

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Шутов Олег Леонтьевич

Должность: Директор

Дата подписания: 06.06.2026 11:49:43

Уникальный программный ключ:

2ee6ded937fc2877009a3b03e0f0a7f33d8083d5

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ПРОФЕССИОНАЛЬНАЯ
ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ**

**«КУБАНСКИЙ ИНСТИТУТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ»
(АНПО «КУБАНСКИЙ ИПО»)**

ОТДЕЛЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

РАБОЧАЯ ПРОГРАММА

учебной дисциплины

ОП.11 ОСНОВЫ ВЕБ-РАЗРАБОТКИ

по специальности

**09.02.11 РАЗРАБОТКА И УПРАВЛЕНИЕ ПРОГРАММНЫМ
ОБЕСПЕЧЕНИЕМ**

направленность программы: Веб-разработка

Краснодар, 2026

СОГЛАСОВАНО

Зам. директора по КОД и МР

_____/ Т.В. Першакова
28.05.2026 г.**УТВЕРЖДАЮ**

Директор АНПОО «Кубанский ИПО»

_____/ О.Л. Шутов
Приказ №38-О от 28.05.2026 г.**ОДОБРЕНО**

Педагогическим советом

Протокол №6 от 28.05.2026 г

РАССМОТРЕНО

на заседании УМО

«Информационные системы и
программирование»

Протокол № 5 от 15.05.2026г.

Председатель _____ / С.А. Пясецкий

Рабочая программа учебной дисциплины ОП.11 Основы веб-разработки предназначена для реализации образовательной программы подготовки специалистов среднего звена.

Разработана на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением (Приказ Министерства Просвещения Российской Федерации от 24 февраля 2025 г. № 138, зарегистрированного Министерством Юстиции России 31 марта 2025 г. № 81696) с учетом примерной образовательной программы, разработанной Федеральным учебно-методическим объединением в системе среднего профессионального образования по укрупненным группам профессий, специальностей 09.00.00 Информатика и вычислительная техника, с учетом профессиональных стандартов: «Программист» (Приказ Министерства труда и социальной защиты РФ от 20 июля 2022 г. № 424н, зарегистрирован Министерством юстиции Российской Федерации от 22 августа 2022г. №69720); «Разработчик Web и мультимедийных приложений» (Приказ Министерства труда и социальной защиты РФ от 18 января 2017 г. № 44н, зарегистрирован Министерством юстиции Российской Федерации от 31 января 2017 г. № 45481) и компетенции «Веб- технологии».

Организация-разработчик: АНПОО «Кубанский ИПО»

Разработчик:

Пясецкий С.А., преподаватель АНПОО «Кубанский ИПО»

Рецензенты:

1. Варкентин В.Ф. – преподаватель, АНПОО «Кубанский ИПО»

Квалификация по диплому: преподаватель информатики

2. Маслиев Р.О, генеральный директор ООО «Старт Эксперт»

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	5
3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	10
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ ..	12

1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.11 ОСНОВЫ ВЕБ-РАЗРАБОТКИ

1.1 Место дисциплины в структуре основной образовательной программы:

Учебная дисциплина ОП.11 Основы веб-разработки является вариативной частью общепрофессионального цикла образовательной программы в соответствии с ФГОС по специальности 09.02.11 Разработка и управление программным обеспечением.

Особое значение дисциплина имеет при формировании и развитии общих и профессиональных компетенций:

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

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ПК 2.1. Проектировать модули программного обеспечения;

ПК 2.2. Разрабатывать модули программного обеспечения;

ПК 2.3. Выполнять интеграцию модулей и компонентов программного обеспечения.

1.2 Цель и планируемые результаты освоения дисциплины:

В рамках программы учебной дисциплины обучающимися осваиваются умения и знания.

