

Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет»

УТВЕРЖДАЮ

Ректор Белорусско-Российского
университета

М.Е. Лустенков

Документ подписан
простой электронной подписью

Информация о владельце:

ФИО: Лустенков Михаил Евгеньевич

Должность: Ректор

Дата: 09.09.2026

Регистрационный № УД-380302/Б.1.026/р

ТЕХНОЛОГИИ ИНТЕРНЕТ-ПРОГРАММИРОВАНИЯ

(наименование дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Направление подготовки 38.03.02 – Менеджмент

Направленность (профиль) Управление электронным бизнесом

Квалификация Бакалавр

	Форма обучения
	Очная
Курс	2
Семестр	3, 4
Лекции, часы	32
Лабораторные занятия, часы	68
Экзамен, семестр	3, 4
Курсовая работа, семестр	4
Контактная работа по учебным занятиям, часы	100
Самостоятельная работа, часы	152
Всего часов / зачетных единиц	252/7

Кафедра-разработчик программы: Программное обеспечение информационных технологий

(название кафедры)

Составитель: Ю. В. Вайнилович, канд. техн. наук

Могилев, 2026

Рабочая программа составлена в соответствии с федеральным государственным образовательными стандартами высшего образования - бакалавриат по направлению подготовки 38.03.02 Менеджмент № 970 от 12.08.2020, учебным планом рег. № 380302-2 от 18.02.2026.

Рассмотрена и рекомендована к утверждению кафедрой «Программное обеспечение информационных технологий»

«02» марта 2026 г., протокол № 8.

Зав. Кафедрой  В. В. Кутузов

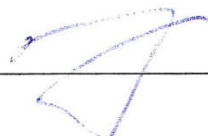
Одобрена и рекомендована к утверждению Научно-методическим советом
Белорусско-Российского университета
(протокол № 6 от «15» апреля 2026 г.)

Рецензент:

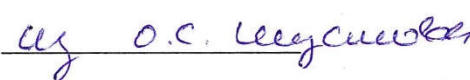
Заведующий кафедрой программного обеспечения информационных технологий
УО «Могилевский государственный университет им. А.А. Кулешова»,
к.т.н., доцент И.В. Акиншева

Рабочая программа согласована:

Зав. кафедрой «Маркетинг и менеджмент»
(название выпускающей кафедры)

 А. В. Александров

Ведущий библиотекарь

 О. С. Кузнецов

Начальник учебно-методического
отдела

 О. Е. Печковская

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1. Цель дисциплины

Цель дисциплины состоит в подготовке студентов к профессиональной деятельности в области веб-разработки путём формирования компетенций, связанных с проектированием пользовательских интерфейсов, разработкой программного обеспечения, его отладкой и тестированием.

1.2. Планируемые результаты изучения дисциплины

В результате освоения учебной дисциплины студент должен:

знать:

- принципы построения и функционирования веб-приложений;
- архитектуру клиент-серверного взаимодействия;
- основные стандарты, технологии и средства разработки пользовательских интерфейсов;
- принципы структурирования и стилизации веб-документов;
- основы программирования на языке сценариев веб-страниц;
- методы организации интерактивного взаимодействия пользователя с веб-интерфейсом;
- принципы работы с сетевыми запросами и обменом данными между клиентской и серверной частями приложения;
- основы компонентного подхода при разработке интерфейсов;
- принципы управления состоянием приложения;
- основы организации серверной части веб-приложений;
- принципы работы с системами хранения данных;
- методы контроля версий, отладки, тестирования и сопровождения программного обеспечения.

уметь:

- использовать современные программные средства для разработки веб-приложений;
- проектировать пользовательские интерфейсы по готовому образцу или концепции;
- создавать структурированные веб-документы и оформлять их в соответствии с требованиями макета;
- разрабатывать адаптивные и интерактивные пользовательские интерфейсы;
- реализовывать программную логику клиентской части приложения;
- организовывать обмен данными с внешними и внутренними источниками;
- разрабатывать компоненты интерфейса и обеспечивать их взаимодействие;
- применять средства управления состоянием приложения;
- разрабатывать серверную часть веб-приложения и реализовывать взаимодействие с базой данных;
- применять инструменты контроля версий в индивидуальной и командной разработке;
- выполнять отладку, тестирование и базовую оптимизацию веб-приложений.

