

Министерство науки и высшего образования Российской Федерации

ЧИТИНСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«БАЙКАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДЕН

на заседании кафедры информационных техно-
логий и высшей математики

24 февраля 2025 г. протокол № 6

Заведующий кафедрой

Л.И. Трухина



**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
(ФОНД ОЦЕНОЧНЫХ СРЕДСТВ)
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
Б1.У.8 Интернет-технологии**

Направление подготовки: 38.03.05 Бизнес-информатика

Направленность (профиль): Цифровая экономика

Квалификация выпускника: бакалавр

Чита, 2025 г.

**Структура
фонда оценочных средств
по дисциплине «Интернет-технологии»**

№ п/п	Этапы формирования компетенций	Перечень формируемых компетенций	ЗУНы (З.1, У1, Н1...)	Контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы	Описание показателей и критериев оценивания компетенций на различных этапах формирования, описания шкал оценивания
1	Принципы работы интернета. DNS и система доменных имен	ПК-3	З.Знать способы и методы управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов У.Уметь управлять, разрабатывать, поддерживать процессы создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов Н.Владеть навыками управления, разработки, поддержки процессов создания, модернизации и про-	Тест №1. Основы веб-разработки	Тест 20 вопросов. по 1 баллу за каждый верный ответ (20)

			движения Интернет-ресурсов и Интернет-сервисов		
2	Основы HTML и CSS	ПК-3	З.Знать способы и методы управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов У.Уметь управлять, разрабатывать, поддерживать процессы создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов Н.Владеть навыками управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов	Лабораторная работа №1. Основы web технологий	Полностью выполненная лабораторная работа - 10 баллов. Частично - доля выполненных заданий*5 Если доля меньше 0.3 - 0 баллов (10)
3	Введение в JavaScript. Работа с DOM и событиями в JavaScript. Работа с DOM и событиями в JavaScript. Основы одностраничных приложе-	ПК-3	З.Знать способы и методы управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов У.Уметь	Лабораторная работа №2. Разработка клиентской части веб-приложений	Полностью выполненная лабораторная работа - 10 баллов. Частично - доля выполненных заданий*5 Если доля меньше 0.3 - 0 баллов (10)

	ний (SPA)		управлять, разрабатывать, поддерживать процессы создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов Н. Владеть навыками управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов		
4	CSS фреймворки и препроцессоры. Введение в адаптивный дизайн	ПК-3	3. Знать способы и методы управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов У. Уметь управлять, разрабатывать, поддерживать процессы создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов Н. Владеть навыками управления, разработки, поддержки процессов создания, модернизации и про-	Лабораторная работа №3. Разработка клиентской части веб-приложений с использованием фреймворков	Полностью выполненная лабораторная работа - 15 баллов. Частично - доля выполненных заданий*5 Если доля меньше 0.3 - 0 баллов (15)

			движения Интернет-ресурсов и Интернет-сервисов		
5	Основы серверной разработки. Создание RESTful API. Введение в базы данных. Аутентификация и авторизация	ПК-3	З.Знать способы и методы управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов У.Уметь управлять, разрабатывать, поддерживать процессы создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов Н.Владеть навыками управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов	Лабораторная работа №4. Проектирование и архитектура веб-приложений	Полностью выполненная лабораторная работа - 15 баллов. Частично - доля выполненных заданий*5 Если доля меньше 0.3 - 0 баллов (15)
6	Веб-серверы и их настройка	ПК-3	З.Знать способы и методы управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов У.Уметь	Лабораторная работа №5: Разработка серверной части и работа с данными	Полностью выполненная лабораторная работа - 15 баллов. Частично - доля выполненных заданий*5 Если доля меньше 0.3 - 0 баллов (15)

			управлять, разрабатывать, поддерживать процессы создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов Н. Владеть навыками управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов		
7	Контейнеризация с использованием Docker. Gitlab. Настройка CI/CD пайплайнов	ПК-3	3. Знать способы и методы управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов У. Уметь управлять, разрабатывать, поддерживать процессы создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов Н. Владеть навыками управления, разработки, поддержки процессов создания, модернизации и про-	Лабораторная работа №6: Развёртывание и автоматизация процессов непрерывной интеграции и доставки	Полностью выполненная лабораторная работа - 15 баллов. Частично - доля выполненных заданий*5 Если доля меньше 0.3 - 0 баллов (15)

			движения Интернет-ресурсов и Интернет-сервисов		
8	Итого по текущей аттестации	ПК-3	З.Знать способы и методы управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов У.Уметь управлять, разрабатывать, поддерживать процессы создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов Н.Владеть навыками управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов	Итоговый тест	100
9	Промежуточная аттестация	ПК-3	З.Знать способы и методы управления, разработки, поддержки процессов создания, модернизации и продвижения Интернет-ресурсов и Интернет-сервисов У.Уметь	зачет	100

			управлять, разрабаты- вать, поддерживать процессы создания, модернизации и про- движения Интернет- ресурсов и Интернет- сервисов Н.Владеть на- выками управления, разработки, поддержки процессов создания, модернизации и про- движения Интернет- ресурсов и Интернет- сервисов		
--	--	--	---	--	--

Министерство науки и высшего образования Российской Федерации

Читинский институт (филиал)

ФГБОУ ВО «БАЙКАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра информационных технологий и высшей математики

Тест: Основы веб-разработки

1. **Что такое HTML?**
 - a) Язык программирования
 - b) Язык разметки для создания веб-страниц
 - c) Протокол передачи данных
 - d) Фреймворк для разработки
2. **Какой тег HTML используется для создания заголовка первого уровня?**
 - a) <h1>
 - b) <header>
 - c) <title>
 - d) <h6>
3. **Что такое CSS?**
 - a) Язык программирования
 - b) Язык разметки
 - c) Язык стилей для оформления веб-страниц
 - d) Протокол передачи данных
4. **Какой селектор CSS используется для выбора всех элементов с определенным классом?**
 - a) #class
 - b) .class
 - c) class
 - d) *class
5. **Что такое JavaScript?**
 - a) Язык разметки
 - b) Язык программирования для добавления интерактивности на веб-страницы
 - c) Протокол передачи данных
 - d) Система управления базами данных
6. **Какой метод JavaScript используется для получения элемента по его ID?**
 - a) getElementById()
 - b) querySelector()
 - c) getElementsByClassName()
 - d) getElementByClass()

7. **Что такое DOM (Document Object Model)?**
- a) Стандартный язык разметки
 - b) Объектная модель документа, представляющая структуру HTML-документа
 - c) Протокол передачи данных
 - d) Фреймворк для разработки
8. **Какой атрибут используется в теге <a> для указания URL ссылки?**
- a) href
 - b) src
 - c) link
 - d) url
9. **Что такое responsive design (адаптивный дизайн)?**
- a) Дизайн, который не изменяется в зависимости от устройства
 - b) Дизайн, который адаптируется под разные размеры экранов и устройства
 - c) Дизайн, который использует только текстовые элементы
 - d) Дизайн, который требует использования Flash
10. **Какой тег HTML используется для вставки изображений на страницу?**
- a)
 - b) <picture>
 - c) <image>
 - d) <src>
11. **Что такое фреймворк в контексте веб-разработки?**
- a). Набор инструментов и библиотек для упрощения разработки приложений.
 - b). Язык программирования.
 - c). Протокол передачи данных.
 - d). Система управления базами данных.
12. **Какой из следующих языков является языком программирования на стороне клиента?**
- a). PHP.
 - b). Python.
 - c). JavaScript.
 - d). SQL.
13. **Что такое API (Application Programming Interface)?**
- a). Интерфейс пользователя.
 - b). Набор правил и протоколов для взаимодействия между различными программными компонентами.
 - c). Язык разметки.
 - d). Фреймворк для разработки.
14. **Какой метод HTTP используется для получения данных с сервера?**
- a). POST.
 - b). GET.
 - c). PUT.
 - d). DELETE.
15. **Что такое Bootstrap?**
- a). Язык программирования.

