы данных и/или web-приложения, методы сбора и анализа детальной информации; **Уметь:** - проводить предпроектное и информационное обследование организации; - выявлять информационные потребности пользователей к web-приложениям; - выявлять информационные потребности предприятия в СУБД, **Владеть:**	**3 семестр** Тема 1. Реляционная модель баз данных Тема 2. Введение в JavaScript	- устный опрос по темам 1, 2 проверка выполнения заданий по теме 1, 2 - ответ на теоретический вопрос на экзамене

		- навыками сбора и анализа информации о предприятии; - навыками подготовки отчета о проведенном предпроектном обследовании организации;		
	ПК-1.2. Формализует требования заказчика к разработке web-приложения	**Знать:** - структуру и порядок построения технического задания на разработку web-приложения **Уметь:** - формировать техническое задание к разрабатываемому web- приложению; **Владеть:** - навыками документирования требований заказчика; - навыками составления технического задания на разработку web-приложения и проектов.	**4 семестр** Тема 10. Технологии разработки программных продуктов. Основные определения и подходы Тема 11. Создание сценариев на PHP. Типы данных, переменные, операторы	-устный опрос по теме 10 - выполнение практического задания по темам 10-11 - ответ на теоретический вопрос на экзамене
ПК-2. Способен проектировать web-приложения	ПК-2.1. Разрабатывает архитектуру web-приложения с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы	**Знать:** - процесс и средства разработки web-приложений; - факторы сложности разработки web-приложений, технологии, подходы и принципы к исследованию и созданию информационных систем, эргономические требования к ним **Уметь:** - определять факторы сложности разработки программных систем, выбирать адекватные технологии для их проектирования **Владеть:** - навыками разработки архитектуру web-приложения с учетом	**3 семестр** Тема 1. Реляционная модель баз данных Тема 2. Введение в JavaScript Тема 7.Свойства объекта, их конфигурация. Прототипы, наследование, классы. **4 семестр** Тема 10. Технологии разработки программных продуктов. Основные определения и подходы Тема 11. Создание сценариев на PHP. Типы данных, переменные,	- устный опрос по темам 1-2,7, 10-11 - выполнение практического задания по темам 1-2,7, 10-11 - ответ на теоретический вопрос на экзамене

		внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы	операторы	
	ПК-2.2. Проектирует базы данных, в том числе используя специализированные цифровые платформы	**Знать:** - содержание этапов разработки и принципы организации проектирования web-приложений, методики сравнения проектных решений; - информационную модель данных и ее структуру, логические модели баз данных и их типы, формы нормализации данных в базе; **Уметь:** - разрабатывать концептуальную, логическую и физическую модели web-приложений; - определять необходимый инструментарий по уровню обеспечения информационных систем при разработке web-приложений; **Владеть:** - навыками проектирования концептуальной модели БД и ее модификации; - навыками разработки модели web-приложения на основе проектного решения; - навыками подготовки и демонстрации прототипа решения;	**3 семестр** Тема 1. Реляционная модель баз данных **4 семестр** Тема 10. Технологии разработки программных продуктов. Основные определения и подходы Тема 11. Создание сценариев на PHP. Типы данных, переменные, операторы Тема 19. Сценарии сайта	- устный опрос по темам 1,10-11 - выполнение практического задания по темам 1, 10-11,19 - ответ на теоретический вопрос на экзамене

	ПК-2.3. Проектирует пользовательский интерфейс с учетом тенденций развития сквозных цифровых технологий и с помощью цифровых инструментов	**Знать:** - принципы разработки пользовательского интерфейса web-приложения; **Уметь:** - проектировать пользовательский интерфейс; **Владеть:** - навыками проектирования и разработки пользовательского интерфейса web-приложения	**4 семестр** Тема 11. Создание сценариев на PHP. Типы данных, переменные, операторы Тема 19. Сценарии сайта	- выполнение практического задания по темам 11,19 - ответ на теоретический вопрос на экзамене
	ПК-2.4. Проводит анализ современных программно-технических средств, специализированных цифровых платформ и облачных решений, сквозных цифровых технологий и отраслевых решений на их основе с целью выбора средств разработки web-приложений	**Знать:** - методы анализа рынка программно-технических средств, информационных продуктов и услуг; - основные источники профессионально-ориентированной информации, предназначенной для проведения анализа рынка программно-технических средств; **Уметь:** - составлять аналитические обзоры рынка программно-технических средств, информационных продуктов и услуг; - проводить сравнительный анализ программно-технических средств и информационных продуктов в соответствии с полученным заданием; **Владеть:** - навыками работы с программно-технологическими средствами (поисковыми машинами, справочно-правовыми и	**3 семестр** Тема 2. Введение в JavaScript Тема 4. Объекты: основы **4 семестр** Тема 10. Технологии разработки программных продуктов. Основные определения и подходы Тема 11. Создание сценариев на PHP. Типы данных, переменные, операторы	- устный опрос по темам 2,4,10 - выполнение практического задания по темам 2,4,10 - ответ на теоретический вопрос на экзамене

		экспертными системами, открытыми информационными ресурсами и т.п.) для выполнения задач исследовательского и аналитического характера;		
ПК-3. Способен разрабатывать web-приложения	ПК-3.1. Осуществляет разработку программного кода и компонентов web-приложения, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем	**Знать:** - язык определения и манипулирования данными; - основные этапы построения web-приложений, понятие кроссбраузерности, кроссплатформености и адаптивной верстки; - термины, синтаксис, управляющие конструкции и типы данных скриптового языка общего назначения PHP; - основы JavaScript; - инструменты и принципы разработки web-приложения; **Уметь:** - обеспечить передачу информации между приложением и веб-сервером; - осуществлять подключение к серверу и выбор базы данных; - разрабатывать динамические приложения на языке JavaScript и PHP для решения прикладных задач; **Владеть:** - навыками создания приложения с помощью языков программирования (HTML, JavaScript, PHP, CSS); - навыками создания сценариев с помощью операций и управляющих	**3 семестр** Тема 3. Качество кода Тема 4. Объекты: основы Тема 5. Типы данных Тема 6. Продвинутая работа с функциями Тема 7.Свойства объекта, их конфигурация. Прототипы наследование, классы Тема 8. Промисы, async/await **4 семестр** Тема 9. Генераторы, продвинутая итерация Тема 11. Создание сценариев на PHP. Типы данных, переменные, операторы Тема 12. Операции и управляющие конструкции Тема 13. Функции и повторное использование кода Тема 14. Массивы Тема 15. Передача данных через HTML-формы Тема 16. Работа с файлами Тема 17. Строковые функции и	- устный опрос по темам 4-5,7-9,17,22 - тестирование по теме 5 - выполнение практического задания по темам 3-8,11,19-20 - работа в малых группах по теме 12-16,18,21 - ответ на теоретический вопрос на экзамене

		конструкций PHP; - навыками работы с различными формами в PHP; - навыками работы с сессиями и cookie в PHP - навыками осуществления передачи данных через HTML-формы	регулярные выражения Тема 18. Cookies и управление сессиями Тема 19. Сценарии сайта Тема 20. Расширение mysqli для работы с базами данных Тема 21. XML и PHP Тема 22. Новостная лента RSS. Адаптивная верстка	
	ПК-3.2. Обеспечивает интеграцию программных модулей и компонентов программного обеспечения, в том числе используя отраслевые облачные решения и цифровые платформы	**Знать:** - назначение тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования **Уметь:** - проводить процедуру тестирования баз данных и web-приложений; **Владеть:** - навыками протоколирования результатов тестирования	**3 семестр** Тема 3. Качество кода Тема 5. Типы данных Тема 6. Продвинутая работа с функциями Тема 7.Свойства объекта, их конфигурация. Прототипы наследование, классы Тема 8. Промисы, async/await **4 семестр** Тема 9. Генераторы, продвинутая итерация Тема 12. Операции и управляющие конструкции Тема 13. Функции и повторное использование кода Тема 14. Массивы Тема 15. Передача данных через HTML-формы Тема 16. Работа с файлами Тема 17. Строковые	- устный опрос по темам 5,7-9,17 - выполнение практического задания по темам 3,5-8 - тестирование по теме 5 - работа в малых группах по теме 12-16,18,21 - ответ на теоретический вопрос на экзамене

			функции и регулярные выражения Тема 18. Cookies и управление сессиями Тема 20. Расширение mysqli для работы с базами данных Тема 21. XML и PHP	
ПК-5. Способен обеспечивать качество работы web-приложения	ПК-5.2. Проводит юзабилити-аудит web-приложения	**Знать:** - назначение юзабилити-аудит web-приложения **Уметь:** - проводить юзабилити-аудит web-приложения; **Владеть:** - навыками протоколирования результатов тестирования	**4 семестр** Тема 15. Передача данных через HTML-формы Тема 22. Новостная лента RSS. Адаптивная верстка	- устный опрос по теме 22 - работа в малых группах по теме 15 - ответ на теоретический вопрос на экзамене
	ПК-5.3. Осуществляет работы по интеграционному тестированию web-приложения с внешним цифровым окружением	**Знать:** - назначение тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования **Уметь:** - проводить процедуру тестирования баз данных и web-приложений; **Владеть:** - навыками протоколирования результатов тестирования	**3 семестр** Тема 3. Качество кода **4 семестр** Тема 15. Передача данных через HTML-формы Тема 21. XML и PHP	- выполнение практического задания по теме 3 - работа в малых группах по темам 15,21 - ответ на теоретический вопрос на экзамене

2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ОБЕСПЕЧИВАЮЩИХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Устные опросы

Примерные вопросы для проведения устного опроса по теме «Реляционная модель баз данных»

1. Области применения баз данных.
2. Виды баз данных.
3. Информационная модель данных и ее структура.
4. Основные понятия реляционных баз данных: поля, записи, индексы, ключи, таблицы, отношения.
5. Уровни модели данных
6. Модель данных в нотации Чена
7. Модель данных в нотации IDEF
8. Декомпозиция отношений
9. Первая и вторая нормальная форма
10. Третья и четвертая нормальная форма
11. Инструментарий для проектирования и документирования баз данных

Примерные вопросы для проведения устного опроса по теме «Введение в JavaScript»

1. Справочники и спецификации. Редакторы кода. Консоль разработчика
2. Структура кода.
3. Строгий режим — "use strict".
4. Преобразование типов.
5. Конструкция "switch".
6. Функции.
7. Function Expression.
8. Стрелочные функции, основы.
9. Особенности JavaScript

Примерные вопросы для проведения устного опроса по теме «Объекты: основы»

1. Объекты.
2. Копирование объектов и ссылки.
3. Методы объекта, "this".
4. Конструкторы, создание объектов через "new".
5. Опциональная цепочка '?.'. Тип данных Symbol.
6. Преобразование объектов в примитивы

Примерные вопросы для проведения устного опроса по теме «Типы данных»

1. Методы примитивов.
2. Числа. Строки.
3. Массивы. Методы массивов.
4. Перебираемые объекты.
5. Map и Set. WeakMap и WeakSet. Object.keys, values, entries.
6. Деструктурирующее присваивание.
7. Дата и время.
8. Формат JSON, метод toJSON

Примерные вопросы для проведения устного опроса по теме «Свойства объекта, их конфигурация. Прототипы, наследование, классы»

1. Флаги и дескрипторы свойств.
2. Свойства - геттеры и сеттеры.
3. Прототипное наследование.
4. Встроенные прототипы.
5. Класс: базовый синтаксис.
6. Наследование классов.
7. Статические свойства и методы.
8. Приватные и защищённые методы и свойства.
9. Расширение встроенных классов.
10. Пользовательские ошибки, расширение Error

Примерные вопросы для проведения устного опроса по теме «Промисы, async/await»

1. Введение: колбэки. Промисы.
2. Цепочка промисов.
3. Промисы: обработка ошибок.
4. Promise API.
5. Промисификация.
6. Микрозадачи.
7. Async/await

Примерные вопросы для проведения устного опроса

по теме «Генераторы, продвинутая итерация»

1. Генераторы.
2. Асинхронные итераторы и генераторы.
3. Модули. Модули, введение.
4. Экспорт и импорт.
5. Динамические импорты.

Примерные вопросы для проведения устного опроса
по теме «Технологии разработки программных продуктов. Основные определения и подходы»

1. Особенности создания программного продукта.
2. Понятие «Локального сервера».
3. Разновидности программных средств для разработки ПО.
4. ПО «Open Server» (Apache, PHP, MySQL, Perl).
5. Программная оболочка для разработки сайтов на локальной машине.
6. Оценка стоимости ошибок.

Примерные вопросы для проведения устного опроса
по теме «Строковые функции и регулярные выражения»

1. Строки в PHP.
2. Строковые операции.
3. Регулярные выражения.
4. Динамическое включение файлов.

Примерные вопросы для проведения устного опроса
по теме «Новостная лента RSS»

1. Новостная лента RSS в формате 2.0.
2. Обязательные и необязательные элементы канала.
3. Кроссбраузерность.
4. Кроссплатформеность.
5. Адаптивная верстка.
6. Media запросы.
7. Адаптивное меню.
8. Адаптивная галерея.

9. Требования к макету.

Критерии оценивания устного опроса:

- оценка «*отлично*» выставляется обучающемуся, если он ответил развернуто на один из заданных вопросов, активно дополнял ответы других обучающихся или задавал им дополнительные вопросы;
- оценка «*хорошо*» выставляется обучающемуся, если он ответил развернуто на один из заданных вопросов или ответил кратко на ряд вопросов;
- оценка «*удовлетворительно*» выставляется обучающемуся, если он кратко ответил на один из задаваемых вопросов или просто дополнял ответы других обучающихся, задавал им дополнительные вопросы;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если он не смог ответить правильно на заданный вопрос, либо не отвечал на вопросы и не участвовал в их обсуждении.

Практические работы

Практическое задание по теме «Реляционная модель баз данных»

Задание: Запускаем OpenServer. Создаем новую БД. Создаем нового пользователя БД, присваиваем логин и пароль, а также назначаем привилегии. Создаем новую таблицы в БД. Задаем таблицам названия и указываем количество полей в этой таблице. Заполняем таблицу данными, получая навыки работы в phpMyAdmin.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если задание было выполнено полном объеме, база данных создана, новый пользователь заведен в БД, присвоен логин и пароль, назначены все привилегии, созданы таблицы в БД, все таблицы заполнены данными;
- оценка «*хорошо*» выставляется обучающемуся, если задание было выполнено частично, база данных создана, новый пользователь заведен в БД, присвоен логин и пароль, назначены все привилегии, созданы таблицы в БД;
- оценка «*удовлетворительно*» выставляется обучающемуся, если задание было выполнено частично, но база данных создана, новый пользователь заведен в БД, присвоен логин и пароль, назначены все привилегии;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если база данных создана, но пользователей нет, таблиц нет.

Практическое задание по теме «Введение в JavaScript»

Задача. Создайте переменную **str** и присвойте ей значение 'abcde'. Обращаясь к отдельным символам этой строки выведите на экран символ 'a', символ 'b', символ 'e'.

Решение: будем обращаться к отдельным символам этой строки, к примеру, буква 'a' имеет номер 0 и ее можно вывести так - **str[0]**, буква 'b' имеет номер 1 и так далее:

```
var str = 'abcde';

alert(str[0]); //выведем букву 'a'

alert(str[1]); //выведем букву 'b'
```

```
alert(str[4]); //выведем букву 'e'
```

Задача. Напишите скрипт, который считает **количество секунд** в часе.

Решение: так как в минуте **60** секунд, а в часе - **60** минут, то умножив **60** на **60** мы получим **количество секунд в часе**:

```
alert(60 * 60);
```

Если нам нужно получить количество секунд в **сутках**, то умножим еще и на **24** (так как в сутках **24** часа):

```
alert(60 * 60 * 24);
```

Задача. Переделайте приведенный код так, чтобы в нем использовались операции **+=, -=, *=, /=, ++, --**. Количество строк кода при этом не должно измениться. Код для переделки:

```
var num = 1;

num = num + 12;

num = num - 14;

num = num * 5;

num = num / 7;

num = num + 1;

num = num - 1;

alert(num);
```

Решение: заменим все подходящие места на сокращенную форму записи. К примеру, вместо **num = num + 12** можно написать **num += 12**, а вместо **num = num + 1** будет **num++**. Результат переделки будет выглядеть так:

```
var num = 1;

num += 12;

num -= 14;

num *= 5;

num /= 7;

num++;

num--;

alert(num);
```

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, обучающийся корректно объясняет их решение;
- оценка «*хорошо*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, но есть несущественные замечания к качеству кода, обучающийся корректно объясняет их решение;
- оценка «*удовлетворительно*» выставляется обучающемуся, если задание выполнено в полном объеме, задачи решены с замечаниями и обучающийся затрудняется отвечать на часть вопросов;

- оценка «*неудовлетворительно*» выставляется обучающемуся, если задачи не решены или решены неверно, обучающийся не знает решение.

Практическое задание по теме «Качество кода»

Задание: написать HTML-страничку, на которой нужно прописать пути к библиотекам, размещенным на сайте https://cdnjs.cloudflare.com. На эту же HTML-страницу помещают тестируемый код и ссылку на программу-скрипт, выполняющую тестирование (тестируемый код и тестирующий скрипт написаны на JavaScript). Запуск тестирования происходит при загрузке этой HTML-странички в браузер. К каждой написанной функции можно приложить тестирующий ее скрипт с тестовыми данными и разнообразными проверками. При внесении в код функции изменений достаточно запустить тестирующий скрипт (в котором могут быть сотни или даже тысячи разных проверок с разнообразными тестовыми данными) и просмотреть результаты всех проверок. При этом отпадает необходимость при каждом изменении вручную тестировать измененную функцию, не требуется фиксировать наборы тестовых данных и список проверок.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если задание выполнено в полном объеме, страница создана, тестирование пройдено, обучающийся корректно объясняет процедуру проверки;
- оценка «*хорошо*» выставляется обучающемуся, если задание выполнено в полном объеме, страница создана, тестирование пройдено, обучающийся корректно объясняет процедуру проверки с небольшими замечаниями;
- оценка «*удовлетворительно*» выставляется обучающемуся, задание выполнено в полном объеме, страница создана, тестирование пройдено с замечаниями и обучающийся затрудняется отвечать на часть вопросов;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если задание выполнено с критическими ошибками или не выполнено.

