Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный индустриальный университет»

Кафедра прикладных информационных технологий и программирования

УТВЕРЖДАЮ
Директор института
информационных технологий и
автоматизированных систем
_____ А.В. Зимин
подпись
«____» _____ 20___ г.

# РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Проектирование и разработка веб-приложений

09.02.11 «Разработка и управление программным обеспечением»
(направленность (профиль): «Разработка и управление программным обеспечением»)

Квалификация выпускника
Программист

Форма обучения
Очная форма

Срок обучения: 3 года 10 месяцев

Год начала подготовки 2026

Новокузнецк
2026

СибГИУ
Подлинник электронного документа хранится в системе электронного согласования документов СибГИУ
ДОКУМЕНТ ПОДПИСАН ЭП
Сертификат: e95695aa2c5fd4a2ad858d663abfeefbdf88f7d3
Владелец: Директор ИИТиАС Зимин Алексей Валерьевич
Действителен: с 2 апреля 2026 г. по 26 июня 2027 г.
Дата и время: 20-05-2026 10:12:40 ПРОВЕРИТЬ ПОДПИСЬ

## 1 Цели и задачи освоения учебной дисциплины

Целями учебной дисциплины являются:

– формирование у обучающихся системных знаний и практических навыков по проектированию, разработке и тестированию веб-приложений с использованием современных клиентских и серверных технологий, обеспечивающих выполнение профессиональных задач в соответствии с техническим заданием заказчика.

Задачами учебной дисциплины являются:

– изучение серверных и клиентских языков программирования и знакомство с бэкенд-платформами для создания полноценного веб-приложения;

– формирование навыков создания пользовательского интерфейса с использованием JavaScript, фреймворков, а также взаимодействия с данными через REST API;

– получение практических умений по составлению технического задания, реализации CRUD-логики, тестированию, работе с CMS и выполнению операций по развертыванию (build, deploy) веб-приложения.

## 2 Место учебной дисциплины в структуре ООП по специальности

Учебная дисциплина входит в состав профессионального модуля ПМ.03 «Проектирование и разработка веб-приложений» профессионального цикла ООП по специальности 09.02.11 «Разработка и управление программным обеспечением».

Учебная дисциплина базируется на предварительном усвоении обучающимися учебных дисциплин:

– Проектная деятельность 1;
– Разработка программных модулей;
– Основы алгоритмизации и программирования;
– Управление ИТ-проектами;
– Проектная деятельность 2;
– Проектная деятельность 3;
– Информатика.

Учебная дисциплина дополняет умения и знания, получаемые по одновременно изучаемым и последующим учебным дисциплинам:

– Оптимизация веб-приложений;
– Обеспечение безопасности веб-приложений;
– Проектная деятельность 4.

## 3 Планируемые результаты обучения по учебной дисциплине

Процесс изучения учебной дисциплины направлен на формирование следующих компетенций:

**Общие компетенции**

– ОК 01.: Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

– ОК 02.: Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности

– ОК 05.: Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

– ОК 09.: Пользоваться профессиональной документацией на государственном и иностранном языках

**Профессиональные компетенции**

– ПК 3.1.: Разрабатывать техническое задание на веб-приложение в соответствии с требованиями заказчика

– ПК 3.2.: Разрабатывать веб-приложения в соответствии с техническим заданием

– ПК 3.4.: Производить тестирование разработанного веб-приложения

– ПК 3.7.: Реализовывать мероприятия по продвижению приложения

Обучающийся должен быть готов к выполнению основных видов профессиональной деятельности:

– Проектирование и разработка веб-приложений.

В результате освоения учебной дисциплины обучающийся должен уметь, знать, иметь практический опыт:

| Код ОК, ПК | Уметь | Знать | Иметь практический опыт |
|---|---|---|---|
| ОК 01.<br>ОК 02.<br>ОК 05.<br>ОК 09.<br>ПК 3.1.<br>ПК 3.2.<br>ПК 3.4.<br>ПК 3.7. | - распознавать задачу и/или проблему в профессиональном и/или социальном контексте;<br>анализировать задачу и/или проблему и выделять её составные части;<br>определять этапы решения задачи;<br>выявлять и эффективно искать информацию, необходимую для решения задачи | - актуальный профессиональный и социальный контекст, в котором приходится работать и жить;<br>основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте;<br>алгоритмы выполнения работ в профессиональной и смежных областях;<br>методы работы в профессиональной и | - сбора предварительных данных для выявления требований к веб-приложению;<br>- определения первоначальных требований заказчика к веб-приложению и возможности их реализации;<br>- подбора оптимальных вариантов реализации задач и согласование их с |

| | | | |
|---|---|---|---|
| | и/или проблемы;<br>- составлять план действия; определять необходимые ресурсы;<br>- владеть актуальными методами работы в профессиональной и смежных сферах; реализовывать составленный план;<br>- оценивать результат и последствия своих действий (самостоятельно или с помощью наставника);<br>- определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; - структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска, применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение; использовать различные | смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности;<br>- номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации, современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности в том числе с использованием цифровых средств;<br>- особенности социального и культурного контекста; правила оформления документов и построения устных сообщений;<br>- правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной | заказчиком; разработки технического задания на веб-приложение в соответствии с требованиями заказчика;<br>- выполнения верстки страниц веб приложений;<br>- кодирования на языках веб программирования;<br>- разработки базы данных;<br>- умения использовать специальные готовые технические решения при разработке веб приложений;<br>- выполнения разработки информационных систем;<br>- разработки интерфейса пользователя;<br>- разработки анимационных эффектов;<br>- разработки интерфейсов пользователя, используя существующие наборы стилей, такие как Bootstrap или Foundation, для создания привлекательного и согласованного визуального оформления;<br>- применения предустановленных элементов управления, таких как кнопки, формы, |

| | | | |
|---|---|---|---|
| | цифровые средства для решения профессиональных задач;<br>- грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе;<br>- понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснять свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы;<br>- проводить анкетирование и интервьюирование для выявления требований заказчика;<br>- оформлять техническую | деятельности; особенности произношения; правила чтения текстов профессиональной направленности;<br>- инструменты и методы выявления требований заказчика к веб-приложению;<br>- типовые решения по разработке веб-приложений;<br>- нормы и стандарты оформления технической документации; принципы проектирования и разработки информационных систем;<br>- языки программирования и разметки для разработки клиентской и серверной части веб-приложений;<br>- принципы работы объектной модели веб-приложений и браузера;<br>- основы технологии клиент-сервер;<br>- технологии разработки серверной части;<br>- особенности отображения веб приложений в размерах рабочего пространства устройств;<br>- особенности отображения элементов ИР в различных браузерах;<br>- особенности выбранной среды программирования и системы управления | меню и т.д., предоставляемых в выбранных наборах стилей;<br>- адаптации и настройки стилей и элементов управления с использованием CSS и JavaScript;<br>- использования инструментальных средств контроля версий и баз данных, учета дефектов;<br>- тестирования веб-приложений с точки зрения логической целостности; тестирования интеграции веб-приложения с внешними сервисами и учетными системами;<br>-- реализации мероприятий по продвижению веб-приложений в сети Интернет;<br>- сбора и предварительного анализа статистическую информации о работе веб-приложений;<br>- сбора статической статистики о работе веб-приложений, такой как время отклика, количество запросов и ошибок, использование ресурсов и т.д.;<br>- анализа собранной статистики для определения эффективности |

