

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

УТВЕРЖДАЮ

Проректор по учебной работе
учреждения образования «Белорусский
государственный университет
информатики и радиоэлектроники»

_____ Е.Н. Шнейдеров
28.10.2025

Регистрационный № 2-420/уч.

«ИНТЕРНЕТ-ТЕХНОЛОГИИ И ВЕБ-ПРОГРАММИРОВАНИЕ»

**Учебная программа учреждения образования по учебной дисциплине
для специальности**

6-05-0612-03 Системы управления информацией

2025 г.

Учебная программа учреждения образования составлена на основе образовательного стандарта ОСВО 6-05-0612-03-2023 и учебных планов специальности 6-05-0612-03 Системы управления информацией.

СОСТАВИТЕЛЬ:

Н.В.Хаджинова, старший преподаватель кафедры информационных технологий автоматизированных систем учреждения образования «Белорусский государственный университет информатики и радиоэлектроники».

РЕЦЕНЗЕНТЫ:

Кафедра интеллектуальных информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 7 от 13.10.2025);

А.А.Процкевич, технический директор общества с ограниченной ответственностью «Пи-консалт бай».

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой информационных технологий автоматизированных систем учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 4 от 13.10.2025);

Научно-методическим советом учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 2 от 17.10.2025).

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа рассчитана на 308 учебных часов (9 з.е.)

План учебной дисциплины в дневной форме получения образования:

Код специальности	Название специальности	Курс	Семестр	Аудиторных часов (в соответствии с учебным планом уо)					Академ. часов на курс. работу (проект)	Форма промежуточной аттестации
				Всего	Лекции	Лабораторные занятия	Практические занятия	Семинарские занятия		
6-05-0612-03	Системы управления информацией	3	6	48	16	32	-	-	-	Экзамен
		4	7	88	48	32	8	-	30	Экзамен

План учебной дисциплины в дистанционной форме получения образования:

Код специальности	Название специальности	Курс	Семестр	Всего	Количество работ			Академ. часов на курс. работа (проект)	Форма промежуточной аттестации
					Контрольные работы	Лабораторные занятия	Индивидуальные практические работы		
6-05-0612-03	Системы управления информацией	4	8	108	1	-	1	-	Экзамен
		5	9	200	2	-	2	30	Экзамен

Место учебной дисциплины.

Учебная программа учреждения образования по учебной дисциплине «Интернет-технологии и веб-программирование» разработана для обучения студентов по специальности 6-05-0612-03 «Системы управления информацией» профилизация «Автоматизированные системы обработки информации».

Изучение дисциплины является актуальным, так как веб-технологии являются основой для создания современных информационных систем, порталов, сервисов электронной коммерции и средств коммуникации. Студенты получают необходимые знания и навыки для полноценной разработки и внедрения веб-решений.

Воспитательное значение учебной дисциплины «Интернет-технологии и веб-программирование» заключается в формировании критического, логического и абстрактного мышления, исследовательских навыков, креативности, воспитании трудолюбия и ответственности, формировании

навыков самообразования и саморазвития, развитию коммуникативных навыков, воспитанию информационной культуры и цифровой грамотности.

Изучение данной учебной дисциплины способствует созданию условий для формирования интеллектуально развитой личности обучающегося, которой присущи стремление к профессиональному совершенствованию, активному участию в экономической и социально-культурной жизни страны, гражданская ответственность и патриотизм.

Цель преподавания учебной дисциплины: освоение комплекса теоретических знаний и практических навыков в области проектирования, разработки и развертывания современных веб-приложений с использованием актуальных интернет-технологий, клиентских и серверных языков программирования, систем управления базами данных и инструментов обеспечения качества и безопасности.

Задачи учебной дисциплины:

приобретение знаний о принципах работы сетевых протоколов, клиент-серверной архитектуры, современных фреймворков и методологий разработки;

освоение практических навыков создания адаптивного пользовательского интерфейса, серверной логики, работы с базами данных и API;

изучение инструментов и процессов, обеспечивающих качество, безопасность и эффективное развертывание веб-приложений (контроль версий, тестирование, контейнеризация, CI/CD).

В результате изучения учебной дисциплины формируются следующие компетенции:

универсальные:

быть способным к саморазвитию и совершенствованию в профессиональной деятельности;

проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности;

владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации;

специализированная: разрабатывать программы (скрипты) на современных языках для web-программирования, работать с web-серверами и серверными приложениями.