Практическое задание по теме «Объекты: основы»

Задача

Задача. Дан массив с элементами 'Привет, ', 'мир' и '!'. Необходимо вывести на экран фразу 'Привет, мир!'.

Решение:

//Выведем фразу 'Привет, мир!':

```
var arr = ['Привет, ', 'мир', '!'];

alert(arr[0] + arr[1] + arr[2]);
```

Разберем это решение: слово 'Привет, ' хранится под номером **0**, это значит, что для доступа к нему мы должны написать **arr[0]**. Для доступа к слову 'мир' мы должны написать **arr[1]**, а **arr[2]** содержит в себе '!'. Далее с помощью оператора **'+'** мы сложим три наши строки ('Привет, ', 'мир' и '!') в одну строку таким образом **arr[0] + arr[1] + arr[2]**, и выведем на экран с помощью **alert**.

Задача

Задача. Решим немного другую задачу: дан массив с элементами 'Привет, ', 'мир' и '!'. Необходимо записать в переменную **text** фразу 'Привет, мир!', а затем вывести на экран содержимое этой переменной.

Решение:

```
//Выведем фразу 'Привет, мир!':

var arr = ['Привет, ', 'мир', '!'];

var str = arr[0] + arr[1] + arr[2]; //в переменной text теперь лежит строка 'Привет, мир!'

alert(str);
```

Задача

Задача. Дан массив ['Привет, ', 'мир', '!']. Необходимо записать в нулевой элемент **этого** массива слово 'Пока, ' (то есть вместо слова **'Привет, '** будет **'Пока, '**).

Решение:

```
var arr = ['Привет, ', 'мир', '!'];

arr[0] = 'Пока, '; //перезапишем нулевой элемент массива

alert(arr); //посмотрим на массив и убедимся в том, что он изменился
```

Задача . Объекты (ассоциативные массивы)

Задача. Создайте ассоциативный массив (объект) заработных плат **obj**. Выведите на экран зарплату Пети и Коли.

```
//Этот объект дан:

var obj = {'Коля':'1000', 'Вася':'500', 'Петя':'200'};
```

Решение: Чтобы вывести зарплату Коли следует вывести значение элемента объекта **с ключом 'Коля'**. Сделаем это:

```
var obj = {'Коля':'1000', 'Вася':'500', 'Петя':'200'};

alert(obj['Коля']); //выведет 1000
```

Задача . Способы создания массива

Задача. Создайте массив **arr** с элементами *1, 2, 3, 4, 5* двумя различными способами.

Решение:

Первый способ создать массив - объявить его через **[ ]**:

```
var arr = [1, 2, 3, 4, 5];
```

Второй способ создания массива - это поступить таким образом:

```
var arr = [];
```

```
arr[0] = 1;
arr[1] = 2;
arr[2] = 3;
arr[3] = 4;
arr[4] = 5;
```

Задача . Многомерный массив

Задача. Дан многомерный массив **arr**:

```
var arr = {
        'ru':['голубой', 'красный', 'зеленый'],
        'en':['blue', 'red', 'green'],
};
```

Выведите с его помощью слово **'голубой'**.

Решение: так как массив многомерный (в нашем случае двухмерный), то нам придется написать несколько квадратных скобок подряд. Поясню это по шагам. Давайте сделаем так:

```
alert(arr['ru']);
```

В этом случае результатом будет массив **['голубой','красный', 'зеленый']**, который является частью нашего многомерного массива. Чтобы вывести слово **'голубой'**, необходимо дописать еще одну квадратную скобку с ключом, соответствующим этому элементу (у него нет явного ключа - значит его ключ **0**):

```
alert(arr['ru'][0]); //выведет 'голубой'
```

Выведем теперь слово **'красный'**:

```
alert(arr['ru'][1]); //выведет 'красный'
```

Выведем **'red'**:

```
alert(arr['en'][1]); //выведет 'red'
```

Работа с массивами

Создайте массив **arr = ['a', 'b', 'c']**. Выведите его на экран с помощью функции **alert**.

С помощью массива **arr** из предыдущего номера выведите на экран содержимое **первого**, **второго** и **третьего** элементов.

Создайте массив **arr = ['a', 'b', 'c', 'd']** и с его помощью выведите на экран строку **'a+b, c+d'**.

Создайте массив **arr** с элементами **2, 5, 3, 9**. Умножьте первый элемент массива на второй, а третий элемент на четвертый. Результаты сложите, присвойте переменной **result**. Выведите на экран значение этой переменной.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, обучающийся корректно объясняет их решение;
- оценка «*хорошо*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, но есть несущественные замечания к качеству кода, обучающийся корректно объясняет их решение;

- оценка «*удовлетворительно*» выставляется обучающемуся, если задание выполнено в полном объеме, задачи решены с замечаниями и обучающийся затрудняется отвечать на часть вопросов;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если задачи не решены или решены неверно, обучающийся не знает решение.

Практическое задание по теме «Типы данных»

Работа с JSON.parse

Дана JSON строка '["Коля", "Вася", "Петя"]'. Преобразуйте ее в массив JavaScript и выведите на экран элемент "Петя".

Решение:

```
var name = '["Коля", "Вася", "Петя"]';

name = JSON.parse(name);

alert(name[2]);
```

Работа с JSON.stringify

Дан объект {a: 'aaa', b: 'bbb'}. Преобразуйте его в JSON.

Решение:

```
var obj = {a: 'aaa', b: 'bbb'};

alert( JSON.stringify(obj, ["a", "b"]));
```

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, обучающийся корректно объясняет их решение;
- оценка «*хорошо*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, но есть несущественные замечания к качеству кода, обучающийся корректно объясняет их решение;
- оценка «*удовлетворительно*» выставляется обучающемуся, если задание выполнено в полном объеме, задачи решены с замечаниями и обучающийся затрудняется отвечать на часть вопросов;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если задачи не решены или решены неверно, обучающийся не знает решение.

Тестирование «Типы данных»

1. Назначение JavaScript

написание frontend-приложений
написание backend-приложений
написание мобильных приложений
все вышеперечисленное

2. Каким образом можно подключить скрипты к HTML?

прописать внутри кода страницы
подключить отдельным файлом
оба варианты верны

3. Необходимо выбрать неверный пункт. Для чего нужен JavaScript?

Для создания анимации
Для обработки данных HTML
Для работы с облачными хранилищами
Для взаимодействия с пользователем
Для математических вычислений

4. Какой оператор вернет тип переменной, переданной ему в качестве аргумента?

native
instanceof
typeof

5. Что определяет переменная?

Символическое имя для значения
Тип данных
Тип значения

6. Что означает значение undefined?

Ошибочный результат вычислений
Что значение не было присвоено
Ссылкой на несуществующий объект

7. К какому типу относится значение NaN?

К числовому
К строковому
К логическому

8. Есть ли разница в JavaScript между двойными и одинарными кавычками при записи строковых литералов?

Нет
Да
Только при работе с jQuery

9. Что представляет собой значение Infinity?

Вычислительную ошибку
Математическую бесконечность
Неопределённое значение

10. К какому типу относится строка в JavaScript?

Объект
Простой тип

11. К какому типу относится значение null?

К символьному
К строковому
К числовому
К логическому
Ни к одному из перечисленных

12. Чем в JavaScript является функция?
Объектом
Последовательностью
Строкой

13. Существует ли в JavaScript символьный тип данных?
Нет
Да
Только при работе с Node.js

14. Можно ли в JavaScript в переменную, в которой ранее хранилось целое число записать объект?
Можно
Нельзя

15. К какой типизации относится JavaScript?
К динамической и слабой
К сильной и статической
К явной и динамической

16. Сколько значений в JavaScript может принимать логический тип?
Одно
Два
Три

17. Существует ли в JavaScript отдельный тип данных для чисел с плавающей запятой?
Да, это тип float
Да, это два типа float и double
Нет, за целочисленные значения и числа с плавающей точкой в JavaScript отвечает тип number

Критерии оценивания тестирования:

- оценка «*отлично*» выставляется обучающемуся, если он ответил верно на 15-17 вопросов;
- оценка «*хорошо*» выставляется обучающемуся, если он ответил верно на 12-14 вопросов;
- оценка «*удовлетворительно*» выставляется обучающемуся, если он ответил верно на 9-11 вопросов;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если ответил верно менее, чем на 9 вопросов.

Практическое задание по теме «Продвинутая работа с функциями»

Работа с рекурсией

Дан массив с числами. Выведите последовательно его элементы используя рекурсию и не используя цикл. Скрыть решение.

Решение:

```
function func(arr) {
	alert(arr[0]);
	arr.splice(0, 1);
	if (arr.length > 0) {
		func(arr);
	}
}
```

Дано число. Сложите его цифры. Если сумма получилась более 9-ти, опять сложите его цифры. И так, пока сумма не станет однозначным числом (9 и менее).

На forEach

Для решения задач данного блока вам понадобятся следующие методы: forEach.

Дан массив с числами. Создайте новый массив, состоящий из квадратов этих чисел.

Решение:

```
var arr = [5, 6, 7, 8, 9];
var newArr = [];
arr.forEach(function(elem) {
	newArr.push(elem * elem);
});
alert(newArr)
```

Дан массив с числами. Найдите сумму этих чисел.

На map

Для решения задач данного блока вам понадобятся следующие методы: map.

Дан массив с числами. Сделайте из него массив, состоящий из квадратов этих чисел.

Решение:

```
var arr = [5, 6, 7, 8, 9];
var newArr = arr.map(function(elem) {
	return elem * elem;
});
alert(newArr);
```

На every, some

Для решения задач данного блока вам понадобятся следующие методы: every, some.

Дан массив с числами. Проверьте то, что все элементы в массиве больше нуля

Решение:

```
var arr = [5, 6, 7, 8, 9];

function func(elem){return elem > 0};

alert(arr.every(func));
```

Дан массив с числами. Проверьте то, что в нем есть отрицательные элементы.

Решение:

```
var arr = [5, 6, -7, 8, 9];

function func(elem){return elem < 0};

alert(arr.some(func));
```

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, обучающийся корректно объясняет их решение;
- оценка «*хорошо*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, но есть несущественные замечания к качеству кода, обучающийся корректно объясняет их решение;
- оценка «*удовлетворительно*» выставляется обучающемуся, если задание выполнено в полном объеме, задачи решены с замечаниями и обучающийся затрудняется отвечать на часть вопросов;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если задачи не решены или решены неверно, обучающийся не знает решение.

Практическое задание по теме «Свойства объекта, их конфигурация. Прототипы, наследование, классы»

Задача:

Дан элемент. Сделайте так, чтобы по клику на него он красился в красный цвет, но только если в момент клика нажата клавиша Ctrl.

Задача:

Дан элемент. Сделайте так, чтобы при клике на него и нажатой клавише Ctrl - в его текст записывалось '1', при нажатой клавише Alt - '2', а при нажатой клавише Shift - '3'.

Задача:

Дан инпут. В него вводится текст и нажимается клавиша Enter (ее код имеет номер 13). Сделайте так, чтобы по нажатию Enter введенный текст попадал в абзац под инпутом, а содержимое инпута очищалось.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, обучающийся корректно объясняет их решение;

- оценка «*хорошо*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, но есть несущественные замечания к качеству кода, обучающийся корректно объясняет их решение;
- оценка «*удовлетворительно*» выставляется обучающемуся, если задание выполнено в полном объеме, задачи решены с замечаниями и обучающийся затрудняется отвечать на часть вопросов;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если задачи не решены или решены неверно, обучающийся не знает решение.

Практическое задание по теме «Промисы, async/await»

Задача:

Сделайте промис, который через 5 секунд случайным образом выполнится или с успехом, или с ошибкой. Примените изученный метод catch для отлавливания ошибок.

Задача:

Сделайте цепочку из трех промисов. Пусть первый промис возвращает число. Сделайте так, чтобы каждый последующий промис через 3 секунды возводил в квадрат результат предыдущего промиса. После окончания манипуляций выведите число алертом на экран.
Задача:

Переделайте предыдущую задачу так, чтобы один из промисов в цепочке выполнился с ошибкой. В конце цепочки расположите метод catch, который поймает эту ошибку.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, обучающийся корректно объясняет их решение;
- оценка «*хорошо*» выставляется обучающемуся, если задание выполнено в полном объеме, все задачи решены верно, но есть несущественные замечания к качеству кода, обучающийся корректно объясняет их решение;
- оценка «*удовлетворительно*» выставляется обучающемуся, если задание выполнено в полном объеме, задачи решены с замечаниями и обучающийся затрудняется отвечать на часть вопросов;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если задачи не решены или решены неверно, обучающийся не знает решение.

Практическое задание по теме «Технологии разработки программных продуктов. Основные определения и подходы»

Задание: Провести аналитический обзор программных средств для разработки ПО и программных оболочек для разработки сайтов на локальной машине. Результат работы представить в виде аналитического отчета.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если составлен в полном объеме аналитический обзор рынка программных средств для разработки ПО и программных оболочек, а также проведён сравнительный анализ программно-технических средств и информационных продуктов в соответствии с полученным заданием;
- оценка «*хорошо*» выставляется обучающемуся, если аналитический обзор рынка программных средств для разработки ПО и программных оболочек выполнен с незначительными замечаниями, а также проведён сравнительный анализ программно-технических средств и информационных продуктов в соответствии с полученным заданием;
- оценка «*удовлетворительно*» выставляется обучающемуся, если аналитический обзор рынка программных средств для разработки ПО и программных оболочек выполнен с критическими ошибками и замечаниями, частично проведён сравнительный анализ программно-технических средств и информационных продуктов в соответствии с полученным заданием;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если аналитический отчет не составлен.

Практическое задание по теме «Создание сценариев на PHP. Типы данных, переменные, операторы»

Задание 1: Запустить «Denwer». После запуска создается диск Z. На диске Z в папке www создать папку myphp. Создать файл в Notepad++ с именем index.php. Наполнить файл html-тэгами так, чтобы получился каркас странички html. Вписать в тело странички php скрипт для вывода на экран строки текста. Тут же создаем файл style.css, заполняем его информацией о стилях отображения странички. Теперь открываем браузер и в адресной строке набираем адрес нового сайта и полный путь до запускаемого файла, в нашем случае этот путь будет выглядеть вот так: http://localhost/myphp/index.php и нажимаем Enter. Если Вы все сделали правильно, то на мониторе (в браузере) Вы увидите свою первую страничку.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если страничка сверстана правильно, используются стили, php-скрипт обработан препроцессором и выводит на экран нужную информацию;
- оценка «*хорошо*» выставляется обучающемуся, если страничка сверстана правильно. Php-скрипт обработан препроцессором и выводит на экран нужную информацию;
- оценка «*удовлетворительно*» выставляется обучающемуся, если страничка сверстана правильно, используются стили. (ПК-1, ПК-22, пороговый уровень освоения компетенции);
- оценка «*неудовлетворительно*» выставляется обучающемуся, если страничка сверстана правильно, нет понимания работы препроцессора.

Задание 2:

1. Необходимо создать свою папку хранения интернет - страниц на web - сервере OpenServerZ:\home\localhost\www. Имя папке даем произвольное. Выполненные работы необходимо сохранять с расширением «.php».

2. Изучить материал о переменных и о выводе информации на экран в PHP. Составить код программы в связке html и php для вывода на экран значений переменных, строк, а также выводим одновременно переменные и текстовые строки.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если все созданные файлы с расширением php. Файлы обрабатываются перепроцессором правильно и выводится нужная информация. Студент понимает принципы работы с переменными и командами отображения информации на экране;
- оценка «*хорошо*» выставляется обучающемуся, если все созданные файлы с расширением php, файлы обрабатываются перепроцессором правильно и выводится нужная информация, студент понимает принципы работы с командами отображения информации на экране;
- оценка «*удовлетворительно*» выставляется обучающемуся, если все созданные файлы с расширением php, файлы обрабатываются перепроцессором правильно и выводится нужная информация, студент понимает принципы работы с переменными;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если студент не понимает принципов работы с переменными и командами отображения информации на экране.

Практическое задание по теме «Операции и управляющие конструкции»

Задание:

1. Необходимо создать свою папку хранения интернет - страниц на web - сервере OpenServerZ:\home\localhost\www. Имя папке даем произвольное. Выполненные работы необходимо сохранять с расширением «.php».

2. Изучить материал об операторах и операциях в PHP. Составить коды программ в связке html и php для решения заданий:

- написать страничку php, где вычисляется текущая дата и время;
- написать код программы, позволяющий выводить на экран сумму кубов и квадратов натуральных чисел, начиная с 10 до 20;
- составить код программы, позволяющий выводить на экран только четные числа квадратов и кубов натуральных чисел с 10 до 20;
- составить код программы, позволяющий выводить на экран только нечетные числа квадратов и кубов чисел, начиная с 10 до 20;

- составить код программы, позволяющий выводить на экран разность, умножение и деление кубов и квадратов натуральных чисел, начиная с 20 до 30;
- составить код программы, где переменной $a присваивается значение 4, а переменной $b значение 5 и вывести на экран переменные и значения в них.
- составить код программы, где используются операции инкремента и декремента;
- составить код программы, где используются битовые операции;
- составить код программы, где используются операции сравнения.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если в архиве содержаться 9 файлов php, где один из них играет ведущую роль, в каждом файле составлен рабочий код программы, согласно заданиям, все файлы имеют презентабельный вид;
- оценка «*хорошо*» выставляется обучающемуся, если в архиве содержаться 7 файлов php, где один из них играет ведущую роль, в каждом файле составлен рабочий код программы, согласно заданиям, все файлы имеют презентабельный вид;
- оценка «*удовлетворительно*» выставляется обучающемуся, если в архиве содержаться 6 файлов php, где один из них играет ведущую роль или 8 простых файлов php, в каждом файле составлен рабочий код программы, согласно заданиям;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если в архиве содержаться файлы php. В файлах составлен код программ, который не работоспособен.

Практическое задание по теме «Функции и повторное использование кода»

Задание:

1. Необходимо создать свою папку хранения интернет - страниц на web - сервере OpenServer. Имя папке даем произвольное. Выполненные работы необходимо сохранять с расширением «.php».