| | | | |
|---|---|---|---|
| | документацию в соответствии с нормами и стандартами;<br>- осуществление выбора одного из типовых решений по разработке веб-приложений;<br>работы со специализированным программным обеспечением для планирования времени и организации работы с клиентами;<br>- проводить анкетирование и интервьюирование для выявления требований заказчика;<br>- оформлять техническую документацию в соответствии с нормами и стандартами;<br>- осуществление выбора одного из типовых решений по разработке веб-приложений;<br>работы со специализированным программным обеспечением для планирования времени и организации работы с клиентами;<br>- выполнять отладку и тестирование программного кода (в том числе с использованием инструментальных средств);<br>- выполнять оптимизацию и рефакторинг | базами данных;<br>- языки программирования и разметки для разработки клиентской и серверной части веб приложений;<br>- принципы работы объектной модели веб-приложений и браузера;<br>- технологии для разработки анимации;<br>- способы манипуляции элементами страницы веб-приложения;<br>- виды анимации и способы ее применения;<br>- знакомство с существующими наборами стилей, такими как Bootstrap, Foundation, Material UI и другие;<br>- понимание основных концепций и возможностей предоставляемых наборами стилей и элементов управления;<br>- знание CSS и JavaScript для настройки и расширения стилей и элементов управления в выбранных наборах;<br>- сетевые протоколы и основы web-технологий;<br>- современные методики тестирования;<br>- эргономику пользовательских интерфейсов;<br>- основные принципы отладки и тестирования программных | работы веб-приложения и выявления возможных проблем или узких мест;<br>- применения методов и инструментов для анализа производительности веб-приложений, таких как мониторинг систем, аналитика данных и профилирование кода;<br>- реализации мероприятий по продвижению веб-приложений в сети Интернет;<br>- сбора и предварительного анализа статистической информации о работе веб-приложений;<br>- разработки и реализации стратегии продвижения приложения в соответствии с целями и потребностями бизнеса;<br>- проведения маркетинговых исследований для определения целевой аудитории и конкурентной среды;<br>- создания и оптимизации контента для привлечения и удержания пользователей, включая описания приложения, видео |

| | | | |
|---|---|---|---|
| | программного кода;<br>- кодировать на скриптовых языках программирования;<br>- тестировать веб-приложения с использованием тест-планов;<br>- применять инструменты подготовки тестовых данных;<br>- выбирать и комбинировать техники тестирования веб-приложений;<br>- работать с системами контроля версий в соответствии с регламентом использования системы контроля версий;<br>выполнять проверку веб-приложения по техническому заданию;<br>- подключать и настраивать системы мониторинга работы Веб-приложений и сбора статистики его использования;<br>- составлять отчет по основным показателям использования Веб-приложений (рейтинг, источники и поведение пользователей, конверсия и др.);<br>- способность настроить сбор статистики о работе веб-приложений, используя различные инструменты и технологии, такие как мониторинг системы, | продуктов;<br>- методы организации работы при проведении процедур тестирования;<br>- возможности используемой системы;<br>- контроль версий и вспомогательных инструментальных программных средств для обработки исходного текста программного кода;<br>- регламент использования системы контроля версий; предметную область проекта для составления тест-планов;<br>- основные показатели использования;<br>- веб-приложения и способы их анализа;<br>- различные методы и инструменты для сбора статистики о работе веб-приложений, такие как мониторинг систем (Nagios, Zabbix), аналитические инструменты (Google Analytics, ELK Stack) и инструменты профилирования кода (Xdebug, Blackfire);<br>- основные метрики и показатели производительности веб-приложений, таких как время отклика, пропускная способность, использование ресурсов и т.д.;<br>- методы оптимизации и улучшения производительности веб-приложений на | обзоры, блоги и социальные медиа публикации;<br>- разработки и реализации рекламных кампаний для повышения видимости приложения, включая контекстную рекламу, рекламу в социальных сетях и партнерские программы;<br>- анализа эффективности мероприятий по продвижению и оптимизация стратегии на основе полученных результатов;<br>разработки и реализации рекламных кампаний для повышения видимости приложения, включая контекстную рекламу, рекламу в социальных сетях и партнерские программы. |

| | | | |
|---|---|---|---|
| | журналы сервера, инструменты аналитики и т.д.;<br>- анализировать собранную статистику для выявления проблем и оптимизации производительности веб-приложений;<br>- умение предоставлять отчеты и рекомендации по улучшению работы веб-приложений на основе собранной статистики;<br>- подключать и настраивать системы мониторинга работы Веб-приложений и сбора статистики его использования;<br>- работать с системами продвижения веб приложений;<br>- публиковать информации о веб приложении в специальных справочниках и каталогах;<br>- осуществлять подбор и анализ ключевых слов и фраз для соответствующей предметной области с использованием специализированных программных средств;<br>- составлять тексты, включающие ссылки на продвигаемый сайт, для размещения на сайтах партнеров;<br>- осуществлять оптимизацию | основе анализа собранной статистики;<br>- принципы функционирования поисковых сервисов;<br>- виды и методы расчета индексов цитируемости веб-приложений (ТИЦ, ВИЦ);<br>- стратегии продвижения веб-приложенийв сети Интернет;<br>- виды поисковых запросов пользователей в интернете;<br>- программные средства и платформы для подбора ключевых словосочетаний, отражающих специфику сайта;<br>- инструменты сбора и анализа поисковых запросов;<br>- основные принципы маркетинга и продвижения приложений;<br>- целевую аудиторию и конкурентную среду в сфере приложений;<br>- различные инструменты и платформы для создания и оптимизации контента, таких как WordPress;<br>- основные методы рекламы и продвижения в интернете, включая контекстную рекламу, рекламу в социальных сетях и партнерские программы; методы анализа эффективности мероприятия по | |

| | | | |
|---|---|---|---|
| | приложений с целью повышения его рейтинга в сети интернет;<br>- умение разрабатывать и реализовывать стратегии продвижения приложений, учитывая цели бизнеса и потребности целевой аудитории;<br>- проводить маркетинговые исследования для определения целевой аудитории и конкурентной среды;<br>- создавать качественный контент для привлечения и удержания пользователей, включая описания приложения, видео обзоры, блоги и социальные медиа публикации;<br>- анализировать эффективность мероприятий по продвижению и оптимизировать стратегию на основе полученных результатов. | продвижению и оптимизации стратегии на основе полученных результатов. | |

## 4 Объем и содержание учебной дисциплины

Учебная деятельность обучающихся предусматривает учебные занятия (лекция, практическое занятие, консультация), самостоятельную работу, а также другие виды учебной деятельности, определенные учебным планом и календарным планом воспитательной работы.

### Объем учебной дисциплины

| Семестр / курс | ***ИТОГО*** | ***7 семестр*** |
|---|---|---|
| Форма промежуточной | | *экзамен* |

| | | |
|---|---|---|
| аттестации | | |
| Трудоёмкость, *академ. час.* | ***132*** | *132* |
| Лекции, *академ. час.* | ***16*** | *16* |
| в форме практической подготовки | ***0*** | *0* |
| Лабораторные работы, *академ. час.* | ***0*** | *0* |
| в форме практической подготовки | ***0*** | *0* |
| Практические занятия, *академ. час.* | ***48*** | *48* |
| в форме практической подготовки | ***0*** | *0* |
| Курсовая работа / проект, *академ. час.* | ***0*** | *0* |
| в форме практической подготовки | ***0*** | *0* |
| Консультации, *академ. час.* | ***1*** | *1* |
| в форме практической подготовки | ***0*** | *0* |
| Самостоятельная работа, *академ. час.* | ***61*** | *61* |
| в форме практической подготовки | ***0*** | *0* |
| Контроль, *академ. час.* | ***6*** | *6* |
| в форме практической подготовки | ***0*** | *0* |

## Содержание учебной дисциплины

Раздел 1 Технологии серверной разработки и интеграции данных;
Тема 1.1 Введение. Основы PHP;
Тема 1.2 Формы;
Тема 1.3 Cookie. HTTP-заголовки ответа сервера. Сессии;
Тема 1.4 Работа с файловой системой;
Тема 1.5 Основы работы с базами данных;

Тема 1.6 Связь с базами данных MySQL;

Тема 1.7 Объектно-ориентированное программирование на PHP;

Тема 1.8 PHP и JSON;

Тема 1.9 Сокеты и сетевые функции;

Тема 1.10 CRUD;

Тема 1.11 Бэкенд: Node.js (Express), Python (Django, Flask), Ruby on Rails;

Тема 1.12 Работа с REST API и GraphQL;

Тема 1.13 Получение данных и их распарсивание;

Раздел 2 Клиентские технологии, CMS и эксплуатация веб-приложений;

Тема 2.1 Фронтенд: React, Vue.js, Angular;

Тема 2.2 Язык сценариев JavaScript. Объектно-ориентированное программирование;

Тема 2.3 CMS;

Тема 2.4 Жизненный цикл веб-приложения: build, deploy, production.