Код ПК, ОК	Умения	Знания
ПК.2.1., ПК.2.2., ПК.2.3., ОК.02, ОК.04, ОК.07	<ul style="list-style-type: none">– создавать веб-страницы с использованием HTML;– оформлять стили с помощью CSS;– программировать на языке JavaScript;– работать с DOM и событиями;– выполнять запросы к API (fetch / AJAX);– создавать и запрашивать данные из простой БД;– проводить ручное тестирование веб-страницы;– оформлять техническую документацию (README, руководство);– использовать Git для хранения кода;– презентовать результат;	<ul style="list-style-type: none">– методы алгоритмизации и формализации задач;– основы языков разметки и стилей– основы программирования на языке JavaScript;– основы типографики и работы с графикой;– основы работы с базами данных– клиент-серверное взаимодействие (HTTP, API);– методы тестирования;– стандарты оформления кода и документации;– человеко-ориентированный подход к интерфейсам;– основы управления версиями (Git);

2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1 Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем в часах
<i>ОБЪЕМ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ДИСЦИПЛИНЫ</i>	89
<i>в том числе вариативная часть</i>	89
<i>- теоретическое обучение</i>	30
<i>- практические занятия</i>	50
<i>в т.ч. в форме практической подготовки</i>	50
<i>- промежуточная аттестация</i>	9
<i>в том числе:</i>	
<i>консультации</i>	6
<i>экзамен</i>	3

Тематический план учебной дисциплины

Наименование разделов и тем	Количество аудиторных часов				
	Всего	в т.ч. в форме практической подготовки	самост. работа	теоретич. обучение	практич. занятия
Раздел 1. Введение в веб-разработку. Инструменты	8	4		4	4
<i>Тема 1.1 Введение в веб-разработку. Инструменты разработчика и</i>	8	4		4	4
Раздел 2. Основы HTML. Структура веб-страницы	18	10		8	10
<i>Тема 2.1 HTML-теги и структура документа</i>	18	10		8	10
Раздел 3. Основы CSS. Стилизация и вёрстка	18	12		6	12
<i>Тема 3.1 Стилизация и верстка веб-страниц с помощью каскадных таблиц стилей</i>	18	12		6	12
Раздел 4. Основы JavaScript. Динамика и события	22	14		8	14
<i>Тема 4.1 Основы программирования на языке JavaScript</i>	22	14		8	14
Раздел 5. Работа с данными. API и хранение	6	4		2	4
<i>Тема 5.1 Работа с API и локальным хранилищем</i>	6	4		2	4
Раздел 6. Документирование, тестирование и презентация проекта	8	6		2	6
<i>Тема 6.1 Тестирование проекта, составление документации и презентация</i>	8	6		2	6
Консультации	6				
Экзамен	3				
ВСЕГО	89	50		30	50

2.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
СЕМЕСТР 5			
Раздел 1. Введение в веб-разработку. Инструменты		8	
Тема 1.1 Введение в веб-разработку. Инструменты разработчика	Содержание учебного материала	8	ПК.2.1., ОК.02, ОК.04, ОК.07
	1 Принцип работы веб: клиент-серверная архитектура Что такое веб-разработка: фронтенд, бэкенд, полный стек. Клиент-серверное взаимодействие: HTTP/HTTPS, запросы и ответы (GET, POST). Основные протоколы: URL, DNS, порты (80, 443). Понятие веб-сервера. Статические vs динамические сайты.	2	
	2 Инструменты разработчика Редакторы кода: VS Code (расширения, Live Server, Emmet). Браузеры и инструменты разработчика (Chrome DevTools): вкладки Elements, Console, Network. Система контроля версий Git: репозиторий, коммиты, ветки (основная идея). Платформы: GitHub (создание репозитория, README).	2	
	в том числе, практических занятий	4	
	ПЗ №1. Установка и настройка среды. Создание первого HTML-файла	2	
	ПЗ №2. Git и GitHub: создание собственного репозитория	2	
Раздел 2. Основы HTML. Структура веб-страницы		18	
Тема 2.1 HTML-теги и структура документа	Содержание учебного материала	18	ПК.2.1., ОК.02, ОК.04, ОК.07
	3 Базовые теги и структура документа DOCTYPE, html, head, body. Метатеги (charset, viewport), заголовок (title). Теги заголовков (h1–h6), абзацы (p), переносы строк (br), горизонтальные линии (hr).	2	
	4 Ссылки и изображения Гиперссылки: тег <a>, относительные/абсолютные пути, якоря). Изображения: img (src, alt, width, height). Форматы изображений: PNG, JPEG, GIF, SVG, WebP (когда что использовать).	2	
	5 Форматирование текста и списки. Теги: strong, em, mark, small, del, ins. Маркированные (ul/li) и нумерованные (ol/li) списки. Вложенные списки.	2	
	6 Таблицы и формы Таблицы: table, tr, td, th, caption, объединение ячеек (colspan, rowspan). Формы: form (method, action), input (text, password, email, number, radio, checkbox), select, textarea, button.	2	
	в том числе, практических занятий	10	
	ПЗ №3. Создание страницы-визитки компании	2	