В результате изучения учебной дисциплины обучающийся должен:

знать:

базовые принципы организации сетей, протоколы HTTP/HTTPS, архитектуру "клиент-сервер";

современные стандарты и технологии клиентской разработки (HTML5, CSS3, JavaScript ES6+);

принципы работы одностраничных приложений (SPA) и основные клиентские фреймворки (React);

основы серверной разработки на Node.js и Python, принципы построения RESTful API
 основы работы с реляционными и нереляционными базами данных;
 основные угрозы безопасности веб-приложений (OWASP Top 10) и методы защиты;

современные инструменты и практики разработки: Git, Docker, CI/CD, менеджеры пакетов;

уметь:

создавать семантическую, доступную и адаптивную верстку;
 разрабатывать интерактивные клиентские приложения на JavaScript и React;
 проектировать и реализовывать серверную логику и REST API;
 работать с базами данных, выполняя основные CRUD-операции;
 использовать систему контроля версий Git для командной работы;
 тестировать, контейнеризировать и развертывать веб-приложения на облачных платформах.

иметь навык:

разработки полнофункционального веб-приложения, включая фронтенд и бэкенд;

использования современных инструментов сборки, развертывания и мониторинга.

Перечень учебных дисциплин, усвоение которых необходимо
 для изучения данной учебной дисциплины

№ п.п.	Название учебной дисциплины	Раздел, темы
1	Информационные системы и технологии	Концепция открытых систем
2	Базы данных	Основные понятия и технологии работы с реляционными базами данных.
3	Объектно-ориентированное программирование	В полном объеме
4	Технологии проектирования программного обеспечения	Вопросы тестирования программного обеспечения
5	Основы информационной безопасности	В полном объеме
6	Системы управления базами данных	СУБД MySQL Server

1. Содержание учебного материала

№ тем	Наименование тем	Содержание тем
Часть 1. Интернет-технологии		
Раздел 1. Основы веб-технологий и разработки		
1	Архитектура Интернет и сетевые протоколы	Модель OSI/TCP-IP. Протоколы HTTP/HTTPS (методы, статусы, заголовки, cookies). DNS-система. Клиент-серверная архитектура. Принципы работы браузера и веб-серверов (Apache, Nginx).
2	Процесс веб-разработки и рабочее окружение	Жизненный цикл веб-разработки (SDLC). Проектирование структуры веб-приложения. Настройка рабочего окружения: IDE (VS Code), браузерные инструменты разработчика. Прототипирование интерфейсов (Figma). Система контроля версий Git (основы работы).
3	Семантическая HTML-разметка и основы доступности	HTML5: семантические теги, формы, мультимедиа. Валидация разметки. Основы доступности (WCAG, ARIA-атрибуты). Микроразметка Schema.org.
4	Современный CSS: макеты и адаптивный дизайн	CSS3: Flexbox, Grid Layout, позиционирование. Адаптивная и отзывчивая верстка (mobile-first, медиа-запросы). Методология БЭМ. CSS-переменные.
5	JavaScript: основы языка и работа с DOM	Синтаксис ES6+. Типы данных, функции, объекты. Работа с DOM: выборка элементов, манипуляция контентом и стилями. Обработка событий.
6	JavaScript: асинхронность и хранение данных	Механизм событийного цикла. AJAX, Fetch API. Форматы JSON/XML. Хранение данных на клиенте (LocalStorage, sessionStorage).
7	Клиентские фреймворки: концепции и обзор	Концепция SPA (Single Page Application). Virtual DOM. Обзор фреймворков: React, Vue, Angular. Progressive Web Apps (PWA).
8	Тестирование фронтенда и инструменты разработчика	Виды тестирования веб-интерфейсов. Инструменты браузера для отладки. Юнит-тесты (Jest). Инструменты для анализа производительности.
Часть 2. Веб-программирование		
Раздел 2. Серверная разработка на Node.js		
9	Платформа Node.js и среда выполнения	Установка Node.js. Модульная система (CommonJS, ES Modules). Глобальные объекты. Менеджер пакетов npm.
10	Фреймворк Express.js: основы	Создание сервера. Маршрутизация (Routing). Обработка параметров URL и строк запроса. Статические файлы.
11	Middleware и шаблонизаторы	Концепция middleware. Встроенные и кастомные middleware. Шаблонизаторы (EJS). Обработка ошибок.
12	Работа с базами данных. ORM Sequelize	Подключение к PostgreSQL/MySQL. Определение моделей. CRUD-операции. Ассоциации между моделями. Миграции.
13	Проектирование и реализация REST API	Принципы REST. Проектирование эндпоинтов. Коды состояния HTTP. Реализация CRUD API. Документирование API.
14	Аутентификация и авторизация	Сессии vs JWT-токены. Регистрация и аутентификация пользователей. Хеширование паролей (bcrypt). Middleware для проверки прав доступа.