## 5 Перечень тем лекций

| № раздела / темы дисциплины | Темы лекций | Трудоемкость, *академ.час* | |
|---|---|---|---|
| | | всего | в форме практической подготовки |
| Раздел 1. | Технологии серверной разработки и интеграции данных | 8 | |
| Раздел 2. | Клиентские технологии, CMS и эксплуатация веб-приложений | 8 | |
| **Итого:** | | **16** | **0** |

## 6 Перечень тем практических занятий (семинаров)

| № раздела / темы дисциплины | Темы практических занятий (семинаров) | Трудоемкость, *академ.час* | |
|---|---|---|---|
| | | всего | в форме практической подготовки |
| Тема 1.1. | Создание серверных сценариев с использованием технологии PHP | 4 | |
| Тема 1.2. | Обработка данных из формы | 4 | |
| Тема 1.3. | Отслеживание сеансов (session) | 4 | |
| Тема 1.4. | Организация файлового ввода-вывода | 4 | |

| | | | |
|---|---|---|---|
| Тема 1.6. | Организация поддержки базы данных в PHP. Облачные БД (Firebase, Supabase) | 4 | |
| Тема 1.11. | Использование фреймворка для создания серверной части сайта | 4 | |
| Тема 1.12. | Создание REST API | 4 | |
| Тема 1.13. | Интеграция фронтенда и бэкенда | 4 | |
| Тема 2.1. | Использование фреймворка для создания клиентской части сайта. Разработка фронтенда с использованием AI-агентов | 4 | |
| Тема 2.2. | Разработка SPA. Автоматизированное тестирование фронтенда с помощью Jest, React Testing Library | 4 | |
| Тема 2.3. | Создание сайта на CMS | 4 | |
| Тема 2.4. | Жизненный цикл веб-приложения: build, deploy, production. Облачные серверы и хостинги. Публикация сайта на бесплатном хостинге.Docker: развёртывание сторонних сервисов, контейнеризация разработанного приложения | 4 | |
| **Итого:** | | **48** | **0** |

## 7 Перечень тем лабораторных работ

| **№ раздела / темы дисциплины** | **Темы лабораторных работ** | **Трудоемкость, *академ.час*** | |
|---|---|---|---|
| | | **всего** | **в форме практической подготовки** |
| | *Отсутствуют* | | |
| **Итого:** | | **0** | **0** |

## 8 Перечень тем курсовых работ (проектов)

| **№ раздела / темы дисциплины** | **Темы курсовых работ (проектов)** | **Трудоемкость, *академ.час*** | |
|---|---|---|---|
| | | **всего** | **в форме практической подготовки** |
| | *Отсутствуют* | | |
| **Итого:** | | **0** | **0** |

## 9 Виды самостоятельной работы

| № раздела / темы дисциплины | Виды самостоятельной работы | Трудоемкость, *академ.час* | |
|---|---|---|---|
| | | **всего** | **в форме практической подготовки** |
| Раздел 1. | 1. Изучение лекционного материала;<br>2. Подготовка к практическому занятию;<br>3. Прохождение тестирования. | 31 | |
| Раздел 2. | 1. Изучение лекционного материала;<br>2. Подготовка к практическому занятию;<br>3. Прохождение тестирования. | 30 | |
| | *Консультации* | 1 | |
| *Контроль* | *Подготовка к экзамену* | 6 | |
| **Итого:** | | **68** | **0** |

## 10 Матрица компетентностных задач по учебной дисциплине

| Контролируемые разделы (темы) дисциплины | Контролируемые компетенции | Оценочные средства |
|---|---|---|
| Раздел 1. | ОК 01.;<br>ОК 02.;<br>ОК 05.;<br>ОК 09.;<br>ПК 3.1.;<br>ПК 3.2.;<br>ПК 3.4.;<br>ПК 3.7. | 1. Наблюдение за работой на практическом занятии;<br>2. Тестовые задания |
| Раздел 2. | ОК 01.;<br>ОК 02.;<br>ОК 05.;<br>ОК 09.;<br>ПК 3.1.;<br>ПК 3.2.;<br>ПК 3.4.;<br>ПК 3.7. | 1. Наблюдение за работой на практическом занятии;<br>2. Тестовые задания |
| Экзамен. | ОК 01.;<br>ОК 02.;<br>ОК 05.;<br>ОК 09.;<br>ПК 3.1.;<br>ПК 3.2.;<br>ПК 3.4.;<br>ПК 3.7. | 1. Вопросы к экзамену;<br>2. Задачи к экзамену |

## 11 Входной контроль

**11.1 Цель входного контроля** − определить начальный уровень подготовленности обучающихся и выстроить индивидуальную траекторию обучения. В условиях личностно-ориентированной образовательной среды результаты входного оценивания обучающегося используются как начальные значения в индивидуальном профиле академической успешности обучающегося.

### 11.2 Описание оценочных средств

Форма проведения входного контроля – тестирование.

Длительность тестирования – 20 минут.

Тестирование проводится в течение первых двух недель изучения учебной дисциплины в СУО СибГИУ.

Для проведения тестирования используются тестовые задания для входного контроля (для ответа на 1 вопрос, как правило, выделяется 1 минута).

### Шкала оценивания результатов тестирования

За каждый правильный ответ выставляется 1 балл. Оценка выставляется в зависимости от процента правильных ответов:

Оценка формируется в соответствии с критериями таблицы:

| Процент результативности (правильных ответов) | Оценка |
|---|---|
| 86 до 100 | «отлично» |
| от 67до 85 | «хорошо» |
| от 50 до 66 | «удовлетворительно» |
| менее 50 | «неудовлетворительно» |

Основываясь на результатах тестирования, формируется индивидуальная траектория обучающегося (группы обучающихся), исходя из следующих соображений:

– 86 – 100 % – обучающийся готов к изучению учебной дисциплины без дополнительного повторения изученного ранее материала;

– 50 – 67 % – обучающийся в целом готов к изучению учебной дисциплины, требуется повторение отдельных вопросов(тем) изученного ранее материала, необходимого для успешного освоения учебной дисциплины;

– 49% и менее – обучающийся имеет слабую подготовку к изучению учебной дисциплины, необходимы корректирующие занятия и консультации.

**Тестовые задания для входного контроля знаний обучающихся приведены в приложении.**

## 12 Текущий контроль

**12.1 Текущий контроль знаний** используется для оперативного и регулярного управления учебной деятельностью (в том числе самостоятельной) обучающихся. Данный вид контроля стимулирует у обучающихся стремление к систематической самостоятельной работе по изучению учебной дисциплины и позволяет оценить степень освоения изученного материала.

Текущий контроль успеваемости осуществляется в течение семестра, в ходе повседневной учебной работы по индивидуальной инициативе педагогического работника, в том числе посредством использования СУО СибГИУ.

Формы проведения текущего контроля – наблюдение за работой на практическом занятии; тестовые задания.