- b). Фреймворк CSS для создания адаптивных веб-сайтов.
 - c). Протокол передачи данных.
 - d). Система управления базами данных.
16. **Какой атрибут в теге указывает альтернативный текст изображения?**
- a). src
 - b). alt
 - c). title
 - d). href
17. **Что такое семантический HTML?**
- a.) Использование HTML-тегов, которые имеют смысл и описывают содержание страницы.
 - b.) Использование только визуальных элементов без структуры.
 - c.) Использование JavaScript для изменения структуры страницы.
 - d.) Использование CSS для стилизации элементов.
18. **Какой тег HTML используется для создания нумерованного списка?**
- a.)
 - b.)
 - c.)
 - d.) <list>
19. **Что такое AJAX (Asynchronous JavaScript and XML)?**
- a.) Метод асинхронного обмена данными с сервером без перезагрузки страницы.
 - b.) Язык программирования.
 - c.) Протокол передачи данных.
 - d.) Фреймворк для работы с базами данных.
20. **Какое расширение файла обычно используется для файлов стилей CSS?**
- a.) .html
 - b.) .css
 - c.) .js
 - d.) .json

Критерии оценки:

За каждый правильный ответ – 1 балл. Максимальное количество баллов – 20 баллов.

Лабораторная работа №1. Основы web технологий

Задание 1

1. Откройте в текстовом редакторе файл index.html. Это файл-заготовка (см. рис.1).

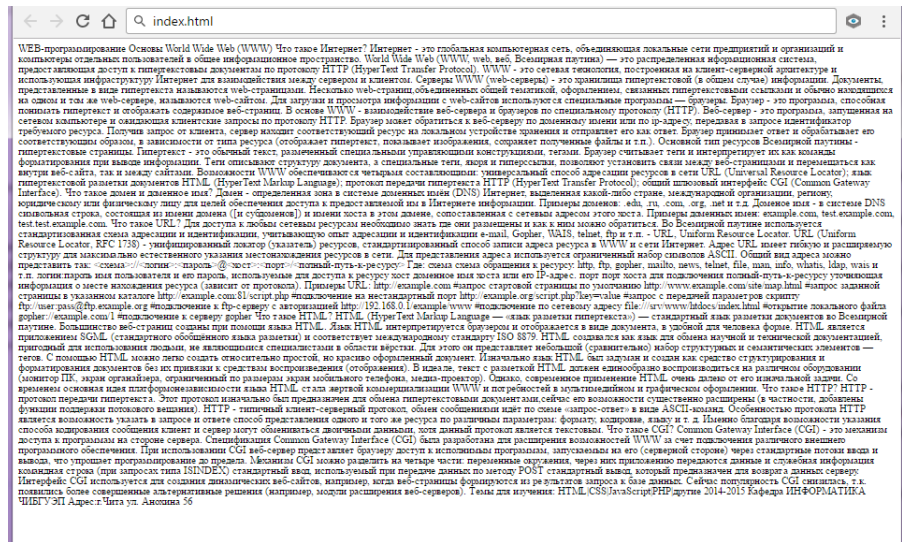


Рис. 1. Файл index.html открытый в браузере

2. Откройте для просмотра файлы 1.jpg, 2.jpg, 3.jpg, оригинал.jpg (см.рис 2, 3, 4, 5). Это результат, который должен получиться.