	<i>ПЗ №4. Работа со ссылками и изображениями</i>	2	
	<i>ПЗ №5. Вёрстка простой таблицы (расписание)</i>	2	
	<i>ПЗ №6. Создание формы обратной связи</i>	2	
	<i>ПЗ №7. Разработка мини-проекта «Страница о себе»</i>	2	
	Итого за семестр	26	
СЕМЕСТР 6			
Раздел 3. Основы CSS. Стилизация и вёрстка		18	
Тема 3.1 Стилизация и вёрстка веб-страниц с помощью каскадных таблиц стилей	Содержание учебного материала	18	ПК.2.1., ОК.02, ОК.04, ОК.07
	7 Подключение CSS. Селекторы Способы подключения: инлайн, в теге style, внешний файл (link). Селекторы: по тегу, классу (.), идентификатору (#), универсальный (*), потомки, дочерние, соседние. Каскадность и специфичность (базовое понимание).	2	
	8 Блочная модель. Отступы и рамки Блочные (div, p, h1) и строчные (span, a) элементы. Блочная модель: margin, padding, border. Разница между margin и padding. width, height, box-sizing (border-box vs content-box).	2	
	9 Свойства оформления текста и фона. Позиционирование Flexbox Цвета (HEX, RGB, RGBA, названия). Фон: background-color, background-image. Шрифты: font-family, font-size (px, rem, em), font-weight, text-align, line-height, text-decoration. Цвета (HEX, RGB, RGBA, названия). Фон: background-color, background-image. Шрифты: font-family, font-size (px, rem, em), font-weight, text-align, line-height, text-decoration. Позиционирование: static, relative, absolute, fixed (базовые сценарии). Flexbox: flex-container, flex-direction, justify-content, align-items, gap.	2	
	в том числе, практических занятий	12	
	<i>ПЗ №8. Подключение CSS. Стилизация страницы-визитки</i>	2	
	<i>ПЗ №9. Работа с селекторами и классами</i>	2	
	<i>ПЗ №10. Работа с блочной моделью и ее свойствами: margin, padding, border</i>	2	
	<i>ПЗ №11. Работы с гибкой вёрткой Flexbox</i>	2	
	<i>ПЗ №12. Разработка сайта-портфолио: вёрстка шапки и основной секции</i>	2	
<i>ПЗ №13. Разработка сайт-портфолио: секция проектов и подвал</i>	2		
Раздел 4. Основы JavaScript. Динамика и события		22	
Тема 4.1 Основы программирования на языке JavaScript	Содержание учебного материала	22	ПК.2.2., ОК.02, ОК.04, ОК.07
	10 Основы синтаксиса JavaScript Переменные: var, let, const. Типы данных: number, string, boolean, null, undefined. Операторы: арифметические, сравнения, логические. Условные конструкции: if/else, switch. Циклы: for, while.	2	
	11 Функции и массивы Объявление функций (function declaration, function expression, стрелочные). Параметры и return. Область видимости. Массивы: создание, доступ, методы (push, pop, shift, unshift, forEach, map, filter).	2	
	12 Объекты и DOM Объекты: литерал, свойства, методы. JSON (stringify/parse). DOM (Document Object Model): получение элементов (getElementById, querySelector, querySelectorAll), изменение содержимого (innerHTML,	2	