### 12.2 Описание оценочных средств

### Шкалы оценивания (методики оценки)

#### 12.2.1 Рекомендации по оцениванию результатов тестирования обучающихся

Тестирование проводится по изученным разделам (темам) учебной дисциплины для выявления уровня умений и знаний обучающегося, мотивирования обучающегося к активизации самостоятельной работы по усвоению учебного материала, выявления и устранения пробелов в знаниях, повышения дисциплины и организации учебной деятельности обучающегося.

Тестирование проводится в СУО СибГИУ в рамках часов, отведенных на самостоятельную работу обучающихся. Для тестирования используется 25 вопросов из базы типовых тестовых заданий по изученным разделам (темам) учебной дисциплины. Обучающийся выбирает один ответ из предложенных вариантов ответа. При этом тест может содержать вопросы различных типов: с выбором одного правильного ответа, с выбором нескольких правильных ответов, с открытым ответом (ввод текста) и на установление соответствия. В вопросах с выбором нескольких ответов и на соответствие ответ засчитывается как правильный только при полном и точном выполнении (указаны все верные варианты или установлены все соответствия). В вопросах с открытым ответом ответ признаётся верным при совпадении с эталоном. За правильный ответ на каждый вопрос (независимо от его типа) обучающийся получает 1 балл, за неправильный – 0 баллов.

Оценка формируется в соответствии с критериями таблицы:

| Максимальный балл | Проходной балл | Оценка |
|---|---|---|
| 100% | 50% | Зачтено |

| | | |
|---|---|---|
| 49% | - | Не зачтено |

### 12.2.2 Рекомендации по оцениванию в процессе наблюдения за работой на практическом занятии

Наблюдение за работой на практическом занятии – важнейший компонент контроля деятельности обучающихся и качества их обучения.

**Критерии оценки в процессе наблюдения за работой на практическом занятии:**

– самостоятельность выполнения задания и работы с методическими указаниями (учитывается работа в течение занятия, быстрота и способность нахождения в методических указаниях нужной информации);

– правильность выполнения работы (учитывается последовательность выполняемых действий, умение работать с инструментами и/или средствами вычислительной техники);

– рациональность использования времени, отведенного на задание (не одобряется затянутость выполнения работы с учетом индивидуальных особенностей обучающихся);

– активность обучающегося при выполнении практической работы (учитывается индивидуальная работа и работа в группе).

Критерии оценки по результатам наблюдения за работой на практическом занятии

| Наименование параметра оценки | Максимальное количество баллов |
|---|---|
| Самостоятельно выполняет задания; умеет работать с методическими указаниями. | 2,5 |
| Указанные в методических указаниях действия выполняет в правильной последовательности; умеет работать с инструментами и/или средствами вычислительной техники. | 1,25 |
| Рационально использует время, отведенное на задание; проявляет активность при выполнении практической работы. | 1,25 |
| **Итого** | **5** |

Максимальное количество – 5 баллов соответствует оценке «отлично», 4 балла – «хорошо», 3 балла – «удовлетворительно», 2 балла – «неудовлетворительно».

**Задания и вопросы для текущего контроля знаний обучающихся по разделам (темам) учебной дисциплины приведены в приложении.**

## 13 Промежуточная аттестация

**13.1 Оценочные материалы для промежуточной аттестации** обучающихся по учебной дисциплине предназначены для оценки

степени достижения запланированных результатов обучения по завершению изучения учебной дисциплины в установленной учебным планом форме и позволяют определить качество усвоения изученного материала.

Итоговой формой контроля сформированности компетенций у обучающихся по учебной дисциплине является: экзамен.

Оценочные материалы для промежуточной аттестации обучающихся включают: вопросы к экзамену; задачи к экзамену.

### 13.2 Оценивание обучающегося на экзамене

| Оценка | Требования к знаниям |
|---|---|
| отлично | Оценка «отлично» выставляется обучающемуся, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал различной литературы, правильно обосновывает принятое нестандартное решение, обладает приемами выполнения практических задач по формированию компетенций. |
| хорошо | Оценка «хорошо» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, обладает необходимыми умениями и приемами их выполнения. |
| удовлетворительно | Оценка «удовлетворительно» выставляется обучающемуся, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, испытывает сложности при выполнении практических работ и затрудняется связать теорию вопроса с практикой. |
| неудовлетворительно | Оценка «неудовлетворительно» выставляется обучающемуся, который не знает значительной части программного материала, неуверенно отвечает, допускает серьезные ошибки, не имеет представлений по методике выполнения практической работы. Как правило, оценка «неудовлетворительно» ставится обучающимся, которые не могут продолжить обучение без дополнительных занятий по данной учебной дисциплине. |

**Задания и вопросы для промежуточной аттестации обучающихся приведены в приложении.**

## 14 Учебно-методическое и информационное обеспечение учебной дисциплины

**а) основная литература:**

1 Диков, А. В. Клиентские технологии веб-дизайна. HTML5 и CSS3 / А. В. Диков. — 2-е изд., стер. — Санкт-Петербург : Лань, 2023. — 188 с. — ISBN 978-5-507-46740-2. – URL: https://e.lanbook.com/book/318443 (дата обращения: 07.05.2026);

2 Полуэктова, Н. Р. Разработка веб-приложений : учебник для среднего профессионального образования / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 204 с. — ISBN 978-5-534-18644-4. – URL: https://urait.ru/bcode/588469 (дата обращения: 07.05.2026);

3 Сергеев, А. Н. Создание сайтов на основе WordPress : учебное пособие для СПО / А. Н. Сергеев. — 5-е изд., стер. — Санкт-Петербург : Лань, 2025. — 120 с. — ISBN 978-5-507-51004-7. – URL: https://e.lanbook.com/book/497675 (дата обращения: 07.05.2026).

**б) дополнительная литература:**

1 Заяц, А. М. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js / А. М. Заяц, Н. П. Васильев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 120 с. — ISBN 978-5-507-45423-5. – URL: https://e.lanbook.com/book/269867 (дата обращения: 07.05.2026);

2 Тузовский, А. Ф. Проектирование и разработка web-приложений : учебник для среднего профессионального образования / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 219 с. — ISBN 978-5-534-16767-2. – URL: https://urait.ru/bcode/565693 (дата обращения: 07.05.2026);

**в) ресурсы информационно-телекоммуникационной сети «Интернет»:**

1 Электронный каталог : сайт / Научно-техническая библиотека СибГИУ. – Новокузнецк, [199 – ]. – URL: http://libr.sibsiu.ru. – URL: https://libr.sibsiu.ru;

2 Электронная библиотека // Научно-техническая библиотека СибГИУ : сайт. – Новокузнецк, [200 – ]. – URL: http://library.sibsiu.ru/LibrELibraryFullText.asp. – Режим доступа: для авторизир. пользователей. – URL: https://library.sibsiu.ru/LibrELibraryFullText.asp;

3 Консультант студента : электронно-библиотечная система / ООО «КОНСУЛЬТАНТ СТУДЕНТА». – Москва, [200 – ]. – URL: http://www.studentlibrary.ru. – Режим доступа: для авторизир. пользователей;

4 ЛАНЬ : электронно-библиотечная система : [Единая профессиональная база знаний издательства Лань для СПО - Издательство Лань (СПО)] / ООО «Издательство ЛАНЬ». – Санкт-Петербург, [200 – ]. – URL: http://e.lanbook.com. – Режим доступа: для авторизир. пользователей;

5 Образовательная платформа ЮРАЙТ / ООО «Электронное издательство ЮРАЙТ». – Москва, [200 – ]. – URL: https://urait.ru. – Режим доступа: для авторизир. пользователей;

6 Университетская библиотека онлайн : электронно-библиотечная система / ООО «Директ-Медиа». – Москва, [200 – ]. – URL: https://biblioclub.ru. – Режим доступа: для авторизир. пользователей;

7 НАУЧНАЯ ЭЛЕКТРОННАЯ БИБЛИОТЕКА eLIBRARY.RU : база данных / ООО «НЭБ». – Москва, [200 – ]. – URL: http://elibrary.ru. – Режим доступа: по подписке;

8 Электронные периодические издания ИВИС : универсальная база данных / ООО «ИВИС». – Москва, [200 – ]. – URL: http://eivis.ru. – Режим доступа: по подписке.