№ тем	Наименование тем	Содержание тем
15	Валидация данных и безопасность	Валидация входящих данных. Основы безопасности: инъекции, XSS, CSRF. Защитные меры: санитизация, CORS.
Раздел 3. Клиентская разработка на React		
16	Введение в React и JSX	Концепция компонентного подхода. Синтаксис JSX. Функциональные компоненты. Props.
17	Управление состоянием и жизненный цикл	Хуки useState и useEffect. Локальное состояние компонента. Сайд-эффекты и их обработка.
18	Маршрутизация в React-приложениях	Библиотека React Router. Создание SPA. Динамические маршруты. Навигация и защита маршрутов.
19	Взаимодействие с сервером	Отправка HTTP-запросов (fetch/axios). Обработка состояний загрузки и ошибок. Обновление интерфейса по данным с сервера.
20	Управление состоянием приложения	Проблема проп drilling. Context API. Хук useReducer. Комплексное управление состоянием приложения.
Раздел 4. Альтернативные технологии		
21	Серверная разработка на Python и FastAPI	Сравнение с Node.js. Создание API на FastAPI. Автоматическая документация OpenAPI. Валидация данных.
22	NoSQL базы данных: MongoDB	Документная модель данных. MongoDB Atlas. ODM Mongoose. Схемы и модели. CRUD-операции.
23	Веб-сокеты и реальное время	Ограничения HTTP. Протокол WebSocket. Библиотека Socket.IO. Создание чата и уведомлений в реальном времени.
Раздел 5. Инструменты сборки и тестирование		
24	Инструменты сборки фронтенда	Модульные системы. Сборщики Webpack и Vite. Настройка конфигурации. Создание production-сборки.
25	Тестирование серверных приложений	Юнит-тесты (Jest). Интеграционные тесты (Supertest). E2E-тесты (Playwright). Покрытие кода тестами.
Раздел 6. Деплой и DevOps		
26	Контейнеризация с Docker	Создание Dockerfile. Образы и контейнеры. Docker Hub. Оптимизация образов.
27	Оркестрация контейнеров	Docker Compose. Запуск многоконтейнерных приложений. Настройка сетей и томов.
28	Непрерывная интеграция и развертывание	Принципы CI/CD. Настройка пайплайна в GitHub Actions. Автоматическое тестирование и деплой.
29	Деплой на облачные платформы	Бесплатные облачные решения: Vercel, Heroku, GitHub Pages. Настройка доменов и SSL.
Раздел 7. Производительность и мониторинг		
30	Оптимизация производительности	Метрики Core Web Vitals. Анализ производительности. Кэширование. Оптимизация изображений и ресурсов.
31	Мониторинг и аналитика	Инструменты мониторинга (Google Analytics, бесплатные аналоги). Логирование ошибок.
32	Архитектурные паттерны	Монолитная vs микросервисная архитектура. Паттерны проектирования. Рефакторинг кода.

