

Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет»

УТВЕРЖДАЮ

Ректор Белорусско-Российского
университета

Документ подписан
простой электронной подписью

Информация о владельце:

ФИО: Лустенков Михаил Евгеньевич

Должность: Ректор

Дата: 09.09.2026

М.Е. Лустенков

22.06 2026

Регистрационный № УД-5.1.0.29.В.18.2/р

РАЗРАБОТКА WEB-ИНТЕРФЕЙСА ПРИЛОЖЕНИЙ

(наименование дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Направление подготовки 09.03.01 – Информатика и вычислительная техника

09.03.04 – Программная инженерия

Направленность (профиль) Программно-аппаратные средства информационных систем

Разработка программно-информационных систем

Квалификация Бакалавр

	Форма обучения
	Очная
Курс	3
Семестр	5, 6
Лекции, часы	16
Лабораторные занятия, часы	50
Экзамен, семестр	5
Контактная работа по учебным занятиям, часы	66
Курсовая работа, семестр	6
Самостоятельная работа, часы	114
Всего часов / зачетных единиц	180/5

Кафедра-разработчик программы: Программное обеспечение информационных технологий

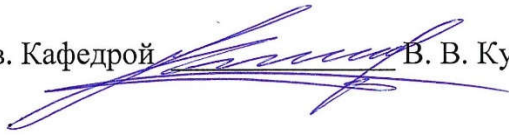
(название кафедры)

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

Рабочая программа составлена в соответствии с федеральным государственным образовательными стандартами высшего образования – бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника № 929 от 19.09.2017, учебным планом рег. № 090301-2.2, от 18.02.2026, в соответствии с федеральными государственными образовательными стандартами высшего образования - бакалавриат по направлению подготовки 09.03.04 Программная инженерия № 920 от 19.09.2017, учебным планом рег. № 090304-2.2 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.)

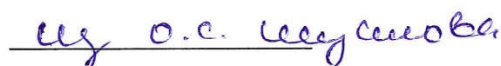
Рецензент:

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

(И.О. Фамилия, должность, ученая степень, ученое звание рецензента)

Рабочая программа согласована:

Ведущий библиотекарь



Начальник учебно-методического
отдела

 О. Е. Печковская

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1. Цель дисциплины

Цель дисциплины – формирование у обучающихся знаний, умений и практических навыков в области проектирования и разработки пользовательских веб-интерфейсов с использованием современных веб-технологий, инструментов и библиотек, а также подготовки к созданию полнофункциональных веб-приложений с использованием актуальных стеков технологий.

1.2. Планируемые результаты изучения дисциплины

В результате освоения учебной дисциплины студент должен:

знать:

- архитектуру клиент-серверных веб-приложений и принципы их функционирования;
- основные стандарты и технологии разработки пользовательских интерфейсов;
- принципы работы с сетевыми запросами и обмена данными между клиентской и серверной частями приложения;
- методы организации безопасности веб-приложений (аутентификация, авторизация);
- принципы работы с реляционными и документо-ориентированными базами данных;
- современные средства автоматизации разработки и контроля версий.

уметь:

- разрабатывать адаптивные и интерактивные веб-страницы с использованием современных технологий верстки;
- создавать клиентскую часть веб-приложений с применением компонентного подхода;
- разрабатывать серверную логику и программные интерфейсы (API) веб-приложений;
- интегрировать клиентскую и серверную части веб-приложения;
- применять системы контроля версий для управления разработкой;
- проводить отладку и тестирование созданного программного обеспечения.

владеть:

- навыками разработки адаптивных и интерактивных пользовательских интерфейсов;
- навыками использования современных инструментов и библиотек веб-разработки;
- навыками компонентной организации кода;
- навыками проведения базового юзабилити-анализа интерфейсов;
- навыками интеграции интерфейсных решений в структуру веб-приложения;
- навыками самостоятельного освоения новых программных средств и технологий.

1.3 Место дисциплины в системе подготовки студента

Дисциплина относится:

- для 09.03.01 к блоку 1 «Дисциплины (модули)» (Обязательная часть блока 1, Элективные дисциплины), дисциплина по выбору;
- для 09.03.04 к блоку 1 «Дисциплины (модули)» (Часть блока 1, формируемая участниками образовательных отношений), дисциплина по выбору.

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

- Информатика;
- Программирование;
- Практика написания программного кода;
- Теория алгоритмов.

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

- Командная разработка программно-аппаратных комплексов (только для 09.03.01);

- Программирование мобильных приложений;
- Технологии промышленного программирования;
- Технологии командной разработки приложений (только для 09.03.04);
- Архитектура программных систем (только для 09.03.04);
- Технологии Интернет-вещей (только для 09.03.01);
- Управление IT-проектами.