2. Составить код программы PHP всех операций и вывести результаты вычислений на экран для конструкций:

- if-else;
- двойного условия if-else;
- вложенных if-else;
- break;
- switch-case.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если в архиве содержаться 6 файлов php, каждый из них выводит результаты вычислений на экран, в каждом файле составлен рабочий код программы, согласно заданиям, все файлы имеют презентабельный вид;

- оценка «*хорошо*» выставляется обучающемуся, если в архиве содержаться 4 файла php, каждый из них выводит результаты вычислений на экран, в каждом файле составлен рабочий код программы, согласно заданиям, все файлы имеют презентабельный вид;
- оценка «*удовлетворительно*» выставляется обучающемуся, если в архиве содержаться 3 файла php каждый из них выводит результаты вычислений на экран, в каждом файле составлен рабочий код программы, согласно заданиям;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если в архиве содержаться файлы php, в файлах составлен код программ, который не работоспособен.

Практическое задание по теме «Массивы»

Задание:

1. Необходимо создать свою папку хранения интернет - страниц на web - сервере OpenServerZ:\home\localhost\www. Имя папке даем произвольное. Выполненные работы необходимо сохранять с расширением «.php».

2. Изучить материал о разновидности массивов в PHP. Составить коды программ для реализации работы массивов:

- простого массива;
- автомассива;
- ассоциативного массива;
- многомерного массива.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если в архиве содержаться 6 файлов php, каждый из которых отображает работу массива, все массивы в заданиях разновидны, в каждом файле составлен рабочий код программы, согласно заданиям, все файлы имеют презентабельный вид;
- оценка «*хорошо*» выставляется обучающемуся, если в архиве содержаться 5 файлов php, каждый из которых отображает работу массива, все массивы в заданиях разновидны, в каждом файле составлен рабочий код программы, согласно заданиям, все файлы имеют презентабельный вид;
- оценка «*удовлетворительно*» выставляется обучающемуся, если в архиве содержаться 4 файлов php, каждый из которых отображает работу массива, все массивы в заданиях разновидны, в каждом файле составлен рабочий код программы, согласно заданиям;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если в архиве содержаться файлы php, в файлах составлен код программ, который не работоспособен.

Практическое задание по теме «Передача данных через HTML-формы»

Задание:

1. Необходимо создать свою папку хранения интернет - страниц на web - сервере OpenServerZ:\home\localhost\www. Имя папке даем произвольное. Выполненные работы необходимо сохранять с расширением «.php».

2. Изучить материал о формах в HTML. Разработать формы HTML с различными элементами управления.

3. Изучить материал о передачи данных из форм HTML средствами php. Разработать формы и сценарий для передачи данных из формы методами POST, GET.

Критерии оценивания:

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если в архиве содержаться 18 файлов php, 2 из которых отображает построение форм с различными элементами управления, все массивы в заданиях разновидны, в каждом файле составлен рабочий код программы, согласно заданиям, все файлы имеют презентабельный вид;
- оценка «*хорошо*» выставляется обучающемуся, если в архиве содержаться 5 файлов php, каждый из которых отображает работу массива, все массивы в заданиях разновидны, в каждом файле составлен рабочий код программы, согласно заданиям. Все файлы имеют презентабельный вид;
- оценка «*удовлетворительно*» выставляется обучающемуся, если в архиве содержаться 4 файлов php, каждый из которых отображает работу массива, все массивы в заданиях разновидны, в каждом файле составлен рабочий код программы, согласно заданиям;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если в архиве содержаться файлы php. В файлах составлен код программ, который не работоспособен.

Практическое задание по теме «Работа с файлами»

Задание:

1. Необходимо создать свою папку хранения интернет - страниц на web - сервере OpenServerZ:\home\localhost\www. Имя папке даем произвольное. Выполненные работы необходимо сохранять с расширением «.php».

2. Получение практических навыков работы с файлами. Создать панель управления файлами согласно рисунку.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если набранный код программы имеет вид панели управления файлами, панель управления файлами функционирует, в панели отображаются имена файлов и размер, выбранные файлы в панели можно удалить, переименовать, выгрузить;
- оценка «*хорошо*» выставляется обучающемуся, если набранный код программы имеет вид панели управления файлами, панель управления файлами функционирует, в панели

отображаются имена файлов и размер, выбранные файлы в панели можно удалить, переименовать;

- оценка «*удовлетворительно*» выставляется обучающемуся, если набранный код программы имеет вид панели управления файлами, панель управления файлами частично функционирует, в панели отображаются имена файлов и размер;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если набранный код программы не функционирует.

Практическое задание по теме «Cookies и управление сессиями»

Задание:

1. Необходимо создать свою папку хранения интернет - страниц на web - сервере OpenServerZ:\home\localhost\www. Имя папке даем произвольное. Выполненные работы необходимо сохранять с расширением «.php».

2. Получение практических навыков работы с cookie. Работа с cookies файлами. Создать панель управления cookie согласно рисунку.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если набранный код программы имеет вид панели управления cookie, панель управления cookie функционирует, в панели отображаются количество посещения страницы пользователем, цвета фона, новый цвет текста и возможность удаления куки;
- оценка «*хорошо*» выставляется обучающемуся, если набранный код программы имеет вид панели управления cookie, панель управления cookie функционирует, в панели отображаются количество посещения страницы пользователем, цвета фона, новый цвет текста;
- оценка «*удовлетворительно*» выставляется обучающемуся, если набранный код программы имеет вид панели управления cookie, панель управления cookie частично функционирует;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если набранный код программы не функционирует.

Практическое задание по теме «Сценарии сайта»

Задание: создать домашнюю страницу сайта с возможностью выбора и заказа тура. Также создать страницы описаний стран, городов и отелей. Учитывать функцию администрирования сайта.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если набранный код программы соответствует теме, страницы работают и наполнены контентом, учтен вход администратора сайта;
- оценка «*хорошо*» выставляется обучающемуся, набранный код программы соответствует теме, страницы работают и наполнены контентом с незначительными замечаниями, учтен вход администратора сайта;
- оценка «*удовлетворительно*» выставляется обучающемуся, если набранный код программы соответствует теме, страницы работают и наполнены контентом со значительными замечаниями, учтен вход администратора сайта;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если код сайта не функционирует.

Практическое задание по теме «Расширение mysqli для работы с базами данных»

Задание: создать скрипт для работы с MySQL. Осуществить подключение к серверу и выбор базы данных. Выполнить запросы к базе данных.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если верно и в полном объеме написан крипт для работы с MySQL, осуществлено подключение к серверу и произведён выбор базы данных, запросы выполнены к базе данных;
- оценка «*хорошо*» выставляется обучающемуся, верно и в полном объеме написан крипт для работы с MySQL, осуществлено подключение к серверу и произведён выбор базы данных, запросы выполнены к базе данных с незначительными замечаниями;
- оценка «*удовлетворительно*» выставляется обучающемуся, если верно и в полном объеме написан крипт для работы с MySQL, осуществлено подключение к серверу и произведён выбор базы данных, запросы выполнены к базе данных со значительными замечаниями;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если скрипт и подключение не выполнено.

Практическое задание по теме «XML и PHP»

Задание:

1. Необходимо создать свою папку хранения интернет - страниц на web - сервере OpenServerZ:\home\localhost\www. Имя папке даем произвольное. Выполненные работы необходимо сохранять с расширением «.php».
2. Получение практических навыков работы с XML. Организация работы с данными в xml-формате. Для этого изучить теоретический материал и исходные тексты примеров серверных скриптов
3. Выполните верстку дополнительных страниц сайта.
4. Протестировать и отладить скрипты, динамически формирующие страницы сайта.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если набранный код программы содержит данные в xml-формате, сайт полностью функционирует, результаты тестирования запротоколированы;
- оценка «*хорошо*» выставляется обучающемуся, если набранный код программы содержит данные в xml-формате, сайт полностью функционирует, результаты тестирования зафиксированы;
- оценка «*удовлетворительно*» выставляется обучающемуся, если набранный код программы содержит данные в xml-формате, сайт частично функционирует;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если набранный код программы не функционирует.

Задание для работы в малых группах,

выполняемого в 4 семестре:

Цель: *разработка web-приложения для заданной предметной области.*

Общее задание: Обучающиеся разбиваются на небольшие группы из 2-3 человек. По выданному преподавателем техническому заданию разрабатывается каркас сайта, дизайн сайта и его контент. Приветствуется реальное техническое задание. В противном случае тема может быть выбрана обучающимися произвольно.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если он понимает и воспроизводит принципы построения Web-приложений, составляет код программы и воспроизводит взаимодействие PHP, HTML и MySQL, составленный код программы работает без сбоев, организована проверка на ошибки при вводе данных, предусмотрена защита данных;
- оценка «*хорошо*» выставляется обучающемуся, если он понимает и воспроизводит принципы построения Web-приложений, воспроизводит взаимодействие PHP, HTML и MySQL, составленный код программы работает без сбоев, организована проверка на ошибки при вводе данных;
- оценка «*удовлетворительно*» выставляется обучающемуся, если он понимает принципы построения Web-приложений, воспроизводит взаимодействие каркаса и базы данных web-приложений, составленный код программы работает без сбоев;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если не понимает принципы построения Web-приложений, не знает определение баз данных и характеристику средств разработки, программный код не составляет

3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

ОЦЕНКА ЗНАНИЙ ОБУЧАЮЩИХСЯ НА ЭКЗАМЕНЕ

Список вопросов для подготовки к экзамену

ПК-1. Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку web-приложения

ПК-1.1. Проводит обследование организаций, выявляет информационные потребности заказчика, используя цифровые инструменты и облачные решения для сбора данных

Обучающийся знает: методику проведения обследования организации и выявления информационных потребностей пользователей; виды требований, предъявляемых к web-приложения; виды информации, необходимой для формализации требований заказчика для разработки базы данных и/или web-приложения, методы сбора и анализа детальной информации

1. Распределённая база данных
2. Экспорт и импорт баз данных
3. Синхронизация данных
4. Общий слой баз данных
5. Реестр баз данных
6. Семантические данные
7. Анализ методов проектирования и управления распределёнными базами данных
8. Пакет программных средств для автоматизированного проектирования структур распределённых баз данных
9. Области применения баз данных.
10. Виды баз данных.
11. Информационная модель данных и ее структура.
12. Основные понятия реляционных баз данных: поля, записи, индексы, ключи, таблицы, отношения.

13. Уровни модели данных

ПК-1. Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку web-приложения

ПК-1.2. Формализует требования заказчика к разработке web-приложения

Обучающийся знает: структуру и порядок построения технического задания на разработку web-приложения

1. Инструментарий для проектирования и документирования баз данных
2. Выборка информации из БД.
3. Синтаксис php при выборке.
4. Резервное копирование. Восстановление базы данных.
5. Защита базы данных резервного копирования или каталога
6. Создание и поддержка открытых отчетов, отчетов об открытых проблемах
7. Жизненный цикл программы

ПК-2. Способен проектировать web-приложения

ПК-2.1. Разрабатывает архитектуру web-приложения с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы

Обучающийся знает: процесс и средства разработки web-приложений; факторы сложности разработки web-приложений, технологии, подходы и принципы к исследованию и созданию информационных систем, эргономические требования к ним

1. Справочники и спецификации. Редакторы кода. Консоль разработчика
2. Структура кода.
3. Строгий режим — "use strict".
4. Преобразование типов.
5. Конструкция "switch".
6. Функции.
7. Function Expression.
8. Стрелочные функции, основы.
9. Особенности JavaScript

ПК-2. Способен проектировать web-приложения

ПК-2.2. Проектирует базы данных, в том числе используя специализированные цифровые платформы

Обучающийся знает: содержание этапов разработки и принципы организации проектирования web-приложений, методики сравнения проектных решений; информационную модель данных и ее структуру, логические модели баз данных и их типы, формы нормализации данных в базе

1. Назначение и виды информационных систем, создаваемых на основе современных СУБД
2. Основные функции СУБД в разных типах информационных систем
3. Информационная модель данных и ее структура
4. Логические модели баз данных и их типы
5. Формы нормализация данных в базе
6. Инструменты и принципы разработки Web- приложения
7. Жизненный цикл программы
8. Этапы создания программных продуктов

ПК-2. Способен проектировать web-приложения

ПК-2.3. Проектирует пользовательский интерфейс с учетом тенденций развития сквозных цифровых технологий и с помощью цифровых инструментов

Обучающийся знает: принципы разработки пользовательского интерфейса web-приложения

1. Организационные процессы жизненного цикла программного продукта
2. Этапы создания программных продуктов
3. Модели жизненного цикла
4. Тестирование
5. Техническое задание. ГОСТ 34.602-89, 15.016-2016

ПК-2. Способен проектировать web-приложения
ПК-2.4. Проводит анализ современных программно-технических средств, специализированных цифровых платформ и облачных решений, сквозных цифровых технологий и отраслевых решений на их основе с целью выбора средств разработки web-приложений
Обучающийся знает: методы анализа рынка программно-технических средств, информационных продуктов и услуг; основные источники профессионально-ориентированной информации, предназначенной для проведения анализа рынка программно-технических средств

1. Информационная модель данных и ее структура
2. Инструменты и принципы разработки Web- приложения
3. Методы анализа рынка программно-технических средств, информационных продуктов и услуг
4. Основные источники профессионально-ориентированной информации, предназначенной для проведения анализа рынка программно-технических средств
5. Использование информационно-библиографических и справочно-правовых систем

ПК-3. Способен разрабатывать web-приложения
ПК-3.1. Осуществляет разработку программного кода и компонентов web-приложения, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем
Обучающийся знает: язык определения и манипулирования данными; основные этапы построения web-приложений, понятие кроссбраузерности, кроссплатформености и адаптивной верстки; термины, синтаксис, управляющие конструкции и типы данных скриптового языка общего назначения PHP; основы JavaScript; инструменты и принципы разработки web-приложения

1. Структура кода.
2. Строгий режим — "use strict".
3. Переменные.
4. Типы данных.
5. Взаимодействие: alert, prompt, confirm.
6. Преобразование типов.
7. Базовые операторы, математика.
8. Операторы сравнения.
9. Условное ветвление.
10. Логические операторы.
11. Циклы while и for.
12. Конструкция "switch".
13. Архитектура клиент-серверного приложения и технологии его разработки.
14. Язык определения и манипулирования данными.
15. Основные этапы построения web-приложений.
16. Понятие кроссбраузерности.
17. Понятие кроссплатформенности.
18. Понятие адаптивной верстки
19. Термины, синтаксис, управляющие конструкции PHP.

20. Типы данных скриптового языка общего назначения PHP.
21. Основы конструирования запросов в СУБД.
22. Назначение и команды структурированного языка запросов (SQL)

ПК-3. Способен разрабатывать web-приложения
ПК-3.2. Обеспечивает интеграцию программных модулей и компонентов программного обеспечения, в том числе используя отраслевые облачные решения и цифровые платформы
Обучающийся знает: назначение тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования

1. Назначение тестирования.
2. Виды и способы тестирования баз данных и Web-приложений
3. Функциональное тестирование веб-приложений
4. Нагрузочное тестирование и тестирование производительности
5. Тестирование безопасности

ПК-5. Способен обеспечивать качество работы web-приложения
ПК-5.2. Проводит юзабилити-аудит web-приложения
Обучающийся знает: назначение юзабилити-аудит web-приложения

1. Понятие юзабилити
2. Руководство пользователя, администратора
3. Инструменты разработки руководства пользователя
4. Правила подготовки презентации.
5. Техническое задание. ГОСТ 34.602-89, 15.016-2016
6. Основные источники профессионально-ориентированной информации, предназначенной

ПК-5. Способен обеспечивать качество работы web-приложения
ПК-5.3. Осуществляет работы по интеграционному тестированию web-приложения с внешним цифровым окружением
Обучающийся знает: назначение тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования

1. Назначение тестирования
2. Виды и способы тестирования баз данных и web-приложений
3. Процедура тестирования
4. Качество кода

ОЦЕНКА УМЕНИЙ И НАВЫКОВ ОБУЧАЮЩИХСЯ НА ЭКЗАМЕНЕ

ПК-1. Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку web-приложения
ПК-1.1. Проводит обследование организаций, выявляет информационные потребности заказчика, используя цифровые инструменты и облачные решения для сбора данных
Обучающийся умеет: проводить предпроектное и информационное обследование организации; выявлять информационные потребности пользователей к web-приложениям; выявлять информационные потребности предприятия в СУБД
Обучающийся владеет: навыками сбора и анализа информации о предприятии; навыками подготовки отчета о проведенном предпроектном обследовании организации
Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам практического задания, в котором обучающийся демонстрирует проведенное

предпроектное и информационное обследование организации, определяет информационные потребности пользователей к web-приложению

ПК-1. Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку web-приложения
ПК-1.2. Формализует требования заказчика к разработке web-приложения
Обучающийся умеет: формировать техническое задание к разрабатываемому web-приложению
Обучающийся владеет: навыками документирования требований заказчика; навыками составления технического задания на разработку web-приложения и проектов..
Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам практического задания, в котором обучающийся формирует техническое задание к проекту.

ПК-2. Способен проектировать web-приложения
ПК-2.1. Разрабатывает архитектуру web-приложения с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы
Обучающийся умеет: определять факторы сложности разработки программных систем, выбирать адекватные технологии для их проектирования
Обучающийся владеет: навыками разработки архитектуру web-приложения с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы
Оценка достижения запланированного результата обучения осуществляется по результатам индивидуального проектного задания и практических заданий в малых группах, в котором обучающийся обосновывает выбор инструментария для реализации поставленных задач во время разработки web-приложения на основе проектного решения

ПК-2. Способен проектировать web-приложения
ПК-2.2. Проектирует базы данных, в том числе используя специализированные цифровые платформы
Обучающийся умеет: разрабатывать концептуальную, логическую и физическую модели web-приложений; определять необходимый инструментарий по уровню обеспечения информационных систем при разработке web-приложений
Обучающийся владеет: навыками проектирования концептуальной модели БД и ее модификации; навыками разработки модели web-приложения на основе проектного решения; навыками подготовки и демонстрации прототипа решения.
Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам практического задания, в котором обучающийся демонстрирует спроектированную базы данных на основе реляционной модели данных, осуществляет передачу информации между приложением и веб-сервером, адаптирует СУБД под конкретные потребности предприятия.