2. Информационно-методический раздел

2.1 Литература

2.1.1 Основная

- 2.1.1.1. Новиков, В. А. Web-программирование : учебное пособие / В. А. Новиков. – Минск : Адукацыя і выхаванне, 2024. – 351 с.
- 2.1.1.2. Тузовский, А. Ф. Проектирование и разработка Web-приложений : учебное пособие / А. Ф. Тузовский. – Москва : Юрайт, 2021. – 218 с.
- 2.1.1.3. Рошин, С. М. Современные интернет-технологии : семь главных трендов / С. М. Рошин. – 3-е изд. – Москва : Дашков и К, 2023. – 124 с.
- 2.1.1.4. Дронов, В. А. HTML и CSS : 25 уроков для начинающих / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2023. – 400 с.
- 2.1.1.5. Робсон, Э. Изучаем HTML, XHTML и CSS / Э. Робсон, Э. Фримен. – 2-е изд. – Санкт-Петербург : Питер, 2021. – 720 с.
- 2.1.1.6. Фрэйн, Б. Отзывчивый дизайн на HTML5 и CSS3 для любых устройств / Б. Фрэйн. – 3-е изд. – Санкт-Петербург : Питер, 2022. – 336 с.
- 2.1.1.7. Сидельников, Г. Наглядный CSS / Г. Сидельников. – Санкт-Петербург : Питер, 2022. – 224 с.
- 2.1.1.8. Грант, К. CSS для профи / К. Грант. – Санкт-Петербург : Питер, 2021. – 496 с.
- 2.1.1.9. Хавербеке, М. Выразительный JavaScript : современное веб-программирование / М. Хавербеке. – 3-е изд. – Санкт-Петербург : Питер, 2022.
- 2.1.1.10. Фрисби, М. JavaScript для профессиональных веб-разработчиков / М. Фрисби. – 4-е изд. – Санкт-Петербург : Питер, 2022. – 1168 с.
- 2.1.1.11. Фримен, Э. Изучаем программирование на JavaScript / Э. Фримен, Э. Робсон. – Санкт-Петербург : Питер, 2021. – 640 с.
- 2.1.1.12. Дронов, В. А. Java Script. 20 уроков для начинающих / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2022. – 352 с.
- 2.1.1.13. Чиннатхамби, К. Java Script с нуля / К. Чиннатхамби. – Санкт-Петербург : Питер, 2022. – 400 с.
- 2.1.1.14. Дронов, В. А. React 17. Разработка веб-приложений на JavaScript / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2022. – 384 с.
- 2.1.1.15. Гриффитс, Д. React. Сборник рецептов / Д. Гриффитс. – Санкт-Петербург : БХВ-Петербург, 2023. – 528 с.
- 2.1.1.16. Ганди, Р. Head First. Git / Р. Ганди. – Санкт-Петербург : БХВ-Петербург, 2024. – 464 с.
- 2.1.1.17. Стаяно, Ф. Figma : проектирование и прототипирование интерфейсов в соответствии с принципами UX/UI / Ф. Стаяно. – 2-е изд. – Москва : ДМК Пресс, 2024. – 370 с.
- 2.1.1.18. Коста, Л. Тестирование JavaScript / Л. Коста. – Санкт-Петербург : Питер, 2023. – 592 с.
- 2.1.1.19. Фленов, М. Е. Web-сервер глазами хакера / М. Е. Фленов. – 3-е изд. – Санкт-Петербург : БХВ-Петербург, 2021. – 256 с.

2.1.2 Дополнительная

2.1.2.1. Хорстман, К. JavaScript для начинающих / К. Хорстман. – Санкт-Петербург : Питер, 2021. – 560 с.

2.1.2.2. Флэнаган, Д. JavaScript. Карманный справочник / Д. Флэнаган. – 3-е изд. – Санкт-Петербург : Питер, 2022. – 320 с.

2.1.2.3. Ларсон, Э. Node.js. Разработка серверных веб-приложений на JavaScript / Э. Ларсон. – Санкт-Петербург : Диалектика, 2022. – 448 с.

2.1.2.4. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – 5-е изд. – Санкт-Петербург : Питер, 2022. – 815 с.

2.1.2.5. Роббинс, Дж. Веб-дизайн для начинающих. HTML, CSS, JavaScript и веб-графика / Дж. Роббинс. – 5-е изд. – Санкт-Петербург : БХВ-Петербург, 2021. – 956 с.

2.1.2.6. Кириченко, А. В. Web на практике. CSS, HTML, JavaScript, MySQL, PHP для fullstack-разработчиков / А. В. Кириченко, А. П. Никольский, Е. В. Дубовик. – Санкт-Петербург : Наука и Техника, 2021. – 432 с.

2.1.2.7. Дронов, В. А. JavaScript. Дополнительные уроки для начинающих / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2022. – 352 с.

2.1.2.8. Янг, А. Node.js в действии / А. Янг, Б. Мек, М. Кантелон. – 2-е изд. – Санкт-Петербург : Питер, 2018. – 432 с.

2.1.2.9. Прохорёнок, Н. А. JavaScript и Node.js для веб-разработчиков / Н. А. Прохорёнок, В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2022. – 768 с.

2.1.2.10. Бэнкс, А. React : современные шаблоны для разработки приложений / А. Бэнкс, Е. Порселло. – 2-е изд. – Санкт-Петербург : Питер, 2022. – 320 с. 1

2.1.2.11. Стефанов, С. React. Быстрый старт / С. Стефанов. – 2-е изд. – Санкт-Петербург : Питер, 2023. – 304 с.

2.1.2.12. Бэнкс, А. GraphQL : язык запросов для современных веб-приложений / А. Бэнкс. – Санкт-Петербург : Питер, 2021. – 240 с.

2.1.2.13. Тепляков, А. Паттерны проектирования на JavaScript / А. Тепляков. – Санкт-Петербург : Питер, 2021. – 256 с.