		<i>textContent</i>), изменение стилей и атрибутов.		
	13	События и обработчики (2 ч) События: <i>click, input, submit, mouseover, keydown</i> . Добавление обработчиков: <i>addEventListener, inline onclick</i> . Взаимодействие с формами: получение значений, валидация (<i>alert, console.log</i>).	2	
		в том числе, практических занятий	14	
		ПЗ №14. Работа с переменными, условиями, циклами в консоли браузера	2	
		ПЗ №15. Создание собственных функций. Работа с массивами и обработка данных	2	
		ПЗ №16. Работа с объектами. Знакомство с DOM	2	
		ПЗ №17. Обработка событий элементов	2	
		ПЗ №18. Взаимодействие с формами: валидация	2	
		ПЗ №19. Динамическое создание элементов (список задач)	2	
		ПЗ №20. Разработка мини-игры «Угадай число»	2	
Раздел 5. Работа с данными. API и хранение			6	
Тема 5.1 Работа с API и локальным хранилищем	Содержание учебного материала		6	ПК.2.2., ПК.2.3., ОК.02, ОК.04, ОК.07
	14	Работа с API. Fetch. Хранение данных на клиенте Что такое API, REST, эндпоинты. Асинхронность: промисы, <i>async/await</i> (базовое понимание). Функция <i>fetch</i> : GET-запрос, обработка ответа (<i>json</i>), отображение ошибок. <i>LocalStorage</i> и <i>SessionStorage</i> : <i>setItem, getItem, removeItem</i> . Хранение объектов (<i>JSON.stringify / JSON.parse</i>). Сохранение настроек темы или списка задач.	2	
		в том числе, практических занятий	4	
		ПЗ №21. Fetch и отображение данных из API	2	
		ПЗ №22. Сохранение списка задач в <i>LocalStorage</i>	2	
Раздел 6. Документирование, тестирование и презентация проекта			8	
Тема 6.1 Тестирование проекта, составление документации и презентация	Содержание учебного материала		8	ПК.2.2., ПК.2.3., ОК.02, ОК.04, ОК.07
	15	Ручное тестирование и документация Что такое тестирование: тест-кейсы, чек-листы, баг-репорт. Составление чек-листа для веб-страницы. Техническая документация: README (структура: название, описание, установка, использование, технологии). Оформление скриншотов: правила, подписи.	2	
		в том числе, практических занятий	6	
		ПЗ №23. Составление чек-листа	2	
		ПЗ №24. Ручное тестирование проекта	2	
	ПЗ №25. Оформление README и презентация итогового проекта	2		
Консультации к экзамену	Содержание		6	ПК.2.1., ПК.2.2., ПК.2.3., ОК.02, ОК.04, ОК.07
		1. HTML-теги и структура документа	2	
		2. Основы программирования на языке JavaScript	2	
		3. Работа с API и локальным хранилищем	2	
Экзамен			3	
ВСЕГО:			89	

3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1 Для реализации программы учебной дисциплины предусмотрены следующие специальные помещения:

Кабинет «Разработки информационных систем»

оснащен оборудованием:

- рабочее место педагога (преподавательский стол (1 шт.), стул (1 шт.))
- рабочие места обучающихся (парты ученические, стулья ученические – по количеству обучающихся);
- доска учебная (меловая трех-секционная) (1 шт.)
- книжный шкаф – 1 шт.;
- учебно-методическая литература по дисциплине;
- комплект учебно-наглядных пособий;

техническими средствами обучения:

– персональный компьютер, подключение к сети Интернет с модулем контентной фильтрации Traffic Inspector, NetPolice и YandexDNS, возможность трансляции на экран аудио и видео информации (1 шт.)

– программное обеспечение на ПК – Microsoft Windows 10 или аналог, Microsoft Office (Word, Excel, PowerPoint) или аналог, 7Zip, 24PDF, Яндекс Браузер (1 шт.)

– программное обеспечение Android Studio, Visual Studio, Visual Studio Code или налоги

- монитор (1 шт.)
- клавиатура (1 шт.)
- мышь (1 шт.)
- телевизор (1 шт.)
- кабель для подключения HDMI (1 шт.)

3.2 Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд института имеет печатные и электронные образовательные и информационные ресурсы, в том числе рекомендованные ФУМО, для использования в образовательном процессе. Список дополнен новыми изданиями.

3.2.1 Основные источники

1. Полуэктова, Н. Р. Разработка веб-приложений : учебник для среднего профессионального образования / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 204 с. — (Профессиональное образование). — ISBN 978-5-534-18644-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588469>

2. Тузовский, А. Ф. Проектирование и разработка web-приложений : учебник для среднего профессионального образования / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 219 с. — (Профессиональное образование). — ISBN 978-5-534-16767-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/565693> – Доп. лит. ПООП.