**г) лицензионное и свободно распространяемое программное обеспечение:**

– 7-Zip;
– Adobe Acrobat Reader;
– Astra Linux Special Edition;
– GIMP;
– Kaspersky Endpoint Security;
– Microsoft Office;
– Microsoft Windows;
– OnlyOffice;
– PyCharm;
– Visual Studio;
– Р7-Офис.

**д) базы данных и информационно-справочные системы:**

1 ГАРАНТ : справочно-правовая система / ООО «Правовой центр «Гарант». – Кемерово, [200 – ]. – Режим доступа: компьютерная сеть Сиб. гос. индустр. ун-та.;

2 КонсультантПлюс : справочно-правовая система / ООО «Информационный центр АНВИК». – Новокузнецк, [199 – ]. – Режим доступа: компьютерная сеть библиотеки Сиб. гос. индустр. ун-та.;

3 Техэксперт : информационно-справочная система / ООО «Группа компаний «Кодекс». – Кемерово, [200 – ]. – Режим доступа: компьютерная сеть Сиб. гос. индустр. ун-та.

## 15 Материально-техническое обеспечение учебной дисциплины

Материально-техническое обеспечение учебной дисциплины включает учебные аудитории для проведения занятий всех видов, предусмотренных учебным планом, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы, мастерские и лаборатории, оснащенные оборудованием, техническими средствами обучения и материалами, учитывающими требования

международных стандартов, научно-техническую библиотеку СибГИУ.

Для проведения занятий лекционного типа (лекций) предусмотрена аудитория, оборудованная компьютерной техникой, экраном и мультимедийным проектором, учебной доской (ауд. 503М); Для проведения практических занятий предусмотрена лаборатория (ауд. 401М), оснащенная оборудованием и техническими средствами обучения: рабочее место преподавателя, посадочные места обучающихся (по количеству обучающихся), компьютеры с лицензионным программным обеспечением, проектором и экраном; доской.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к информационно-телекоммуникационной сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС СПО по направлению подготовки (специальности) 09.02.11 «Разработка и управление программным обеспечением».

Составитель(и):

доцент Маслова Елена Владимировна (кафедра прикладных информационных технологий и программирования).

Рабочая программа дисциплины рассмотрена и утверждена на заседании кафедры.

Согласована:

Системный аналитик группы
системной архитектуры департамента
информационных технологий
ООО "СИ СОФТ" Е.А. Шапошникова

## Приложение

## Тестовые задания для входного контроля знаний обучающихся

**Выберите один правильный ответ.**

Вопрос 1
Что такое алгоритм?
- A) Программа на языке Python
- B) Описание последовательности действий для решения задачи
- C) Компьютерная игра
- D) Тип данных в базе данных

Вопрос 2
Какой тип данных используется для хранения целых чисел в большинстве языков программирования?
- A) float
- B) string
- C) int
- D) bool

Вопрос 3
Чему равно значение переменной a после выполнения кода: a = 5 + 3 * 2?
- A) 16
- B) 11
- C) 13
- D) 10

Вопрос 4
Какой оператор используется для проверки равенства в большинстве языков программирования?
- A) =
- B) ==
- C) :=
- D) ===

Вопрос 5
Что такое цикл в программировании?
- A) Ошибка в коде
- B) Конструкция для многократного выполнения блока кода
- C) Тип переменной
- D) Функция вывода на экран

Вопрос 6
Какой тег HTML используется для создания гиперссылки?
- A) <link>
- B) <a>
- C) <href>
- D) <url>

Вопрос 7
Что такое переменная в программировании?
- A) Постоянное значение
- B) Именованная область памяти для хранения данных
- C) Тип ошибки
- D) Название программы
Вопрос 8
Какой из перечисленных является логическим типом данных?
- A) 123
- B) "строка"
- C) true
- D) 3.14
Вопрос 9
Что делает оператор % (остаток от деления) при вычислении 10 % 3?
- A) 3.33
- B) 1
- C) 0
- D) 30
Вопрос 10
Какой тег HTML используется для создания таблицы?
- A) <table>
- B) <grid>
- C) <tab>
- D) <list>
Вопрос 11
Что такое функция в программировании?
- A) Математическая формула
- B) Именованный блок кода, который можно вызвать многократно
- C) Тип данных
- D) Оператор цикла
Вопрос 12
Какой протокол используется для передачи веб-страниц в Интернете?
- A) FTP
- B) SMTP
- C) HTTP
- D) TCP
Вопрос 13
Чему равно значение выражения 2^3 (два в степени три) в программировании?
- A) 6
- B) 8
- C) 9
- D) 23
Вопрос 14

Какой символ обычно используется для однострочного комментария в Python?
- A) //
- B) /* */
- C) #
- D) --

Вопрос 15
Что такое массив (список) в программировании?
- A) Одна переменная
- B) Структура данных для хранения нескольких значений под одним именем
- C) Тип цикла
- D) Функция сортировки

Вопрос 16
Какой тег HTML используется для заголовка первого уровня?
- A) <head>
- B) <header>
- C) <h1>
- D) <title>

Вопрос 17
Что такое условный оператор if?
- A) Оператор цикла
- B) Конструкция для выполнения кода при определённом условии
- C) Функция ввода данных
- D) Оператор присваивания

Вопрос 18
Какое значение получит переменная x после выполнения: x = "5" + "3" (конкатенация строк)?
- A) 8
- B) 53
- C) "8"
- D) Ошибка

Вопрос 19
Что такое IP-адрес?
- A) Название веб-сайта
- B) Уникальный числовой идентификатор устройства в сети
- C) Тип файла
- D) Протокол передачи данных

Вопрос 20
Какой из перечисленных НЕ является языком программирования?
- A) Python
- B) Java
- C) HTML
- D) C++

**Задания и вопросы для текущего контроля знаний обучающихся по разделам (темам) учебной дисциплины**

**2 Тестирование по результатам освоения лекционного материала и самостоятельной работы**

Тестирование проводится для определения степени усвоения материалов лекций, а также выполнения самостоятельной работы в рамках часов, отведенных на самостоятельную работу обучающихся, в СУО
СибГИУ.

**Раздел 1 Технологии серверной разработки и интеграции данных**

1. Какое расширение файла используется для PHP-скриптов?
   а) .html
   б) .php
   в) .js
   г) .py
2. Какая конструкция PHP используется для вывода данных на экран?
   а) print_r()
   б) var_dump()
   в) echo
   г) output()
3. Какой HTTP-метод используется для передачи данных формы в теле запроса (без отображения в URL)?
   а) GET
   б) POST
   в) PUT
   г) HEAD
4. Что такое cookie в веб-разработке?
   а) Небольшой фрагмент данных, сохраняемый браузером на стороне клиента
   б) Файл на сервере с данными пользователя
   в) Зашифрованный токен авторизации
   г) Специальный HTTP-метод
5. Какая PHP-функция используется для установки cookie?
   а) session_start()
   б) header()
   в) setcookie()
   г) cookie_set()
6. Что означает аббревиатура CRUD?
   а) Create, Read, Update, Delete
   б) Connect, Retrieve, Upload, Drop
   в) Code, Run, Update, Deploy
   г) Configure, Register, Use, Delete
7. Какой порт по умолчанию использует СУБД MySQL?

