МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Воронежский государственный технический университет»

УТВЕРЖДАЮ

Декан факультета информационных технологий и компьютерной безопасности

_____ /П. Ю. Гусев/

24 января 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Бэкенд-разработка»

Направление подготовки 09.03.01 Информатика и вычислительная техника

Профиль Технологии интеллектуальных автоматизированных систем

Квалификация выпускника бакалавр

Нормативный период обучения 4 года / 4 года и 11 м.

Форма обучения очная / заочная

Год начала подготовки 2023

Автор программы _____ Д.Е. Пачевский

Заведующий кафедрой Компьютерных интеллектуальных технологий проектирования _____ М.И. Чижов

Руководитель ОПОП _____ М.И. Чижов

Воронеж 2023

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Формирование у студентов теоретических знаний и практических навыков в области бэкенд-разработки, необходимых для проектирования, разработки и поддержки серверной части программных систем.

1.2. Задачи освоения дисциплины

1 Изучение основных принципов и технологий бэкенд-разработки.

2 Освоение методов проектирования и реализации серверной части приложений.

3 Развитие навыков работы с базами данных, API, сетевыми протоколами и системами управления версиями.

4 Формирование умений по обеспечению безопасности и оптимизации производительности серверных приложений.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Бэкенд-разработка» относится к дисциплинам части, формируемой участниками образовательных отношений блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Бэкенд-разработка» направлен на формирование следующих компетенций:

ПК-5 - Способен обеспечивать производственный процесс предприятия программным обеспечением в соответствии с предъявляемыми требованиями

ПК-6 - Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности.

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-5	Знать: основные принципы и подходы к разработке программного обеспечения для производственных процессов; технологии и инструменты, используемые для создания серверных приложений; стандарты и требования к качеству и безопасности программного обеспечения.
	Уметь: анализировать требования к программному обеспечению для производственных процессов; проектировать и реализовывать серверные приложения в соответствии с требованиями; проводить тестирование и отладку разработанного программного обеспечения.
	Владеть: навыками работы с инструментами и технологиями для разработки серверных приложений; методами обеспечения качества и безопасности программного обеспечения; опытом разработки и внедрения программного обеспечения для производственных процессов.

ПК-6	Знать: принципы и методы концептуального, функционального и логического проектирования систем; подходы к проектированию архитектуры бэкенд-систем; паттерны проектирования и их применение в разработке систем среднего и крупного масштаба.
	Уметь: проводить анализ требований и определять концепции проектирования систем; разрабатывать функциональные и логические модели систем; выбирать оптимальные подходы и инструменты для проектирования систем.
	Владеть: навыками проектирования архитектуры бэкенд-систем; опытом применения паттернов проектирования в разработке систем; инструментами для моделирования и визуализации проектных решений.

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Бэкенд-разработка» составляет 5 з.е.
Распределение трудоемкости дисциплины по видам занятий

очная форма обучения

Виды учебной работы	Всего часов	Семестры
		6
Аудиторные занятия (всего)	64	64
В том числе:		
Лекции	32	32
Лабораторные работы (ЛР)	32	32
Самостоятельная работа	89	89
Курсовая работа	+	+
Часы на контроль	27	27
Виды промежуточной аттестации - экзамен	+	+
Общая трудоемкость: академические часы зач.ед.	 180 5	 180 5

заочная форма обучения

Виды учебной работы	Всего часов	Семестры
		7

Аудиторные занятия (всего)	20	20
В том числе:		
Лекции	8	8
Лабораторные работы (ЛР)	12	12
Самостоятельная работа	151	151
Курсовая работа	+	+
Часы на контроль	9	9
Виды промежуточной аттестации - экзамен	+	+
Общая трудоемкость: академические часы зач.ед.	 180 5	 180 5