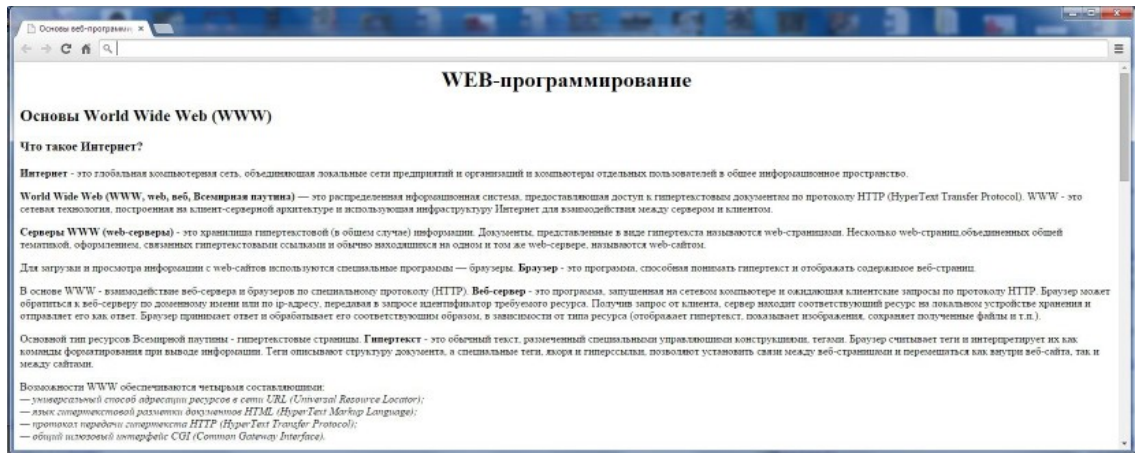


Рис. 2

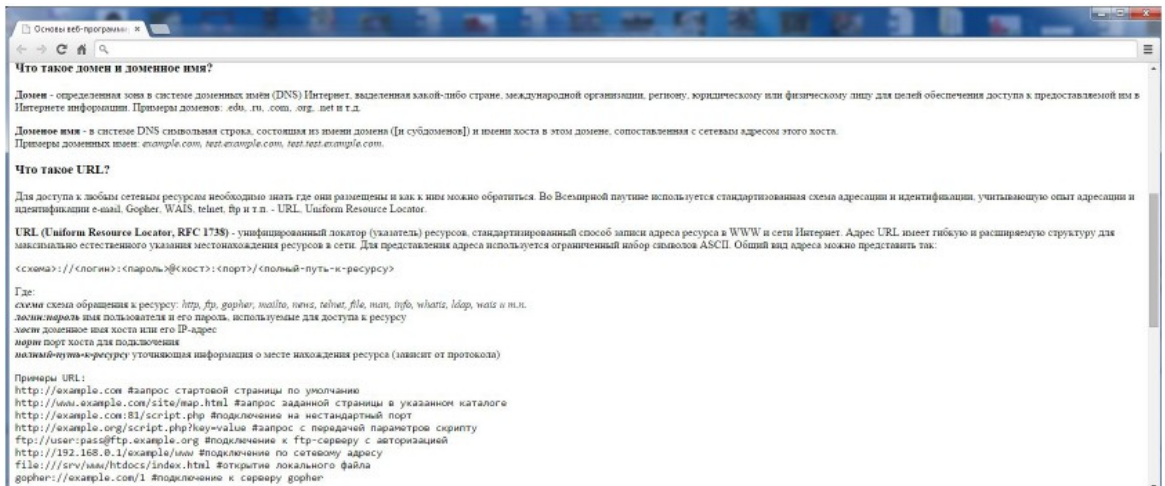


Рис. 3

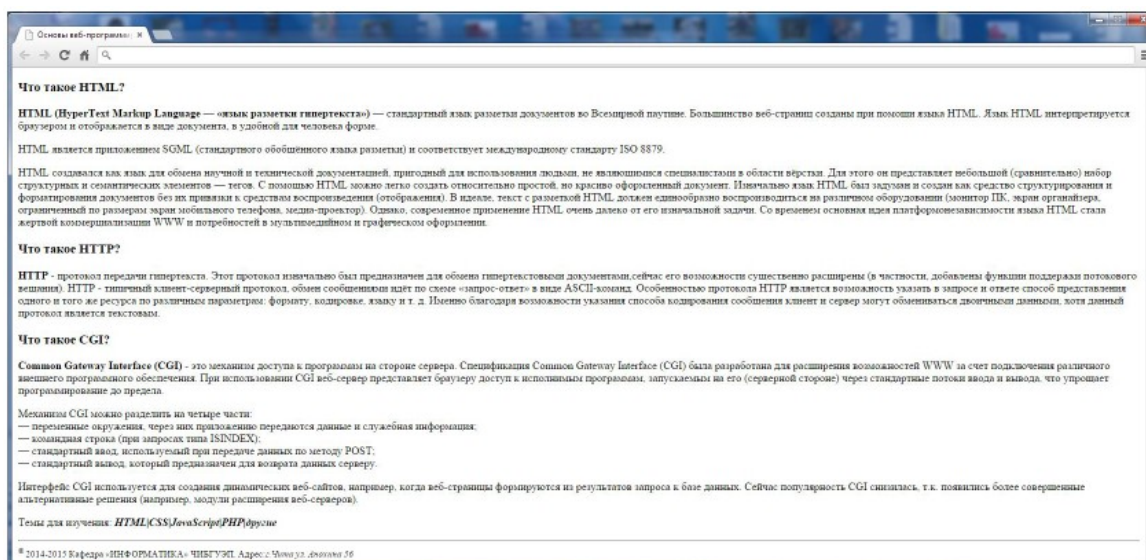


Рис. 4

3. Напишите необходимый html-код. Запустите файл index.html в браузере

Задание 2

1. Откройте в редакторе файл Index.html.
2. Откройте в браузере HTML валидатор (<http://validator.w3.org/>).
3. Проверьте на валидность данный файл – HTML 4.1, HTML 5 (разберите и исправьте ошибки).
4. Добавьте любое изображение в верхний левый угол страницы, подпишите изображение. Добавьте гиперссылку на изображении (при нажатии открывается любая поисковая система в новой вкладке).
5. Внизу страницы добавьте переход в начало страницы.
6. Добавьте на веб-страницу текст: «Отзывы направляйте по адресу: **укажите адрес электронной почты**». Добавьте гиперссылку на e-mail (при нажатии открывается почтовый агент).
7. Добавьте гиперссылку на папку с заданиями по информатике на диске N.

Задание 3

1. Создайте файл style.css
2. Подключите данный файл к файлу index.html
3. Новый стиль:
 - ✓ Фон страницы (изображение);
 - ✓ Цвет текста (произвольный);
 - ✓ Цвет ссылок (произвольный);

- ✓ Тип шрифта (произвольный);
 - ✓ Цвет заголовков (произвольный);
 - ✓ Выравнивание заголовков по центру;
 - ✓ Выравнивание текста по ширине;
 - ✓ Для всех абзацев «красная строка»;
 - ✓ Междустрочный интервал – 1,15.
4. Проверьте другие свойства CSS.
 5. Самостоятельно разобрать списки HTML (маркированные, нумерованные) и CSS свойства для списков.

Задание 4

1. Создайте таблицы (образцы – t1.jpg, t2.jpg, t3.jpg – см. рис 5-7).
2. Создайте *html-файл*. Результат просмотра файла в браузере представлен на рисунке *sq.jpg* (рис. 8).

План занятий

№п/п	Раздел дисциплины	Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)		
			лекции	л/р	с/р
1	Введение в Web-конструирование (HTML, CSS)	3.2	10	10	6
2	Программирование на JavaScript	3.2	10	10	10
3	Программирование на PHP. MySQL & PHP	3.3	20	20	12
ИТОГО			40	40	28

Рис. 5

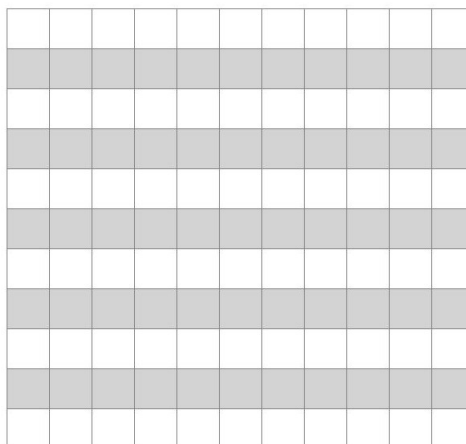


Рис. 6

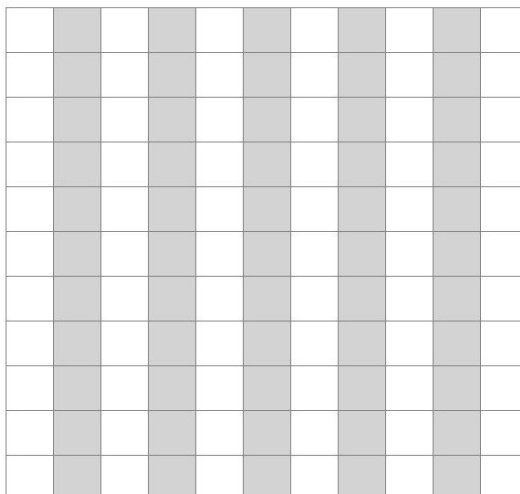


Рис. 7

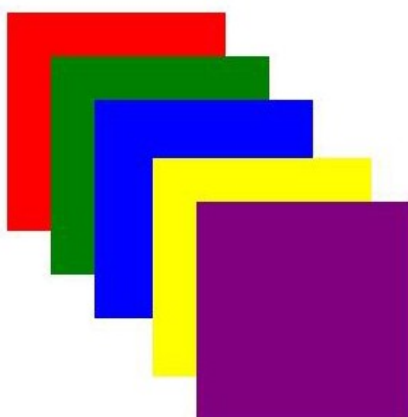


Рис. 8

3. Откройте файл ***index_maket.html***. Образец оформления представлен на рисунке ***Макет.jpg*** (размещение блоков на странице – рис.9). Создать файл ****.css***.

Задание 5

Поиск

Искать

JavaScript

PHP

Python

Perl

Java

Веб-программирование

Веб-программирование — раздел программирования, ориентированный на разработку веб-приложений (программ, обеспечивающих функционирование динамических сайтов Всемирной паутины). Языки веб-программирования — это языки, которые в основном предназначены для работы с веб-технологиями. Языки веб-программирования можно условно разделить на две пересекающиеся группы: клиентские и серверные..

Программы на клиентских языках обрабатываются на стороне пользователя, как правило, их выполняет браузер. Это и создает главную проблему клиентских языков — результат выполнения программы (скрипта) зависит от браузера пользователя. То есть если пользователь запретил выполнять клиентские программы, то они исполняться не будут, как бы ни желал этого программист. Кроме того, может произойти такое, что в разных браузерах или в разных версиях одного и того же браузера один и тот же скрипт будет выполняться по-разному. С другой стороны, если программист возлагает надежды на серверные программы, то он может упростить их работу и снизить нагрузку на сервер за счет программ, исполняемых на стороне клиента, поскольку они не всегда требуют перезагрузки (генерацию) страницы.

Когда пользователь дает запрос на какую-либо страницу (переходит на нее по ссылке или вводит адрес в адресной строке своего браузера), то вызванная страница сначала обрабатывается на сервере, то есть выполняются все программы, связанные со страницей, и только потом возвращается к посетителю по сети в виде файла. Этот файл может иметь расширения: HTML, PHP, ASP, ASPX, Perl, SSI, XML, DHTML, XHTML.

Полезные ссылки

CSS Validation Service:
<http://jigsaw.w3.org/css-validator/>

World Wide Web Consortium
W3C: <http://w3c.org/>

HTML5 W3C Recommendation
: <http://www.w3.org/TR/html5/>

Рис. 9

1. В файле index_maket.html ():
при нажатии на кнопку «Искать» в новой вкладке открывается поисковая система Google с найденными результатами (см. рис. 9).
2. Создайте стиль для печати на принтере (выводить на печать только основной текст index_maket.html, фон – белый, цвет текста – черный; без ссылок и др. элементов).
3. Создайте файл html с примерами вставки внешних объектов (теги Embed и Object), элементы HTML5 для добавления аудио- и видеофайлов.
4. Изучить CSS-правила настройки курсора (продемонстрировать на примере).

Критерии оценки:

За каждое задание – 2 балла. Максимальное количество баллов – 10 баллов. Частично - доля выполненных заданий*10. Если доля меньше 0.3 - 0 баллов (10)

Лабораторная работа №2. Разработка клиентской части веб-приложений

Язык JavaScript

Упражнение 1.