владеть:

- навыками использования современных средств разработки программного обеспечения;
- навыками проектирования и реализации пользовательских интерфейсов;
- навыками вёрстки веб-страниц и построения адаптивных интерфейсов;
- навыками разработки клиентской логики веб-приложений;
- навыками реализации компонентного подхода в разработке интерфейсов;
- навыками организации взаимодействия клиентской и серверной частей веб-приложения;
- навыками работы с системами контроля версий;
- навыками разработки, отладки и тестирования программного обеспечения;
- навыками интеграции программных компонентов в единое веб-приложение;
- навыками сопровождения и доработки программных решений.

1.3 Место учебной дисциплины в системе подготовки студента

Дисциплина относится к блоку 1 «Дисциплины (модули)» (Обязательная часть блока 1).

Перечень учебных дисциплин, изучаемых ранее, усвоение которых необходимо для изучения данной дисциплины:

- Информатика.

Перечень учебных дисциплин (циклов дисциплин), которые будут опираться на данную дисциплину:

- Графический дизайн и прототипирование;
- Основы машинного обучения;
- Системы искусственного интеллекта.

Кроме того знания, полученные при изучении дисциплины на лекционных занятиях будут применимы при прохождении технологической (проектно-технологической) практики, а также при подготовке выпускной квалификационной работы и дальнейшей профессиональной деятельности.

1.4 Требования к освоению дисциплины

Освоение данной учебной дисциплины должно обеспечивать формирование следующих компетенций

Коды формируемых компетенций	Наименование формируемых компетенций
ОПК-5	Способен использовать при решении профессиональных задач современные информационные технологии и программные средства, включая управление крупными массивами данных и их интеллектуальный анализ
ПК-6	Способен планировать и организовывать выполнение проектов по разработке программного обеспечения, в том числе анализировать требования клиентов к постпродажному обслуживанию и сервису

2 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Вклад дисциплины в формирование результатов обучения выпускника (компетенций) и достижение обобщенных результатов обучения происходит путём освоения содержания обучения и достижения частных результатов обучения, описанных в данном разделе.

2.1 Содержание дисциплины

Номера тем	Наименование тем	Содержание	Коды формируемых компетенций
1	2	3	4
1	Введение в web-программирование	Архитектура клиент-сервер. Жизненный цикл запроса (DNS → TCP → HTTP → рендеринг). Роли HTML, CSS, JavaScript. SPA vs MPA. VS Code: расширения, Emmet, Live Server. DevTools (Elements, Console, Network, Application). Стандарты W3C/WHATWG, кроссбраузерность, Can I Use	ОПК-5 ПК-6
2	Система контроля версий Git и GitHub	Понятие системы контроля версий, варианты СКВ. Терминология Git, настройка конфигурации, создание репозитория. Основные команды (add, commit, log, diff, checkout, reset). Ветвление, слияние, разрешение конфликтов. Работа с удалённым сервером: push, pull, clone. Pull request и code review. .gitignore, README.md, Markdown. GitHub Pages	ОПК-5 ПК-6