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Введение в бэкенд-разработку.	Основные понятия и задачи бэкенд-разработки. Обзор технологий и инструментов. Жизненный цикл серверного приложения.	4	4	8	16
2	Архитектура серверных приложений	Монолитная и микросервисная архитектура. Паттерны проектирования (MVC, MVP, MVVM). Работа с сетевыми протоколами (HTTP, WebSocket).	4	4	12	20
3	Работа с базами данных	Реляционные и нереляционные базы данных. ORM (Object-Relational Mapping). Оптимизация запросов и индексация.	4	4	12	20
4	Разработка RESTful API	Принципы REST. Проектирование API. Документирование API (Swagger, OpenAPI).	4	4	12	20
5	Аутентификация и авторизация	Методы аутентификации (JWT, OAuth2). Ролевая модель доступа. Защита от атак (CSRF, XSS).	4	4	12	20
6	Оптимизация производительности серверных приложений	Кэширование данных. Балансировка нагрузки. Асинхронная обработка запросов.	4	4	12	20
7	Безопасность в бэкенд-разработке	Защита данных (шифрование, хэширование). Защита от SQL-инъекций. Мониторинг и предотвращение атак.	4	4	12	20

8	Развертывание и мониторинг серверных приложений	Контейнеризация (Docker). Оркестрация (Kubernetes). Мониторинг и логирование (Prometheus, Grafana	4	4	9	17
		Итого	**32**	**32**	**89**	**153**

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Введение в бэкенд-разработку.	Основные понятия и задачи бэкенд-разработки. Обзор технологий и инструментов. Жизненный цикл серверного приложения.	1	1	16	18
2	Архитектура серверных приложений	Монолитная и микросервисная архитектура. Паттерны проектирования (MVC, MVP, MVVM). Работа с сетевыми протоколами (HTTP, WebSocket).	1	1	20	22
3	Работа с базами данных	Реляционные и нереляционные базы данных. ORM (Object-Relational Mapping). Оптимизация запросов и индексация.	1	2	20	23
4	Разработка RESTful API	Принципы REST. Проектирование API. Документирование API (Swagger, OpenAPI).	1	2	20	23
5	Аутентификация и авторизация	Методы аутентификации (JWT, OAuth2). Ролевая модель доступа. Защита от атак (CSRF, XSS).	1	2	20	23
6	Оптимизация производительности серверных приложений	Кэширование данных. Балансировка нагрузки. Асинхронная обработка запросов.	1	2	20	23
7	Безопасность в бэкенд-разработке	Защита данных (шифрование, хэширование). Защита от SQL-инъекций. Мониторинг и предотвращение атак.	1	1	20	22
8	Развертывание и мониторинг серверных приложений	Контейнеризация (Docker). Оркестрация (Kubernetes). Мониторинг и логирование (Prometheus, Grafana	1	1	15	17
		Итого	**8**	**12**	**151**	**171**

5.2 Перечень лабораторных работ

1. Создание простого серверного приложения на Python/Java/Node.js.
2. Работа с базой данных (CRUD-операции).
3. Разработка RESTful API.
4. Разработка GraphQL API.
5. Реализация аутентификации и авторизации.
6. Оптимизация запросов к базе данных.
7. Контейнеризация приложения с использованием Docker.
8. Настройка CI/CD для проекта.

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсовой работы в 6 семестре для очной формы обучения, в 7 семестре

для заочной формы обучения.

Примерная тематика курсовой работы:

1. Разработка серверной части для системы управления задачами.
2. Создание API для интернет-магазина.
3. Разработка микросервисной архитектуры для социальной сети.
4. Реализация системы аутентификации с использованием JWT.
5. Оптимизация и масштабирование серверного приложения.

Курсовая работа по бэкенд-разработке направлена на закрепление теоретических знаний и развитие практических навыков студентов в области проектирования, разработки и сопровождения серверной части программных систем. В процессе выполнения курсовой работы студент решает следующие задачи:

1. Анализ предметной области и постановка задачи
2. Проектирование архитектуры системы
3. Разработка серверной части приложения
4. Работа с базами данных
5. Реализация аутентификации и авторизации
6. Обеспечение безопасности
7. Тестирование и отладка
8. Оптимизация производительности
9. Документирование проекта
10. Контейнеризация и развертывание
11. Подготовка отчета и презентации