2.1.2.14. Молочков, В. П. Создание сайта, его SEO-продвижение и монетизация. Самоучитель / В. П. Молочков. – Санкт-Петербург : БХВ-Петербург, 2022. – 512 с.

2.2 Перечень компьютерных программ, наглядных и других пособий, методических указаний и материалов, технических средств обучения, оборудования для выполнения лабораторных работ

2.2.1. Аппаратное обеспечение: ЭВМ (Процессор: Intel Core i3 или аналогичный, Оперативная память (RAM): 4 Гб или более) с установленными операционной системой Microsoft® Windows® 10 и выше и браузером: Google Chrome, Mozilla Firefox

2.2.2. Программное обеспечение:

Текстовый редактор: Visual Studio Code;

Node.js (версия LTS);
 Git;
 Docker Desktop;
 Браузер Google Chrome (последняя версия);
 PostgreSQL;
 MongoDB Community Edition;
 Редактор прототипов Figma (бесплатная версия);

2.2.3. Методические указания и наглядный материал:

Комплект описания лабораторных и практических работ в электронной форме;

Примеры кода и шаблоны проектов;

Видео-инструкции по настройке рабочего окружения.

2.3. Перечень тем практических занятий, их название

Целью практических занятий является закрепление теоретического курса, приобретение навыков решения задач, активизация самостоятельной работы студентов.

№ темы по п.1	Название практического занятия	Содержание	Обеспеченность по пункту 2.2
11	Middleware и шаблонизаторы	Разработка кастомного middleware для обработки запросов. Создание шаблонов страниц с использованием EJS. Реализация механизма обработки ошибок.	2.2.1, 2.2.3
13	Проектирование и реализация REST API	Проектирование REST API для системы управления данными. Реализация эндпоинтов для операций CRUD. Документирование API.	2.2.1, 2.2.2
18	Маршрутизация в React-приложениях	Настройка React Router для SPA. Создание защищенных маршрутов. Реализация навигации между страницами.	2.2.1, 2.2.2
20	Управление состоянием приложения	Реализация глобального состояния с использованием Context API и useReducer. Оптимизация перерисовок компонентов.	2.2.1, 2.2.2

2.4. Перечень тем лабораторных занятий, их название

Основная цель проведения лабораторных занятий состоит в закреплении теоретического материала курса, приобретении навыков выполнения эксперимента, обработки экспериментальных данных, анализа результатов, грамотного оформления отчетов.

№ темы по п.1	Наименование лабораторной работы	Содержание	Обеспеченность по пункту 2.2
1	Анализ сетевых протоколов и инструменты разработчика	Исследование HTTP/HTTPS запросов через браузерные инструменты. Анализ заголовков и кодов состояния. Работа с сетевыми панелями разработчика.	2.2.1, 2.2.3
2	Настройка рабочего окружения для веб-разработки	Установка и настройка Node.js, Git, VS Code. Создание репозитория проекта. Настройка окружения для разработки.	2.2.1, 2.2.2
3	Создание семантической HTML-разметки	Разработка валидной HTML-разметки для заданного макета с учетом доступности. Применение семантических тегов.	2.2.1, 2.2.2
4	Создание адаптивного макета с использованием CSS	Применение Flexbox и Grid Layout для создания адаптивного дизайна. Реализация mobile-first подхода.	2.2.1, 2.2.2
5	Разработка интерактивного интерфейса на JavaScript	Создание интерактивных элементов на JavaScript. Работа с DOM и обработка событий.	2.2.1, 2.2.2
6	Работа с API браузера и асинхронными операциями	Использование localStorage, Fetch API. Обработка асинхронных операций. Работа с JSON.	2.2.1, 2.2.2
7	Создание компонентного интерфейса с использованием React	Разработка React-компонентов. Работа с JSX, props и состоянием компонентов.	2.2.1, 2.2.2
8	Тестирование фронтенд-приложения	Настройка Jest для unit-тестирования. Создание тестов для JavaScript кода. Использование инструментов разработчика.	2.2.1, 2.2.2
9	Разработка серверного приложения на Node.js	Создание сервера на Express.js. Настройка маршрутизации. Обработка HTTP-запросов.	2.2.1, 2.2.2
12	Работа с базами данных и ORM Sequelize	Подключение и настройка PostgreSQL. Создание моделей с использованием Sequelize. Реализация CRUD-операций.	2.2.1, 2.2.2
14	Реализация системы аутентификации и авторизации	Создание системы регистрации и аутентификации пользователей. Реализация JWT-токенов. Хеширование паролей.	2.2.1, 2.2.2
17	Управление состоянием в React-приложении	Реализация локального состояния компонентов с использованием useState. Обработка side-эффектов с useEffect.	2.2.1, 2.2.2
19	Взаимодействие React-приложения с сервером	Настройка HTTP-запросов к REST API. Обработка состояний загрузки и ошибок. Обновление интерфейса.	2.2.1, 2.2.2