Вычислить на ЭВМ значения переменных, указанных в таблице (в соответствии с вариантом), по заданным расчетным формулам и наборам исходных данных. Подготовить тестовый вариант исходных данных и вычислить для них вручную значения вычисляемых в программе величин.

Вариант задания	Расчетные формулы	Значения исходных данных
1	$a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}$ $b = 1 + \frac{z^2}{3 + z^2/5}$	$x = 1,426$ $y = -1,220$ $z = 3,5$
2	$\gamma = x^{y/x} - \sqrt[3]{y/x} $ $\phi = (y - x) \frac{y - z/(y - x)}{1 + (y - x)^2}$	$x = 1,825$ $y = 18,225$ $z = -3,298$

Вариант задания	Расчетные формулы	Значения исходных данных
3	$s = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$ $\phi = x(\sin x^3 + \cos^2 y)$	$x=0,335 \ y=0,025$
4	$y = e^{-bt} \sin(at+b) - \sqrt{ bt+a }$ $s = b \sin(at^2 \cos 2t) - 1$	$a=-0,5 \ b=1,7 \ t=0,44$
5	$w = \sqrt{x^2+b} - b^2 \sin^3(x+a)/x$ $y = \cos^2 x^3 - x/\sqrt{a^2+b^2}$	$a=1,5 \ b=15,5 \ x=-2,9$
6	$s = x^3 tg^2(x+b)^2 + a/\sqrt{x+b}$ $q = \frac{bx^2 - a}{e^{ax} - 1}$	$a=16,5 \ b=3,4 \ x=0,61$
7	$r = x^2(x+1)/b - \sin^2(x+a)$ $s = \sqrt{xb/a} + \cos^2(x+b)^3$	$a=0,7 \ b=0,05 \ x=0,5$
8	$y = \sin^3(x^2+a)^2 - \sqrt{x/b}$ $z = \frac{x^2}{a} + \cos(x+b)^3$	$a=1,1 \ b=0,004 \ x=0,2$
9	$f = \sqrt{mtg t} + c \sin t $ $z = m \cos(bt \sin t) + c$	$m=2 \ c=-1 \ t=1,2$ $b=0,7$
10	$y = b tg^2 x - \frac{a}{\sin^2(x/a)}$ $d = a e^{-\sqrt{a}} \cos(bx/a)$	$a=3,2 \ b=17,5 \ x=-4,8$

Упражнение 2

Вычислить значение функции заданной в таблице (в соответствии с вариантом задания). Подготовить тесты (число тестов равно числу ветвей вычислительного процесса) для проверки правильности функционирования программ.

Вариант	Функция	Условие	Исходные данные
1	$y = \begin{cases} at^2 \ln t & 1 \leq t \leq 2 \\ 1 & t < 1 \\ e^{at} \cos bt & t > 2 \end{cases}$	$\begin{cases} 1 \leq t \leq 2 \\ t < 1 \\ t > 2 \end{cases}$	$\begin{aligned} a &= -0,5 \\ b &= 2 \end{aligned}$
2	$y = \begin{cases} \pi x^2 - 7/x^2 & x < 1,3 \\ ax^3 + 7\sqrt{x} & x = 1,3 \\ \ln(x + 7\sqrt{x}) & x > 1,3 \end{cases}$	$\begin{cases} x < 1,3 \\ x = 1,3 \\ x > 1,3 \end{cases}$	$a = 1,5$
3	$\omega = \begin{cases} ax^2 + bx + c & x < 1,2 \\ a/x + \sqrt{x^2 + 1} & x = 1,2 \\ (a + bx)/\sqrt{x^2 + 1} & x > 1,2 \end{cases}$	$\begin{cases} x < 1,2 \\ x = 1,2 \\ x > 1,2 \end{cases}$	$\begin{aligned} a &= 2,8 \\ b &= -0,3 \\ c &= 4 \end{aligned}$
4	$q = \begin{cases} \pi x^2 - 7/x^2 & x < 1,4 \\ ax^3 + 7\sqrt{x} & x = 1,4 \\ \ln(x + 7\sqrt{ x+a }) & x > 1,4 \end{cases}$	$\begin{cases} x < 1,4 \\ x = 1,4 \\ x > 1,4 \end{cases}$	$a = 1,65$
5	$y = \begin{cases} 1,5 \cos^2 x & x < 1 \\ 1,8ax & x = 1 \\ (x-2)^2 + 6 & 1 < x < 2 \\ 3 \operatorname{tg} x & x > 2 \end{cases}$	$\begin{cases} x < 1 \\ x = 1 \\ 1 < x < 2 \\ x > 2 \end{cases}$	$a = 2,3$
6	$\omega = \begin{cases} x \sqrt[3]{x-a} & x > a \\ x \sin ax & x = a \\ e^{-ax} \cos ax & x < a \end{cases}$	$\begin{cases} x > a \\ x = a \\ x < a \end{cases}$	$a = 2,5$
7	$q = \begin{cases} bx - \lg bx & bx < 1 \\ 1 & bx = 1 \\ bx + \lg bx & bx > 1 \end{cases}$	$\begin{cases} bx < 1 \\ bx = 1 \\ bx > 1 \end{cases}$	$b = 1,5$
8	$y = \begin{cases} \sin x \lg x & x > 3,5 \\ \cos^2 x & x \leq 3,5 \end{cases}$	$\begin{cases} x > 3,5 \\ x \leq 3,5 \end{cases}$	
9	$f = \begin{cases} \lg(x+1) & x > 1 \\ \sin^2 \sqrt{ ax } & x \leq 1 \end{cases}$	$\begin{cases} x > 1 \\ x \leq 1 \end{cases}$	$a = 20,3$
10	$z = \begin{cases} (\ln^3 x + x^2)/\sqrt{x+t} & x < 0,5 \\ \sqrt{x+t} + 1/x & x = 0,5 \\ \cos x + t \sin^2 x & x > 0,5 \end{cases}$	$\begin{cases} x < 0,5 \\ x = 0,5 \\ x > 0,5 \end{cases}$	$t = 2,2$

Упражнение 3

Составить программу для вычисления и вывода на экран таблицы значений функции y при изменении аргументах.

№ варианта	Функция	Диапазон и шаг изменения аргумента
1	$y = \frac{\cos^2 x - \sin^2 x}{25e}$	$x \in [0; 10]$ $\Delta x = 0,25$
2	$y = \operatorname{tg} x^2 - e^x + \ln x$	$x \in [1; 7]$ $\Delta x = 0,25$
3	$y = \frac{\sin(6,5+x^2)}{2} e^x$	$x \in [0; 10]$ $\Delta x = 0,5$
4	$y = \frac{3 - \sqrt{x}}{4 - 2\sqrt{x+x^2}}$	$x \in [0; 5]$ $\Delta x = 0,25$
5	$y = \sqrt[3]{\sin x - \cos^2 x + 1}$	$x \in [0; 10]$ $\Delta x = 0,2$
6	$y = \sqrt{\frac{x^2}{\sin^2 x + 1.5}}$	$x \in [3; 9]$ $\Delta x = 0,5$
7	$y = x \cos x^2 + 5x$	$x \in [-10; 10]$ $\Delta x = 0,5$
8	$y = \frac{(\sin x + e^x)}{\cos x}$	$x \in [6; 8]$ $\Delta x = 0,2$
9	$y = e^{\sin x} - x^2$	$x \in [-5; 5]$ $\Delta x = 0,5$
10	$y = \frac{\sin x - \cos x + e^x}{3}$	$x \in [-10; 0]$ $\Delta x = 0,2$

Упражнение 4

Составить программу решения задачи

Вариант	Задание
1	Вычислить сумму и количество положительных элементов массива X .
2	В массив Y переписать элементы массива X , вместо отрицательных элементов записать 0, подсчитать их количество.
3	Среди отрицательных элементов массива X найти максимальный

<i>Вариант</i>	<i>Задание</i>
	элемент.
4	Вычислить среднее геометрическое положительных элементов массива X.
5	Вычислить сумму и количество элементов массива X, удовлетворяющих условию: $-1 \leq x_i \leq 1$.
6	Переписать в массив Y все положительные элементы массива X и подсчитать их количество
7	Вычислить среднее арифметическое элементов массива X.
8	Среди положительных элементов массива X найти минимальный элемент.
9	Найти сумму и количество элементов массива, которые делятся на 4 без остатка.
10	Вычислить сумму и количество элементов массива X, удовлетворяющих условиям: $10 \leq x_i \leq 100$ или $-100 \leq x_i \leq -10$.