Кроме того знания, полученные при изучении дисциплины на лекционных занятиях будут применимы при прохождении первой технологической (проектно-технологической) практики и второй 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
для специальности 09.03.01	
ПК-3	Способен разрабатывать и проектировать программное обеспечение
ОПК-9	Способен осваивать методики использования программных средств для решения практических задач
для специальности 09.03.04	
ПК-8	Способность создавать программные интерфейсы

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 учебной дисциплины

Номер а тем	Наименование тем	Содержание	Коды формируемых компетенций	
			09.03.01	09.03.04
1	2	3	4	5
1	Серверная платформа Node.js и фреймворк Express	Node.js: отличие от браузерного JS, Event Loop, модули (CommonJS, ESM), встроенные модули (fs, 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.	ПК-3 ОПК-9	ПК-8
2	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, proxy/CORS, запросы через Axios.	ПК-3 ОПК-9	ПК-8
3	Безопасность веб-приложений:	Хеширование паролей (bcrypt). JWT: access token, refresh token, подпись, верификация. Middleware	ПК-3 ОПК-9	ПК-8

	аутентификация и авторизация	для проверки токена. Защищённые маршруты (backend). Хранение токена на клиенте (localStorage vs httpOnly cookie). Роли и разграничение прав. Регистрация, логин, логアウト. Защищённые страницы (React: PrivateRoute). Axios: настройка baseURL, интерсепторы для автоматической подстановки токена и обработки 401.		
4	Статическая типизация в веб-разработке	TypeScript: зачем, tsconfig.json. Типы данных. Type alias, interface. Объединение, пересечение. Generics. Enum. Type narrowing. Utility types. TS с React (типизация props, useState, событий).	ПК-3 ОПК-9	ПК-8
5	Серверный рендеринг и статическая генерация веб-приложений	Зачем SSR/SSG: SEO, производительность, UX. Create Next App. Pages Router: файловая маршрутизация, динамические маршруты, getStaticProps, getStaticPaths, getServerSideProps. Link, Image, Head. CSS Modules. Layout. API Routes. App Router: app/, page.js, layout.js, loading.js, error.js, not-found.js. Server Components vs Client Components. Route Handlers, Middleware, ISR, Fetch и кеширование.	ПК-3 ОПК-9	ПК-8
6	Фреймворк Vue.js и его экосистема	Сравнение с React. Создание проекта. SFC. Шаблонный синтаксис: интерполяция, директивы. Компоненты: props, emit. Options API. Composition API. Vue Router: маршруты, router-link, router-view, динамические маршруты, вложенные маршруты, навигационные хуки. Pinia: store, state, getters, actions. Слоты. Provide/inject. Transition. Работа с API через Axios. Библиотеки UI-компонентов.	ПК-3 ОПК-9	ПК-8
7	Архитектурный фреймворк серверной разработки Nest.js	Nest.js: зачем, сравнение с Express. @nestjs/cli. Архитектура: модули (@Module), контроллеры (@Controller), сервисы (@Injectable). Декораторы (@Get, @Post, @Body, @Param). Dependency Injection. DTO, class-validator, ValidationPipe. Pipes, Guards, Interceptors, Exception Filters. TypeORM (Entity, Repository, CRUD, связи). JWT-аутентификация (@nestjs/jwt, @nestjs/passport). Swagger	ПК-3 ОПК-9	ПК-8

2.2 Учебно-методическая карта учебной дисциплины

№ недели	Лекции (наименование тем)	Часы	Лабораторные работы	Часы	Самостоятельная работа, часы	Форма контроля знаний	Баллы (max)
5 семестр							
Модуль 1							
1	Тема 1. Серверная платформа Node.js и фреймворк Express	2	ЛР №1. Платформа Node.js.	2	2	ЗЛР	3
			ЛР №2. Серверный фреймворк Express	2		ЗЛР	3
2			ЛР №3. Клиентское приложение React	2	3		
3		2	ЛР №3. Клиентское приложение React	2	2	ЗЛР	4