а) 80
б) 443
в) 3306
г) 5432

8. Какой формат данных обычно используется в REST API для обмена информацией?
а) XML
б) CSV
в) JSON
г) YAML

9. Что такое сессия (session) в веб-программировании?
а) Временное хранилище данных о пользователе на стороне сервера
б) Долгосрочный cookie с токеном
в) Соединение по протоколу WebSocket
г) JWT-токен авторизации

10. Какой PHP-метод объекта PDO применяется для подготовки SQL-запроса с параметрами?
а) query()
б) exec()
в) prepare()
г) run()

11. Как называется PHP-функция, которая преобразует JSON-строку в массив или объект PHP?

12. Как называется HTML-тег, используемый для создания интерактивной формы?

13. Как называется SQL-команда для добавления новой записи в таблицу?

14. Как называется широко применяемая система контроля версий, используемая при разработке веб-приложений?

15. Как называется тип архитектуры базы данных, при которой данные хранятся в таблицах, связанных между собой ключами?

16. Что из перечисленного относится к инструментам мониторинга и аналитики веб-приложений?
а) Nagios
б) Zabbix
в) Google Analytics
г) ELK Stack
д) Bootstrap

17. Какие составляющие включает в себя REST API?
а) URL-адрес ресурса (endpoint)
б) HTTP-метод (GET, POST, PUT, DELETE)
в) Код состояния HTTP (status code)
г) Тело запроса/ответа (обычно JSON)
д) Компилятор

18. Установите соответствие между HTTP-методами и их назначением:

| | |
|---|---|
| 1. GET | а) Создание новых данных на сервере |
| 2. POST | б) Получение данных с сервера |
| 3. PUT | в) Удаление данных |
| 4. DELETE | г) Полное обновление существующих данных |

19. Установите соответствие между PHP-функциями и их назначением:

| | |
|---|---|
| 1. mysqli_connect() | а) Преобразование массива PHP в JSON-строку |
| 2. json_encode() | б) Инициализация сессии |
| 3. session_start() | в) Подключение к базе данных MySQL |
| 4. setcookie() | г) Установка cookie-файла |

20. Установите соответствие между операциями CRUD и SQL-командами:

| | |
|---|---|
| 1. Create | а) SELECT |
| 2. Read | б) UPDATE |
| 3. Update | в) INSERT |
| 4. Delete | г) DELETE |

21. Установите соответствие между фреймворками и языками программирования:

| | |
|---|---|
| 1. Express | а) Python |
| 2. Django | б) Ruby |
| 3. Laravel | в) Node.js (JavaScript) |
| 4. Ruby on Rails | г) PHP |

22. Установите соответствие между кодами состояния HTTP и их значением:

| | |
|---|---|
| 1. 200 | а) Ошибка на сервере |
| 2. 404 | б) Успешный запрос |
| 3. 401 | в) Ресурс не найден |
| 4. 500 | г) Требуется авторизация |

23. Установите соответствие между понятиями и их определениями:

| | |
|---|---|
| 1. Сессия | а) Данные, сохраняемые браузером на стороне клиента |
| 2. Cookie | б) Текстовый формат обмена данными на основе JavaScript |
| 3. REST API | в) Хранилище данных пользователя на сервере, идентифицируемое по ID |
| 4. JSON | г) Архитектурный стиль взаимодействия компонентов системы через HTTP |

24. Как называется PHP-функция, которую необходимо вызвать перед началом работы с сессией?

25. Какой глобальный массив PHP содержит данные, переданные методом POST?

**Раздел 2 Клиентские технологии, CMS и эксплуатация веб-приложений**

1. Что означает аббревиатура SPA применительно к веб-приложениям?
    а) Server PHP Application
    б) Single Page Application – одностраничное приложение
    в) Static Page Application
    г) Style Page Adapter
2. Какой хук React используется для хранения и изменения состояния функционального компонента?
    а) useEffect()
    б) useState()
    в) useContext()
    г) useRef()
3. Что такое CMS в веб-разработке?
    а) Система управления контентом
    б) Язык программирования для клиентской части
    в) Протокол передачи данных
    г) Инструмент автоматического тестирования
4. Какая npm-команда используется для создания production-сборки React-приложения?
    а) npm start
    б) npm test
    в) npm run build
    г) npm deploy
5. Что такое компонент в React?
    а) CSS-класс для стилизации элемента
    б) Независимый, переиспользуемый фрагмент пользовательского интерфейса
    в) Функция-обработчик HTTP-запроса
    г) Модуль базы данных
6. Какой инструмент используется для контейнеризации веб-приложений?
    а) Git
    б) npm
    в) Docker
    г) Webpack
7. Что такое Bootstrap?
    а) Серверный фреймворк на JavaScript
    б) CSS-фреймворк для создания адаптивных интерфейсов
    в) Система управления базами данных
    г) Язык программирования серверной части
8. Какой метод JavaScript применяется для добавления обработчика событий к элементу DOM?
    а) attachEvent()
    б) on()
    в) addEventListener()
    г) bindEvent()

9. Что такое Virtual DOM в React?
    а) База данных для хранения элементов страницы
    б) Легковесное представление реального DOM в памяти для оптимизации обновлений
    в) CSS-класс для анимации элементов
    г) Метод оптимизации SQL-запросов
10. Какой фреймворк использует директивы v-if и v-for для условного рендеринга и перебора данных?
    а) React
    б) Angular
    в) Vue.js
    г) Svelte
11. Как называется популярный инструмент для unit-тестирования JavaScript-кода, разработанный Facebook?
12. Как называется популярная CMS с открытым исходным кодом, на которой работает более 40% сайтов в интернете?
13. Как называется процесс публикации (размещения) веб-приложения на сервере, делающий его доступным пользователям?
14. Какие из перечисленных платформ являются популярными CMS?
    а) WordPress
    б) Joomla
    в) Drupal
    г) Redux
    д) 1С-Битрикс
15. Какие концепции ООП реализованы в современном JavaScript (ES6+)?
    а) Классы (class)
    б) Наследование (extends)
    в) Инкапсуляция (приватные поля #)
    г) Статическая типизация
    д) Прототипное наследование
16. Что из перечисленного относится к инструментам тестирования фронтенда?
    а) Jest
    б) React Testing Library
    в) Cypress
    г) Postman
    д) Selenium
17. Какие утверждения о Docker являются верными?
    а) Docker позволяет создавать изолированные контейнеры для приложений
    б) Dockerfile содержит инструкции для сборки образа контейнера
    в) Docker Compose позволяет управлять несколькими контейнерами одновременно
    г) Docker является языком программирования

д) Docker Hub – это публичный реестр образов контейнеров

18. Установите соответствие между фреймворком и его характерной особенностью:

1. React а) Использует TypeScript, двустороннее связывание данных
2. Vue.js б) Использует JSX и концепцию Virtual DOM
3. Angular в) CSS-фреймворк для адаптивной вёрстки
4. Bootstrap г) Использует директивы v-if, v-for, двустороннее связывание

19. Установите соответствие между этапами жизненного цикла и их описанием:

1. Development а) Сборка оптимизированной версии приложения
2. Build б) Написание и отладка кода
3. Deploy в) Работающее приложение, доступное конечным пользователям
4. Production г) Публикация приложения на удалённом сервере

20. Установите соответствие между хуками React и их назначением:

1. useState а) Выполнение побочных эффектов после рендеринга
2. useEffect б) Получение ссылки на DOM-элемент без вызова рендеринга
3. useContext в) Управление локальным состоянием компонента
4. useRef г) Получение значения из React-контекста

21. Установите соответствие между командами npm и их назначением:

1. npm install а) Запуск тестов
2. npm start б) Установка зависимостей проекта
3. npm run build в) Запуск приложения в режиме разработки
4. npm test г) Создание production-сборки

22. Установите соответствие между понятиями асинхронного JavaScript и их описанием:

1. Promise а) Функция, передаваемая как аргумент другой функции
2. async/await б) Объект, представляющий результат асинхронной операции
3. Callback в) Механизм обработки событий JavaScript
4. Event Loop г) Синтаксический способ работы с промисами

23. Установите соответствие между видами тестирования и их описанием:

1. Unit-тестирование а) Проверка взаимодействия нескольких компонентов
2. Интеграционное тестирование б) Проверка отдельных функций и компонентов в изоляции
3. E2E-тестирование в) Проверка, что новые изменения не нарушили работу

4. Регрессионное тестирование г) Проверка сценариев работы пользователя от начала до конца

24. Как называется встроенный менеджер пакетов, поставляемый вместе с Node.js?