3	Язык разметки гипертекста HTML	HTTP, URL. Структура документа (DOCTYPE, head, body, meta). Семантические теги. Ссылки, якоря, списки, таблицы. Формы: элементы, типы input, валидация (required, pattern). Мультимедиа. Форматы графики. SVG. Доступность: ARIA, alt, роли. SEO-основы.	ОПК-5 ПК-6
4	Каскадные таблицы стилей CSS	Подключение стилей. Селекторы. Специфичность. Каскад и наследование. Блочная модель. Единицы измерения. Позиционирование. Z-index. Overflow. Flexbox. Grid. Медиазапросы, Mobile First. Адаптивные изображения. CSS-переменные. Анимации. Псевдоклассы и псевдоэлементы. Шрифты. Оптимизация изображений.	ОПК-5 ПК-6
5	Профессиональная вёрстка	Чтение макета в Figma (Inspect / Dev Mode, экспорт ассетов). Методология ВЕМ. Препроцессоры: Sass/SCSS. CSS-фреймворк Bootstrap. Tailwind CSS.	ОПК-5 ПК-6
6	JavaScript	Подключение скриптов. Переменные. Типы данных. Операторы. Условные конструкции. Циклы. Функции: объявление, выражение, стрелочные, параметры, return. Область видимости. Регулярные выражения. Массивы: CRUD-методы, методы перебора. Объекты: создание, перебор, вложенность. Деструктуризация. Spread/rest. JSON. Замыкания. Контекст (this), call, apply, bind. Колбэки, функции высшего порядка. ES6-классы: constructor, методы, наследование (extends, super). Модули ES6. DOM-дерево. Поиск, создание, удаление элементов. DocumentFragment. textContent, innerHTML, classList, setAttribute. addEventListener, объект event. Всплытие и погружение. Делегирование событий. Работа с формами. Синхронный vs асинхронный код. Event Loop. Таймеры. requestAnimationFrame. Promises. Fetch API. Async/await. Try/catch/finally. cookie, localStorage, sessionStorage.	ОПК-5 ПК-6
7	Инструменты разработки и сборки	Node.js – установка и назначение. Пакетный менеджер NPM. Понятие о бандлерах: Vite, Webpack. Минификация, Babel – транспиляция. Линтинг и форматирование: ESLint, Prettier. Подключение сторонних библиотек через NPM. ESLint, Prettier	ОПК-5 ПК-6
8	Работа с сетью и фейковым API	Протокол HTTP/HTTPS: методы, заголовки, коды ответов. REST API: принципы, ресурсы, эндпоинты. Формат JSON. Фейковые серверы: json-server, MockAPI.io, JSONPlaceholder. CORS. Практика CRUD через Fetch/async-await. Библиотека Axios: установка, синтаксис, сравнение с Fetch, интерсепторы, обработка ошибок. Индикация загрузки и обработка ошибок	ОПК-5 ПК-6
9	Библиотека React	Понятие фреймворка. Компонентный подход. Создание проекта. JSX-синтаксис. Функциональные компоненты. Props. Состояние. Обработка событий. Условный рендеринг. Хуки. Управляемые формы. Маршрутизация. Библиотеки UI-компонентов	ОПК-5 ПК-6
10	Архитектурный паттерн Redux	Проблема prop drilling, зачем нужен глобальный стейт. React Redux, Redux Toolkit, Redux thunk. Действия и редьюсеры. Middleware. Асинхронные операции.	ОПК-5 ПК-6
11	Серверные технологии web-разработки	Node.js: отличие от браузерного JS, Event Loop, модули (CommonJS, ESM), встроенные модули (fs,	ОПК-5 ПК-6

		path, http), async/await, process.env. Express.js: создание сервера, маршрутизация (params, query, body), middleware (встроенные, пользовательские, cors, morgan), обработка ошибок, структура проекта (routes → controllers → services), nodemon, dotenv.	
12	Реляционные базы данных и full-stack интеграция	PostgreSQL: реляционная модель, типы данных, SQL, связи, индексы, pgAdmin. Подключение: библиотека pg (Pool, параметризованные запросы). Sequelize: модели, миграции, валидация, связи, include. Интеграция с React: структура full-stack проекта, проху/CORS, запросы через Axios.	ОПК-5 ПК-6