	Тема 2. Реляционные базы данных и full-stack интеграция		ЛР №4. ORM Sequelize и база данных PostgreSQL	2		ЗЛР	3
4			ЛР №5. Full-stack приложение на стеке PERN	2	3		
5	Тема 3. Безопасность веб-приложений: аутентификация и авторизация	2	ЛР №5. Full-stack приложение на стеке PERN	2	2	ЗЛР	4
			ЛР №6. JWT-аутентификация	2		ЗЛР	3
6			ЛР №7. Язык программирования TypeScript	2	3	ЗЛР	4
7	Тема 4. Статическая типизация в веб-разработке	2	ЛР №8. Типизация React-приложения	2	2	ЗЛР	3
			ЛР №9. . Фреймворк Next.js и серверный рендеринг	2		ЗЛР	3
8			ЛР №10 Архитектура App Router в Next.js	2	3	ТКУ	30
Модуль 2							
9	Тема 5. Серверный рендеринг и статическая генерация веб-приложений	2	ЛР №10. Архитектура App Router в Next.js	2	2	ЗЛР	3
			ЛР №11. Интеграция Next.js с базой данных	2			
10			ЛР №11 Интеграция Next.js с базой данных	2	3	ЗЛР	3
11	Тема 6. Фреймворк Vue.js и его экосистема	2	ЛР №12. Фреймворк Vue.js	2	2	ЗЛР	3
			ЛР №12. Фреймворк Vue.js	2			
12			ЛР №13. Реактивность и Composition API	2	2	ЗЛР	3
13	Тема 6. Фреймворк Vue.js и его экосистема	2	ЛР №14. Маршрутизация Vue Router и хранилище Pinia	2	2		
			ЛР №14. Маршрутизация Vue Router и хранилище Pinia	2		ЗЛР	3
14			ЛР №15. Компонентная библиотека Vuetify	2	3	ЗЛР	3
15	Тема 7. Архитектурный фреймворк серверной разработки Nest.js	2	ЛР №16. Модульная архитектура серверного приложения Nest.js	2	2		
			ЛР №16. Модульная архитектура серверного приложения Nest.js	2		ЗЛР	3
16			ЛР №17. Валидация данных и взаимодействие с базой данных в Nest.js	2	2	ЗЛР ТЗ	3 3
17			ЛР №18.Безопасность и документирование API в Nest.js	2	4	ЗЛР ТКУ	3 30
18-20					36	ПА (экза- мен	40
	Итого	16		50	78		100
6 семестр							
1-17	Выполнение курсовой работы				36		

	Итого				36		
--	--------------	--	--	--	-----------	--	--

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	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	
1	С использованием ЭВМ	1-7	1-15	66
	ИТОГО	16	50	66

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	1/18
2	Экзаменационные билеты	1
3	Перечень тем курсовых работ	1
4	Тестовое задание	1

5 МЕТОДИКА И КРИТЕРИИ ОЦЕНКИ КОМПЕТЕНЦИЙ СТУДЕНТОВ

5.1 Формируемые компетенции и индикаторы их достижения

для специальности 09.03.01

<i>ПК-3. Способен разрабатывать и проектировать программное обеспечение</i>
<i>ИПК-3.4. Разрабатывает пользовательские интерфейсы (Web-интерфейсы) для систем управления, используя современные веб-технологии и библиотеки, обеспечивая удобство взаимодействия.</i>
Знает: принципы проектирования пользовательских интерфейсов веб-приложений; современные веб-технологии и библиотеки компонентов; требования к удобству и эргономике пользовательского интерфейса; методы структурирования и организации интерфейсного кода; принципы адаптивного и компонентного проектирования.
Умеет: разрабатывать пользовательские интерфейсы веб-приложений по заданным требованиям; использовать библиотеки компонентов для построения интерфейсов; реализовывать адаптивные и интерактивные элементы управления; обеспечивать удобство и логичность взаимодействия пользователя с системой; интегрировать пользовательский интерфейс с серверной логикой.
Владет: навыками проектирования и реализации Web-интерфейсов; навыками применения современных веб-библиотек и фреймворков; навыками организации структуры интерфейсного проекта; навыками оценки качества пользовательского интерфейса.
<i>ОПК-9. Способен осваивать методики использования программных средств для решения практических задач</i>
<i>ИОПК-9.3 Осваивает и применяет методики юзабилити-тестирования и инструменты прототипирования для создания эргономичных интерфейсов</i>
Знает: основные принципы юзабилити и эргономики пользовательских интерфейсов; методы тестирования пользовательского взаимодействия; инструменты прототипирования интерфейсов; критерии оценки удобства использования веб-приложений.
Умеет: применять инструменты прототипирования для создания интерфейсов; проводить анализ удобства пользовательского взаимодействия; выявлять и устранять проблемы интерфейса; адаптировать интерфейс с учетом требований пользователей.
Владет: навыками прототипирования пользовательских интерфейсов; навыками анализа и оценки юзабилити; навыками использования современных программных средств для создания эргономичных интерфейсов; навыками совершенствования пользовательских интерфейсов на основе анализа.