Упражнение 5

Функции для работы с таймерами

setTimeout, clearTimeout, setInterval, clearInterval.

Данные функции являются глобальными (являются методами объекта Window) и используются для того, чтобы выполнить задержку перед тем, как вызвать определенный метод или для того, чтобы задать интервал между повторяющимися вызовами одного и того же метода.

Разберите скрипты.

```

<!DOCTYPE html>
<html>
<head>
  <title>setTimeout()/</title>
  <script type="text/javascript">
    window.onload = function () {
      setTimeout(showMessage, 5000)
      function showMessage() {
        alert("Hello world");
      }
    }
  </script>
</head>
<body>
  <p>Через 5 секунд будет показано сообщение</p>

```

```
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
  <title>setInterval</title>
  <script type="text/javascript">
    window.onload = function () {
      var counter = 0;
      function Count() {
        counter++;
        document.getElementById("output").innerHTML = counter;
      }
      setInterval(Count, 1000);
    }
  </script>
</head>
<body>
  <p id="output">0</p>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
  <title>setTimeout and clearTimeout</title>
  <script>
    var timer;
    window.onload = function () {
      timer = window.setTimeout(showMessage, 5000);
    }
    function showMessage() {
      alert("Hello world");
    }
    function cancel() {
      clearTimeout(timer);
    }
  </script>
</head>
<body>
  <p>Сообщение будет отображено через 5 секунд</p>
```

```

    <a href="javascript:cancel()">Отмена</a>
</body>
</html>

<!DOCTYPE html>
<html>
<head>
    <title>setInterval and clearInterval</title>
    <script>
        window.onload = function () {
            var counter = 0;
            var intervalHandler;
            function count(){
                get("output").innerHTML = counter;
                counter++;
            }
            get("startButton").onclick = function () {
                intervalHandler = setInterval(count, 5);
            }
            get("stopButton").onclick = function () {
                clearInterval(intervalHandler);
            }
        }
        var get = function (id){
            return document.getElementById(id);
        }
    </script>
</head>
<body>
    <p id="output">-</p>
    <input id="startButton" type="button" value="Start" />
    <input id="stopButton" type="button" value="Stop" />
</body>
</html>

```

Разработайте программу секундомер. Секундомер должен иметь три кнопки «Старт», «Стоп», «Сбросить». Секундомер должен выводить время в формате 00:00:00.

Упражнение 6

JavaScript и CSS

Для того, чтобы через JavaScript изменить оформление элемента следует использовать следующий код:

```
[элемент].[style].[имя_css_атрибута] = "значение"
```

Например:

```
e.style.color = "green";  
e.style.width = "100px";
```

При использовании CSS атрибутов с именем, которое содержит «-» следует использовать правило – символ дефиса в названии следует удалить, а следующую букву в названии CSS атрибута, которая идет после дефиса, написать в верхнем регистре. Такой стиль именования называется Camel Casing.

Рассмотрите приведенные ниже скрипты. Доработайте скрипты следующим образом: при нажатии на кнопку цвет квадрата чередуется (красный – зеленый), во втором скрипте цвет квадрата изменяется плавно (все цвета радуги).

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>CSS и JavaScript</title>  
  <script type="text/javascript">  
    window.onload = function () {  
      var button = document.getElementsByTagName("input")[0];  
      button.onclick = function () {  
        var div = document.getElementsByTagName("div")[0];  
        div.style.backgroundColor = "green";  
      }  
    }  
  </script>  
</head>  
<body>  
  <div style="width: 200px; height: 200px; background-color: Red;">  
  </div>  
  <input type="button" value="Сменить цвет" />  
</body>  
</html>
```

```
<!DOCTYPE html>
```

```

<html>
<head>
  <title>Простая анимация цвета</title>
</head>
<body>
  <div id="rect">
  </div>
  <script type="text/javascript">
    var e = document.getElementById("rect");
    e.style.width = "100px";
    e.style.height = "100px";
    e.style.border = "solid black 1px";
    var colors = ["white", "yellow", "orange", "red"]
    var nextColor = 0;
    setInterval(function () {
      e.style.backgroundColor = colors[nextColor++ % colors.length];
    }, 500);
  </script>
</body>
</html>

```

Рассмотрите скрипт.

```

<!DOCTYPE html>
<html>
<head>
  <title>Анимация</title>
  <script type="text/javascript">
    var div1;
    var leftPos = 0;
    window.onload = function () {
      div1 = document.getElementById("div1");
      setInterval(animate, 10);
    }
    function animate() {
      div1.style.left = leftPos + "px";
      leftPos += 1;
    }
  </script>

```