2.2 Учебно-методическая карта дисциплины

№ недели	Лекции (наименование тем)	Часы	Лабораторные работы	Часы	Самостоятельная работа, часы	Форма контроля знаний	Баллы (max)
3 семестр							
Модуль 1							
1	Тема 1. Введение в web-программирование	2	ЛР №1. Среда разработки, DevTools, основы HTML	2	1	ЗЛР	6
2			ЛР №2. Система контроля версий Git и GitHub	2	1	ЗЛР	6
3	Тема 2. Система контроля версий Git и GitHub	2	ЛР №3. Фиксированная вёрстка проекта пользовательского интерфейса	2	1		
4			ЛР №3. Фиксированная вёрстка проекта пользовательского интерфейса	2	1	ЗЛР	6
5	Тема 3. Язык разметки гипертекста HTML	2	ЛР №4. Адаптивная вёрстка проекта пользовательского интерфейса	2	1		
6			ЛР №4. Адаптивная вёрстка проекта пользовательского интерфейса	2	1	ЗЛР	6
7	Тема 4. Каскадные таблицы стилей CSS	2	ЛР №5. Анимации и визуальные эффекты в CSS	2	1	ЗЛР	6
8			ЛР №6. Динамическое управление DOM-деревом	2	1	ТКУ	30
Модуль 2							
9	Тема 5. Профессиональная вёрстка	2	ЛР №7. Обработка событий и валидация форм	2	1	ЗЛР	5
10			ЛР №8. Интерактивные компоненты web-интерфейса	2	1		
11	Тема 6. JavaScript	2	ЛР №8. Интерактивные компоненты web-интерфейса	2	1	ЗЛР	5
12			ЛР №9. Клиентское хранилище данных	2	1		

13	Тема 7. Инструменты разработки и сборки	2	ЛР №9. Клиентское хранилище данных	2	2	ЗЛР	5
14			ЛР №10. Сборка и автоматизация проекта	2	2	ЗЛР	5
15	Тема 8. Работа с сетью и фейковым API	2	ЛР №11. Взаимодействие с REST API	2	2		
16			ЛР №11. Взаимодействие с REST API	2	2	ТЗ	5
17			ЛР №12. Интеграция сторонних веб-сервисов	2	2	ЗЛР ТКУ	5 30
18-20					36	ПА (экзамен)	40
	Итого	16		34	58		100
4 семестр							
Модуль 1							
1	Тема 9. Библиотека React	2	ЛР №13. Архитектура React-приложения	2	1		
2			ЛР №13. Архитектура React-приложения	2	1	ЗЛР	6
3	Тема 9. Библиотека React	2	ЛР №14. Маршрутизация в React-приложениях	2	1	ЗЛР	6
4			ЛР №15. Управление состоянием компонентов React	2	1		
5	Тема 9. Библиотека React	2	ЛР №15. Управление состоянием компонентов React	2	1	ЗЛР	6
6			ЛР №16. Нуки React. Работа с API	2	1		
7	Тема 10. Архитектурный паттерн Redux	2	ЛР №16. Нуки React. Работа с API	2	1	ЗЛР	6
8			ЛР №17. Библиотека MUI	2	1	ЗЛР ТКУ	6 30
Модуль 2							
9	Тема 11. Серверные технологии web-разработки	2	ЛР №18. Менеджер состояний Redux Toolkit	2	1		
10			ЛР №18. Менеджер состояний Redux Toolkit	2	1	ЗЛР	4
11	Тема 12. Реляционные базы данных и full-stack интеграция	2	ЛР №19. Платформа Node.js.	2	1	ЗЛР	4
12			ЛР №20. Серверный фреймворк Express	2	2	ЗЛР	4
13	Тема 12. Реляционные базы данных и full-stack интеграция	2	ЛР №21. ORM Sequelize и база данных PostgreSQL	2	2		
14			ЛР №21. ORM Sequelize и база данных PostgreSQL	2	1	ЗЛР	4

15	Тема 12. Реляционные базы данных и full-stack интеграция	2	ЛР №22. Full-stack приложение на стеке PERN	2	2		
16			ЛР №22. Full-stack приложение на стеке PERN	2	2	ЗЛР ТЗ	4 6
17			ЛР №23. Интеграция сторонних web-сервисов	2	2	ЗЛР ТКУ	4 30
1-17	Выполнение курсовой работы				36		
18-20					36	ПА (экзамен)	40
	Итого	16		34	94		100
	ИТОГО	32		68	152		

Принятые обозначения:

ТКУ – текущий контроль успеваемости;

ПА – промежуточная аттестации.

ЗЛР – защита лабораторной работы;

ТЗ – тестовое задание.

Итоговая оценка определяется как сумма текущего контроля и промежуточной аттестации и соответствует баллам:

Экзамен

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Баллы	87-100	65-86	51-64	0-50

2.3. Требования к курсовой работе

Целью курсового проектирования является приобретение практических навыков разработки web-приложений.