3. Соколова, В. В. Разработка мобильных приложений : учебник для среднего профессионального образования / В. В. Соколова. — Москва : Издательство Юрайт, 2025. — 160 с. — (Профессиональное образование). — ISBN 978-5-534-16868-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/566082>

4. Бессмертный, И. А. Интеллектуальные системы : учебник и практикум для среднего профессионального образования / И. А. Бессмертный, А. Б. Нугуманова, А. В. Платонов. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 250 с. — (Профессиональное образование). — ISBN 978-5-534-20730-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/586775>

5. Замятина, О. М. Инфокоммуникационные системы и сети. Основы моделирования : учебник для среднего профессионального образования / О. М. Замятина. — Москва : Издательство Юрайт, 2025. — 167 с. — (Профессиональное образование). — ISBN 978-5-534-17558-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/566086>

6. Зараменских, Е. П. Информационные системы: управление жизненным циклом : учебник и практикум для среднего профессионального образования / Е. П. Зараменских. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 486 с. — (Профессиональное образование). — ISBN 978-5-534-21416-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/571329>

3.2.2 Дополнительные источники

7. Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 196 с. — (Профессиональное образование). — ISBN 978-5-534-18760-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/584552>

8. Гниденко, И. Г. Технология разработки программного обеспечения : учебник для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 248 с. — (Профессиональное образование). — ISBN 978-5-534-18131-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/585518>

4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Результаты обучения	Критерии оценки	Методы оценки
<ul style="list-style-type: none"> – <i>методы алгоритмизации и формализации задач;</i> – <i>основы языков разметки и стилей</i> – <i>основы программирования на языке JavaScript;</i> – <i>основы типографики и работы с графикой;</i> – <i>основы работы с базами данных</i> – <i>клиент-серверное взаимодействие (HTTP, API);</i> – <i>методы тестирования;</i> – <i>стандарты оформления кода и документации;</i> – <i>человеко-ориентированный подход к интерфейсам;</i> – <i>основы управления версиями (Git);</i> 	<p>Оценка «5»:</p> <p>Самостоятельно строит алгоритмы средней сложности (поиск максимума, сортировка пузырьком). Обосновывает выбор алгоритмической конструкции. Понимает разницу между линейными, разветвляющимися и циклическими алгоритмами. Применяет формализацию для описания бизнес-логики веб-приложения.</p> <p>Анализирует и объясняет работу блочной модели (box-sizing, margin/padding/border). Понимает принципы адаптивной вёрстки (flexbox, медиазапросы). Объясняет, в каких случаях использовать разные форматы изображений (WebP vs PNG vs JPEG).</p> <p>Разбирается в семантической вёрстке и её влиянии на SEO/доступность.</p> <p>Объясняет замыкания и контекст this (на базовом уровне). Понимает асинхронность (промисы, async/await). Разбирается в методах работы с DOM (querySelector, addEventListener). Объясняет, как работают события всплытия (bubbling) и делегирование. Может прочитать и объяснить чужой код средней сложности.</p> <p>Анализирует макет с точки зрения читаемости и иерархии. Обосновывает выбор формата изображения под конкретную задачу (WebP, SVG, адаптивные изображения srcset).</p> <p>Понимает принципы доступности текста (размер шрифта, контрастность согласно WCAG). Объясняет, как анимация и графика влияют на восприятие интерфейса.</p> <p>Объясняет принципы нормализации (1НФ, 2НФ, 3НФ — базово). Понимает индексы и их влияние на производительность. Различает типы связей (один-к-одному, один-ко-многим, многие-ко-многим). Может спроектировать простую схему БД для веб-приложения (например, блог: пользователи, статьи, комментарии).</p> <p>Объясняет работу fetch API и асинхронных запросов. Понимает концепцию REST (ресурсы, эндпоинты, идемпотентность). Различает коды состояния клиентских и серверных ошибок (4xx, 5xx). Объясняет, как работают куки, сессии, токены (базовый уровень). Анализирует сетевые запросы в DevTools.</p> <p>Применяет техники тест-дизайна (классы эквивалентности, граничные значения) для составления тест-кейсов. Понимает роль регрессионного тестирования. Объясняет, как тестирование вписывается в жизненный цикл разработки. Различает критичность дефектов (блокер, критический, мажорный, минорный).</p>	<p>Текущий контроль:</p> <p>Компьютерное тестирование по основным разделам программы.</p> <p>Решение задач по основным разделам программы</p> <p>Промежуточная аттестация:</p> <p>экзамен</p>