<i>ПК-8. Способность создавать программные интерфейсы</i>
<i>ИПК-8.2. Разрабатывает программные интерфейсы Web-приложений с использованием библиотек компонентов</i>
Знает: принципы создания программных интерфейсов веб-приложений; современные библиотеки компонентов и их назначение; архитектурные подходы к построению пользовательских интерфейсов; требования к функциональности и удобству интерфейсов.
Умеет: разрабатывать программные интерфейсы Web-приложений с использованием библиотек компонентов; структурировать пользовательский интерфейс на основе компонентного подхода; реализовывать интерактивные элементы управления; интегрировать интерфейсные компоненты в общую архитектуру приложения.
Владеет: навыками разработки и проектирования Web-интерфейсов; навыками использования библиотек компонентов; навыками организации структуры интерфейсного проекта; навыками создания удобных и функциональных пользовательских интерфейсов.

5.2 Критерии оценки лабораторных работ

1) вопросы к лабораторным работам

Балл	Критерии
3-4	Студент демонстрирует полное понимание темы, грамотно использует терминологию, даёт исчерпывающие ответы на вопросы, способен обосновать принятые решения и связать их с теоретическими положениями
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

Баллы	Критерии
3	Выполнено 85 % заданий предложенного теста
2	Выполнено 75 % заданий предложенного теста
1	Выполнено 51 % заданий предложенного теста
0	Выполнено менее 51 % заданий предложенного теста

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,

Баллы	Критерии оценивания
	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.
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 студентов (СРС) направлена на закрепление и углубление освоения учебного материала, развитие практических умений. СРС включает следующие виды самостоятельной работы студентов:

- самостоятельное изучение материала по учебникам и другим источникам;
- решение задач и упражнений по образцу.

Перечень контрольных вопросов и заданий для самостоятельной работы студентов 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	Тузовский, А. Ф. Проектирование и разработка web-приложений : учебник для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 219 с.	УМО ВО	https://urait.ru/bcode/561176

7.2 Дополнительная литература

№ п/п	Библиографическое описание	Гриф	Количество экземпляров / URL
1.	Веб-разработка : учебник для вузов / под общей редакцией О. В. Ратановой, Н. А. Ребус, А. Ю. Анисимова. — Москва : Издательство Юрайт, 2026. — 217 с.	УМО ВО	https://urait.ru/bcode/590626

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;
<https://nodejsdev.ru/> – русскоязычный справочник по Node.js: основы Event Loop, модули (CommonJS / ESM), встроенные модули fs, path, http.
https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Statements/async_function – MDN, теория и примеры async/await и промисов в JavaScript.
<https://www.expressjs.com.cn/ru/guide/using-middleware.html> – официальная документация Express: маршрутизация, middleware, обработка ошибок, cors, Morgan.
<https://ru.hexlet.io/qna/javascript/questions/v-chem-otlichiya-i-preimushchestva-ispolzovaniya-mjs-i-js-v-node-js> – разбор различий между CommonJS и ESM (require vs import/export, *.js vs *.mjs).
<https://postgrespro.ru/docs/postgresql/16/tutorial> – официальный учебный курс по PostgreSQL: типы данных, SQL, связи, индексы.
<https://node-postgres.com/> – документация библиотеки pg для Node.js: подключение к PostgreSQL, Pool, параметризованные запросы.
<https://sequelize.org/> – официальная документация Sequelize: модели, миграции, валидация, связи, include.
<https://axios-http.com/ru/> – документация Axios: примеры запросов из React-клиента, настройка baseURL, интерсепторы, обработка 401.
https://developer.mozilla.org/ru/docs/Web/Security/Information_security_concepts – MDN, базовые концепции безопасности веб-приложений: аутентификация, авторизация, сессии, токены.
<https://www.npmjs.com/package/bcrypt> – документация bcrypt для Node.js: хеширование паролей, примеры bcrypt.hash / bcrypt.compare.
<https://jwt.io/introduction> – введение в JWT: структура access/refresh-токенов, подпись, верификация, примеры декодирования.

<https://developer.mozilla.org/ru/docs/Web/HTTP/Cookies> – MDN, описание httpOnly-кук, безопасность хранения токенов на клиенте.

<https://docs.mongodb.com/ru/manual/> – документация MongoDB: документная модель, BSON, коллекции, CRUD-операции.

<https://www.mongodb.com/ru/docs/compass/> – официальная поддержка MongoDB Compass: визуальная работа с коллекциями и документами.

<https://mongoosejs.com/> – документация Mongoose: схемы, модели, типы данных, валидация, ref/populate, pre/post-хуки.