№ темы по п.1	Наименование лабораторной работы	Содержание	Обеспеченность по пункту 2.2
22	Интеграция с NoSQL базой данных MongoDB	Настройка подключения к MongoDB. Работа с документами через Mongoose. Реализация CRUD-операций.	2.2.1, 2.2.2
23	Реализация реального времени с WebSocket	Создание чата в реальном времени с использованием Socket.IO. Обработка событий подключения и обмена сообщениями.	2.2.1, 2.2.2
26	Контейнеризация приложения с Docker	Создание Dockerfile для backend и frontend. Сборка и запуск контейнеров. Настройка Docker Compose.	2.2.1, 2.2.2

2.5 Курсовая работа, её характеристика

Курсовая работа по учебной дисциплине «Интернет-технологии и веб-программирование» представляет собой самостоятельную разработку полнофункционального веб-приложения, включающего клиентскую часть (React), серверную часть (Node.js/Express), базу данных (PostgreSQL/MongoDB), систему аутентификации и не менее 3 основных функциональных модулей.

Цель работы: закрепление полученных знаний и навыков в области клиентской и серверной разработки, проектирования веб-приложений и их развертывания.

Рекомендуемый объем пояснительной записки 25–30 страниц. Количество часов на выполнение курсовой работы – 30, зачетных единиц – 1.

Перечень тем курсовых работ

1. Разработка системы управления задачами (TODO-приложение) с функциями проектов и тегами.
2. Создание блогговой платформы с системой комментариев и рейтингов.
3. Разработка интернет-магазина с корзиной покупок и системой заказов.
4. Создание системы бронирования ресурсов (переговорных комнат, оборудования).
5. Разработка платформы для онлайн-курсов с системой прогресса и тестирования.
6. Создание системы учета личных финансов с аналитикой и отчетами.
7. Разработка социальной сети с лентой новостей и системой сообщений.
8. Создание сервиса для организации мероприятий с регистрацией участников.
9. Разработка системы управления контентом (CMS) для новостного сайта.
10. Создание платформы для совместной работы над документами.

2.6 Перечень рекомендуемых средств диагностики результатов учебной деятельности

Для диагностики результатов учебной деятельности могут использоваться следующие формы:

- тестирование;
- отчет по практическому занятию (ПЗ);
- отчет по лабораторному занятию (ЛЗ);
- отчет по лабораторной работе (ЛР);
- защита контрольной работы;
- защита индивидуальной практической работы (ИПР).

2.7 Контрольная работа

№ темы по п.1	Наименование контрольной работы	Содержание	Обеспеченность по пункту 2.2
1-8	Контрольная работа №1 "Основы веб-технологий"	Разработка статического веб-сайта с использованием HTML, CSS, JavaScript. Создание адаптивного дизайна и интерактивных элементов.	2.2.1, 2.2.2
9-15	Контрольная работа №2 "Серверная разработка"	Разработка REST API на Node.js с аутентификацией и работой с базой данных. Реализация CRUD-операций.	2.2.1, 2.2.2
16-32	Контрольная работа №3 "Интеграция и оптимизация"	Интеграция клиентской и серверной частей приложения. Оптимизация производительности и настройка деплоя.	2.2.1, 2.2.2

2.8 Индивидуальная практическая работа

№ темы по п.1	Наименование индивидуальной практической работы	Содержание	Обеспеченность по пункту 2.2
1-8	ИПР №1 "Прототипирование и тестирование интерфейса"	Создание прототипа веб-интерфейса в Figma. Реализация адаптивного дизайна и написание unit-тестов для JavaScript кода.	2.2.1, 2.2.2
9-15	ИПР №2 "Разработка безопасного REST API"	Создание REST API с системой аутентификации, валидацией данных и защитой от уязвимостей. Реализация ролевой модели доступа.	2.2.1, 2.2.2
16-32	ИПР №3 "Деплой и мониторинг приложения"	Настройка CI/CD пайплайна. Деплой приложения на облачную платформу. Настройка мониторинга и логирования.	2.2.1, 2.2.2

2.9 Характеристика (описание) инновационных подходов к преподаванию учебной дисциплины

В процессе обучения применяются современные образовательные технологии:

интерактивные методы: дискуссии на основе реальных проектов;

смешанное обучение: теория в СЭО (lms.bsuir.by), практика - в аудитории;

индивидуальный подход: дифференцированные задания и персональные траектории;

проектная работа: групповые проекты в GitHub с использованием agile-методологий;

современный инструментарий: VS Code, системы контроля версий, облачные сервисы;

автоматизированный контроль: тестирование в СЭО, рубричное оценивание работ.