	<p>Применяет стандарты форматирования кода (отступы, порядок свойств в CSS). Понимает, что такое самодокументируемый код. Оформляет техническую документацию с оглавлением, гиперссылками, структурой (по ГОСТ или шаблону). Различает документацию для пользователя и техническую документацию. Анализирует интерфейс с точки зрения пользовательского опыта (UX). Объясняет принципы F-паттерна и Z-паттерна для чтения страниц. Понимает важность иерархии информации и когнитивной нагрузки. Обосновывает, почему одни интерфейсы интуитивны, а другие — нет. Знает базовые принципы юзабилити (по Якобу Нильсену, хотя бы 3–4). Объясняет модель GitFlow (основные ветки main/develop, feature-ветки). Понимает принцип работы pull request (или merge request) на GitHub. Умеет откатывать изменения (reset, revert). Объясняет, что такое gitignore и зачем он нужен.</p> <p>Оценка «4»: Умеет формализовать простую задачу в виде алгоритма. Применяет псевдокод для описания решения. Различает основные нотации (блок-схемы, псевдокод). Объясняет семантику HTML-тегов (header, nav, main, section, article, footer). Понимает каскадность и специфичность селекторов. Знает основные CSS-свойства (цвет, шрифт, фон, отступы, рамки). Различает блочные и строчные элементы. Объясняет разницу между var, let, const. Понимает области видимости. Умеет создавать и вызывать функции (в том числе стрелочные). Работает с массивами (push, pop, forEach, map). Понимает основы работы с объектами. Объясняет правила выбора шрифтов: гарнитура (serif/sans-serif), кегль, интерлиньяж, цветовая контрастность. Различает форматы по сценариям использования (PNG для логотипов с прозрачностью, JPEG для фотографий). Понимает теорию цвета (RGB, HEX, контрастность). Объясняет основные операции CRUD (INSERT, SELECT, UPDATE, DELETE). Понимает первичный ключ, внешний ключ. Знает, как выполнить JOIN двух таблиц (базовый уровень). Объясняет структуру HTTP-запроса и ответа (заголовки, тело, коды состояния: 200, 404, 500). Понимает разницу между GET, POST, PUT, DELETE. Разбирается в структуре JSON. Объясняет разницу между модульным, интеграционным и UI-тестированием. Умеет составлять чек-лист для проверки веб-страницы. Понимает, что такое баг-репорт и из каких</p>	
--	---	--

	<p>полей он состоит.</p> <p>Объясняет основные правила именования переменных (camelCase, осмысленные имена). Знает, как оформлять комментарии в HTML, CSS, JS. Умеет оформлять скриншоты и подписи к ним в документации.</p> <p>Объясняет принципы эргономики: размер кликабельных элементов, понятная навигация, обратная связь (hover, активные состояния).</p> <p>Понимает основы доступности (alt-тексты для изображений, контрастность).</p> <p>Объясняет, что такое ветка (branch), как создать ветку, переключиться между ветками.</p> <p>Понимает, как разрешать простые конфликты.</p> <p>Знает, как посмотреть историю коммитов (log, diff).</p> <p>Оценка «3»:</p> <p>Знает, что такое алгоритм. Может нарисовать простую блок-схему (например, «проверить чётность числа»). Понимает базовые конструкции: следование, ветвление, цикл.</p> <p>Знает базовые HTML-теги (html, head, body, h1–h6, p, a, img, ul, ol, li, table, form, input).</p> <p>Понимает разницу между тегами и атрибутами.</p> <p>Может назвать способы подключения CSS.</p> <p>Знает базовый синтаксис: переменные (let, const), типы данных (числа, строки, булевы), условные операторы (if/else), простые циклы (for).</p> <p>Знает разницу между растровой и векторной графикой. Может назвать основные форматы (PNG, JPEG, GIF, SVG). Понимает единицы измерения шрифтов (px, em, rem).</p> <p>Знает, что такое база данных, таблица, запись, поле. Понимает разницу между SQL и NoSQL (на уровне «SQL — таблицы, NoSQL — документы»). Может написать простой SELECT-запрос.</p> <p>Знает, что такое HTTP/HTTPS. Различает методы GET и POST. Понимает, что такое API и для чего он нужен. Знает, что такое JSON.</p> <p>Знает, зачем нужно тестировать веб-приложения. Может назвать виды тестирования (ручное, автоматизированное). Понимает, что такое тест-кейс.</p> <p>Знает, что существуют стандарты оформления кода (единообразие). Понимает, зачем нужна документация. Может перечислить основные разделы README (название, описание, установка).</p> <p>Знает, что интерфейс должен быть удобным для пользователя. Может назвать основные элементы интерфейса (кнопки, поля ввода, ссылки).</p> <p>Знает, зачем нужна система контроля версий. Может назвать базовые команды: init, add, commit, push, pull. Понимает разницу между</p>	
--	---	--