Примерная тематика курсовых работ хранится на кафедре.

Содержание курсовой работы включает три части:

- 1) техническое задание – анализ предметной области, сбор требований к web-приложению;
- 2) проектирование клиентской и серверной компоненты web-приложения – разработка дизайн-макетов всех страниц приложения, проектирование архитектуры клиентской и серверной компоненты в соответствии с требованиями;
- 3) реализация web-приложения – разработка клиентской компоненты web-приложения.

Разбивка этапов курсовой работы, определение количества минимальных и максимальных баллов за каждый из них производится преподавателем. Примерный перечень этапов выполнения курсовой работы и количество баллов за каждый из них представлен в таблице.

№	Этап выполнения	Минимум	Максимум
1	Техническое задание	9	15
2	Проектирование клиентской и серверной компоненты web-приложения	12	20
3	Реализация web-приложения	12	20
4	Оформление пояснительной записки и графического материала	3	5
	Итого за выполнение курсового проекта	36	60
	Защита курсовой работы	15	40

3 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При изучении дисциплины используется модульно-рейтинговая система оценки знаний студентов. Применение форм и методов проведения занятий при изучении различных тем курса представлено в таблице.

№ п/п	Форма проведения занятия	Вид аудиторных занятий		Всего часов
		Лекции	Лабораторные занятия	
2	С использованием ЭВМ	1-12	1-23	100
	ИТОГО	32	68	100

4. ОЦЕНОЧНЫЕ СРЕДСТВА

Используемые оценочные средства по учебной дисциплине представлены в таблице и хранятся на кафедре.

№ п/п	Вид оценочных средств*	Количество комплектов
1	Вопросы к экзамену / лабораторным работам	1 / 21
2	Экзаменационные билеты	1
3	Перечень тем курсовых работ	1
4	Тестовое задание	2

5 МЕТОДИКА И КРИТЕРИИ ОЦЕНКИ КОМПЕТЕНЦИЙ СТУДЕНТОВ

5.1 Формируемые компетенции и индикаторы их достижения

<i>ОПК-5. Способен проектировать пользовательские интерфейсы по готовому образцу или концепции интерфейса</i>
<i>ИОПК-5.2. Использует современные программные средства при решении профессиональных задач</i>
Знать: Архитектуру клиент-серверных веб-приложений и жизненный цикл HTTP-запроса; структуру HTML-документа, семантические теги и принципы веб-доступности; каскадные таблицы стилей CSS, модели раскладки Flexbox и Grid, принципы адаптивной вёрстки; синтаксис и возможности языка JavaScript, структуру DOM-дерева, механизмы обработки событий; принципы компонентного подхода и архитектуру библиотеки React; паттерны управления состоянием приложения; серверные технологии Node.js и фреймворк Express; принципы работы с реляционными базами данных и ORM
Уметь: Настраивать среду разработки и использовать инструменты разработчика браузера; создавать семантическую HTML-разметку по готовым макетам из Figma; разрабатывать адаптивные пользовательские интерфейсы; программировать интерактивные элементы интерфейса; разрабатывать одностраничные приложения (SPA); реализовывать управление состоянием приложений; создавать серверные приложения и REST API; работать с базами данных
Владеть: Навыками работы с редактором кода VS Code и его расширениями; навыками вёрстки адаптивных интерфейсов по макетам; навыками разработки интерактивных веб-приложений на JavaScript; навыками создания компонентов React и работы с хуками; навыками интеграции UI-библиотек (MUI, Bootstrap); навыками разработки full-stack приложений на стеке PERN
<i>ПК-6. Способен планировать и организовывать выполнение проектов по разработке программного обеспечения, в том числе анализировать требования клиентов к постпродажному обслуживанию и сервису</i>
<i>ИПК-6.2. Способен разрабатывать программное обеспечение, проводить его эффективную отладку и тестирование</i>
Знать: Принципы работы систем контроля версий Git и платформы GitHub; процессы организации командной разработки; протокол HTTP/HTTPS, методы запросов, коды ответов и принципы REST API; механизмы обработки ошибок в JavaScript и на сервере; инструменты отладки клиентского кода