ПК-2. Способен проектировать web-приложения
ПК-2.3. Проектирует пользовательский интерфейс с учетом тенденций развития сквозных цифровых технологий и с помощью цифровых инструментов
Обучающийся умеет: проектировать пользовательский интерфейс
Обучающийся владеет: навыками проектирования и разработки пользовательского интерфейса web-приложения.
Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам практического задания, в котором обучающийся демонстрирует разработанный пользовательский интерфейс web-приложения

ПК-2. Способен проектировать web-приложения

ПК-2.4. Проводит анализ современных программно-технических средств, специализированных цифровых платформ и облачных решений, сквозных цифровых технологий и отраслевых решений на их основе с целью выбора средств разработки web-приложений
Обучающийся умеет: составлять аналитические обзоры рынка программно-технических средств, информационных продуктов и услуг; проводить сравнительный анализ программно-технических средств и информационных продуктов в соответствии с полученным заданием
Обучающийся владеет: навыками работы с программно-технологическими средствами (поисковыми машинами, справочно-правовыми и экспертными системами, открытыми информационными ресурсами и т.п.) для выполнения задач исследовательского и аналитического характера
Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам практического задания, в котором обучающийся демонстрирует работу с программно-технологическими средствами (поисковыми машинами, справочно-правовыми и экспертными системами, открытыми информационными ресурсами и т.п.) для выполнения задач исследовательского и аналитического характера

ПК-3. Способен разрабатывать web-приложения
ПК-3.1. Осуществляет разработку программного кода и компонентов web-приложения, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем
Обучающийся умеет: обеспечить передачу информации между приложением и веб-сервером; осуществлять подключение к серверу и выбор базы данных; разрабатывать динамические приложения на языке JavaScript и PHP для решения прикладных задач
Обучающийся владеет: навыками создания приложения с помощью языков программирования (HTML, JavaScript, PHP, CSS); навыками создания сценариев с помощью операций и управляющих конструкций PHP; навыками работы с различными формами в PHP; навыками работы с сессиями и cookie в PHP, навыками осуществления передачи данных через HTML-формы.
Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения практических заданий и практических заданий в малых группах, в котором обучающийся разрабатывает web-приложение на основе HTML, JavaScript, PHP, CSS, создает сценарии с помощью операций и управляющих конструкций PHP, работает с различными формами, с сессиями и cookie в PHP

ПК-3. Способен разрабатывать web-приложения
ПК-3.2. Обеспечивает интеграцию программных модулей и компонентов программного обеспечения, в том числе используя отраслевые облачные решения и цифровые платформы
Обучающийся умеет: проводить процедуру тестирования баз данных и web-приложений
Обучающийся владеет: навыками протоколирования результатов тестирования.
Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам практических заданий и практического задания в малых группах, в котором обучающийся проводит и протоколирует результаты тестирования.

ПК-5. Способен обеспечивать качество работы web-приложения
ПК-5.2. Проводит юзабилити-аудит web-приложения
Обучающийся умеет: проводить юзабилити-аудит web-приложения
Обучающийся владеет: навыками протоколирования результатов тестирования
Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам практических заданий в малых группах, в котором обучающийся проводит и протоколирует результаты тестирования.

ПК-5. Способен обеспечивать качество работы web-приложения

ПК-5.3. Осуществляет работы по интеграционному тестированию web-приложения с внешним цифровым окружением
Обучающийся умеет проводить процедуру тестирования баз данных и web-приложений
Обучающийся владеет: навыками протоколирования результатов тестирования
Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам практических заданий и практических заданий в малых группах, в котором обучающийся проводит и протоколирует результаты тестирования.

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНОВАНИЯ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

ПРОЦЕДУРА ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Для контроля усвоения обучающимися данной дисциплины и достижения запланированных результатов обучения учебным планом предусмотрен экзамен, проводимый в форме устного ответа на теоретический вопрос и выполнение практического задания. При выставлении итоговой оценки по дисциплине учитываются результаты текущего контроля, отражающего сформированность практических умений и навыков обучающегося. Во время обучения обучающиеся участвуют в разных мероприятиях текущего контроля: участвуют в устных теоретических опросах, выполняют практические задания и практические задания в малых группах. Для допуска к экзамену обучающемуся необходимо выполнить все практические задания и выполнить к экзамену задание в малых группах. Допустимо совместить защиту практического задания в малых группах и ответ на теоретический вопрос экзамена.

Экзамен в третьем семестре

Отметка выставляется обучающемуся, который выполнил все практические задания на отметку не ниже «удовлетворительно», выполнил все практические задания, ответил на вопросы по различным заданиям, ответил на устный вопрос преподавателя из списка вопросов по одной из пройденных тем.

Отметка не выставляется обучающемуся, который не выполнил в полном объеме все практические задания.

Экзамен в четвертом семестре

Отметка выставляется обучающемуся, который выполнил все практические задания на отметку не ниже «удовлетворительно», выполнил и защитил групповое задание, ответил на устный вопрос экзаменационного билета и выполнил практическое задание.

Отметка не выставляется обучающемуся, который не выполнил в полном объеме все практические задания и групповое задание.

Примерный перечень вопросов для подготовки к экзамену в 4 семестре:

1. Фильтрация записей в БД.
2. Работа со связанными таблицами.
3. Основные сведения о языке SQL.

4. Индексы в таблицах БД.
5. Отбор данных из таблицы при реляционном способе доступа.
6. Безопасность в БД
7. Вложенные запросы при реляционном способе доступа.
8. Изменение состава полей таблицы при использовании языка SQL.
9. Соединение таблиц при использовании языка SQL.
10. Модификация записей БД при использовании языка SQL.
11. Привилегии в БД
12. Статические и динамические запросы при использовании языка SQL.
13. Архитектура клиент-серверного приложения.
14. Понятие «Локального сервера».
15. Программные средства для разработки ПО.
16. Нормализация отношений БД
17. ПО «Denwer».
18. Классификация и функции веб-сайтов.
19. Этапы создания веб-сайта.
20. Доменное имя, хостинг.
21. Методы и способы доступа к данным
22. Синтаксис языка php.
23. Типы данных в языке php.
24. Циклы.
25. Форма в HTML.
26. Строки в php.
27. Методы передачи данных в сценариях php.
28. Подключение к БД в php.
29. Условные операторы.
30. Конструкции включений.
31. Сессии в PHP.
32. Элементы управления в HTML.
33. Массив.
34. Атрибуты тега.
35. Каскадные таблицы стилей.
36. Селекторы, свойства и значения.

37. Сортировка и объединение массивов.
38. Методы GET и POST.
39. Инструкции и конструкции php.
40. Сессии в php.

Примеры практических заданий к экзамену в 4 семестре:

1. Разработать формы с различными элементами управления и сценарий для передачи данных из формы методами POST, GET.
2. Составить коды программ для реализации работы массивов: простого массива, автомассива, ассоциативного массива, многомерного массива.

КРИТЕРИИ ОЦЕНИВАНИЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ДЛЯ ФОРМИРОВАНИЯ КОМПЕТЕНЦИИ И КРИТЕРИИ ИХ ОЦЕНИВАНИЯ

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
ПК-1. Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку web-приложения **ПК-1.1.** Проводит обследование организаций, выявляет информационные потребности заказчика, используя цифровые инструменты и облачные решения для сбора данных					
Знать: методику проведения обследования организации и выявления информационных потребностей пользователей; виды требований, предъявляемых к web-приложениям; виды информации, необходимой для формализации требований заказчика для разработки базы данных и/или	Не знает	Фрагментарные знания о методике проведения обследования организации и выявления информационных потребностей пользователей; о видах требований, предъявляемых к web-приложениям; о видах информации, необходимой для формализации требований заказчика для	Общие, но не структурированные знания о методике проведения обследования организации и выявления информационных потребностей пользователей; о видах требований, предъявляемых к web-приложениям; о видах информации, необходимой для формализации требований заказчика для разработки базы данных и/или	Сформированные, но содержащие отдельные пробелы знания о методике проведения обследования организации и выявления информационных потребностей пользователей; о видах требований, предъявляемых к web-приложениям; о видах информации, необходимой для формализации требований заказчика для	Сформированные систематизированные прочные знания о методике проведения обследования организации и выявления информационных потребностей пользователей; о видах требований, предъявляемых к web-приложениям; о видах информации, необходимой для формализации требований заказчика для разработки базы данных и/или web-приложения,

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
web-приложения, методы сбора и анализа детальной информации		разработки базы данных и/или web-приложения, методы сбора и анализа детальной информации	web-приложения, методы сбора и анализа детальной информации	разработки базы данных и/или web-приложения, методы сбора и анализа детальной информации	методы сбора и анализа детальной информации
Уметь: -проводить предпроектное и информационное обследование организации; - выявлять информационные потребности пользователей к информационной системе; - выявлять информационные потребности предприятия в СУБД;	Не умеет	Частично освоенные умения проводить предпроектное и информационное обследование организации; выявлять информационные потребности пользователей к web-приложениям; выявлять информационные потребности предприятия в СУБД	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения проводить предпроектное и информационное обследование организации; выявлять информационные потребности пользователей к web-приложениям; выявлять информационные потребности предприятия в СУБД	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения проводить предпроектное и информационное обследование организации; выявлять информационные потребности к web-приложениям; выявлять информационные потребности предприятия в СУБД	Успешно сформированные, самостоятельно и осознанно выполняемые умения проводить предпроектное и информационное обследование организации; выявлять информационные потребности пользователей к web-приложениям; выявлять информационные потребности предприятия в СУБД
Владеть: -навыками сбора и анализа информации о предприятии; - навыками подготовки отчета о проведенном предпроектном обследовании организации;	Не владеет	Фрагментарно сформированные навыки сбора и анализа информации о предприятии, подготовки отчета о проведенном предпроектном обследовании организации	В целом сформированные, но выполняемые на элементарном уровне навыки сбора и анализа информации о предприятии, подготовки отчета о проведенном предпроектном обследовании организации	Сформированные, но не устойчивые в сложных ситуациях навыки сбора и анализа информации о предприятии, подготовки отчета о проведенном предпроектном обследовании организации	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки сбора и анализа информации о предприятии, подготовки отчета о проведенном предпроектном обследовании организации
ПК-1. Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку web-приложения **ПК-1.2.** Формализует требования заказчика к разработке web-приложения					
Знать: структуру и порядок построения технического задания на	Не знает	Фрагментарные знания о структуре и порядке построения технического	Общие, но не структурированные знания о структуре и порядке построения	Сформированные, но содержащие отдельные пробелы знания о структуре и порядке	Сформированные систематизированные прочные знания о структуре и порядке построения

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
разработку web-приложения		задания на разработку web-приложения	технического задания на разработку web-приложения	построения технического задания на разработку web-приложения	технического задания на разработку web-приложения
Уметь: формировать техническое задание к разрабатываемому web- приложению	Не умеет	Частично освоенные умения формировать техническое задание к разрабатываемому web-приложению	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения формировать техническое задание к разрабатываемому web-приложению	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения формировать техническое задание к разрабатываемому web-приложению	Успешно сформированные, самостоятельно и осознанно выполняемые умения формировать техническое задание к разрабатываемому web- приложению
Владеть: - навыками документирования требований заказчика; - навыками составления технического задания на разработку web-приложения и проектов	Не владеет	Фрагментарно сформированные навыки документирования требований заказчика и навыки составления технического задания на разработку web-приложения и проектов	В целом сформированные, но выполняемые на элементарном уровне навыки документирования требований заказчика и навыки составления технического задания на разработку web-приложения и проектов	Сформированные, но не устойчивые в сложных ситуациях навыки документирования требований заказчика и навыки составления технического задания на разработку web-приложения и проектов	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки документирования требований заказчика и навыки составления технического задания на разработку web-приложения и проектов
ПК-2. Способен проектировать web-приложения **ПК-2.1.** Разрабатывает архитектуру web-приложения с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы					
Знать: - процесс и средства разработки web-приложений; - факторы сложности разработки web-приложений, технологии, подходы и принципы к исследованию и созданию информационных систем,	Не знает	Фрагментарные знания процесса и средств разработки web-приложений; факторов сложности разработки web-приложений, технологий, подходов и принципов к	Общие, но не структурированные знания процесса и средств разработки web-приложений; факторов сложности разработки web-приложений, технологий, подходов и принципов к исследованию и созданию	Сформированные, но содержащие отдельные пробелы знания процесса и средств разработки web-приложений; факторов сложности разработки web-приложений, технологий, подходов и принципов к	Сформированные систематизированные прочные знания процесса и средств разработки web-приложений; факторов сложности разработки web-приложений, технологий, подходов и принципов к исследованию и

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
эргономические требования к ним		исследованию и созданию информационных систем, эргономические требования к ним	информационных систем, эргономические требования к ним	исследованию и созданию информационных систем, эргономические требования к ним	созданию информационных систем, эргономические требования к ним
Уметь: - определять факторы сложности разработки программных систем, выбирать адекватные технологии для их проектирования	Не умеет	Частично освоенные умения определять факторы сложности разработки программных систем, выбирать адекватные технологии для их проектирования	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения определять факторы сложности разработки программных систем, выбирать адекватные технологии для их проектирования	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения определять факторы сложности разработки программных систем, выбирать адекватные технологии для их проектирования	Успешно сформированные, самостоятельно и осознанно выполняемые умения определять факторы сложности разработки программных систем, выбирать адекватные технологии для их проектирования
Владеть: - навыками разработки архитектуру web-приложения с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы	Не владеет	Фрагментарно сформированные навыки разработки архитектуру web-приложения с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы	В целом сформированные, но выполняемые на элементарном уровне навыки разработки архитектуру web-приложения с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы	Сформированные, но не устойчивые в сложных ситуациях навыки разработки архитектуру web-приложения с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки разработки архитектуру web-приложения с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы
ПК-2. Способен проектировать web-приложения ПК-2.2. Проектирует базы данных, в том числе используя специализированные цифровые платформы					
Знать: - содержание этапов разработки и принципы организации проектирования web-приложений,	Не знает	Фрагментарные знания о содержании этапов разработки и принципах организации проектировани	Общие, но не структурированные знания о содержании этапов разработки и принципах организации	Сформированные, но содержащие отдельные пробелы знания о содержании этапов разработки и принципах организации	Сформированные систематизированные прочные знания о содержании этапов разработки и принципах организации

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
методики сравнения проектных решений; - информационную модель данных и ее структуру, логические модели баз данных и их типы, формы нормализации данных в базе		я web-приложений, о методиках сравнения проектных решений; информационной модели данных и ее структуре, логических моделей баз данных и их типы, форм нормализации данных в базе	проектирования web-приложений, о методиках сравнения проектных решений; информационной модели данных и ее структуре, логических моделей баз данных и их типы, форм нормализации данных в базе	проектирования web-приложений, о методиках сравнения проектных решений; информационной модели данных и ее структуре, логических моделей баз данных и их типы, форм нормализации данных в базе	проектирования web-приложений, о методиках сравнения проектных решений; информационной модели данных и ее структуре, логических моделей баз данных и их типы, форм нормализации данных в базе
Уметь: - разрабатывать концептуальную, логическую и физическую модели web-приложений; - определять необходимый инструментарий по уровню обеспечения информационных систем при разработке web-приложений	Не умеет	Частично освоенные умения разрабатывать концептуальную, логическую и физическую модели web-приложений; определять необходимый инструментарий по уровню обеспечения информационных систем при разработке web-приложений	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения разрабатывать концептуальную, логическую и физическую модели web-приложений; определять необходимый инструментарий по уровню обеспечения информационных систем при разработке web-приложений	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения разрабатывать концептуальную, логическую и физическую модели web-приложений; определять необходимый инструментарий по уровню обеспечения информационных систем при разработке web-приложений	Успешно сформированные, самостоятельно и осознанно выполняемые умения разрабатывать концептуальную, логическую и физическую модели web-приложений; определять необходимый инструментарий по уровню обеспечения информационных систем при разработке web-приложений
Владеть: - навыками проектирования концептуальной модели БД и ее модификации; - навыками разработки модели web-приложения на основе проектного решения; - навыками подготовки и демонстрации прототипа	Не владеет	Фрагментарно сформированные навыки проектирования концептуальной модели БД и ее модификации; разработки модели web-приложения на основе проектного решения; навыками подготовки и	В целом сформированные, но выполняемые на элементарном уровне навыки проектирования концептуальной модели БД и ее модификации; разработки модели web-приложения на основе проектного решения; навыками подготовки и	Сформированные, но не устойчивые в сложных ситуациях навыки проектирования концептуальной модели БД и ее модификации; разработки модели web-приложения на основе проектного решения; навыками подготовки и демонстрации	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки проектирования концептуальной модели БД и ее модификации; разработки модели web-приложения на основе проектного решения; навыками

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
решения		демонстрации прототипа решения	демонстрации прототипа решения	прототипа решения	подготовки и демонстрации прототипа решения
ПК-2. Способен проектировать web-приложения **ПК-2.3.** Проектирует пользовательский интерфейс с учетом тенденций развития сквозных цифровых технологий и с помощью цифровых инструментов					
Знать: - принципы разработки пользовательского интерфейса web-приложения	Не знает	Фрагментарные знания о принципах разработки пользовательского интерфейса web-приложения	Общие, но не структурированные знания о принципах разработки пользовательского интерфейса web-приложения	Сформированные, но содержащие отдельные пробелы знания о принципах разработки пользовательского интерфейса web-приложения	Сформированные систематизированные прочные знания о принципах разработки пользовательского интерфейса web-приложения
Уметь: - проектировать пользовательский интерфейс	Не умеет	Частично освоенные умения проектировать пользовательский интерфейс	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения проектировать пользовательский интерфейс	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения проектировать пользовательский интерфейс	Успешно сформированные, самостоятельно и осознанно выполняемые умения проектировать пользовательский интерфейс
Владеть: - навыками проектирования и разработки пользовательского интерфейса web-приложения	Не владеет	Фрагментарно сформированные навыки проектирования и разработки пользовательского интерфейса web-приложения	В целом сформированные, но выполняемые на элементарном уровне навыки проектирования и разработки пользовательского интерфейса web-приложения	Сформированные, но не устойчивые в сложных ситуациях навыки проектирования и разработки пользовательского интерфейса web-приложения	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки проектирования и разработки пользовательского интерфейса web-приложения
ПК-2. Способен проектировать web-приложения **ПК-2.4.** Проводит анализ современных программно-технических средств, специализированных цифровых платформ и облачных решений, сквозных цифровых технологий и отраслевых решений на их основе с целью выбора средств разработки web-приложений					
Знать: -методы анализа рынка программно-	Не знает	Фрагментарные знания о методах анализа рынка	Общие, но не структурированные знания о методах анализа	Сформированные, но содержащие отдельные пробелы знания о	Сформированные систематизированные прочные знания о методах