Данные подходы формируют практические навыки, соответствующие требованиям современной веб-разработки, и развивают компетенции для успешной работы в IT-индустрии.

3.1 Учебно-методическая карта учебной дисциплины в дневной форме получения образования

Номер раздела,	Название раздела, темы	Количество аудиторных часов				Самостоятельная работа, часы	Форма контроля знаний
		ЛК	Лаб. зан	ПЗ	СЗ		
6-й семестр							
Часть 1. Интернет-технологии							
Раздел 1. Основы веб-технологий и разработки		16	32			60	
1	Архитектура Интернет и сетевые протоколы	2	4	-	-	4	Отчет по ЛР
2	Процесс веб-разработки и рабочее окружение	2	4	-	-	6	Отчет по ЛР
3	Семантическая HTML-разметка и основы доступности	2	4	-	-	8	Отчет по ЛР
4	Современный CSS: макеты и адаптивный дизайн	2	4	-	-	8	Отчет по ЛР
5	JavaScript: основы языка и работа с DOM	2	4	-	-	8	Отчет по ЛР
6	JavaScript: асинхронность и хранение данных	2	4	-	-	8	Отчет по ЛР
7	Клиентские фреймворки: концепции и обзор	2	4	-	-	8	Отчет по ЛР
8	Тестирование фронтенда и инструменты разработчика	2	4	-	-	10	Отчет по ЛР
Промежуточная аттестация							экзамен
Итого в 6-м семестре		16	32	-	-	60	
7-й семестр							
Часть 2. Веб-программирование							
Раздел 2. Серверная разработка на Node.js		12	12	4		38	
9	Платформа Node.js и среда выполнения	2	4	-	-	6	Отчет по ЛР
10	Фреймворк Express.js: основы	2	-	-	-	6	Тестирование
11	Middleware и шаблонизаторы	2	-	2	-	6	Отчет по ПЗ
12	Работа с базами данных. ORM Sequelize	2	4	-	-	6	Отчет по ЛР
13	Проектирование и реализация REST API	2	-	2	-	6	Отчет по ПЗ
14	Аутентификация и авторизация	2	4	-	-	8	Отчет по ЛР
Раздел 3. Клиентская разработка на React		12	8	4		32	
15	Валидация данных и безопасность	2	-	-	-	4	Тестирование
16	Введение в React и JSX	2	-	-	-	6	Тестирование
17	Управление состоянием и жизненный цикл	2	4	-	-	6	Отчет по ЛР
18	Маршрутизация в React-приложениях	2	-	2	-	6	Отчет по ПЗ
19	Взаимодействие с сервером	2	4	-	-	6	Отчет по ЛР
20	Управление состоянием приложения	2	-	2	-	4	Отчет по ПЗ
Раздел 4. Альтернативные технологии		6	8			10	
21	Серверная разработка на Python и FastAPI	2	-	-	-	4	Тестирование

Номер раздела.	Название раздела, темы	Количество аудиторных часов				Самостоятельная работа, часы	Форма контроля знаний
		ЛК	Лаб. зан	ПЗ	СЗ		
22	NoSQL базы данных: MongoDB	2	4	-	-	2	Отчет по ЛР
23	Веб-сокеты и реальное время	2	4	-	-	4	Отчет по ЛР
Раздел 5. Инструменты сборки и тестирование		4	-			6	
24	Инструменты сборки фронтенда	2	-	-	-	2	Тестирование
25	Тестирование серверных приложений	2	-	-	-	4	Тестирование
Раздел 6. Деплой и DevOps		8	4			14	
26	Контейнеризация с Docker	2	4	-	-	2	Отчет по ЛР
27	Оркестрация контейнеров	2	-	-	-	4	Тестирование
28	Непрерывная интеграция и развертывание	2	-	-	-	4	Тестирование
29	Деплой на облачные платформы	2	-	-	-	4	Тестирование
Раздел 7. Производительность и мониторинг		6				12	
30	Оптимизация производительности	2	-	-	-	4	Тестирование
31	Мониторинг и аналитика	2	-	-	-	4	Тестирование
32	Архитектурные паттерны	2	-	-	-	4	Тестирование
	Промежуточная аттестация						Экзамен, курсовая работа
	Итого в 7-м семестре	48	32	8		112	
	Итого по учебной дисциплине	64	64	8		172	