(DevTools, React DevTools) и серверного кода; принципы тестирования API и инструменты для тестирования (Postman); структуру и организацию full-stack проекта
Уметь: Инициализировать репозиторий, выполнять коммиты, работать с ветками и разрешать конфликты слияния; организовывать совместную работу над проектом через GitHub; настраивать сборку проекта; выполнять отладку JavaScript-кода и React-компонентов через инструменты разработчика; тестировать REST API с помощью Postman и Thunder Client; обрабатывать ошибки на клиенте и сервере, реализовывать индикацию загрузки; документировать проект с помощью
Владеть: Навыками работы с системой контроля версий Git и платформой GitHub; навыками организации структуры проекта; навыками настройки инструментов сборки и автоматизации; навыками отладки клиентских и серверных приложений; навыками тестирования API и обработки ошибок; навыками публикации проектов; навыками ведения документации проекта

5.2 Критерии оценки лабораторных работ

1) вопросы к лабораторным работам

Балл	Критерии
6-4	Студент демонстрирует полное понимание темы, грамотно использует терминологию, даёт исчерпывающие ответы на вопросы, способен обосновать принятые решения и связать их с теоретическими положениями
3	Студент демонстрирует хорошее понимание темы, правильно использует терминологию, отвечает на вопросы с незначительными неточностями, способен обосновать основные решения
2	Студент демонстрирует базовое понимание темы, допускает ошибки в терминологии, отвечает на вопросы частично, затрудняется в обосновании принятых решений
1	Студент не понимает сути выполненной работы, не владеет терминологией, не способен ответить на основные вопросы по теме

2) тестовые задания

Балл	Критерии
5-6	85–100% правильных ответов
4	70–84% правильных ответов
3	55–69% правильных ответов
2	Менее 55% правильных ответов

5.3 Критерии оценки курсовой работы

Баллы	Критерии оценивания
36–40	Все задачи проекта решены в полном объёме, содержание полностью соответствует заданию. Приложение полностью работоспособно, все заявленные функции реализованы и корректно работают. Студент демонстрирует полную самостоятельность в решении задач, уверенно ориентируется в коде и архитектуре проекта. Код чистый, структурированный и читаемый, соблюдены принципы организации проекта. Проект содержит оригинальные решения, выходящие за рамки базовых требований. Пояснительная записка оформлена в соответствии с требованиями, графическая часть качественная и информативная. Доклад чёткий, логически выстроенный, студент свободно владеет профессиональной терминологией. Студент уверенно и аргументированно отвечает на все заданные вопросы.
28–35	Основные задачи проекта решены, содержание в целом соответствует заданию. Приложение работоспособно, основные функции реализованы, имеются незначительные ошибки. Студент выполнил работу в основном самостоятельно, ориентируется в коде, но затрудняется с объяснением отдельных решений. Код в целом читаемый, структура проекта логична, имеются незначительные нарушения стиля. Проект содержит отдельные элементы, демонстрирующие инициативность студента. Оформление в целом соответствует требованиям, имеются незначительные отклонения. Доклад понятный, терминология используется с незначительными неточностями. Студент отвечает на большинство вопросов, но отдельные ответы недостаточно полные.

Баллы	Критерии оценивания
20–27	Задачи проекта решены частично, содержание недостаточно полно раскрывает тему. Приложение частично работоспособно, часть заявленных функций не реализована или работает некорректно. Самостоятельность выполнения работы вызывает сомнения, студент слабо ориентируется в отдельных частях проекта. Код плохо структурирован, имеется значительное дублирование, отсутствует единый стиль. Проект выполнен строго по заданию без дополнительной инициативы. Оформление имеет существенные нарушения требований, графическая часть неполная. Доклад сбивчивый, нарушена логика изложения, допускаются ошибки в терминологии. Студент затрудняется с ответами на вопросы, демонстрирует поверхностное понимание материала.
15–19	Задачи проекта решены фрагментарно, содержание не соответствует заданию. Приложение неработоспособно или имеет критические ошибки. Работа выполнена несамостоятельно, студент не ориентируется в собственном проекте. Код нечитаемый, структура проекта отсутствует. Элементы творчества и инициативы отсутствуют. Пояснительная записка оформлена с грубыми нарушениями. Доклад не раскрывает содержание проекта. Студент не может ответить на вопросы.