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
технических средств, информационных продуктов и услуг; - основные источники профессионально-ориентированной информации, предназначенной для проведения анализа рынка программно-технических средств		программно-технических средств, информационных продуктов и услуг, а также об основных источниках профессионально-ориентированной информации, предназначенной для проведения анализа рынка программно-технических средств	рынка программно-технических средств, информационных продуктов и услуг, а также об основных источниках профессионально-ориентированной информации, предназначенной для проведения анализа рынка программно-технических средств	методах анализа рынка программно-технических средств, информационных продуктов и услуг, а также об основных источниках профессионально-ориентированной информации, предназначенной для проведения анализа рынка программно-технических средств	анализа рынка программно-технических средств, информационных продуктов и услуг, а также об основных источниках профессионально-ориентированной информации, предназначенной для проведения анализа рынка программно-технических средств
Уметь: - составлять аналитические обзоры рынка программно-технических средств, информационных продуктов и услуг; - проводить сравнительный анализ программно-технических средств и информационных продуктов в соответствии с полученным заданием;	Не умеет	Частично освоенные умения составлять аналитические обзоры рынка программно-технических средств, информационных продуктов и услуг; проводить сравнительный анализ программно-технических средств и информационных продуктов в соответствии с полученным заданием	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения составлять аналитические обзоры рынка программно-технических средств, информационных продуктов и услуг; проводить сравнительный анализ программно-технических средств и информационных продуктов в соответствии с полученным заданием	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения составлять аналитические обзоры рынка программно-технических средств, информационных продуктов и услуг; проводить сравнительный анализ программно-технических средств и информационных продуктов в соответствии с полученным заданием	Успешно сформированные, самостоятельно и осознанно выполняемые умения составлять аналитические обзоры рынка программно-технических средств, информационных продуктов и услуг; проводить сравнительный анализ программно-технических средств и информационных продуктов в соответствии с полученным заданием
Владеть: - навыками работы с программно-технологическими средствами (поисковыми машинами, справочно-	Не владеет	Фрагментарно сформированные навыки работы с программно-технологическими средствами (поисковыми	В целом сформированные, но выполняемые на элементарном уровне навыки работы с программно-технологическим	Сформированные, но не устойчивые в сложных ситуациях навыки работы с программно-технологическими средствами (поисковыми	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки работы с программно-технологическими

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
правовыми и экспертными системами, открытыми информационными ресурсами и т.п.) для выполнения задач исследовательского и аналитического характера;		машинами, справочно-правовыми и экспертными системами, открытыми информационными ресурсами и т.п.) для выполнения задач исследовательского и аналитического характера	и средствами (поисковыми машинами, справочно-правовыми и экспертными системами, открытыми информационными ресурсами и т.п.) для выполнения задач исследовательского и аналитического характера	машинами, справочно-правовыми и экспертными системами, открытыми информационными ресурсами и т.п.) для выполнения задач исследовательского и аналитического характера	средствами (поисковыми машинами, справочно-правовыми и экспертными системами, открытыми информационными ресурсами и т.п.) для выполнения задач исследовательского и аналитического характера
ПК-3. Способен разрабатывать web-приложения **ПК-3.1**. Осуществляет разработку программного кода и компонентов web-приложения, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем					
Знать: - язык определения и манипулирования данными; - основные этапы построения web-приложений, понятие кроссбраузерности, кроссплатформенности и адаптивной верстки; - термины, синтаксис, управляющие конструкции и типы данных скриптового языка общего назначения PHP; - основы JavaScript; - инструменты и принципы разработки web-приложения	Не знает	Фрагментарные знания языка определения и манипулирования данными, основных этапов построения web-приложений, понятия кроссбраузерности, кроссплатформенности и адаптивной верстки; терминов, синтаксиса, управляющих конструкций и типов данных скриптового языка общего назначения PHP; основах JavaScript; инструментах и принципах разработки web-приложения	Общие, но не структурированные знания языка определения и манипулирования данными, основных этапов построения web-приложений, понятия кроссбраузерности, кроссплатформенности и адаптивной верстки; терминов, синтаксиса, управляющих конструкций и типов данных скриптового языка общего назначения PHP; основах JavaScript; инструментах и принципах разработки web-приложения	Сформированные, но содержащие отдельные пробелы знания языка определения и манипулирования данными, основных этапов построения web-приложений, понятия кроссбраузерности, кроссплатформенности и адаптивной верстки; терминов, синтаксиса, управляющих конструкций и типов данных скриптового языка общего назначения PHP; основах JavaScript; инструментах и принципах разработки web-приложения	Сформированные систематизированные прочные знания языка определения и манипулирования данными, основных этапов построения web-приложений, понятия кроссбраузерности, кроссплатформенности и адаптивной верстки; терминов, синтаксиса, управляющих конструкций и типов данных скриптового языка общего назначения PHP; основах JavaScript; инструментах и принципах разработки web-приложения
Уметь: - обеспечить	Не умеет	Частично освоенные	В целом освоенные, но не	В целом сформированные	Успешно сформированные,

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
передачу информации между приложением и веб-сервером; - осуществлять подключение к серверу и выбор базы данных; - разрабатывать динамические приложения на языке JavaScript и PHP для решения прикладных задач		умения обеспечивать передачу информации между приложением и веб-сервером; осуществлять подключение к серверу и выбор базы данных; разрабатывать динамические приложения на языке JavaScript и PHP для решения прикладных задач	подкрепляемые знаниями и самостоятельностью выполнения умения обеспечивать передачу информации между приложением и веб-сервером; осуществлять подключение к серверу и выбор базы данных; разрабатывать динамические приложения на языке JavaScript и PHP для решения прикладных задач	и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения обеспечивать передачу информации между приложением и веб-сервером; осуществлять подключение к серверу и выбор базы данных; разрабатывать динамические приложения на языке JavaScript и PHP для решения прикладных задач	самостоятельно и осознанно выполняемые умения обеспечивать передачу информации между приложением и веб-сервером; осуществлять подключение к серверу и выбор базы данных; разрабатывать динамические приложения на языке JavaScript и PHP для решения прикладных задач
Владеть: - навыками создания приложения с помощью языков программирования (HTML, JavaScript, PHP, CSS); - навыками создания сценариев с помощью операций и управляющих конструкций PHP; - навыками работы с различными формами в PHP; - навыками работы с сессиями и cookie в PHP	Не владеет	Фрагментарно сформированные навыки создания приложения с помощью языков программирования (HTML, JavaScript, PHP, CSS); создания сценариев с помощью операций и управляющих конструкций PHP; работы с различными формами в PHP; работы с сессиями и cookie в PHP навыками осуществления передачи данных через HTML-формы	В целом сформированные, но выполняемые на элементарном уровне навыки создания приложения с помощью языков программирования (HTML, JavaScript, PHP, CSS); создания сценариев с помощью операций и управляющих конструкций PHP; работы с различными формами в PHP; работы с сессиями и cookie в PHP навыками осуществления передачи данных через HTML-формы	Сформированные, но не устойчивые в сложных ситуациях навыки создания приложения с помощью языков программирования (HTML, JavaScript, PHP, CSS); создания сценариев с помощью операций и управляющих конструкций PHP; работы с различными формами в PHP; работы с сессиями и cookie в PHP навыками осуществления передачи данных через HTML-формы	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки создания приложения с помощью языков программирования (HTML, JavaScript, PHP, CSS); создания сценариев с помощью операций и управляющих конструкций PHP; работы с различными формами в PHP; работы с сессиями и cookie в PHP навыками осуществления передачи данных через HTML-формы

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
- навыками осуществления передачи данных через HTML-формы					
ПК-3. Способен разрабатывать web-приложения **ПК-3.2.** Обеспечивает интеграцию программных модулей и компонентов программного обеспечения, в том числе используя отраслевые облачные решения и цифровые платформы					
Знать: - назначение тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования	Не знает	Фрагментарные знания назначений тестирования, видов и способов тестирования баз данных и web-приложений, процедур тестирования	Общие, но не структурированные знания назначений тестирования, видов и способов тестирования баз данных и web-приложений, процедур тестирования	Сформированные, но содержащие отдельные пробелы знания назначений тестирования, видов и способов тестирования баз данных и web-приложений, процедур тестирования	Сформированные систематизированные прочные знания назначений тестирования, видов и способов тестирования баз данных и web-приложений, процедур тестирования
Уметь: - проводить процедуру тестирования баз данных и web-приложений	Не умеет	Частично освоенные умения проводить процедуру тестирования баз данных и web-приложений	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения проводить процедуру тестирования баз данных и web-приложений	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения проводить процедуру тестирования баз данных и web-приложений	Успешно сформированные, самостоятельно и осознанно выполняемые умения проводить процедуру тестирования баз данных и web-приложений
Владеть: - навыками протоколирования результатов тестирования	Не владеет	Фрагментарно сформированные навыки протоколирования результатов тестирования	В целом сформированные, но выполняемые на элементарном уровне навыки протоколирования результатов тестирования	Сформированные, но не устойчивые в сложных ситуациях навыки протоколирования результатов тестирования	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки протоколирования результатов тестирования
ПК-5. Способен обеспечивать качество работы web-приложения **ПК-5.2.** Проводит юзабилити-аудит web-приложения					
Знать: - назначение юзабилити-аудита web-приложения	Не знает	Фрагментарные знания назначения юзабилити-аудита web-приложения	Общие, но не структурированные знания назначения юзабилити-аудита web-приложения	Сформированные, но содержащие отдельные пробелы знания назначения юзабилити-аудита web-	Сформированные систематизированные прочные знания назначения юзабилити-аудита web-приложения

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
				приложения	
Уметь: - проводить юзабилити-аудит web-приложения	Не умеет	Частично освоенные умения проводить юзабилити-аудит web-приложения	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения проводить юзабилити-аудит web-приложения	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения проводить юзабилити-аудит web-приложения	Успешно сформированные, самостоятельно и осознанно выполняемые умения проводить юзабилити-аудит web-приложения
Владеть: навыками протоколирования результатов тестирования	Не владеет	Фрагментарно сформированные навыки протоколирования результатов тестирования	В целом сформированные, но выполняемые на элементарном уровне навыки протоколирования результатов тестирования	Сформированные, но не устойчивые в сложных ситуациях навыки протоколирования результатов тестирования	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки протоколирования результатов тестирования
ПК-5. Способен обеспечивать качество работы web-приложения **ПК-5.3.** Осуществляет работы по интеграционному тестированию web-приложения с внешним цифровым окружением					
Знать: - назначение тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования	Не знает	Фрагментарные знания назначения тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования	Общие, но не структурированные знания назначения тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования	Сформированные, но содержащие отдельные пробелы знания назначения тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования	Сформированные систематизированные прочные знания назначения тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования
Уметь: - проводить процедуру тестирования баз данных и web-приложений	Не умеет	Частично освоенные умения проводить процедуру тестирования баз данных и web-приложений	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения проводить процедуру тестирования баз данных и web-приложений	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения проводить процедуру тестирования баз данных и web-	Успешно сформированные, самостоятельно и осознанно выполняемые умения проводить процедуру тестирования баз данных и web-приложений

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	**Критерии оценивания результатов обучения**				
	1	2	3	4	5
				приложений	
Владеть: навыками протоколирования результатов тестирования	Не владеет	Фрагментарно сформированные навыки протоколирования результатов тестирования	В целом сформированные, но выполняемые на элементарном уровне навыки протоколирования результатов тестирования	Сформированные, но не устойчивые в сложных ситуациях навыки протоколирования результатов тестирования	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки протоколирования результатов тестирования

КРИТЕРИИ ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

При выставлении итоговой оценки за дисциплину необходимо учитывать результаты текущего контроля и критерии оценивания сформированности компетенций.

Оценка «**отлично**» выставляется обучающемуся, если:

- при ответе на теоретический вопрос были продемонстрированы глубокие, систематизированные и прочные знания (в соответствии с критериями сформированности компетенций), умения логично и обоснованно излагать материал, быстро ориентироваться в дополнительных вопросах и давать полные ответы;
- задание в малых группах выполнено безошибочно и в полном соответствии с требованиями на оценку «отлично», обучающийся в работе продемонстрировал устойчивые навыки работы и осознанно исполняемые действия по построению web-приложений, составляет код программы и воспроизводит взаимодействие PHP, JavaScript, HTML и MySQL, составленный код программы работает без сбоев, организована проверка на ошибки при вводе данных, предусмотрена защита данных;
- обучающийся участвовал во всех видах текущего контроля и получал по ним максимальные баллы (допустимо отдельные отметки «хорошо»).

Все вышеперечисленное говорит о сформированности запланированных компетенций на высоком уровне.

Оценка «**хорошо**» выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано хорошее, но недостаточно полное изложение материала, с отдельными пробелами в знаниях (возможно, незначительными неточностями в употреблении терминов), логичное изложение материала, но недостаточно быстрая ориентация в нем при ответе на дополнительные вопросы;
- индивидуальное проектное задание выполнено на оценку «отлично» или «хорошо» (с незначительными замечаниями и/или ошибками), обучающийся продемонстрировал хорошие навыки работы и осознанно исполняемые действия по построению web-приложений, составляет код программы и воспроизводит взаимодействие PHP, JavaScript, HTML и MySQL, составленный код программы работает без сбоев, организована проверка на ошибки при вводе данных, предусмотрена защита данных;

- обучающийся участвовал во всех видах текущего контроля и получал по ним в основном отметки «хорошо».

  Все вышеперечисленное говорит о сформированности запланированных компетенций на хорошем уровне.

Оценка «**удовлетворительно**» выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано поверхностное владение материалом и фрагментарные знания, отсутствие стройного и логично выстроенного ответа, затруднение в употреблении терминов и приведении примеров, плохая ориентация в материале при ответе на дополнительные вопросы;
- индивидуальное проектное задание выполнено на оценку «хорошо» или «удовлетворительно» (с ошибками и замечаниями), с отклонениями от требований, обучающийся затруднялся в объяснении применяемых средств построения web-приложений, составляет код программы и воспроизводит взаимодействие PHP, JavaScript, HTML и MySQL, составленный код программы работает со сбоями, не организована проверка на ошибки при вводе данных, не предусмотрена защита данных;
- обучающийся участвовал не во всех мероприятиях текущего контроля и получал по ним в основном удовлетворительные отметки.

  Все вышеперечисленное говорит о сформированности запланированных компетенций на низком уровне.

Оценка «**неудовлетворительно**» выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано незнание значительного объема материала, «фрагментарность имеющихся знаний, отсутствие стройного и логично выстроенного ответа, затруднение в употреблении терминов и приведении примеров, неумение выделить главное, сделать выводы и обобщения, неспособность отвечать на дополнительные вопросы;
- индивидуальное проектное задание выполнено на оценку «удовлетворительно» со значительными отклонениями от требований, с грубыми ошибками и множеством замечаний, обучающийся продемонстрировал частично сформированные навыки работы, затрудняется в объяснении применяемых средств по построению web-приложений, не составляет код программы и не воспроизводит взаимодействие PHP, JavaScript, HTML и MySQL, составленный код программы не работает, не организована проверка на ошибки при вводе данных, не предусмотрена защита данных;
- обучающийся участвовал не во всех мероприятиях текущего контроля и получал по ним в основном удовлетворительные отметки.

  Все вышеперечисленное говорит о частичной сформированности запланированных компетенций.

5. ДИАГНОСТИЧЕСКИЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ УРОВЕНЬ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Компетенция ПК-1. Способен выявлять информационные потребности пользователей и составлять техническое задание на разработку web-приложения

Обучающийся знает:

- методику проведения обследования организации и выявления информационных потребностей пользователей;
- виды требований, предъявляемых к web-приложения;
- виды информации, необходимой для формализации требований заказчика для разработки; базы данных и/или web-приложения, методы сбора и анализа детальной информации;
- структуру и порядок построения технического задания на разработку web-приложения

Обучающийся умеет:

- проводить предпроектное и информационное обследование организации;
- выявлять информационные потребности пользователей к web-приложениям;
- выявлять информационные потребности предприятия в СУБД,
- формировать техническое задание к разрабатываемому web- приложению;

Обучающийся владеет:

- навыками сбора и анализа информации о предприятии;
- навыками подготовки отчета о проведенном предпроектном обследовании организации;
- навыками документирования требований заказчика;
- навыками составления технического задания на разработку web-приложения и проектов.