```
</head>
<body>
<div id="div1"
style="height: 30px; width: 30px; background-color: Green; position: rela-
tive;">
</div>
</body>
</html>
```

1. Поместите на странице 5 кнопок: «вверх», «вниз», «влево», «вправо», «стоп» и div с текстом. При нажатии на кнопки div должен двигаться в соответствующем направлении.

2. Напишите сценарий, который каждую секунду будет случайным образом менять положение трех элементов div (с заданной высотой и шириной).

3. Создайте страницу для авторизации (поля ввода для логина и пароля, кнопка «Вход», флажок «Запомнить»). Разработать сценарий, срабатывающий при нажатии «Вход»:

- ✓ если поля пустые выводится сообщение (на странице, а не через Alert): «Заполните поля!!!», цвет полей – красный.
- ✓ если введен логин Admin и пароль pass, выводится сообщение: «Авторизация прошла успешно»
- ✓ в остальных случаях выводится сообщение: «У вас недостаточно прав, обратитесь к администратору».

Упражнение 7

Создайте на странице калькулятор для арифметических операций над данными. Реализуйте задачу двумя способами – с использованием модели DOM Level 0 (через свойства) и DOM Level 2.

Добавьте к калькулятору следующие возможности:

1. Кнопки, вставляющие в текстовое поле число π и число e .
2. Кнопку `backspace`, удаляющую из текстового поля последнюю введенную цифру.
3. Кнопку «+/-», меняющую знак числа на противоположный.
4. Вычисление числа $1/x$.
5. Вычисление синуса и косинуса числа.

6. Вычисление натурального логарифма числа.
7. Вычисление остатка от деления.
8. Возведение x в степень n .

Упражнение 8

Работа с *Canvas*

Элемент *Canvas*. Добавление элемента *canvas* на страницу. Рисование прямоугольников, линий и дуг, кривых Безье. Добавление цвета.

Полностью выполненная лабораторная работа - 10 баллов. Частично - доля выполненных заданий*10. Если доля меньше 0.3 - 0 баллов.

Лабораторная работа №3: Разработка клиентской части веб-приложения с использованием фреймворков

Цель работы:

Научиться разрабатывать клиентскую часть веб-приложения с использованием одного из популярных JavaScript-фреймворков (React, Vue.js или Angular).

Задание:

Создайте простое веб-приложение "Список задач" (To-Do List), которое позволяет пользователям добавлять, удалять и отмечать задачи как выполненные.

Требования к приложению:

1. Структура приложения:

1. Используйте выбранный фреймворк для создания одностраничного приложения (SPA).
2. Приложение должно иметь следующие компоненты:
 1. Компонент для отображения списка задач.
 2. Компонент для добавления новой задачи.
 3. Компонент для фильтрации задач (все, выполненные, невыполненные).

2. Функциональность:

1. Пользователь должен иметь возможность:
 1. Добавлять новую задачу через текстовое поле и кнопку "Добавить".
 2. Удалять задачу из списка.
 3. Отмечать задачу как выполненную (например, при нажатии на задачу).
 4. Фильтровать задачи по статусу (все, выполненные, невыполненные).

3. Хранение данных:

1. Используйте локальное состояние компонента или глобальное состояние (например, Redux для React или Vuex для Vue.js) для хранения списка задач.
2. При необходимости можно использовать Local Storage для сохранения задач между сессиями.

4. Стилизация:

1. Приложение должно быть стилизовано с использованием CSS или CSS-фреймворка (например, Bootstrap или Tailwind CSS).
2. Обратите внимание на адаптивность интерфейса.

5. Документация:

1. Напишите краткую документацию к вашему приложению, описывающую его функциональность и структуру кода.
2. Укажите инструкции по запуску приложения.

Дополнительные задания (по желанию):

1. Реализуйте возможность редактирования существующих задач.
2. Добавьте возможность сортировки задач по алфавиту или дате создания.
3. Реализуйте анимации при добавлении и удалении задач.

Оценивание:

Работа будет оцениваться по следующим критериям (15 баллов):

1. Полнота выполнения задания (все требования выполнены) – 50%
2. Качество кода (читаемость, структура) – 30%
3. Стилизация и пользовательский интерфейс – 20%

Срок выполнения:

Укажите срок выполнения лабораторной работы (например, 1-2 недели).

Рекомендации:

1. Ознакомьтесь с документацией выбранного фреймворка.
2. Используйте примеры из официальных руководств и учебных материалов.

Лабораторная работа №4 Проектирование и архитектура веб-приложений

Цели работы:

1. Ознакомиться с основами проектирования веб-приложений.
2. Научиться разрабатывать архитектурные решения для веб-приложений.

3. Освоить принципы проектирования пользовательского интерфейса и взаимодействия с пользователем.

Задание:

1. Выбор темы проекта:

1. Выберите тему для вашего веб-приложения (например, блог, интернет-магазин, система управления задачами, социальная сеть и т.д.).

2. Определите целевую аудиторию и основные функции приложения.

2. Анализ требований:

1. Составьте список функциональных и нефункциональных требований к вашему приложению.

2. Определите ключевые сценарии использования (user stories) для вашего приложения.

3. Проектирование архитектуры:

1. Разработайте архитектурную схему вашего приложения, включая:

1. Клиентскую часть (frontend).

2. Серверную часть (backend).

3. Базу данных.

4. Взаимодействие между компонентами.

2. Определите используемые технологии (языки программирования, фреймворки, базы данных и т.д.).

4. Создание прототипа пользовательского интерфейса:

1. Создайте прототип пользовательского интерфейса вашего приложения с помощью инструментов проектирования (например, Figma, Adobe XD или Sketch).

2. Убедитесь, что прототип включает основные страницы и элементы интерфейса (формы, кнопки, навигация).

5. Документация архитектуры:

1. Подготовьте документацию по архитектуре вашего приложения:

1. Описание компонентов системы.

2. Схемы взаимодействия между компонентами.

3. Обоснование выбора технологий и архитектурных решений.

6. Презентация проекта:

1. Подготовьте короткую презентацию о вашем проекте (5-10 минут), в которой расскажете о том:

1. Как вы выбрали тему и определили требования.

2. Как вы разработали архитектуру приложения.

3. Какие проблемы возникли в процессе проектирования и как вы их решили.

Оценивание (15 баллов):

1. Полнота выполнения задания (30%)

2. Качество проектирования архитектуры и документации (30%)

3. Качество прототипа пользовательского интерфейса (20%)
4. Презентация проекта (20%)

Дополнительные рекомендации:

1. Рассмотрите возможность использования паттернов проектирования (например, MVC, MVVM) в вашей архитектуре.
2. Обратите внимание на аспекты безопасности при проектировании приложения.
3. Используйте принципы адаптивного дизайна для обеспечения удобства использования на различных устройствах.

Лабораторная работа №5 Разработка серверной части и работа с данными

Цели работы:

1. Ознакомиться с основами разработки серверной части веб-приложений.
2. Научиться работать с базами данных (SQL или NoSQL).
3. Освоить обработку HTTP-запросов и создание RESTful API.

Задание:

1. Выбор технологии:

1. Выберите язык программирования и фреймворк для разработки серверной части (например, Node.js с Express, Python с Flask или Django, Ruby on Rails, Java с Spring Boot и т.д.).

2. Создание проекта:

1. Создайте новый проект и настройте его структуру.
2. Убедитесь, что у вас есть файл для управления зависимостями (например, package.json для Node.js или requirements.txt для Python).

3. Настройка базы данных:

1. Выберите тип базы данных (реляционная, например PostgreSQL или MySQL, или NoSQL, например MongoDB).
2. Создайте базу данных и таблицы/коллекции для хранения данных вашего приложения.
3. Определите структуру данных (например, пользователи, товары, заказы и т.д.).

4. Создание RESTful API:

1. Реализуйте следующие эндпоинты для вашего API:
 1. GET /items — получение списка всех элементов.
 2. GET /items/:id — получение элемента по ID.
 3. POST /items — создание нового элемента.
 4. PUT /items/:id — обновление существующего элемента.
 5. DELETE /items/:id — удаление элемента по ID.

2. Обработайте соответствующие запросы к базе данных в каждом из эндпоинтов.
5. **Валидация данных:**
 1. Реализуйте валидацию входящих данных на стороне сервера (например, проверка обязательных полей, формата данных и т.д.).
 2. Используйте библиотеки для валидации (например, Joi для Node.js или Marshmallow для Python).
6. **Тестирование API:**
 1. Напишите тесты для вашего API (можно использовать Postman или встроенные инструменты тестирования).
 2. Убедитесь, что все эндпоинты работают корректно и возвращают ожидаемые результаты.
7. **Документация:**
 1. Подготовьте документацию по вашему API:
 1. Описание всех доступных эндпоинтов.
 2. Примеры запросов и ответов.
 3. Инструкции по запуску сервера и работе с базой данных.
8. **Презентация:**
 1. Подготовьте короткую презентацию о вашем проекте (5-10 минут), в которой расскажете о том:
 1. Как вы выбрали технологии для разработки.