5.4 Критерии оценки экзамена

Для оценки экзамена используется следующее оценочное средство:

1) Экзаменационные билеты

Экзаменационный билет включает 3 теоретических вопроса. Ответы на вопросы оцениваются по следующим критериям.

Баллы	Критерии оценивания
36–40	Ответ полный, логически выстроенный, демонстрирует глубокое понимание материала. Студент свободно владеет терминологией, корректно раскрывает содержание всех вопросов билета, приводит примеры и обоснования. Допускаются незначительные неточности, не влияющие на общее качество ответа.
28–35	Ответ в целом правильный и достаточно полный. Студент демонстрирует хорошее понимание материала, владеет терминологией, раскрывает основное содержание вопросов. Допускаются отдельные неточности или недостаточная аргументация по некоторым аспектам.
20–27	Ответ неполный, но содержит основные положения по вопросам билета. Студент демонстрирует общее понимание материала, однако допускает ошибки в использовании терминологии или раскрытии отдельных понятий. Ответ требует уточняющих вопросов со стороны преподавателя.
10–19	Ответ фрагментарный, раскрывает лишь отдельные аспекты вопросов билета. Студент допускает существенные ошибки, демонстрирует слабое владение терминологией и недостаточное понимание взаимосвязей между понятиями.
1–9	Ответ неудовлетворительный. Студент демонстрирует отрывочные знания, не раскрывает содержание вопросов билета, допускает грубые ошибки. Ответ не соответствует минимальным требованиям к освоению дисциплины.
0	Ответ отсутствует или полностью не соответствует содержанию вопросов билета.

6 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Самостоятельная работа обучающихся (СР) направлена на закрепление и углубление освоения учебного материала, развитие практических умений. СР включает следующие виды:

- Выполнение тестовых заданий;
- Подготовка к экзамену;
- Изучение нормативных документов;
- Подготовка к аудиторным занятиям.

Перечень контрольных вопросов и заданий для СР приведен в приложении и хранится на кафедре.

Для СРС рекомендуется использовать источники, приведенные в п. 7.

7 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1 Основная литература:

№ п/п	Библиографическое описание	Гриф	Количество экземпляров/URL
1	Полуэктова, Н. Р. Разработка веб-приложений : учебник для вузов / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 204 с.	Гриф УМО ВО	https://urait.ru/bcode/588458

7.2 Дополнительная литература

№ п/п	Библиографическое описание	Гриф	Количество экземпляров/URL
1	Сысолетин, Е. Г. Разработка интернет-приложений : учебник для вузов / Е. Г. Сысолетин, С. Д. Ростунцев ; под научной редакцией Л. Г. Доросинского. — Москва : Издательство Юрайт, 2025. — 80 с. — (Высшее образование).	Гриф УМО ВО	https://urait.ru/bcode/562916

7.3 Перечень ресурсов сети Интернет по изучаемой дисциплине

<http://moodle.bru.by> – Образовательный портал Белорусско-Российского университета;
<http://e.biblio.bru.by/> – Электронная библиотека Белорусско-Российского университета;
<https://znanium.com/> – Электронно-библиотечная система Znanium;
<https://stepik.org/catalog> – Российская образовательная платформа и конструктор бесплатных открытых онлайн-курсов и уроков;
<https://habr.com/ru/> – Хабр. Публикации по ИТ тематикам;
<http://www.ixbt.com/> – содержит достоверную и полную информацию об аппаратном обеспечении компьютера.
<https://www.computer-museum.ru/> – Виртуальный музей компьютерной техники;
<http://ru.wikipedia.org> – Википедия, общедоступная многоязычная универсальная интернет-энциклопедия со свободным контентом;
<https://урокцифры.рф/lessons/bezopasnost-v-internete-2018-2019/materials> – Урок Цифры. Безопасность в Интернете 2018-2019;
<https://урокцифры.рф/lessons/seti-i-oblachnye-tehnologii/materials> – Урок Цифры. Сети и облачные технологии.