<ul style="list-style-type: none"> – создавать веб-страницы с использованием HTML; – оформлять стили с помощью CSS; – программировать на языке JavaScript; – работать с DOM и событиями; – выполнять запросы к API (fetch / AJAX); – создавать и запрашивать данные из простой БД; – проводить ручное тестирование веб-страницы; – оформлять техническую документацию (README, руководство); – использовать Git для хранения кода; – презентовать результат; 	<p>локальным и удалённым репозиторием.</p> <p>Оценка «5»</p> <p>Создаёт полностью семантически верную структуру документа. Использует iframe, аудио/видео теги (audio, video), внедряет сторонние карты (Google Maps через iframe). Форма содержит корректные атрибуты (required, placeholder, pattern). HTML-код проходит валидацию (W3C Validator).</p> <p>Реализует сложную верстку с использованием CSS Grid. Создает анимации и переходы (transition, @keyframes). Использует кастомные свойства (CSS variables). Стилизация полностью соответствует макету (в том числе hover-эффекты, активные состояния). Код семантичен, и не содержит !important без необходимости.</p> <p>Применяет функциональные методы (filter, reduce). Понимает и использует замыкания. Пишет чистые функции без побочных эффектов. Код структурирован, читаем, соответствует стандартам (camelCase, осмысленные имена). Использует строгий режим ('use strict').</p> <p>Применяет делегирование событий (event delegation). Работает с формами: собирает данные, валидирует поля, выводит сообщения об ошибках. Использует localStorage для сохранения состояния между сессиями.</p> <p>Обработывает события клавиатуры (keydown, keyup). Понимает и применяет всплытие (bubbling) и остановку распространения (stopPropagation).</p> <p>Выполняет POST-запрос с отправкой данных (метод, headers, body). Работает с разными эндпоинтами. Реализует загрузку с индикатором (loading spinner). Добавляет повторные попытки (retry) при ошибках. Понимает CORS и знает, как его обрабатывать в учебных проектах.</p> <p>Проектирует простую схему БД (2–3 связанные таблицы). Выполняет JOIN двух таблиц.</p> <p>Использует параметризованные запросы (безопасность, защита от SQL-инъекций — концептуально). Может подключиться к БД из JavaScript через простой бэкенд (например, JSON Server или mock API).</p> <p>Оформляет баг-репорты по структуре: заголовок, шаги воспроизведения, ожидаемый результат, фактический результат, скриншот.</p> <p>Проверяет кроссплатформенность (Chrome, Firefox). Проводит исследовательское тестирование (exploratory testing) без чек-листа.</p> <p>Приоритезирует баги по критичности.</p> <p>README содержит: бейджи (статус, технологии), оглавление с гиперссылками, примеры кода, раздел «Планы по доработке», лицензию. Скриншоты подписаны и обрезаны. Документ сверстан в Markdown с заголовками, списками, таблицами, ссылками. Руководство пользователя (отдельный файл) содержит</p>	<p>Текущий контроль:</p> <p>Оценка результатов выполнения практической работы</p> <p>Экспертное наблюдение за ходом выполнения практической работы (деятельностью студента)</p> <p>Оценка выполнения практического задания (работы)</p> <p>Промежуточная аттестация:</p> <p>экзамен</p>
--	---	---