ЗАДАНИЯ ЗАКРЫТОГО ТИПА

1. Что представляет собой предпроектное обследование организации?
   A) Разработка концепции нового продукта
   B) Оценка текущего состояния бизнес-процессов и инфраструктуры организации
   C) Анализ финансовых показателей компании
   D) Изучение потребностей рынка
   Ответ: B) Оценка текущего состояния бизнес-процессов и инфраструктуры организации

2. Для чего необходимо проводить информационное обследование организации?
   A) Для разработки шаблонов документов
   B) Для обучения персонала
   C) Для выявления проблем и уязвимостей
   D) Для организации корпоративных мероприятий
   Ответ: C) Для выявления проблем и уязвимостей

3. Какие методы могут использоваться при проведении предпроектного обследования?
   A) Опрос сотрудников
   B) Анализ финансовых отчетов
   C) Интервью с клиентами
   D) Все вышеперечисленные
   Ответ: D) Все вышеперечисленные

4. Какие данные можно использовать при проведении информационного обследования организации?
   A) Финансовые отчеты
   B) Отзывы клиентов
   C) Статистические данные
   D) Все вышеперечисленные
   Ответ: D) Все вышеперечисленные

5. Что необходимо делать после завершения предпроектного и информационного обследования организации?
   A) Разработать новую стратегию развития
   B) Провести анализ конкурентов
   C) Подготовить отчет о результатах обследования
   D) Начать реализацию проекта
   Ответ: C) Подготовить отчет о результатах обследования

6. Какие критерии влияют на выбор пользователя web-приложения для получения информации?
   A) Удобство использования
   B) Качество контента
   C) Скорость загрузки
   D) Все вышеперечисленное
   Правильный ответ: D) Все вышеперечисленное

7. Для каких целей пользователи обычно обращаются к web-приложениям?
   A) Поиск информации
   B) Общение с другими пользователями
   C) Покупки онлайн
   D) Все вышеперечисленное
   Правильный ответ: D) Все вышеперечисленное

8. На каких устройствах пользователи обычно используют web-приложения для получения информации?
   A) ПК
   B) Смартфоны
   C) Планшеты
   D) Все вышеперечисленное
   Правильный ответ: D) Все вышеперечисленное

9. Какие функции web-приложений могут быть полезны для пользователей?
   A) Персонализированные рекомендации
   B) Уведомления о новостях
   C) Возможность сохранения контента для просмотра офлайн
   D) Все вышеперечисленное
   Правильный ответ: D) Все вышеперечисленное

10. Какие факторы могут повлиять на удовлетворенность пользователей информацией, полученной из web-приложений?
    A) Актуальность контента
    B) Легкость навигации
    C) Качество изображений
    D) Все вышеперечисленное
    Правильный ответ: D) Все вышеперечисленное

ЗАДАНИЯ ОТКРЫТОГО ТИПА

11. Задание: Какие основные цели проведения предпроектного обследования организации?
Ответ: Основные цели проведения предпроектного обследования организации включают определение текущего состояния бизнес-процессов, выявление проблем и недостатков, выработка стратегии развития и определение возможных направлений улучшения работы организации.

12. Задание: Какие этапы включает предпроектное обследование организации?

Ответ: Проведение анализа текущего состояния организации, выявление потенциальных проблем и узких мест в бизнес-процессах, разработка стратегии развития, определение необходимых изменений и подготовка рекомендаций для оптимизации работы организации.

13. Задание: Какие методы и инструменты могут использоваться при информационном обследовании организации?

Ответ: Методы интервьюирования сотрудников и руководителей, анализ статистических данных и отчетов, SWOT-анализ для выявления сильных и слабых сторон организации, а также возможностей и угроз.

14. Задание: Какие выводы могут быть сделаны на основе информационного обследования организации?

Ответ: На основе информационного обследования можно сделать выводы о текущем состоянии организации, выявить ключевые проблемы и узкие места, определить потенциальные возможности для развития и роста, а также сформулировать рекомендации по улучшению работы организации.

15. Задание: Каким образом результаты предпроектного и информационного обследования могут помочь организации?

Ответ: Результаты предпроектного и информационного обследования могут помочь организации определить стратегию развития, выработать план действий для улучшения бизнес-процессов, увеличить эффективность работы и конкурентоспособность, а также повысить уровень удовлетворенности клиентов и сотрудников.

16. Задание: Определите основные информационные потребности пользователей, которые могут возникнуть при использовании веб-приложений для заказа и доставки еды.

Ответ: К основным информационным потребностям пользователей, которые возникают при использовании веб-приложений для заказа и доставки еды можно отнести:

1. Информация о доступных блюдах и их составе.
2. Цены на блюда и напитки.
3. Возможность заказа еды на дом или в офис.
4. Время доставки и стоимость доставки.
5. Возможность оплаты заказа онлайн.
6. Отзывы других клиентов о блюдах и сервисе.
7. Контактный телефон и адрес электронной почты ресторана.

17. Задание: Проанализируйте информационные потребности пользователя при использовании веб-приложения для поиска и бронирования отелей.

Ответ: К основным информационным потребностям пользователей при использовании веб-приложения для поиска и бронирования отелей можно отнести:

1. Расположение отеля на карте.
2. Фотографии номеров и территории отеля.
3. Цены на номера и скидки.
4. Наличие свободных номеров.
5. Отзывы других гостей.
6. Услуги, предоставляемые отелем.
7. Контактная информация отеля.

18. Задание: Разработайте опрос для определения информационных потребностей пользователей при использовании веб-приложения для покупки авиабилетов.

Ответ: опрос для определения информационных потребностей пользователей при использовании веб-приложения для покупки авиабилетов может содержать:

1. Как часто вы планируете покупать авиабилеты?
2. Какие направления вас интересуют?
3. Важна ли для вас стоимость авиабилетов?
4. Нужно ли вам удобное мобильное приложение для покупки авиабилетов?
5. Важно ли для вас наличие онлайн-регистрации на рейс?
6. Нужно ли вам получать уведомления об изменении расписания рейсов?
7. Нужно ли вам иметь возможность бронировать отели через то же приложение, в котором вы покупаете авиабилеты?

19. Задание: Оцените информационные потребности пользователей при использовании веб-приложения для изучения иностранных языков и определите возможные способы их удовлетворения.

Ответ: К основным информационным потребностям пользователей при использовании веб-приложения для изучения иностранных языков и определите возможные способы их удовлетворения можно отнести:

1. Доступ к большому количеству материалов для изучения языка (тексты, аудио, видео).
2. Возможность выбора уровня сложности материала.
3. Наличие тестов и упражнений для закрепления знаний.
4. Возможность общения с носителями языка.
5. Доступ к словарю и грамматическим правилам.
6. Возможность отслеживания прогресса в изучении языка.
7. Наличие мобильной версии приложения.

20. Задание: На основе проведённого анализа трех web-приложений, которые решают аналогичные задачи, были выделены следующие сильные и слабые стороны, а также возможности для улучшения:

Приложение	Сильные стороны	Слабые стороны	Возможности для улучшения
Приложение 1	Широкий выбор обучающих материалов	Отсутствие интерактивных элементов	Добавление интерактивных элементов, таких как тесты и задания
Приложение 2	Доступ к обучающим материалам по различным темам	Ограниченные возможности для общения с преподавателями	Расширение возможностей для общения с преподавателями и другими пользователями
Приложение 3	Современный и интуитивно понятный интерфейс	Ограниченный выбор обучающих материалов	Расширение выбора обучающих материалов и добавление новых функций

На основе анализа предложите идеи для улучшения существующего приложения или создания нового.

Ответ:

Для улучшения существующего приложения или создания нового можно предложить:

- добавить интерактивные элементы, такие как тесты и задания, для повышения вовлечённости пользователей;
- расширить возможности для общения с преподавателями и другими пользователями, чтобы создать более активное сообщество;
- расширить выбор обучающих материалов и добавить новые функции, чтобы удовлетворить потребности различных пользователей.

21. Задание: Опишите, какие основные функции для обеспечения требований по функциональности может включать в себя web-приложение для управления проектами.

Ответ: Список основных функций, которые должны быть включены в web-приложение для управления проектами, может включать в себя:

- создание и управление проектами;
- распределение задач между участниками проекта;
- отслеживание прогресса выполнения задач;
- обмен информацией и документами между участниками проекта;
- анализ результатов проекта и принятие решений на основе данных.

Каждая из этих функций может помочь пользователям в управлении проектами, обеспечивая эффективное планирование, координацию и контроль над выполнением задач.

22. Задание: Какие риски могут возникнуть при разработке web-приложения и какие меры для их минимизации можно предпринять?

Ответ: При разработке web-приложения могут возникнуть следующие риски:

- технические риски, связанные с ошибками в коде, сбоями в работе системы и т. д.;
- риски безопасности, связанные с утечкой данных, несанкционированным доступом и т. п.;
- финансовые риски, связанные с перерасходом бюджета, изменением рыночных условий и т. д.

Для минимизации этих рисков следующие меры:

- провести тщательное тестирование приложения перед запуском;
- обеспечить защиту персональных данных пользователей;
- регулярно обновлять систему безопасности;
- контролировать расходы и следить за изменениями рыночных условий.

23. Какие функциональные требования должно удовлетворять web-приложение?

Ответ: Web-приложение должно обеспечивать возможность регистрации и авторизации пользователей, выполнения определенных действий с данными, обработки запросов и отображения информации в удобной форме.

24. Какие интерфейсные требования должно соответствовать web-приложение?

Ответ: Web-приложение должно иметь интуитивно понятный и привлекательный пользовательский интерфейс, который удовлетворяет требованиям юзабилити и обеспечивает удобство использования.

25. Каковы требования к безопасности web-приложения?

Ответ: Web-приложение должно обеспечивать защиту данных пользователей, обеспечивая безопасную передачу информации по защищенному соединению (HTTPS), а также предотвращать возможные атаки, такие как CSRF или SQL инъекции.

ПК-2. Способен проектировать web-приложения

Знать:

- процесс и средства разработки web-приложений;
- факторы сложности разработки web-приложений, технологии, подходы и принципы к исследованию и созданию информационных систем, эргономические требования к ним
- содержание этапов разработки и принципы организации проектирования web-приложений, методики сравнения проектных решений;
- информационную модель данных и ее структуру, логические модели баз данных и их типы, формы нормализации данных в базе;
- принципы разработки пользовательского интерфейса web-приложения;

- методы анализа рынка программно-технических средств, информационных продуктов и услуг;
- основные источники профессионально-ориентированной информации, предназначенной для проведения анализа рынка программно-технических средств;

Уметь:

- определять факторы сложности разработки программных систем, выбирать адекватные технологии для их проектирования
- разрабатывать концептуальную, логическую и физическую модели web-приложений;
- определять необходимый инструментарий по уровню обеспечения информационных систем при разработке web-приложений;
- проектировать пользовательский интерфейс;
- составлять аналитические обзоры рынка программно-технических средств, информационных продуктов и услуг;
- проводить сравнительный анализ программно-технических средств и информационных продуктов в соответствии с полученным заданием;

Владеть:

- навыками разработки архитектуру web-приложения с учетом внедрения сквозных цифровых технологий и передовых ИТ-решений прикладной сферы
- навыками проектирования концептуальной модели БД и ее модификации;
- навыками разработки модели web-приложения на основе проектного решения;
- навыками подготовки и демонстрации прототипа решения;
- навыками проектирования и разработки пользовательского интерфейса web-приложения;
- навыками работы с программно-технологическими средствами (поисковыми машинами, справочно-правовыми и экспертными системами, открытыми информационными ресурсами и т.п.) для выполнения задач исследовательского и аналитического характера

ЗАДАНИЯ ЗАКРЫТОГО ТИПА

1. Что такое MVC в контексте веб-разработки?

a) Модульный веб-компонент
b) Модуль контроля версий
c) Модель-вид-контроллер
d) Многоверсионный контейнер
Правильный ответ: c) Модель-вид-контроллер

2. 2. Какая из следующих технологий является клиентской стороной веб-приложения?

a) Node.js
b) Express.js
c) MongoDB
d) React.js
Правильный ответ: d) React.js

3. Какая роль отвечает за представление данных в MVC архитектуре?

a) Модель
b) Вид
c) Контроллер
d) Сервер
Правильный ответ: b) Вид

4. Что такое RESTful API?

a) Программный интерфейс для доступа к базе данных
b) Модуль управления сетью
c) Архитектурный стиль для построения веб-сервисов
d) Контроллер обработки запросов

Правильный ответ: c) Архитектурный стиль для построения веб-сервисов

5. Какие HTTP методы используются для чтения данных в RESTful API?

a) GET
b) POST
c) PUT
d) DELETE
Правильный ответ: a) GET

6. Какая роль отвечает за бизнес-логику в MVC архитектуре?

a) Модель
b) Вид
c) Контроллер
d) Хранилище данных
Правильный ответ: a) Модель

7. Какая база данных чаще всего используется в web-приложениях?

a) SQL Server
b) NoSQL
c) SQLite
d) PostgreSQL
Правильный ответ: b) NoSQL

8. Какой протокол обмена данными часто используется для взаимодействия между клиентом и сервером в современных веб-приложениях?

a) FTP
b) HTTP
c) SMTP
d) TCP
Правильный ответ: b) HTTP

9. Какая технология позволяет реализовать реактивные интерфейсы в веб-приложениях?

a) AJAX
b) WebSocket
c) GraphQL
d) REST API
Правильный ответ: b) WebSocket

10. Какой тип архитектуры обычно используется для создания масштабируемых веб-приложений?

a) Монолитная архитектура
b) Клиент-серверная архитектура
c) Микросервисная архитектура
d) Централизованная архитектура
Правильный ответ: c) Микросервисная архитектура

ЗАДАНИЯ ОТКРЫТОГО ТИПА

11. Опишите процесс разработки архитектуры веб-приложения.

Ответ: Разработка архитектуры веб-приложения обычно включает в себя несколько этапов. На первом этапе проводится анализ требований, в ходе которого определяются основные функциональные и нефункциональные требования к приложению. Затем происходит проектирование, на этом этапе решается, какие технологии, инструменты и архитектурные шаблоны будут использоваться для реализации приложения. После этого разрабатывается

архитектурный дизайн, включающий в себя построение модели данных, определение компонентов приложения и их взаимосвязей. Наконец, на этапе реализации кодируются компоненты приложения согласно разработанной архитектуре.

12. Какие основные преимущества имеет микросервисная архитектура перед монолитной?

Ответ: Микросервисная архитектура предлагает ряд преимуществ перед монолитной архитектурой. Во-первых, микросервисы могут быть независимо развернуты и масштабированы, что облегчает поддержку и управление приложением. Во-вторых, они обеспечивают гибкость и модульность, позволяя быстрее разрабатывать, тестировать и развертывать новые функции. Кроме того, микросервисы позволяют использовать различные технологии и языки программирования для каждого сервиса, что способствует выбору наилучшего инструмента для конкретной задачи.

13. Что такое RESTful API и почему его часто используют в веб-приложениях?

Ответ: RESTful API - это архитектурный стиль для построения веб-сервисов, который основан на принципах REST (Representational State Transfer). Он позволяет организовать взаимодействие между клиентом и сервером с помощью стандартных HTTP методов, таких как GET, POST, PUT и DELETE, и обмен данными в формате JSON или XML. RESTful API широко используется в веб-приложениях из-за своей простоты, гибкости и масштабируемости. Он упрощает разработку и интеграцию различных компонентов приложения, а также обеспечивает четкую структуру запросов и ответов.

14. Какие основные компоненты входят в MVC архитектуру?

Ответ: MVC (Model-View-Controller) - это архитектурный шаблон, который разделяет приложение на три основных компонента: модель (Model), представление (View) и контроллер (Controller). Модель отвечает за бизнес-логику и работу с данными приложения, представление отображает данные пользователю и обрабатывает пользовательский ввод, а контроллер управляет взаимодействием между моделью и представлением, а также обрабатывает запросы от пользователя.

15. Что такое SPA (Single Page Application) и какие преимущества оно предлагает?

Ответ: SPA (Single Page Application) - это тип веб-приложения, которое загружает единственную HTML страницу и динамически обновляет ее содержимое без перезагрузки страницы при взаимодействии пользователя. Основные преимущества SPA включают более быструю загрузку и отзывчивость интерфейса, так как большая часть ресурсов загружается один раз при запуске приложения, а также улучшенный пользовательский опыт благодаря плавным переходам и анимациям.

16. Какие методы аутентификации могут быть использованы в веб-приложениях?

Ответ: В веб-приложениях могут использоваться различные методы аутентификации, включая аутентификацию на основе токенов (например, JWT), аутентификацию с использованием сессий, аутентификацию через социальные сети, а также аутентификацию с помощью сторонних сервисов (например, OAuth). Выбор метода зависит от требований к безопасности, удобства использования и функциональности приложения.

17. Какие инструменты можно использовать для тестирования производительности веб-приложений?

Ответ: Для тестирования производительности веб-приложений можно использовать различные инструменты, включая Apache JMeter, LoadRunner, Gatling, Siege и другие.

18. Задание: Разработайте модель web-приложения для онлайн-магазина. Опишите структуру базы данных, включающую таблицы для хранения информации о продуктах,

пользователях, заказах и платежах. Предложите схему взаимодействия клиентской и серверной частей приложения, включая функциональные возможности каждой части.

Решение: Для модели web-приложения онлайн-магазина мы можем разработать следующую структуру базы данных:

Таблица «Продукты» с полями: id продукта, название, описание, цена, количество на складе и т.д.

Таблица «Пользователи» с полями: id пользователя, имя, адрес электронной почты, хэш пароля и т.д.

Таблица «Заказы» с полями: id заказа, id пользователя, дата заказа, статус заказа и т.д.

Таблица «Платежи» с полями: id платежа, id заказа, сумма, дата и т.д.

Взаимодействие клиентской и серверной частей приложения будет осуществляться с помощью RESTful API, где клиент будет отправлять запросы на добавление товаров в корзину, оформление заказа и оплату, а сервер будет обрабатывать эти запросы, взаимодействуя с базой данных.

19. Задание: Представьте модель web-приложения для социальной сети. Опишите структуру базы данных для хранения профилей пользователей, сообщений, фотографий, друзей и лайков.

Обсудите архитектурные решения для реализации новостной ленты, системы уведомлений и поиска пользователей.

Решение: Для модели web-приложения социальной сети мы можем разработать следующую структуру базы данных:

Таблица «Пользователи» с полями: id пользователя, имя, дата рождения, фото профиля и т.д.

Таблица «Сообщения» с полями: id сообщения, id отправителя, id получателя, текст сообщения, дата отправки и т.д.

Таблица «Друзья» с полями: id пользователя, id друга и т.д.

Таблица «Лайки» с полями: id пользователя, id записи, дата лайка и т.д.

Архитектурное решение будет включать в себя разделение приложения на клиентскую и серверную части, где клиентская часть будет отвечать за отображение профилей пользователей, новостной ленты и личных сообщений, а серверная часть будет обрабатывать запросы от клиента и взаимодействовать с базой данных.

20. Задание: Разработайте модель web-приложения для онлайн-обучения. Опишите сущности базы данных, такие как курсы, уроки, студенты и преподаватели. Предложите структуру для организации курсов, включая моделирование уровней доступа и рейтинговой системы для оценки успеваемости студентов.