25. Как называется синтаксическое расширение JavaScript, позволяющее писать HTML-подобный код внутри JS-файлов React?

## 3 Выполнение практических работ

Задания для практических работ, теоретический и методический материал для самостоятельного изучения и вопросы для собеседования представлены в методических указаниях к практическим занятиям.

**Тема 1.1. Создание серверных сценариев с использованием технологии PHP**

Установите локальный веб-сервер (OpenServer или XAMPP) с поддержкой PHP и MySQL. Создайте PHP-скрипт, который запрашивает у пользователя через HTML-форму два числа, вычисляет их сумму, разность, произведение и частное, а затем выводит результаты в таблице. Реализуйте проверку деления на ноль. Оформите страницу с использованием встроенных CSS-стилей.

**Тема 1.2. Обработка данных из формы**

Создайте HTML-форму анкеты соискателя с полями: ФИО, дата рождения, email, телефон, желаемая должность (выпадающий список), уровень образования (радиокнопки), навыки (чекбоксы) и файл резюме. Напишите PHP-обработчик, который принимает данные методом POST, проверяет корректность email и телефона регулярными выражениями, фильтрует текст от HTML-тегов и выводит результат в виде структурированной карточки.

**Тема 1.3. Отслеживание сеансов (session)**

Разработайте многостраничный сайт с системой авторизации. Создайте форму входа с проверкой логина и пароля по заранее заданному массиву пользователей. При успешной авторизации запускайте сессию, сохраняйте имя пользователя и время входа. На защищённой странице выводите приветствие и счётчик посещений за текущую сессию. Реализуйте кнопку «Выход» с уничтожением сессии и перенаправлением на страницу входа.

**Тема 1.4. Организация файлового ввода-вывода**

Создайте PHP-приложение «Гостевая книга». Данные из формы (имя, email, сообщение, дата) сохраняйте в CSV-файл с проверкой на дублирование email. Реализуйте постраничный вывод записей из файла по 5 штук на страницу с сортировкой по дате. Добавьте возможность загрузки аватара с проверкой типа (JPG/PNG) и размера (до 1 МБ), сохраняя файл в папку uploads.

**Тема 1.6. Организация поддержки базы данных в PHP. Облачные БД (Firebase, Supabase)**

Создайте локальную базу данных «Библиотека» в MySQL через phpMyAdmin: таблица books (id, title, author, year, genre). Напишите PHP-скрипт на PDO для добавления, вывода списка и удаления книг с использованием подготовленных запросов. Затем зарегистрируйтесь в Supabase, создайте проект с таблицей books и подключитесь к ней через REST API из PHP с помощью cURL, выполнив аналогичные операции CRUD.

**Тема 1.11. Использование фреймворка для создания серверной части сайта**

Установите Node.js и фреймворк Express через npm. Создайте серверное приложение для управления задачами с маршрутами: GET /tasks (список задач из массива), GET /tasks/:id (одна задача по ID), POST /tasks (добавление задачи из JSON-тела запроса), PUT /tasks/:id (обновление статуса), DELETE /tasks/:id (удаление). Протестируйте все endpoints через Postman или Thunder Client и задокументируйте API в файле README.md.

**Тема 1.12. Создание REST API**

Разработайте собственное REST API на PHP для сущности «Товары» (products). Реализуйте endpoints: GET /api/products (список с пагинацией), GET /api/products/{id} (один товар), POST /api/products (создание с валидацией полей), PUT /api/products/{id} (полное обновление), DELETE /api/products/{id} (удаление). Данные храните в JSON-файле или подключите MySQL. Все ответы должны возвращаться в формате JSON с соответствующими HTTP-статусами (200, 201, 404, 400).

**Тема 1.13. Интеграция фронтенда и бэкенда**

Создайте полноценное приложение «Список дел» (Todo List). Фронтенд: HTML-страница с JavaScript, которая отправляет асинхронные запросы (fetch) к бэкенду. Бэкенд: PHP-скрипты, реализующие REST API для операций CRUD над задачами в базе данных MySQL. Реализуйте динамическое добавление задачи без перезагрузки, переключение статуса «выполнено», удаление с подтверждением и фильтрацию по статусу (все/активные/выполненные).

**Тема 2.1. Использование фреймворка для создания клиентской части сайта. Разработка фронтенда с использованием AI-агентов**

Установите React через Vite. Создайте компонент «Каталог товаров» с карточками, поиском и фильтрацией по категории. Используйте AI-ассистент для генерации кода компонента сортировки товаров по цене и рейтингу. Проанализируйте сгенерированный код: исправьте ошибки, оптимизируйте производительность и добавьте обработку пустого результата поиска. Оформите вывод в виде адаптивной сетки.

**Тема 2.2. Разработка SPA. Автоматизированное тестирование фронтенда с помощью Jest, React Testing Library**

Создайте одностраничное приложение (SPA) на React с двумя экранами: «Список пользователей» и «Профиль пользователя» с маршрутизацией (React Router). Данные получайте из публичного API (JSONPlaceholder). Установите Jest и React Testing Library. Напишите минимум три теста: проверка отображения списка пользователей после загрузки, проверка перехода на страницу профиля по клику, проверка отображения сообщения об ошибке при неудачном запросе.

**Тема 2.3. Создание сайта на CMS**

Установите WordPress на локальный сервер (OpenServer). Создайте сайт-визитку для вымышленной IT-компании: настройте тему, создайте страницы «Главная», «Услуги», «Портфолио», «Контакты» с использованием конструктора страниц. Добавьте меню навигации, виджеты в подвале и контактную форму через плагин. Опубликуйте минимум 3 записи в блоге с категориями и метками.

**Тема 2.4. Жизненный цикл веб-приложения: build, deploy, production. Облачные серверы и хостинги. Публикация сайта на бесплатном хостинге. Docker: развёртывание сторонних сервисов, контейнеризация разработанного приложения**

Соберите готовое React-приложение командой npm run build. Создайте Dockerfile для контейнеризации приложения на базе образа nginx:alpine, скопируйте папку build в директорию /usr/share/nginx/html. Соберите Docker-образ и запустите контейнер на порту 8080. Проверьте работу приложения в браузере по адресу localhost:8080.

## 4 Вопросы к экзамену для промежуточной аттестации

1. Что такое веб-приложение? Чем веб-приложение отличается от статического сайта? Приведите примеры веб-приложений.
2. Опишите архитектуру «клиент – сервер». Какую роль играет браузер, а какую – веб-сервер?
3. Что такое язык PHP? Какова его роль в серверной веб-разработке? Назовите основные области применения PHP.
4. Как работают HTML-формы? Какие методы передачи данных существуют? В чём разница между методами GET и POST?
5. Что такое cookie? Что такое сессия? В чём принципиальная разница между ними с точки зрения хранения данных?
6. Как осуществляется работа с файловой системой в PHP? Назовите основные функции для чтения и записи файлов.
7. Что такое реляционная база данных? Назовите основные понятия: таблица, строка, столбец, первичный ключ, внешний ключ.
8. Как организовать подключение к базе данных MySQL из PHP? Чем отличается расширение mysqli от PDO?