	<p>пошаговые инструкции для конечного пользователя.</p> <p>Использует gitignore для исключения ненужных файлов. Работает с удалённым репозиторием (fetch, rebase — базово). Откатывает изменения (git reset, git revert). Оформляет pull request на GitHub с описанием изменений. Демонстрирует понимание GitFlow (ветки main/develop + feature).</p> <p>Использует слайды (3–5 шт.) с визуалами (скриншоты, схемы). Говорит чётко, без воды. Отвечает на сложные вопросы (почему выбран такой подход, что можно улучшить).</p> <p>Демонстрирует знание альтернативных решений и компромиссов. Проводит презентацию без чтения с листа, держит контакт с аудиторией.</p> <p>Оценка «4»</p> <p>Использует семантические теги (header, nav, main, section, article, footer). Создаёт таблицы (таблицу с объединением ячеек — colspan/rowspan). Строит форму с различными полями ввода (text, email, radio, checkbox, select, textarea).</p> <p>Использует блочную модель (margin, padding, border). Применяет Flexbox для выравнивания элементов (justify-content, align-items, gap).</p> <p>Создаёт адаптивные элементы с помощью медиазапросов (media).</p> <p>Использует массивы и их методы (push, pop, forEach, map). Работает с объектами (создание, чтение свойств). Пишет стрелочные функции.</p> <p>Обрабатывает простые ошибки (try/catch на базовом уровне).</p> <p>Динамически создаёт и удаляет элементы (createElement, appendChild, remove). Меняет стили элементов через style или classList.</p> <p>Использует события input, submit, mouseover.</p> <p>Предотвращает стандартное поведение (preventDefault).</p> <p>Обрабатывает ошибки сети и HTTP-статусы (catch, проверка response.ok). Парсит JSON и отображает данные в виде карточек или таблицы. Использует async/await вместо цепочек then.</p> <p>Выполняет запросы с условиями (WHERE, ORDER BY). Создаёт таблицу с первичным ключом. Вставляет (INSERT), обновляет (UPDATE), удаляет (DELETE) записи.</p> <p>Составляет чек-лист из 5–10 пунктов перед тестированием. Проверяет граничные значения полей (пустое поле, слишком длинная строка, специальные символы). Фиксирует баги в письменном виде (что ожидалось, что получили).</p> <p>README содержит разделы: описание, установка и запуск, использование (скриншоты), использованные технологии.</p>	
--	--	--

	<p>Присутствуют оформленные скриншоты интерфейса.</p> <p>Создаёт ветки (branch), переключается между ними (checkout). Делает слияние веток (merge). Разрешает простые конфликты. История коммитов осмысленная (не «update»), а фиксирующая конкретные изменения).</p> <p>Структурирует презентацию: проблема → решение → демонстрация → технологии → выводы. Демонстрирует не только работающий код, но и показывает ключевые фрагменты. Укладывается в 5 минут.</p> <p>Оценка «3»</p> <p>Создаёт простую веб-страницу с базовыми тегами: заголовки h1–h6, абзацы, списки (ul/ol), ссылки (a), изображения (img). Страница отображается в браузере без ошибок.</p> <p>Подключает внешний CSS-файл. Применяет селекторы по тегу и классу. Умеет задавать цвета, шрифты, размеры, фоны. Результат отличается от стилей браузера по умолчанию.</p> <p>Пишет код с использованием переменных (let, const), условных операторов (if/else), циклов (for). Функции объявляются и вызываются корректно. Решает простые вычислительные задачи (сумма, поиск минимума/максимума).</p> <p>Находит элементы на странице (getElementById, querySelector). Изменяет содержимое через innerHTML или textContent. Вешает обработчик события click на кнопку.</p> <p>Выполняет GET-запрос с помощью fetch. Обрабатывает ответ с помощью then() и .json(). Выводит полученные данные на страницу (например, в список).</p> <p>Пишет простой SQL-запрос SELECT * FROM table. Понимает, как подключиться к БД через инструмент командной строки или phpMyAdmin (в рамках учебного окружения).</p> <p>Проверяет, открывается ли страница без ошибок. Кликает по ссылкам — они ведут туда, куда нужно. Заполняет форму — данные отправляются или выводятся в консоль.</p> <p>Сообщает о явных багах устно.</p> <p>Создаёт файл README.md, содержащий название проекта и краткое описание (1–2 предложения).</p> <p>Инициализирует репозиторий (git init).</p> <p>Выполняет базовые команды: add, commit -m "message", push, pull. Код загружен на GitHub (или аналоги).</p> <p>Рассказывает, что делает приложение. Показывает его в действии (live demo). Отвечает на вопросы по функционалу.</p> <p>Продолжительность — 2–3 минуты.</p>	
--	--	--