 2. Как вы организовали структуру проекта.
 3. Какие проблемы возникли в процессе разработки и как вы их решили.

Оценивание (15 баллов):

1. Полнота выполнения задания (30%)
2. Качество реализации API и его функциональности (30%)
3. Корректность работы с базой данных (20%)
4. Документация и презентация (20%)

Дополнительные рекомендации:

1. Рассмотрите возможность использования ORM (Object-Relational Mapping) для упрощения работы с базой данных.
2. Обратите внимание на безопасность вашего API (например, аутентификация пользователей).
3. Используйте инструменты для логирования запросов и ошибок.

Лабораторная работа №6. Развёртывание и автоматизация процессов непрерывной интеграции и доставки (CI/CD)

Цели работы:

1. Ознакомиться с концепциями непрерывной интеграции (CI) и непрерывной доставки (CD).
2. Научиться настраивать автоматизированные процессы сборки, тестирования и развертывания приложений.
3. Получить практический опыт работы с инструментами CI/CD.

Задание:

1. Выбор проекта:

1. Выберите или создайте простой проект веб-приложения (например, на Node.js, Python Flask, Ruby on Rails или другом языке/фреймворке).
2. Убедитесь, что проект имеет базовую структуру и включает в себя как минимум один тест.

2. Настройка системы контроля версий:

1. Создайте репозиторий вашего проекта на GitHub, GitLab или Bitbucket.
2. Загрузите ваш проект в репозиторий.

3. Настройка CI/CD:

1. Выберите инструмент для автоматизации CI/CD (например, GitHub Actions, GitLab CI/CD, Travis CI или CircleCI).
2. Создайте конфигурационный файл для вашего инструмента CI/CD:
 1. Определите шаги для сборки вашего приложения.
 2. Настройте выполнение тестов после сборки.
 3. Настройте процесс развертывания приложения на выбранном вами хостинге (например, Heroku, AWS, DigitalOcean или другом).

4. Автоматизация тестирования:

1. Напишите несколько юнит-тестов для вашего приложения (если они еще не написаны).
2. Убедитесь, что ваши тесты запускаются автоматически при каждом коммите в репозиторий.

5. Развертывание приложения:

1. Настройте автоматическое развертывание приложения на выбранном вами хостинге после успешного прохождения всех тестов.
2. Убедитесь, что ваше приложение доступно по URL-адресу после развертывания.

6. Документация:

1. Подготовьте документацию по вашему проекту:
 1. Описание функциональности приложения.
 2. Инструкции по настройке локальной среды разработки.
 3. Описание процесса настройки CI/CD.
 4. Скриншоты или ссылки на результаты работы (например, URL-адрес развернутого приложения).

7. Презентация:

1. Подготовьте короткую презентацию о вашем проекте и процессе настройки CI/CD (5-10 минут), в которой расскажете о том:
 1. Как вы выбрали инструменты для CI/CD.
 2. Какие проблемы возникли в процессе настройки и как вы их решили.
 3. Как работает ваше приложение и как оно развернуто.

Оценивание (15 баллов):

1. Полнота выполнения задания (30%)
2. Качество реализации проекта и его функциональности (30%)
3. Корректность настройки процессов CI/CD (20%)
4. Документация и презентация (20%)

Дополнительные рекомендации:

1. Рассмотрите возможность использования контейнеризации с Docker для упрощения процесса развертывания.
2. Обратите внимание на безопасность при настройке доступа к вашему приложению и репозиторию.
3. Используйте различные среды (например, staging и production) для тестирования перед окончательным развертыванием.

Итоговый тест

Вопросы:

1. **Что такое HTTP?**
 - a) Протокол передачи данных
 - b) Язык программирования
 - c) Система управления базами данных
 - d) Фреймворк для разработки
2. **Какой метод HTTP используется для отправки данных на сервер?**
 - a) GET
 - b) POST
 - c) PUT
 - d) DELETE
3. **Что такое REST?**
 - a) Протокол передачи данных
 - b) Архитектурный стиль для создания веб-сервисов
 - c) Язык разметки
 - d) Фреймворк для разработки
4. **Какой язык используется для создания стилей веб-страниц?**
 - a) HTML
 - b) CSS

- c) JavaScript
- d) SQL

5. Что такое DOM?

- a) Язык программирования
- b) Объектная модель документа
- c) Протокол передачи данных
- d) Система управления базами данных

6. Какой из следующих фреймворков является клиентским?

- a) Django
- b) Express.js
- c) React
- d) Ruby on Rails

7. Что такое AJAX?

- a) Метод асинхронного обмена данными с сервером без перезагрузки страницы
- b) Язык программирования для создания веб-приложений
- c) Протокол передачи данных
- d) Фреймворк для работы с базами данных

8. Какой из следующих языков является языком программирования на стороне сервера?

- a) HTML
- b) CSS
- c) PHP
- d) JavaScript

9. Что такое SQL?

- a) Язык разметки
- b) Язык запросов к базам данных
- c) Протокол передачи данных
- d) Фреймворк для разработки

10. Какой метод SQL используется для извлечения данных из таблицы?

- a) INSERT
- b) UPDATE
- c) SELECT
- d) DELETE

11. Что такое JSON?

- a) Формат обмена данными, основанный на текстовом формате JavaScript
- b) Язык программирования

- c) Протокол передачи данных
- d) Система управления базами данных

12. Какой из следующих методов используется для обновления существующей записи в базе данных?

- a) INSERT
- b) UPDATE
- c) SELECT
- d) DELETE

13. Что такое фреймворк?

- a) Набор инструментов для разработки приложений
- b) Язык программирования
- c) База данных
- d) Протокол передачи данных

14. Какой из следующих фреймворков является серверным?

- a) Angular
- b) Vue.js
- c) Flask
- d) Bootstrap

15. Что такое MVC в контексте веб-разработки?

- a) Модель-Вид-Контроллер, архитектурный паттерн для организации кода приложения
- b) Модель-Вид-Компонент, подход к разработке интерфейсов
- c) Методология управления проектами
- d) Протокол передачи данных

16. Какой тег HTML используется для создания гиперссылок?

- a) <link>
- b) <a>
- c) <href>
- d) <url>

17. Что такое CDN (Content Delivery Network)?

- a). Сеть доставки контента, которая ускоряет загрузку ресурсов сайта.
- b). Система управления базами данных.
- c). Протокол передачи файлов.
- d). Язык программирования.

18. Какой из следующих методов HTTP является безопасным (idempotent)?

- a). POST.
- b). GET.
- c). DELETE.

d). PUT.

19. Что такое CORS (Cross-Origin Resource Sharing)?

- a). Механизм безопасности браузеров, позволяющий ограничивать доступ к ресурсам между разными доменами.
- b). Протокол передачи файлов.
- c). Язык программирования.
- d). Фреймворк для работы с API.

20. Какой из следующих инструментов используется для тестирования API?

- a). Figma.
- b). Postman.
- c). GitHub.
- d). Visual Studio Code.

21. Что такое WebSocket?

- a). Протокол, обеспечивающий двустороннюю связь между клиентом и сервером в реальном времени.
- b). Метод асинхронного обмена данными с сервером.
- c). Формат обмена данными.
- d). Язык программирования.

22. Какой тег HTML используется для вставки изображений на страницу?

- a)
- b) <picture>
- c) <image>
- d) <src>

23. Что такое ORM (Object-Relational Mapping)?

- a). Технология, позволяющая работать с базами данных через объектно-ориентированные языки программирования.
- b). Методология проектирования интерфейсов.
- c). Протокол передачи данных.
- d). Язык запросов к базам данных.

24. Какой из следующих языков не является языком программирования на стороне сервера?

- a). Python.
- b). Ruby.
- c). JavaScript.
- d). HTML.

25. Что такое Agile в контексте разработки ПО?

- a.) Методология управления проектами, основанная на итеративной разработке и гибкости изменений требований.
 - b.) Стандартный язык разметки веб-страниц.
 - c.) Инструмент для тестирования API.
 - d.) Архитектурный паттерн.
- 26. Какое расширение файла обычно используется для файлов стилей CSS?**
- a.) .html
 - b.) .css
 - c.) .js
 - d.) .json
- 27. Что такое "состояние" в контексте веб-приложений?**
- a.) Информация о текущем взаимодействии пользователя с приложением
 - b.) Статус сервера
 - c.) Версия приложения
 - d.) Тип базы данных
- 28. Какое значение имеет атрибут "alt" в теге ?**
- a.) Определяет ссылку на изображение
 - b.) Описывает изображение и отображается при его отсутствии
 - c.) Указывает размер изображения
 - d.) Задает стиль изображения
- 29. Что означает термин "responsive design"?**
- a.) Дизайн, который адаптируется под разные размеры экранов и устройства
 - b.) Дизайн, который реагирует на действия пользователя
 - c.) Дизайн, который использует только текстовые элементы
 - d.) Дизайн, который не требует использования CSS
- 30. Какая функция JavaScript используется для отправки асинхронных запросов к серверу?**
- a.) fetch()
 - b.) send()
 - c.) request()
 - d.) ajax()

Министерство науки и высшего образования Российской Федерации
ЧИТИНСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФГБОУ ВО «БАЙКАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Кафедра информационных технологий и высшей математики