9. Что такое объектно-ориентированное программирование? Перечислите и объясните его основные принципы: инкапсуляция, наследование, полиморфизм, абстракция.
10. Как реализуется наследование в PHP? Приведите пример: базовый класс и производный класс.
11. Что такое JSON? Какова структура JSON-данных? Как PHP работает с JSON: функции json_encode() и json_decode().
12. Что такое CRUD? Напишите примеры SQL-запросов для каждой из четырёх операций применительно к таблице пользователей.
13. Чем Node.js отличается от PHP как платформы для серверной разработки? Какие задачи лучше решает Node.js?
14. Что такое Express.js? Для чего он используется? Опишите структуру простого Express-сервера.
15. Что такое REST API? Перечислите основные принципы REST. Как REST API используется для взаимодействия фронтенда и бэкенда?
16. Чем GraphQL отличается от REST API? В каких случаях GraphQL может быть предпочтительнее?
17. Что такое HTTP-заголовки? Приведите примеры заголовков запроса и ответа. Для чего используется заголовок Content-Type?
18. Что такое техническое задание (ТЗ) на веб-приложение? Какие обязательные разделы должно включать ТЗ? Какие стандарты регулируют оформление ТЗ?
19. Что такое фронтенд и бэкенд? Как они взаимодействуют? Что такое «полный стек» (full stack) разработки?
20. Что такое React? Перечислите ключевые концепции React: компонент, props, state, Virtual DOM.
21. В чём разница между функциональным и классовым компонентом в React? Почему современная разработка предпочитает функциональные компоненты?
22. Что такое хуки (hooks) в React? Опишите назначение хуков useState, useEffect и useContext. Приведите примеры использования.
23. Что такое Vue.js? Чем Vue.js отличается от React? Объясните понятия: директива, реактивность, однофайловый компонент (.vue).
24. Что такое CMS (система управления контентом)? Приведите примеры. В каких случаях разработка на CMS предпочтительнее самостоятельной разработки?
25. Что такое жизненный цикл веб-приложения? Опишите этапы: разработка, сборка (build), тестирование, деплой (deploy), эксплуатация (production).

## 5 Задачи к экзамену для промежуточной аттестации

Задача 1.
Напишите PHP-скрипт, выводящий на экран строку «Привет, мир!». Объясните, что означает каждая конструкция вашего кода.

Задача 2.
Создайте HTML-форму для регистрации пользователя, содержащую поля: имя (текст), email (email), пароль (password), кнопку «Зарегистрироваться». Данные должны отправляться методом POST на страницу register.php.
Задача 3.
Напишите PHP-скрипт для обработки данных из формы регистрации: получить имя и email, проверить, что поля не пустые, и вывести сообщение «Добро пожаловать, [Имя]! Ваш email: [email]» или сообщение об ошибке.
Задача 4.
Напишите PHP-скрипт, который устанавливает cookie с именем «username» и значением «Иван» сроком на 7 дней. Затем при следующем запросе читает этот cookie и выводит приветствие с именем.
Задача 5.
Напишите PHP-скрипт, реализующий простую авторизацию через сессию: если пользователь отправил логин «admin» и пароль «1234», записать данные в сессию и перенаправить на страницу dashboard.php; иначе – вывести сообщение об ошибке.
Задача 6.
Напишите PHP-скрипт для работы с файлом log.txt: при каждом обращении к скрипту добавляйте в файл строку с текущей датой и временем, затем выводите последние 10 строк файла.
Задача 7.
Напишите SQL-запросы для создания таблицы users со следующими полями: id (первичный ключ, автоинкремент), name (строка, не пустая), email (строка, уникальная), created_at (дата и время, по умолчанию текущие). Объясните назначение каждого атрибута.
Задача 8.
Напишите PHP-код с использованием PDO для: подключения к БД MySQL, выполнения SELECT-запроса для получения всех пользователей из таблицы users, вывода результата в виде HTML-таблицы.
Задача 9.
Создайте PHP-класс User с защищёнными полями $name и $email, конструктором, геттерами и методом getInfo(), возвращающим строку «Пользователь: [имя], Email: [email]». Создайте объект класса и вызовите метод getInfo().
Задача 10.
Создайте PHP-класс AdminUser, наследующий класс User из задачи 9. Добавьте поле $role со значением «Администратор» и переопределите метод getInfo() так, чтобы он также выводил роль. Продемонстрируйте полиморфизм.
Задача 11.

Напишите PHP-функцию, которая принимает массив с данными о товаре (название, цена, количество) и возвращает JSON-строку. Напишите также функцию для обратного преобразования JSON-строки в массив PHP.

Задача 12.

Напишите четыре SQL-запроса (CRUD) для таблицы products (id, name, price, quantity): добавление нового товара, выборка всех товаров дороже 1000 руб., обновление цены товара по id, удаление товара по id.

Задача 13.

Создайте HTML-страницу с формой регистрации (логин, пароль, подтверждение пароля, email). Реализуйте валидацию формы с помощью JavaScript перед отправкой.

Задача 14.

Напишите JavaScript-код с использованием fetch API для получения данных с endpoint /api/products. Обработайте результат: выведите список товаров в виде элементов <li> в HTML-список с id="productList". Обработайте возможные ошибки.

Задача 15.

Создайте React-компонент ProductList, который получает массив товаров через props и отображает каждый товар в виде карточки (название, цена). Если массив пуст, выводится сообщение «Товары не найдены».

Задача 16.

Напишите React-компонент Counter – счётчик с начальным значением 0. На странице должны быть кнопки «+» и «–», изменяющие значение счётчика. Счётчик не должен уходить ниже 0. Используйте хук useState.

Задача 17.

Создайте JavaScript-код для реализации модального окна: при нажатии на кнопку открывается окно с затемненным фоном, при нажатии на крестик или вне окна – закрывается.

Задача 18.

Опишите структуру REST API для приложения «Онлайн-библиотека» (книги и авторы). Укажите эндпоинты, HTTP-методы, формат запросов и ответов для операций: получить список всех книг, добавить новую книгу, обновить данные книги, удалить книгу.

Задача 19.

Составьте техническое задание (ТЗ) на веб-приложение «Список задач» (To-Do List) для студента. ТЗ должно включать: название, цель, описание функционала, требования к интерфейсу, технологический стек, сроки. Объём: 1–2 страницы.

Задача 20.

Составьте тест-план для проверки функции регистрации пользователя в веб-приложении. Включите: позитивные тест-кейсы (корректные данные), негативные тест-кейсы (пустые поля, неверный email, слабый пароль, уже занятый email), ожидаемые результаты.

Задача 21.
Создайте простую страницу на WordPress: установите CMS, настройте тему, создайте 3 страницы (Главная, О нас, Контакты) и меню навигации.
Задача 22.
Реализуйте на JavaScript слайдер изображений с кнопками “вперед”, “назад” и индикаторами (точками) текущего слайда.
Задача 23.
Напишите SQL-запросы для следующих задач по таблице users (id, name, email, role, created_at): выбрать всех пользователей с ролью 'admin', зарегистрированных за последние 30 дней; подсчитать количество пользователей по каждой роли; найти пользователей, email которых содержит '@gmail.com'.
Задача 24.
Напишите PHP-скрипт для серверной валидации формы регистрации: email должен быть в правильном формате, пароль – не менее 8 символов, содержать буквы и цифры, поле имени – не пустое и не более 50 символов. Выведите список ошибок или сообщение об успехе.
Задача 25.
Опишите алгоритм реализации системы аутентификации пользователя на PHP с использованием сессий: регистрация (хэширование пароля), вход (проверка хэша), защита страниц (проверка сессии), выход (уничтожение сессии). Укажите, какие PHP-функции используются на каждом шаге.