<https://www.npmjs.com/package/multer> – официальная документация multer: загрузка файлов (single, array) в Express.

<https://www.typescriptlang.org/docs/> – официальная документация TypeScript: базовые типы, tsconfig.json, interface, type, generics, enum.

<https://www.typescriptlang.org/docs/handbook/react.html> – TypeScript с React: типизация props, useState, обработчиков событий.

<https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/typeof> – MDN, типы данных и type narrowing в JavaScript/TypeScript.

<https://nextjs.org/docs/pages/api-reference> – официальная документация Next.js: getStaticProps, getStaticPaths, getServerSideProps, Pages Router, Link, Image.

<https://nextjs.org/docs/app> – документация App Router: app/, layout.js, page.js, loading.js, error.js, not-found.js, кэширование, fetch.

<https://developer.mozilla.org/ru/docs/Web/SEO> – MDN Web Docs, базовые принципы SEO: теги title, meta, семантическая разметка.

<https://vuejs.org/guide/quick-start.html> – официальное руководство по Vue 3: SFC, компоненты, props, emit, Options API / Composition API.

<https://router.vuejs.org/> – документация Vue Router: маршруты, router-link, router-view, динамические и вложенные маршруты, навигационные хуки.

<https://pinia.vuejs.org/> – официальная документация Pinia: store, state, getters, actions, работа с API через Axios.

https://purpleschool.ru/knowledge-base/vue/state_management/pinia – русскоязычное практическое руководство по Pinia с примерами и разбором типичных паттернов.

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 Разработка web-интерфейса приложений : метод. рек. к лаб. работам для студентов / сост. Ю. В. Вайнилович . - Могилев : Беларус.-Рос. ун-т, 2026. - 23с.

2 Вайнилович Ю.В., Методические рекомендации к курсовому проектированию по дисциплине «Разработка web-интерфейса приложений» для студентов направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» дневной формы обучения, Могилев, 2022 г., 44 стр (электронный вариант).

Перечень программного обеспечения, используемого в учебном процессе (по видам занятий)

№	Наименование	Назначение	Лицензия	Лекции	Лабораторные
1	Visual Studio Code	Редактор исходного кода	Бесплатное (MIT)	1–8	1–14
2	Git	Система контроля версий	Бесплатное (GPL-2.0)	1–8	1–14
3	Google Chrome	Веб-браузер	Бесплатное	1–8	1–14
4	Node.js	Среда выполнения JavaScript	Бесплатное (MIT)	1–8	1–14
5	npm	Пакетный менеджер	Бесплатное	1–8	1–14
6	Express.js	Веб-фреймворк для Node.js	Бесплатное (MIT)	1, 4	1–9
7	React	Библиотека для интерфейсов	Бесплатное (MIT)	2, 4	3–9
8	Axios	HTTP-клиент	Бесплатное (MIT)	2, 3, 4	3–9
9	PostgreSQL	Реляционная СУБД	Бесплатное (PostgreSQL License)	2	4–6
10	pgAdmin	Клиент для PostgreSQL	Бесплатное (PostgreSQL License)	2	4
11	Sequelize	ORM для реляционных БД	Бесплатное (MIT)	2	4–6
12	bcrypt	Хеширование паролей	Бесплатное (MIT)	3	6, 8
13	jsonwebtoken	JWT-токены	Бесплатное (MIT)	3	6, 8
14	cors	Middleware для CORS	Бесплатное (MIT)	1	1–9
15	dotenv	Переменные окружения	Бесплатное (BSD-2-Clause)	1	1–14
16	multer	Загрузка файлов	Бесплатное (MIT)	4	3
17	Mongoose	ODM для MongoDB	Бесплатное (MIT)	4	1–4
18	Postman	Тестирование API	Бесплатное (Free Tier)	1, 4	1–9
19	TypeScript	Типизированный JavaScript	Бесплатное (Apache-2.0)	7	10–14
20	Next.js	Фреймворк для React с SSR/SSG	Бесплатное (MIT)	5	5–7
21	Vue.js	Фреймворк для интерфейсов	Бесплатное (MIT)	6	8–11
22	Vue Router	Маршрутизация во Vue	Бесплатное (MIT)	6	10
23	Pinia	Управление состоянием для Vue	Бесплатное (MIT)	6	10
24	Vuetify	UI-компоненты для Vue	Бесплатное (MIT)	6	11
25	GitHub	Платформа репозитория	Бесплатное	1–8	1–14
26	ESLint	Анализатор кода	Бесплатное (MIT)	1	1–14
27	Prettier	Форматирование кода	Бесплатное (MIT)	1	1–14

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.