Курсовая работа включат в себя графическую часть и расчетно-пояснительную записку.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-5	Знать: основные принципы и подходы к разработке программного обеспечения для производственных процессов; технологии и инструменты, используемые для создания серверных приложений; стандарты и требования к качеству и безопасности программного обеспечения.	Студент демонстрирует понимание теоретических основ разработки ПО. Знает этапы жизненного цикла программного обеспечения. Понимает принципы работы с требованиями заказчика.	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь: анализировать требования к программному обеспечению для производственных процессов; проектировать и реализовывать серверные приложения в соответствии с требованиями; проводить	Студент способен разработать техническое задание на основе требований заказчика. Умеет проектировать и	Выполнение работ в срок, предусмотренный в рабочих	Невыполнение работ в срок, предусмотренный в рабочих

	тестирование и отладку разработанного программного обеспечения.	реализовывать функциональные модули ПО. Демонстрирует навыки тестирования и устранения ошибок в коде.	программах	программах
	Владеть: навыками работы с инструментами и технологиями для разработки серверных приложений; методами обеспечения качества и безопасности программного обеспечения; опытом разработки и внедрения программного обеспечения для производственных процессов.	Навыками работы с инструментами разработки ПО (IDE, системы контроля версий). Методами проектирования и документирования программных решений. Навыками взаимодействия с заказчиком для уточнения требований.	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-6	Знать: принципы и методы концептуального, функционального и логического проектирования систем; подходы к проектированию архитектуры бэкенд-систем; паттерны проектирования и их применение в разработке систем среднего и крупного масштаба.	Студент демонстрирует знание этапов проектирования систем. Понимает принципы работы с базами данных и сетевыми технологиями. Знает современные подходы к проектированию (микросервисы, контейнеризация).	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь: проводить анализ требований и определять концепции проектирования систем; разрабатывать функциональные и логические модели систем; выбирать оптимальные подходы и инструменты для проектирования систем.	Студент способен разработать архитектуру системы на основе требований. Умеет проектировать базы данных и API. Демонстрирует навыки обеспечения масштабируемости системы.	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть: навыками проектирования архитектуры бэкенд-систем; опытом применения паттернов проектирования в разработке систем; инструментами для моделирования и визуализации проектных решений.	Студент уверенно использует инструменты проектирования. Умеет оптимизировать базы данных и запросы. Демонстрирует навыки работы с CI/CD для автоматизации процессов.	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 6 семестре для очной формы обучения, 7 семестре для заочной формы обучения по четырехбалльной системе:

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компе-тенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ПК-5	Знать: основные принципы и подходы к разработке программного обеспечения для производственных процессов; технологии и инструменты, используемые для создания серверных приложений; стандарты и требования к качеству и безопасности программного обеспечения.	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов

	Уметь: анализировать требования к программному обеспечению для производственных процессов; проектировать и реализовывать серверные приложения в соответствии с требованиями; проводить тестирование и отладку разработанного программного обеспечения.	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть: навыками работы с инструментами и технологиями для разработки серверных приложений; методами обеспечения качества и безопасности программного обеспечения; опытом разработки и внедрения программного обеспечения для производственных процессов.	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-6	Знать: принципы и методы концептуального, функционального и логического проектирования систем; подходы к проектированию архитектуры бэкенд-систем; паттерны проектирования и их применение в разработке систем среднего и крупного масштаба.	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	Уметь: проводить анализ требований и определять концепции проектирования систем; разрабатывать функциональные и логические модели систем; выбирать оптимальные подходы и инструменты для проектирования систем.	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть: навыками проектирования архитектуры бэкенд-систем; опытом применения паттернов проектирования в разработке систем; инструментами для моделирования и визуализации проектных решений.	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. **Что такое RESTful API?**

a) Архитектурный стиль для создания веб-сервисов

b) Язык программирования

c) Тип базы данных

d) Протокол передачи данных

Ответ: a) Архитектурный стиль для создания веб-сервисов

2. **Какой метод HTTP используется для создания нового ресурса?**
a) GET
b) POST
c) PUT
d) DELETE
Ответ: b) POST
3. **Что такое JWT?**
a) Язык программирования
b) Формат токена для аутентификации
c) Тип базы данных
d) Протокол передачи данных
Ответ: b) Формат токена для аутентификации
4. **Какой язык программирования чаще всего используется для бэкенд-разработки?**
a) HTML
b) Python
c) CSS
d) JavaScript
Ответ: b) Python
5. **Что такое ORM?**
a) Объектно-реляционное отображение
b) Операционная система
c) Язык запросов
d) Протокол передачи данных
Ответ: a) Объектно-реляционное отображение
6. **Какой протокол используется для защищенного соединения?**
a) HTTP
b) FTP
c) HTTPS
d) SMTP
Ответ: c) HTTPS
7. **Что такое Docker?**
a) Язык программирования
b) Система виртуализации
c) База данных
d) Протокол передачи данных
Ответ: b) Система виртуализации
8. **Какой метод HTTP используется для обновления существующего ресурса?**
a) GET
b) POST
c) PUT
d) DELETE
Ответ: c) PUT
9. **Что такое NoSQL?**