Решение: Для модели web-приложения онлайн-обучения мы можем разработать следующую структуру базы данных:

Таблица «Курсы» с полями: id курса, название, описание, длительность и т.д.

Таблица «Уроки» с полями: id урока, id курса, название, описание, материалы и т.д.

Таблица «Студенты» с полями: id студента, имя, адрес электронной почты, курсы и т.д.

Таблица «Преподаватели» с полями: id преподавателя, имя, адрес электронной почты, преподаваемые курсы и т.д.

Архитектурное решение будет включать в себя создание интерфейса для студентов и преподавателей, где студенты смогут просматривать курсы, регистрироваться на них и проходить уроки, а преподаватели смогут добавлять новые уроки и оценивать успеваемость студентов.

21. Задание: Представьте модель web-приложения для финансового учета. Опишите базу данных для хранения информации о транзакциях, счетах пользователей, бюджетах и категориях расходов. Рассмотрите алгоритмы для автоматического распознавания и категоризации транзакций.

Решение: Для модели web-приложения финансового учета мы можем разработать следующую структуру базы данных:

Таблица «Транзакции» с полями: id транзакции, id пользователя, сумма, категория, дата и т.д.

Таблица «Счета» с полями: id счета, id пользователя, баланс, валюта и т.д.

Таблица «Категории расходов» с полями: id категории, название, описание и т.д.

Архитектурное решение включает разработку интерфейса для пользователей, позволяющего им добавлять новые транзакции, просматривать и анализировать свои расходы и доходы, а также управлять своими счетами и бюджетами.

22. Задание: Разработайте модель web-приложения для электронной медицинской карты. Обсудите структуру базы данных для хранения информации о пациентах, истории болезней, назначениях и результатах анализов. Рассмотрите механизмы обеспечения безопасности и конфиденциальности данных.

Решение: Для модели web-приложения электронной медицинской карты мы можем разработать следующую структуру базы данных:

Таблица «Пациенты» с полями: id пациента, имя, дата рождения, медицинская история и т.д.

Таблица «Диагнозы» с полями: id диагноза, id пациента, дата постановки, описание и т.д.

Таблица «Назначения» с полями: id назначения, id пациента, дата, описание, лекарства и т.д.

Архитектурное решение будет включать в себя создание защищенного интерфейса для врачей и медицинского персонала, где они смогут просматривать и редактировать медицинские записи пациентов.

23. Задание: Представьте модель web-приложения для планирования путешествий. Опишите структуру базы данных для хранения информации о турах, отелях, бронированиях и отзывах пользователей.

Решение: Для модели web-приложения для планирования путешествий нам необходимо разработать структуру базы данных, которая будет содержать информацию о турах, отелях, бронированиях и отзывах пользователей. Давайте опишем возможную структуру базы данных:

Таблица «Туры»:

tour_id: уникальный идентификатор тура
название: название тура
описание: описание тура
стоимость: стоимость тура
дата_начала: дата начала тура
дата_окончания: дата окончания тура
место_посещения: место, которое будет посещено в рамках тура
тип_тура: тип тура (например, экскурсионный, пляжный и т.д.)

Таблица «Отели»:

hotel_id: уникальный идентификатор отеля
название: название отеля
описание: описание отеля
адрес: адрес отеля
рейтинг: рейтинг отеля
цена_за_ночь: стоимость проживания за ночь

Таблица «Бронирования»:

booking_id: уникальный идентификатор бронирования
tour_id: идентификатор тура, на который осуществляется бронирование
hotel_id: идентификатор отеля, в котором осуществляется бронирование
дата_заселения: дата заселения в отель

дата_выселения: дата выселения из отеля
количество_гостей: количество гостей в брони
статус: статус бронирования (подтверждено, ожидает подтверждения и т.д.)
Таблица «Отзывы»:
review_id: уникальный идентификатор отзыва
user_id: идентификатор пользователя, оставившего отзыв
tour_id: идентификатор тура, к которому относится отзыв
отель_id: идентификатор отеля, к которому относится отзыв
оценка: оценка тура или отеля
текст_отзыва: текст отзыва

24. Рассмотрите способы интеграции с сервисами бронирования и картами для отображения местоположения для предметной области из предыдущего задания (задание 23):

Решение:

1. Интеграция с сервисами бронирования:

Мы можем интегрировать API сервисов бронирования, таких как Booking.com или Airbnb, чтобы пользователи могли просматривать доступные отели и совершать бронирование без покидания нашего приложения.

Для этого мы можем использовать асинхронные запросы к API сервисов бронирования для получения доступных вариантов и отображения их на странице приложения. После того как пользователь сделает выбор и заполнит данные для бронирования, мы можем отправить запрос на создание брони через API сервиса бронирования.

2. Интеграция с картами для отображения местоположения:

Мы можем использовать картографические сервисы, такие как Google Maps или Mapbox, для отображения местоположения туров и отелей на карте.

Для этого мы можем использовать JavaScript API картографического сервиса для создания интерактивной карты на странице приложения. Местоположение туров и отелей можно будет отобразить на карте с помощью соответствующих маркеров или областей. Пользователи смогут взаимодействовать с картой, выбирая интересующие их места или маршруты.

25. Задание: Разработайте модель web-приложения для онлайн-заказа еды. Обсудите структуру базы данных для хранения меню, заказов, клиентов и доставок. Предложите интерфейс для администраторов ресторанов для управления меню и заказами.

Для модели web-приложения онлайн-заказа еды нам необходимо разработать структуру базы данных, которая будет содержать информацию о меню, заказах, клиентах и доставках. Предложим также интерфейс для администраторов ресторанов для управления меню и заказами.

Вот предложенная структура базы данных:

1. Таблица "Меню":
item_id: уникальный идентификатор блюда
название: название блюда
описание: описание блюда
цена: цена блюда
2. Таблица «Заказы»:
order_id: уникальный идентификатор заказа
client_id: идентификатор клиента, сделавшего заказ
item_id: идентификатор блюда из меню
количество: количество порций блюда в заказе
статус: статус заказа (принят, готовится, доставляется и т.д.)
дата_заказа: дата и время размещения заказа

3. Таблица "Клиенты":
client_id: уникальный идентификатор клиента
имя: имя клиента
адрес: адрес доставки
телефон: контактный телефон клиента
email: адрес электронной почты клиента
4. Таблица «Доставки»:
delivery_id**: уникальный идентификатор доставки
order_id: идентификатор заказа
статус: статус доставки (в пути, доставлено и т.д.)
дата_доставки: дата и время доставки
Интерфейс для администраторов ресторанов может включать следующие функции:
1. Управление меню:
- Добавление, редактирование и удаление блюд из меню.
- Изменение цен на блюда.
- Добавление описания к блюдам.
2. Управление заказами:
- Просмотр списка всех заказов с информацией о клиенте, составе заказа и его статусе.
- Возможность изменения статуса заказа (например, отметка о том, что заказ готов к доставке).
- Просмотр истории заказов.
3. Управление доставками:
- Просмотр списка всех активных доставок с информацией о заказе, адресе доставки и статусе доставки.
- Возможность отметки о доставке заказа (подтверждение доставки).
- Просмотр истории доставок.
4. Управление клиентами:
- Просмотр списка всех клиентов с контактной информацией.
- Возможность добавления новых клиентов в случае оформления заказа от нового клиента.
- Возможность редактирования контактной информации клиентов.

Таким образом, создав данную структуру базы данных и предоставив администраторам ресторанов удобный интерфейс управления, мы можем эффективно организовать процесс онлайн-заказа еды.

ПК-3. Способен разрабатывать web-приложения

Знать:

- язык определения и манипулирования данными;
- основные этапы построения web-приложений, понятие кроссбраузерности, кроссплатформености и адаптивной верстки;
- термины, синтаксис, управляющие конструкции и типы данных скриптового языка общего назначения PHP;
- основы JavaScript;
- инструменты и принципы разработки web-приложения;
- назначение тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования

Уметь:

- обеспечить передачу информации между приложением и веб-сервером;
- осуществлять подключение к серверу и выбор базы данных;
- разрабатывать динамические приложения на языке JavaScript и PHP для решения прикладных задач;

- проводить процедуру тестирования баз данных и web-приложений;

Владеть:

- навыками создания приложения с помощью языков программирования (HTML, JavaScript, PHP, CSS);
- навыками создания сценариев с помощью операций и управляющих конструкций PHP;
- навыками работы с различными формами в PHP;
- навыками работы с сессиями и cookie в PHP
- навыками осуществления передачи данных через HTML-формы
- навыками протоколирования результатов тестирования

ЗАДАНИЯ ЗАКРЫТОГО ТИПА

1. Какой язык программирования является стандартом для создания гипертекстовых и гипермедийных документов в интернете?
A) HTML
B) JavaScript
C) PHP
D) CSS
Ответ: A) HTML

2. Какой из перечисленных языков программирования отвечает за динамическое изменение содержимого и поведения веб-страницы?
A) HTML
B) JavaScript
C) PHP
D) CSS
Ответ: B) JavaScript

3. Какой язык программирования широко используется для создания серверной части веб-приложений?
A) HTML
B) JavaScript
C) PHP
D) CSS
Ответ: C) PHP

4. Для чего используется язык программирования CSS?
A) Для описания структуры HTML-документа
B) Для добавления интерактивности на веб-странице
C) Для определения внешнего вида элементов HTML
D) Для работы с серверной частью веб-приложения
Ответ: C) Для определения внешнего вида элементов HTML

5. Какой из языков программирования используется для стилизации веб-страницы и задания внешнего вида элементов?
A) HTML
B) JavaScript
C) PHP
D) CSS
Ответ: D) CSS

6. Что означает термин "кроссбраузерность" в веб-разработке?
A) Совместимость веб-сайта с различными операционными системами
B) Совместимость веб-сайта с различными браузерами и их версиями
C) Возможность отображения веб-сайта на разных устройствах
D) Возможность взаимодействия веб-сайта с разными базами данных
Ответ: B) Совместимость веб-сайта с различными браузерами и их версиями

7. Что означает термин "кроссплатформенность" в веб-разработке?
A) Возможность отображения веб-сайта на различных устройствах
B) Возможность взаимодействия веб-сайта с разными базами данных
C) Возможность работы веб-сайта на разных операционных системах
D) Возможность создания резервных копий веб-сайта на разных серверах
Ответ: C) Возможность работы веб-сайта на разных операционных системах

8. Что означает термин "адаптивная верстка"?
A) Возможность веб-сайта работать на различных браузерах
B) Возможность веб-сайта работать на различных операционных системах
C) Возможность веб-сайта отображаться корректно на различных устройствах и разрешениях экранов
D) Возможность веб-сайта адаптироваться под разные языки и культуры
Ответ: C) Возможность веб-сайта отображаться корректно на различных устройствах и разрешениях экранов

9. Что такое медиазапросы в CSS?
A) Элементы CSS, которые определяют внешний вид веб-сайта
B) Конструкции CSS, которые позволяют применять стили в зависимости от различных устройств и условий
C) Параметры CSS, определяющие кроссбраузерное поведение веб-сайта
D) Подход в CSS, который обеспечивает кроссплатформенность веб-сайта
Ответ: B) Конструкции CSS, которые позволяют применять стили в зависимости от различных устройств и условий

10. Какие из следующих методов являются основными для обеспечения кроссбраузерности веб-сайта?
A) Использование вендорных префиксов в CSS
B) Использование JavaScript-библиотек для стандартизации поведения в разных браузерах
C) Тестирование и отладка в различных браузерах и их версиях
D) Все перечисленные методы
Ответ: D) Все перечисленные методы

ЗАДАНИЯ ОТКРЫТОГО ТИПА

1. Напишите JavaScript функцию, которая принимает на вход массив чисел и возвращает сумму всех элементов этого массива.

Решение:

JavaScript функция для вычисления суммы всех элементов массива:

```javascript
function sumArray(array) {
    return array.reduce((sum, current) => sum + current, 0);
```

```
}
// Пример использования:
const numbers = [1, 2, 3, 4, 5];
console.log(sumArray(numbers)); // Выведет: 15
```

2. Создайте PHP-скрипт, который принимает на вход POST-запросом два числа и возвращает их сумму.

Решение:

Пример PHP-скрипта для вычисления суммы двух чисел из POST-запроса:

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $num1 = $_POST["num1"];
    $num2 = $_POST["num2"];
    $sum = $num1 + $num2;
    echo $sum;
}
?>
```

3. Напишите JavaScript функцию, которая проверяет, является ли строка палиндромом (читается одинаково как с начала, так и с конца) без учета регистра символов и пробелов.

Решение:

JavaScript функция для проверки строки на палиндром:

```javascript
function isPalindrome(str) {
    const cleanStr = str.toLowerCase().replace(/[^a-za-яё]/g, '');
    return cleanStr === cleanStr.split('').reverse().join('');
}
// Пример использования:
console.log(isPalindrome("А роза упала на лапу Азора")); // Выведет: true
```

4. Разработайте PHP-скрипт, который принимает на вход GET-параметр "username" и выводит сообщение "Добро пожаловать, [username]!".

Решение:

Пример PHP-скрипта для вывода персонализированного приветствия:

```php
<?php
if (isset($_GET["username"])) {
```

```
    $username = $_GET["username"];
    echo "Добро пожаловать, $username!";
}
?>
```

5. Напишите JavaScript функцию, которая принимает на вход строку и возвращает новую строку, в которой каждое слово начинается с заглавной буквы.

Решение:

JavaScript функция для преобразования каждого слова в строке, чтобы оно начиналось с заглавной буквы:

```javascript
function capitalizeWords(str) {
    return str.replace(/\b\w/g, char => char.toUpperCase());
}
// Пример использования:
console.log(capitalizeWords("hello world")); // Выведет: Hello World
```

6. Создайте PHP-скрипт, который принимает на вход POST-запросом текст и возвращает количество слов в этом тексте.

Решение:

PHP-скрипт для подсчета количества слов в тексте, переданном POST-запросом:

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $text = $_POST["text"];
    $wordCount = str_word_count($text);
    echo $wordCount;
}
?>
```

7. Напишите JavaScript функцию, которая принимает на вход число и возвращает его факториал.

Решение:

JavaScript функция для вычисления факториала числа:

```javascript
function factorial(n) {
    if (n === 0 || n === 1) {
        return 1;
```

```
    }
    return n * factorial(n - 1);
}
// Пример использования:
console.log(factorial(5)); // Выведет: 120
```

8. Разработайте PHP-скрипт, который принимает на вход GET-параметры "width" и "height" и выводит HTML-код для создания прямоугольника с заданными размерами.

Решение:

Пример PHP-скрипта для создания прямоугольника с заданными размерами через GET-параметры:

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Rectangle</title>
</head>
<body>
    <?php
    if (isset($_GET["width"]) && isset($_GET["height"])) {
        $width = $_GET["width"];
        $height = $_GET["height"];
        echo "<div style='width: {$width}px; height: {$height}px; background-color: #ccc;'></div>";
    }
    ?>
</body>
</html>
```

9. Напишите JavaScript функцию, которая принимает на вход массив объектов (каждый объект содержит поля "имя" и "возраст") и возвращает массив имен тех объектов, у которых возраст больше 18 лет.

Решение:

JavaScript функция для фильтрации объектов по возрасту и получения массива их имен:

```javascript
function getAdultNames(objects) {
```

```
    return objects
        .filter(obj => obj.age > 18)
        .map(obj => obj.name);
}
// Пример использования:
const people = [
    { name: 'Alice', age: 25 },
    { name: 'Bob', age: 17 },
    { name: 'Charlie', age: 30 }
];
console.log(getAdultNames(people)); // Выведет: ["Alice", "Charlie"]
```

10. Создайте PHP-скрипт, который принимает на вход POST-запросом массив чисел и возвращает их среднее значение.

Решение:

PHP-скрипт для вычисления среднего значения чисел из массива, переданного POST-запросом:

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $numbers = $_POST["numbers"];
    $average = array_sum($numbers) / count($numbers);
    echo $average;
}
?>
```

11. Напишите PHP-скрипт для создания прямоугольника с заданными размерами через GET-параметры:

Решение:

Пример PHP-скрипта для создания прямоугольника с заданными размерами через GET-параметры:

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Rectangle</title>
```

```
</head>
<body>
    <?php
    if (isset($_GET["width"]) && isset($_GET["height"])) {
        $width = $_GET["width"];
        $height = $_GET["height"];
        echo "<div style='width: {$width}px; height: {$height}px; background-color: #ccc;'></div>";
    } else {
        echo "Параметры width и height не указаны!";
    }
    ?>
</body>
</html>
```

12. Напишите HTML-форму для отправки массива чисел на сервер с использованием метода POST:

Решение:

Пример HTML-формы для отправки массива чисел на сервер с использованием метода POST:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Submit Numbers</title>
</head>
<body>
    <form action="average.php" method="post">
        <label for="numbers">Enter numbers (comma-separated):</label><br>
        <input type="text" id="numbers" name="numbers"><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

Этот HTML-код создает форму, где пользователь может ввести числа, разделенные запятыми, которые будут отправлены на сервер методом POST.

13. Напишите JavaScript функцию для проверки, является ли переданная строка палиндромом без учета пробелов и регистра символов.

Решение:

```javascript
function isPalindromeIgnoringCaseAndSpace(str) {
    const cleanStr = str.toLowerCase().replace(/\s+/g, '');
    return cleanStr === cleanStr.split('').reverse().join('');
}
// Пример использования:
console.log(isPalindromeIgnoringCaseAndSpace("A man a plan a canal Panama")); // Выведет: true
```

Эта функция принимает строку, приводит её к нижнему регистру, удаляет все пробелы и затем сравнивает полученную строку с её обратным представлением.

14. Создайте PHP-скрипт, который принимает на вход POST-запросом массив строк и возвращает массив строк, в которых содержится более двух слов.

Решение:

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $inputStrings = $_POST["strings"];
    $resultStrings = array_filter($inputStrings, function($str) {
        return str_word_count($str) > 2;
    });
    echo json_encode($resultStrings);
}
?>
```

Этот скрипт принимает массив строк через POST-запрос, фильтрует строки, оставляя только те, в которых более двух слов, и возвращает результат в формате JSON.