Примерный перечень вопросов к зачету

1. Какие задачи решаются с помощью системы управления версиями?
2. Какие из перечисленных свойств используются в Flexbox?
3. Какие из перечисленных технологий используются для создания адаптивного дизайна веб-страниц?
4. Какие из следующих технологий относятся к стеку MEAN?
5. Какие инструменты используются для контейнеризации и оркестрации?
6. Какие инструменты используются для мониторинга производительности веб-приложений?
7. Какие инструменты используются для управления версиями кода?
8. Какие компоненты включают в себя современные веб-приложения?
9. Какие компоненты входят в архитектуру микросервисов?
10. Какие методы используются для обеспечения безопасности веб-приложений?
11. Какие методы используются для управления состоянием в клиентских приложениях?
12. Какие основные функции выполняет сервер в клиент-серверной архитектуре?
13. Какие преимущества предоставляет использование CI/CD пайплайнов?
14. Какие преимущества предоставляет использование контейнеров?
15. Какие принципы лежат в основе архитектуры микросервисов?
16. Какие протоколы используются для безопасной передачи данных в интернете?
17. Какие протоколы используются для обмена данными между микросервисами?
18. Какие технологии используются для обработки и хранения данных в реальном времени?
19. Какие технологии используются для разработки серверной части веб-приложений?
20. Какие технологии используются для создания одностраничных приложений (SPA)?
21. Какие типы данных поддерживаются в языке разметки HTML?
22. Какие типы тестирования используются в веб-разработке?
23. Какие шаблоны проектирования используются в веб-разработке?
24. Какие этапы входят в процесс CI/CD?
25. Какой из перечисленных методов не является методом HTTP?
26. Какой метод HTTP используется для отправки данных на сервер с целью их обработки и добавления?
27. Что такое Docker и для чего он используется?
28. Что такое Kubernetes и для чего он используется?
29. Что такое RESTful API?

30. Что такое WebSocket и для чего он используется?

ОБРАЗЕЦ БИЛЕТА

Министерство науки и высшего образования
Российской Федерации
Читинский институт (филиал)
Федерального государственного бюджетного
образовательного учреждения
высшего образования
«БАЙКАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»
(ЧИ ФГБОУ ВО «БГУ»)

Направление - 38.03.05 Бизнес-
информатика
Профиль - Цифровая экономика
Кафедра информационных техно-
логий и высшей математики
Дисциплина - Интернет-техно-
логии

БИЛЕТ № 1

1. Тест (30 баллов).
2. Продемонстрируйте умения разработки программ, тестирования, исправления несоответствий в архитектуре, дизайне, программах и документации ИС, создания пользовательской документации, разработки методологии обучения пользователей, развертывания ИС, интеграции с другими ИС заказчика, оптимизации работы ИС и управления полномочиями пользователей согласно варианту (30 баллов).
3. Разработайте и протестируйте веб сервис или его компонент, согласно варианту (40 баллов).

Составитель _____ Е.О. Богатикова
Заведующий кафедрой _____ Л.И. Трухина

Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Система критериев оценки определяет оценку успеваемости по каждому заданию (вопросу) экзаменационного билета или заданию для зачета с использованием интервальной шкалы баллов, применяемой в привязке к рейтинговой 100-балльной системе.

ОЦЕНКА ОТВЕТА НА ТЕОРЕТИЧЕСКИЙ ВОПРОС В УСТНОЙ ИЛИ ПИСЬМЕННОЙ ФОРМЕ:

Оценка «отлично» / «зачтено» (91-100 баллов) выставляется при соблюдении следующих условий: Ответ отличается глубиной и полнотой, свободным владением понятийно-категориальным (терминологическим) аппаратом изученной дисциплины. Отражает знание не только основной, но и дополнительной литературы. Приведены примеры, отражающие умение связать теорию с практикой. Ответ изложен логически последовательно, грамотно и корректно.

Оценка «хорошо» / «зачтено» (76-90 баллов) выставляется при соблюдении следующих условий: Ответ отличается полнотой, владением понятийно-категориальным

(терминологическим) аппаратом изученной дисциплины, но в ответе могут присутствовать неточности. Отражает знание основной литературы. Приведены примеры, отражающие умение связать теорию с практикой. Ответ изложен логически последовательно, грамотно и корректно, но недостаточно аргументирован.

Оценка «удовлетворительно» / «зачтено» (61-75 баллов) выставляется при соблюдении следующих условий: В ответе отражено знание понятийно-категориального (терминологического) аппарата изучаемой дисциплины, но присутствуют отдельные ошибки и неточности. Ответ характеризуется недостаточным знанием рекомендованной литературы. Примеры, отражающие умение связать теорию с практикой, тривиальны, либо отсутствуют. Ответ неполный, носит фрагментарный, непоследовательный характер.

Оценка «неудовлетворительно» / «не зачтено» (0-60 баллов) выставляется при соблюдении следующих условий: Ответ характеризуется незнанием, либо фрагментарным представлением о понятийно-категориальном аппарате дисциплины, содержит множество ошибок. Примеры и иллюстрации отсутствуют. Ответ логически непоследователен.

ОЦЕНКА ВЫПОЛНЕНИЯ ЗАДАНИЯ В ФОРМЕ CASE-STUDY (СИТУАЦИИ)

Оценка «отлично» / «зачтено» (91-100 баллов) выставляется при соблюдении следующих условий: Четкая формулировка проблемы. Полное и соответствующее ситуации решение, основанное на знании правовых норм и технологий (опыте), применяемых в реальных организациях (известных компаниях). Предполагаемые действия описаны логично и последовательно. Даны дополнительные авторские комментарии и предложения к решению ситуации.

Оценка «хорошо» / «зачтено» (76-90 баллов) выставляется при соблюдении следующих условий: Понимание сути проблемы, но ее формулирование затруднено. Решение соответствует ситуации, отражает знание правовых норм и опыт работы других организаций при решении подобных ситуаций. Логика и последовательность действий не нарушены.

Оценка «удовлетворительно» / «зачтено» (61-75 баллов) выставляется при соблюдении следующих условий: Проблема не сформулирована. Приведен набор действий, потенциально способствующих улучшению ситуации и решению проблемы.

Оценка «неудовлетворительно» / «не зачтено» (0-60 баллов) выставляется при соблюдении следующих условий: Предложенный перечень мероприятий не соответствует ситуации.

ОЦЕНКА РЕШЕНИЯ ЗАДАЧИ

Оценка «отлично» / «зачтено» (91-100 баллов) выставляется при соблюдении следующих условий: Полное верное решение - оценивается в n баллов (n – максимальное количество баллов за решение задачи в структуре экзаменационного билета/задания).

Оценка «хорошо» / «зачтено» (76-90 баллов) выставляется при соблюдении следующих условий: Верное решение; имеются небольшие недочеты, в целом не влияющие на решение – оценивается в диапазоне от $0,76 \cdot n$ баллов до $0,9 \cdot n$ баллов (n – максимальное количество баллов за решение задачи в структуре экзаменационного билета/задания).

Оценка «удовлетворительно» / «зачтено» (61-75 баллов) выставляется при соблюдении следующих условий: Решение в целом верное; однако оно содержит ряд ошибок, либо не учитывает отдельных случаев, но может стать правильным после некоторых исправлений или дополнений – оценивается в диапазоне от $0,61 \cdot n$ баллов до $0,75 \cdot n$ баллов (n – максимальное количество баллов за решение задачи в структуре экзаменационного билета/задания).

Оценка «неудовлетворительно» / «не зачтено» (0-60 баллов) выставляется при соблюдении следующих условий: Решение неверное; изначально выбран неверный ход решения, или решение отсутствует – оценивается в 0 баллов.

ОЦЕНКА ВЫПОЛНЕНИЯ ТЕСТОВОГО ЗАДАНИЯ

Подсчитывается доля набранных баллов в максимальной сумме баллов за все задания теста:

- Каждый правильный ответ на тестовый вопрос (тип выборочный, одинарный, множественный, открытый) оценивается в m баллов (число m определяется путем деления максимального количества баллов за выполнение теста в структуре экзаменационного билета/задания на количество тестовых заданий);
- Каждый частично правильный ответ на тестовый вопрос (тип выборочный, множественный, открытый) оценивается в $m/2$ баллов независимо от соотношения правильно/неправильно выбранных вариантов (число m определяется путем деления максимального количества баллов за выполнение теста в структуре экзаменационного билета/задания на количество тестовых заданий);
- Каждый неправильный ответ на тестовый вопрос (тип выборочный, одинарный) оценивается в 0 баллов.

Оценка «отлично»/ «зачтено» (91-100 баллов) выставляется, если доля набранных баллов составляет 91-100%.

Оценка «хорошо»/ «зачтено» (76-90 баллов), если доля набранных баллов составляет 76-90%.

Оценка «удовлетворительно»/ «зачтено» (61-75 баллов), если доля набранных баллов составляет 61-75%.

Оценка «неудовлетворительно»/ «не зачтено» (0-60 баллов), если доля набранных баллов составляет не более 60%.