a) Язык программирования
b) Тип базы данных
c) Протокол передачи данных
d) Архитектурный стиль
Ответ: b) Тип базы данных

10. **Какой инструмент используется для контроля версий?**
a) Docker
b) Git
c) Jenkins
d) Kubernetes
Ответ: b) Git

7.2.2 Примерный перечень заданий для решения стандартных задач

1. **Какой запрос SQL используется для выборки всех данных из таблицы "users"?**
a) SELECT * FROM users;
b) INSERT INTO users;
c) UPDATE users;
d) DELETE FROM users;
Ответ: a) SELECT * FROM users;
2. **Какой метод HTTP используется для удаления ресурса?**
a) GET
b) POST
c) PUT
d) DELETE
Ответ: d) DELETE
3. **Какой язык используется для написания запросов к базе данных?**
a) Python
b) SQL
c) HTML
d) JavaScript
Ответ: b) SQL
4. **Какой статус HTTP указывает на успешное выполнение запроса?**
a) 200
b) 404
c) 500
d) 302
Ответ: a) 200
5. **Какой инструмент используется для автоматизации сборки и развертывания?**
a) Docker
b) Git
c) Jenkins
d) Kubernetes
Ответ: c) Jenkins

6. **Какой метод HTTP используется для получения данных?**
   a) GET
   b) POST
   c) PUT
   d) DELETE
   Ответ: a) GET
7. **Какой тип базы данных использует документную модель хранения данных?**
   a) MySQL
   b) PostgreSQL
   c) MongoDB
   d) SQLite
   Ответ: c) MongoDB
8. **Какой инструмент используется для оркестрации контейнеров?**
   a) Docker
   b) Git
   c) Jenkins
   d) Kubernetes
   Ответ: d) Kubernetes
9. **Какой статус HTTP указывает на ошибку "Не найдено"?**
   a) 200
   b) 404
   c) 500
   d) 302
   Ответ: b) 404
10. **Какой язык программирования используется для написания серверных скриптов?**
    a) HTML
    b) CSS
    c) Python
    d) JavaScript
    Ответ: c) Python

7.2.3 Примерный перечень заданий для решения прикладных задач

1. **Какой инструмент используется для кэширования данных?**
   a) Redis
   b) MySQL
   c) MongoDB
   d) PostgreSQL
   Ответ: a) Redis
2. **Какой метод HTTP используется для частичного обновления ресурса?**
   a) GET
   b) POST
   c) PATCH
   d) DELETE

Ответ: c) PATCH

3. **Какой инструмент используется для мониторинга производительности приложения?**
   a) Docker
   b) Prometheus
   c) Jenkins
   d) Kubernetes
   Ответ: b) Prometheus
4. **Какой статус HTTP указывает на внутреннюю ошибку сервера?**
   a) 200
   b) 404
   c) 500
   d) 302
   Ответ: c) 500
5. **Какой язык программирования используется для написания микросервисов?**
   a) HTML
   b) Python
   c) CSS
   d) JavaScript
   Ответ: b) Python
6. **Какой инструмент используется для автоматизации тестирования?**
   a) Docker
   b) Selenium
   c) Jenkins
   d) Kubernetes
   Ответ: b) Selenium
7. **Какой метод HTTP используется для перенаправления?**
   a) GET
   b) POST
   c) PUT
   d) DELETE
   Ответ: a) GET
8. **Какой инструмент используется для управления зависимостями в Python?**
   a) npm
   b) pip
   c) Maven
   d) Gradle
   Ответ: b) pip
9. **Какой статус HTTP указывает на успешное создание ресурса?**
   a) 200
   b) 201
   c) 404
   d) 500