15. Напишите JavaScript функцию для нахождения наибольшего общего делителя (НОД) двух чисел.

Решение:

```javascript
function gcd(a, b) {
    if (b === 0) {
        return a;
```

```
    } else {
        return gcd(b, a % b);
    }
}
// Пример использования:
console.log(gcd(48, 18)); // Выведет: 6
```

Эта функция использует алгоритм Евклида для нахождения наибольшего общего делителя двух чисел.

ПК-5. Способен обеспечивать качество работы web-приложения

Знать:
- назначение юзабилити-аудит web-приложения
- назначение тестирования, виды и способы тестирования баз данных и web-приложений, процедуру тестирования

Уметь:
- проводить юзабилити-аудит web-приложения;
- проводить процедуру тестирования баз данных и web-приложений;

Владеть:
- навыками протоколирования результатов тестирования
- навыками протоколирования результатов тестирования

ЗАДАНИЯ ЗАКРЫТОГО ТИПА

1. Какой из следующих факторов оказывает наибольшее влияние на пользовательское восприятие скорости загрузки веб-приложения?
A) Объем данных, передаваемых по сети
B) Размер изображений на странице
C) Сложность JavaScript-кода
D) Количество используемых CSS файлов
Ответ: A) Объем данных, передаваемых по сети

2. Что означает термин «респонсивность» в контексте веб-приложений?
A) Возможность приложения адаптироваться под разные типы устройств
B) Быстрый отклик сервера на запросы пользователя
C) Возможность пользователя взаимодействовать с приложением в реальном времени
D) Способность приложения корректно обрабатывать ошибки и исключения
Ответ: A) Возможность приложения адаптироваться под разные типы устройств

3. Какие из следующих практик способствуют улучшению производительности веб-приложения?
A) Минификация и сжатие файлов CSS и JavaScript
B) Добавление максимального количества изображений на страницу
C) Использование наибольшего возможного числа сторонних библиотек и плагинов
D) Все вышеперечисленные практики
Ответ: A) Минификация и сжатие файлов CSS и JavaScript

4. Что такое "оптимизация изображений" в контексте веб-приложений?
A) Процесс выбора самых красивых изображений для использования на веб-страницах
B) Процесс уменьшения размера файлов изображений без значительной потери качества
C) Процесс добавления дополнительных изображений на веб-страницы для улучшения дизайна
D) Процесс оптимизации цветовой гаммы изображений для лучшего отображения на различных устройствах
Ответ: B) Процесс уменьшения размера файлов изображений без значительной потери качества

5. Что такое "кэширование" веб-ресурсов?
A) Процесс сохранения копий веб-страниц на локальном компьютере пользователя
B) Процесс хранения часто используемых данных на сервере приложения
C) Процесс временного хранения ранее загруженных ресурсов на стороне клиента или сервера
D) Процесс сжатия и упаковки файлов ресурсов для более быстрой загрузки
Ответ: C) Процесс временного хранения ранее загруженных ресурсов на стороне клиента или сервера

6. Какая из следующих характеристик наиболее существенна для оценки юзабилити веб-приложения?
A) Графический дизайн и использование цветовой схемы
B) Возможность масштабирования и адаптивности интерфейса
C) Наличие сложных анимаций и переходов между страницами
D) Корректная работа JavaScript-функционала
Ответ: B) Возможность масштабирования и адаптивности интерфейса

7. Какая из перечисленных функциональных особенностей веб-приложения имеет наибольшее влияние на его юзабилити?
A) Возможность работы с базой данных
B) Функционал поиска и фильтрации данных
C) Наличие слайдеров и каруселей для просмотра контента
D) Автоматическое обновление страницы без перезагрузки
Ответ: B) Функционал поиска и фильтрации данных

8. Какой из следующих аспектов веб-приложения необходимо оценивать при аудите юзабилити?
A) Доступность и навигация
B) Цветовая гамма и графический дизайн
C) Скорость загрузки и производительность
D) Все вышеперечисленные аспекты
Ответ: D) Все вышеперечисленные аспекты

9. Что означает термин «доступность» в контексте юзабилити веб-приложений?
A) Возможность приложения быть доступным для пользователей в разных странах
B) Возможность приложения быть доступным для пользователей с ограниченными возможностями (например, с ограниченным зрением или слухом)
C) Возможность приложения быть доступным для пользователей через мобильные устройства
D) Возможность приложения быть доступным для пользователей с разным уровнем технической подготовки
Ответ: B) Возможность приложения быть доступным для пользователей с ограниченными возможностями (например, с ограниченным зрением или слухом)

10. Какие метрики помогают оценить удобство использования форм на веб-страницах?
A) Время загрузки страницы и скорость отклика сервера

B) Количество полей ввода и длина формы
C) Процент успешно отправленных форм
D) Все вышеперечисленные метрики
Ответ: D) Все вышеперечисленные метрики

ЗАДАНИЯ ОТКРЫТОГО ТИПА

1. Задание: Опишите процесс тестирования производительности веб-приложения. Какие инструменты и методики могут быть использованы для оценки производительности?

Ответ:

Процесс тестирования производительности веб-приложения включает в себя несколько этапов.

Сначала необходимо определить цели тестирования, например, проверка скорости загрузки страниц, времени отклика сервера, производительности базы данных и т.д.

Далее составляется план тестирования, где определяются критерии оценки производительности и методы сбора данных.

Затем проводится непосредственно тестирование, включающее нагрузочное тестирование, стресс-тестирование, тестирование производительности UI и т.д.

После тестирования анализируются полученные данные, выявляются узкие места и проблемы производительности.

На основе результатов анализа принимаются меры по оптимизации приложения.

В качестве инструментов для оценки производительности веб-приложения могут использоваться JMeter, Apache Benchmark, LoadRunner, WebPageTest, Lighthouse и другие. Методики могут включать в себя сравнение средних значений времени ответа сервера, анализ пропускной способности сети, определение времени загрузки страниц и ресурсов, и т.д.

2. Задание: Объясните, что такое автоматизированное тестирование веб-приложений. Какие инструменты и подходы могут быть использованы для автоматизации тестирования?

Ответ:

Автоматизированное тестирование веб-приложений - это процесс использования специальных программных средств для выполнения тестовых сценариев без необходимости вручную повторять действия тестировщика. Это позволяет повысить эффективность тестирования, ускорить процесс выявления ошибок и уменьшить вероятность человеческих ошибок.

Для автоматизации тестирования веб-приложений могут использоваться различные инструменты и подходы:

Фреймворки для написания автотестов, такие как Selenium WebDriver, Puppeteer, Cypress.io.

Инструменты для тестирования API, например, Postman, SoapUI.

Инструменты для генерации тестовых данных, например, Faker.js.

Средства контроля версий для хранения тестовых сценариев и результатов их выполнения, такие как Git, SVN.

Инструменты для непрерывной интеграции и развертывания, например, Jenkins, GitLab CI/CD.

Использование языков программирования, таких как Python, JavaScript, для написания скриптов автоматизации тестирования.

3. Задание: Какие методы и инструменты могут использоваться для тестирования безопасности веб-приложений?

Ответ:

Тестирование безопасности веб-приложений - это процесс проверки наличия уязвимостей и защищенности приложения от различных видов атак, таких как инъекции, переполнения

буфера, утечки информации и другие. Для этого могут использоваться различные методы и инструменты:

Анализ исходного кода приложения с использованием специализированных инструментов, например, SonarQube, ESLint.

Проведение сканирования уязвимостей, например, с помощью Burp Suite, OWASP ZAP.

Тестирование на проникновение, включающее в себя моделирование атак и проверку реакции приложения на них.

Аудит конфигурации серверов и баз данных на предмет наличия уязвимостей.

Обучение персонала безопасности и разработчиков в области лучших практик безопасного программирования и защиты данных.

4. Задание: Какие основные аспекты следует учитывать при тестировании кроссбраузерной совместимости веб-приложения?

Ответ:

При тестировании кроссбраузерной совместимости веб-приложения необходимо учитывать следующие аспекты:

Проверка отображения и работы приложения в различных браузерах (Chrome, Firefox, Safari, Edge) и их версиях.

Тестирование на различных операционных системах (Windows, MacOS, Linux) и устройствах (настольные компьютеры, ноутбуки, планшеты, мобильные устройства).

Проверка корректности отображения и интерактивности пользовательского интерфейса, включая формы, кнопки, меню, анимации и т.д.

Адаптация приложения под разные разрешения экранов и устройства с разной плотностью пикселей.

Проверка работы функциональности приложения, включая AJAX-запросы, анимации, клиентские скрипты, при работе с разными браузерами и устройствами.

5. Задание: Каким образом можно провести тестирование доступности веб-приложения для пользователей с ограниченными возможностями?

Ответ:

Тестирование доступности веб-приложения для пользователей с ограниченными возможностями включает в себя ряд мероприятий:

Проверка соответствия веб-приложения стандартам доступности, таким как WCAG (Web Content Accessibility Guidelines).

Тестирование навигации по приложению с помощью клавиатуры, без использования мыши.

Проверка совместимости с ассистивными технологиями, такими как программы чтения с экрана (screen readers).

Оценка контрастности цветов, используемых на веб-страницах, для обеспечения читаемости для людей с нарушениями зрения.

Тестирование доступности форм и интерактивных элементов управления для пользователей с ограниченными двигательными возможностями.

Проверка использования семантических элементов HTML для правильного описания структуры документа и его контента для программ чтения с экрана.

Аудит доступности изображений, таблиц и мультимедийного контента для пользователей с нарушениями зрения или слуха.

6. Задание: Опишите процесс функционального тестирования веб-приложения. Какие шаги включает в себя функциональное тестирование?

Ответ:

Функциональное тестирование веб-приложения направлено на проверку корректности

работы его функциональности согласно требованиям и ожиданиям пользователей. Этот процесс включает в себя следующие шаги:

1. Подготовка тестовых сценариев. На этом этапе определяются основные функциональные возможности приложения, которые подлежат тестированию, и разрабатываются соответствующие тестовые сценарии.

2. Написание тестовых случаев. Для каждого тестового сценария разрабатываются конкретные тестовые случаи, которые определяют шаги тестирования, ожидаемые результаты и критерии успешного выполнения.

3. Выполнение тестов. На этом этапе тестировщики запускают тестовые случаи, выполняют шаги тестирования в соответствии с тестовыми сценариями и записывают результаты выполнения.

4. Анализ результатов. Полученные результаты тестирования анализируются для выявления ошибок, несоответствий требованиям или других проблем функциональности приложения.

5. Отчетность. На основе анализа результатов тестирования разрабатывается отчет, в котором фиксируются выявленные проблемы, предложения по их устранению и общая оценка качества функциональности приложения.

7.Задание: Как можно проверить надежность и стабильность веб- приложения?

Ответ: Для проверки надежности и стабильности веб-приложения можно использовать следующие методы и инструменты:

Мониторинг производительности. Организация постоянного мониторинга работы веб-приложения с целью выявления узких мест, проблем производительности и недоступности.

Тестирование стрессоустойчивости. Проведение тестовых нагрузок, имитирующих высокую нагрузку на сервер и сеть, для оценки реакции приложения на такие условия.

Тестирование восстановления после сбоев. Проверка возможности восстановления приложения после сбоев, включая проверку работы механизмов резервного копирования, восстановления данных и т.д.

Мониторинг ошибок и исключений. Анализ логов приложения и системных регистров для выявления ошибок, исключений и нештатных ситуаций, которые могут привести к сбоям.

Тестирование безопасности. Проверка приложения на наличие уязвимостей и защищенность от атак, таких как инъекции, переполнения буфера, утечки информации и другие.

Регулярное обновление и тестирование. Проведение регулярных обновлений и тестирования новых версий приложения с целью выявления и исправления проблем до их обнаружения пользователями.

8.Задание: Как можно обеспечить высокую степень защиты данных в веб-приложении?

Ответ: Для обеспечения высокой степени защиты данных в веб-приложении необходимо применять следующие меры:

Шифрование данных. Использование шифрования для защиты конфиденциальных данных при передаче по сети и хранении на сервере.

Механизмы аутентификации и авторизации. Реализация механизмов аутентификации пользователей и управления правами доступа для предотвращения несанкционированного доступа к данным.

Обновление системы и патчи безопасности. Регулярное обновление серверного и клиентского программного обеспечения с целью устранения известных уязвимостей и установки последних патчей безопасности.

Мониторинг безопасности. Постоянный мониторинг системы на предмет обнаружения подозрительной активности, атак и попыток несанкционированного доступа.

Резервное копирование данных. Создание регулярных резервных копий данных с целью восстановления информации в случае ее потери или повреждения.

Обучение сотрудников. Проведение обучения персонала по вопросам безопасности информации, осведомление их о потенциальных угрозах и методах защиты.

9.Задание: Как можно оценить качество кода веб-приложения?

Ответ: Качество кода веб-приложения можно оценить с помощью следующих методов:

Анализ кода статическими анализаторами. Использование специализированных инструментов для анализа и проверки исходного кода на соответствие стандартам кодирования, выявление потенциальных уязвимостей и ошибок.

Code review. Организация процесса код-ревью, в ходе которого другие члены команды проверяют код на соответствие стандартам, наличие ошибок и улучшения.

Тестирование кода. Написание и выполнение автоматизированных тестов для проверки корректности работы функциональности приложения, а также тестов на производительность, безопасность и доступность.

Анализ метрик кода. Использование метрик кода, таких как цикломатическая сложность, покрытие тестами, количественные характеристики кода (количество строк, функций и т.д.), для оценки его качества.

Применение лучших практик программирования. Следование принципам SOLID, DRY, KISS и другим лучшим практикам программирования для создания чистого, эффективного и поддерживаемого кода.

10. Задание: Какие могут быть стратегии и методы тестирования веб-приложений для обеспечения их качества?

Ответ: Для обеспечения качества веб-приложений могут применяться различные стратегии и методы тестирования:

Методы тестирования по уровням. Это включает в себя модульное тестирование (unit testing), интеграционное тестирование (integration testing), системное тестирование (system testing) и приемочное тестирование (acceptance testing).

Методы тестирования по типам.

11. Задание: Проверка соединения с базой данных

Решение:

Шаги:

1. Запустите приложение или скрипт, который должен использовать базу данных.
2. Попытайтесь установить соединение с базой данных, используя предварительно заданные учетные данные и параметры подключения.
3. Проверьте, было ли соединение успешно установлено.
4. Верните сообщение об успешном соединении или сообщение об ошибке, если соединение не удалось.

Ответ:

- Если соединение установлено успешно, выдаётся сообщение: "Соединение с базой данных установлено успешно."
- Если соединение не удалось установить из-за неверных учетных данных или параметров подключения, выдаётся сообщение: "Ошибка: Не удалось установить соединение с базой данных."

12. Задание: Проверка наличия таблицы в базе данных

Решение:

Шаги:

1. Убедитесь, что соединение с базой данных установлено успешно.
2. Выполните запрос к базе данных, чтобы проверить наличие конкретной таблицы.
3. Проверьте результат запроса: наличие таблицы или ошибку, если таблица не существует.

4. Верните сообщение о наличии таблицы или сообщение об ошибке, если таблица отсутствует.

Ответ:

- Если таблица существует, выдаётся сообщение: «Таблица 'название_таблицы' существует в базе данных.»
- Если таблица не существует, выдаётся сообщение: «Ошибка: Таблица 'название_таблицы' отсутствует в базе данных.»

13.Задание: Проверка количества записей в таблице

Решение:

Шаги:

1. Убедитесь, что соединение с базой данных установлено успешно.
2. Выполните запрос к базе данных для подсчета количества записей в указанной таблице.
3. Проверьте результат запроса: количество записей или ошибку, если таблица не существует.
4. Верните сообщение о количестве записей в таблице или сообщение об ошибке, если таблица отсутствует.

Ответ:

- Если таблица существует и содержит записи, выдаётся сообщение с количеством записей: «Таблица 'название_таблицы' содержит 'N' записей.»
- Если таблица существует, но не содержит записей, выдаётся сообщение: «Таблица 'название_таблицы' пуста.»
- Если таблица не существует, выдаётся сообщение об ошибке: «Ошибка: Таблица 'название_таблицы' отсутствует в базе данных.»

14.Задание: Проверка правильности структуры таблицы

Решение:

Шаги:

1. Убедитесь, что соединение с базой данных установлено успешно.
2. Выполните запрос к базе данных для получения описания структуры таблицы.
3. Сравните полученную структуру с ожидаемой схемой таблицы.
4. Верните сообщение о соответствии или несоответствии структуры таблицы ожидаемой схеме.

Ответ:

- Если структура таблицы соответствует ожидаемой схеме, выдаётся сообщение: «Структура таблицы 'название_таблицы' соответствует ожидаемой схеме.»
- Если структура таблицы не соответствует ожидаемой схеме, выдаётся сообщение с указанием различий.
- Если таблица не существует, выдаётся сообщение об ошибке: «Ошибка: Таблица 'название_таблицы' отсутствует в базе данных.»

15.Задание: Проверка наличия индексов в таблице

Решение:

Шаги:

1. Убедитесь, что соединение с базой данных установлено успешно.
2. Выполните запрос к базе данных для получения списка индексов в указанной таблице.
3. Проверьте результат запроса: наличие индексов или их отсутствие.
4. Верните сообщение о наличии индексов в таблице или сообщение об их отсутствии.

Ответ:

- Если в таблице существуют индексы, выдаётся сообщение с перечислением индексов: «Таблица 'название_таблицы' содержит следующие индексы: [индекс1, индекс2, ...]».

- Если в таблице нет индексов, выдаётся сообщение: «Таблица 'название_таблицы' не содержит индексов.»

- Если таблица не существует, выдаётся сообщение об ошибке: «Ошибка: Таблица 'название_таблицы' отсутствует в базе данных.»

6. ЛИСТ СОГЛАСОВАНИЯ

Составил:

В.Н. Маризина, к.п.н.

(подпись)

Заведующий кафедрой

Н.Б.Стрекалова, д.п.н., доцент

(подпись)

Заведующий выпускающей кафедрой

Н.Б.Стрекалова, д.п.н., доцент

(подпись)