7.4 Перечень наглядных и других пособий, методических рекомендаций по проведению учебных занятий, а также методических материалов к используемым в образовательном процессе техническим средствам

Методические рекомендации

1 Вайнилович Ю. В. Технологии интернет-программирования. Методические рекомендации к лабораторным работам для студентов специальностей 38.03.02 – «Менеджмент» очной формы обучения. – Могилев, 2026, - 48 с. [Электронный вариант]

2 Вайнилович Ю. В. Технологии интернет-программирования. Методические рекомендации к курсовому проектированию для студентов специальностей 38.03.02 – «Менеджмент» очной формы обучения. – Могилев, 2026, - 48 с. [Электронный вариант]

Перечень программного обеспечения, используемого в образовательном процессе

№	Наименование	Назначение	Лицензия
1	Visual Studio Code	Редактор исходного кода	Бесплатное (MIT)
2	Git	Распределённая система контроля версий	Бесплатное (GPL-2.0)
3	Node.js	Среда выполнения JavaScript	Бесплатное (MIT)
4	npm	Пакетный менеджер для Node.js	Бесплатное
5	PostgreSQL	Реляционная СУБД	Бесплатное (PostgreSQL License)
6	pgAdmin	Графический клиент для PostgreSQL	Бесплатное (PostgreSQL License)
7	Google Chrome	Веб-браузер с инструментами разработчика	Бесплатное
8	Mozilla Firefox	Веб-браузер с инструментами разработчика	Бесплатное (MPL-2.0)
9	Microsoft Edge	Веб-браузер с инструментами разработчика	Бесплатное
10	Figma	Инструмент проектирования UI/UX	Бесплатное (Free Tier)
11	Postman	Платформа для тестирования API	Бесплатное (Free Tier)
12	Thunder Client	Расширение VS Code для тестирования API	Бесплатное
13	Webpack	Сборщик модулей JavaScript	Бесплатное (MIT)
14	Babel	Транспилатор JavaScript	Бесплатное (MIT)
15	ESLint	Статический анализатор кода	Бесплатное (MIT)
16	Prettier	Инструмент форматирования кода	Бесплатное (MIT)
17	json-server	Локальный REST API сервер	Бесплатное (MIT)
18	Live Server	Локальный сервер с автообновлением	Бесплатное (MIT)
19	GitHub Pages	Хостинг статических сайтов	Бесплатное
20	GitHub	Платформа хостинга репозитория	Бесплатное
21	React	Библиотека для создания интерфейсов	Бесплатное (MIT)
22	React Router	Маршрутизация в React	Бесплатное (MIT)
23	Redux Toolkit	Управление состоянием	Бесплатное (MIT)
24	Axios	HTTP-клиент	Бесплатное (MIT)
25	MUI (Material UI)	UI-компоненты для React	Бесплатное (MIT)
26	Express.js	Веб-фреймворк для Node.js	Бесплатное (MIT)
27	Sequelize	ORM для реляционных баз данных	Бесплатное (MIT)
28	Framer Motion	Библиотека анимаций	Бесплатное (MIT)
29	Recharts	Визуализация данных	Бесплатное (MIT)
30	react-hook-form	Работа с формами	Бесплатное (MIT)
31	react-i18next	Интернационализация	Бесплатное (MIT)
32	react-toastify	Уведомления	Бесплатное (MIT)
33	bcrypt	Хеширование паролей	Бесплатное (MIT)
34	jsonwebtoken	JWT-токены	Бесплатное (MIT)
35	multer	Загрузка файлов	Бесплатное (MIT)
36	cors	Middleware для CORS	Бесплатное (MIT)
37	dotenv	Переменные окружения	Бесплатное (BSD-2-Clause)

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Материально-техническое обеспечение дисциплины содержится в паспорте лаборатории а. 517/2, рег. № паспорта лаборатории № ПУЛ - 4 517/2- 25; в паспорте лаборатории а. 518/2, рег. № паспорта лаборатории № ПУЛ - 4 518/2- 25; в паспорте лаборатории а. 502/2, рег. № паспорта лаборатории № ПУЛ - 4 502/2- 25.