Ответ: b) 201

10. **Какой инструмент используется для управления контейнерами?**

a) Docker

b) Git

c) Jenkins

d) Kubernetes

Ответ: a) Docker

7.2.4 Примерный перечень вопросов для подготовки к зачету

Не предусмотрено учебным планом

7.2.5 Примерный перечень заданий для подготовки к экзамену

1. **Опишите** принципы RESTful API.
2. Какие методы HTTP используются в RESTful API и для чего?
3. Как работает аутентификация с использованием JWT?
4. Опишите процесс проектирования базы данных для веб-приложения.
5. Какие преимущества и недостатки у микросервисной архитектуры?
6. Как обеспечить безопасность серверного приложения?
7. Опишите процесс контейнеризации приложения с использованием Docker.
8. Какие инструменты используются для автоматизации CI/CD?
9. Как оптимизировать запросы к базе данных?
10. Опишите процесс тестирования серверного приложения.

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 6 баллов.

2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 6 до 10 баллов

3. Оценка «Хорошо» ставится в случае, если студент набрал от 11 до 15 баллов.

4. Оценка «Отлично» ставится, если студент набрал от 16 до 20 баллов.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Введение в бэкенд-разработку.	ПК-5, ПК-6	Защита лабораторная работ, защита курсовой работы

2	Архитектура серверных приложений	ПК-5, ПК-6	Защита лабораторная работ, защита курсовой работы
3	Работа с базами данных	ПК-5, ПК-6	Защита лабораторная работ, защита курсовой работы
4	Разработка RESTful API	ПК-5, ПК-6	Защита лабораторная работ, защита курсовой работы
5	Аутентификация и авторизация	ПК-5, ПК-6	Защита лабораторная работ, защита курсовой работы
6	Оптимизация производительности серверных приложений	ПК-5, ПК-6	Защита лабораторная работ, защита курсовой работы
7	Безопасность в бэкенд-разработке	ПК-5, ПК-6	Защита лабораторная работ, защита курсовой работы
8	Развертывание и мониторинг серверных приложений	ПК-5, ПК-6	Защита лабораторная работ, защита курсовой работы

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1 Баланов, А. Н. Бэкенд-разработка веб-приложений: архитектура, проектирование и управление проектами : учебное пособие для вузов / А. Н. Баланов. — 2-е изд., стер. — Санкт-Петербург : Лань, 2025. — 312 с. — ISBN 978-5-507-52472-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/451820 (дата обращения: 28.02.2025). — Режим доступа: для авториз. пользователей.

2 Заяц, А. М. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js : учебное пособие для вузов / А. М. Заяц, Н. П. Васильев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2021. — 120 с. — ISBN 978-5-8114-7042-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/154380 (дата обращения: 28.02.2025). — Режим доступа: для авториз. пользователей.

3 Смоленцева, Т. Е. Проектирование и разработка WEB-приложений: Практикум : учебное пособие / Т. Е. Смоленцева. — Москва : РТУ МИРЭА, 2023. — 68 с. — ISBN 978-5-7339-1759-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/368954 (дата обращения: 28.02.2025). — Режим доступа: для авториз. пользователей.

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

Лицензионное ПО:

- Microsoft Word
- PyCharm

Свободное программное обеспечение:

- LibreOffice
- Jupyter Notebook
- Python
- Visual Studio Code

Отечественное ПО:

- СУБД Линтер
- Яндекс Браузер

Ресурсы информационно-телекоммуникационной сети «Интернет»:

- http://www.edu.ru/
- Образовательный портал ВГТУ
- Open Machine Learning Course https://mlcourse.ai
- Платформа для проведения соревнований по Data Science https://www.kaggle.com

Информационные справочные системы:

- http://window.edu.ru
- https://wiki.cchgeu.ru/

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Специализированная лекционная аудитория, оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой.

Учебные лаборатории (г. Воронеж, ул. Плехановская, д. 11):

- 202/2.
- 215/2.

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Бэкенд-разработка» читаются лекции, проводятся лабораторные работы, выполняется курсовая работа.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсовой работы изложена в учебно-методическом пособии. Выполнять этапы курсовой работы должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсовой работы, защитой курсовой работы.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удается разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомится с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоения учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.

Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.