3.2 Учебно-методическая карта учебной дисциплины в дистанционной форме получения образования

Номер раздела	Название раздела, темы	Количество работ			Самостоятельная работа, часы	Форма контроля знаний
		КР	Лаб. зан.	ИПР		
8-й семестр						
Часть 1. Интернет-технологии						
1	Архитектура Интернет и сетевые протоколы	КР №1		ИПР №1	13	Защита контрольной работы, защита ИПР
2	Процесс веб-разработки и рабочее окружение	КР №1	-	ИПР №1	14	
3	Семантическая HTML-разметка и основы доступности	КР №1	-	ИПР №1	13	
4	Современный CSS: макеты и адаптивный дизайн	КР №1	-	ИПР №1	14	
5	JavaScript: основы языка и работа с DOM	КР №1	-	ИПР №1	13	
6	JavaScript: асинхронность и хранение данных	КР №1	-	ИПР №1	14	
7	Клиентские фреймворки: концепции и обзор	КР №1	-	ИПР №1	13	
8	Тестирование фронтенда и инструменты разработчика	КР №1	-	ИПР №1	14	
	Промежуточная аттестация					экзамен
	Итого в 8-м семестре	1		1	108	
9-й семестр						
Часть 2. Веб-программирование						
9	Платформа Node.js и среда выполнения	КР №2	-	ИПР №2	9	Защита контрольной работы, защита ИПР
10	Фреймворк Express.js: основы	КР №2	-	ИПР №2	8	
11	Middleware и шаблонизаторы	КР №2	-	ИПР №2	9	
12	Работа с базами данных. ORM Sequelize	КР №2	-	ИПР №2	8	
13	Проектирование и реализация REST API	КР №2	-	ИПР №2	9	
14	Аутентификация и авторизация	КР №2	-	ИПР №2	8	
15	Валидация данных и безопасность	КР №2	-	ИПР №2	9	
16	Введение в React и JSX	КР №3	-	ИПР №3	8	
17	Управление состоянием и жизненный цикл	КР №3	-	ИПР №3	9	
18	Маршрутизация в React-приложениях	КР №3	-	ИПР №3	8	
19	Взаимодействие с сервером	КР №3	-	ИПР №3	9	
20	Управление состоянием приложения	КР №3	-	ИПР №3	8	
21	Серверная разработка на Python и FastAPI	КР №3	-	ИПР №3	9	
22	NoSQL базы данных: MongoDB	КР №3	-	ИПР №3	8	

Номер раздела	Название раздела, темы	Количество работ			Самост оятель ная работа, часы	Форма контроля знаний	
		КР	Лаб. зан.	ИПР			
23	Веб-сокеты и реальное время	КР №3	-	ИПР №3	9		
24	Инструменты сборки фронтенда	КР №3	-	ИПР №3	8		
25	Тестирование серверных приложений	КР №3	-	ИПР №3	8		
26	Контейнеризация с Docker	КР №3	-	ИПР №3	8		
27	Оркестрация контейнеров	КР №3	-	ИПР №3	8		
28	Непрерывная интеграция и развертывание	КР №3	-	ИПР №3	8		
29	Деплой на облачные платформы	КР №3	-	ИПР №3	8		
30	Оптимизация производительности	КР №3	-	ИПР №3	8		
31	Мониторинг и аналитика	КР №3	-	ИПР №3	8		
32	Архитектурные паттерны	КР №3	-	ИПР №3	8		
	Промежуточная аттестация						Экзамен, курсовая работа
	Итого в 9-м семестре	2		2	200		
	Всего	3		3	308		

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
С ДРУГИМИ УЧЕБНЫМИ ДИСЦИПЛИНАМИ СПЕЦИАЛЬНОСТИ

Перечень учебных дисциплин	Кафедра, обеспечивающая учебную дисциплину по п.1	Предложения об изменениях в содержании по изучаемой учебной дисциплине	Подпись заведующего кафедрой, обеспечивающей учебную дисциплину по п.1, с указанием номера протокола и даты заседания кафедры
1	2	3	4
Современные системы программирования	ИТАС	нет	<p style="text-align: center;">_____ В.А.Рыбак</p> <p style="text-align: center;">Протокол № 4 от 13.10.2025</p>

Заведующий кафедрой
информационных технологий
автоматизированных систем

В.А.Рыбак