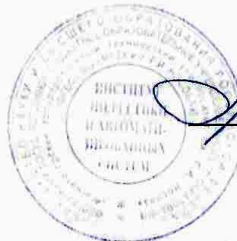




МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И.
Носова»



УТВЕРЖДАЮ
Директор ИЭиАС
В.Р. Храпшин

03.02.2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ BACK-END В WEB РАЗРАБОТКЕ

Направление подготовки (специальность)
09.03.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы
Логика и дизайн пользовательских интерфейсов

Уровень высшего образования - бакалавриат

Форма обучения
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	3
Семестр	6

Магнитогорск
2026 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

Рабочая программа рассмотрена и одобрена на заседании кафедры
Вычислительной техники и программирования
29.01.2026, протокол № 7

Зав. кафедрой



О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭиАС
03.02.2026 г. протокол № 5

Председатель



В.Р. Храмшин

Рабочая программа составлена:
доцент кафедры кафедры ВТиЦ,



Н.А. Дьяконов

Рецензент:

директор НИИ «Промбезопасность», д-р техн. наук



М.Ю. Наркевич

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2029 - 2030 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2030 - 2031 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины (модуля) «Программное обеспечение Back-End в Web разработке» является ознакомление студентов с технологиями, программным обеспечением предназначенным для проектирования, разработки и оптимизации серверной части web-приложений, изучение принципов fullstack разработки web-приложений и применение полученных в процессе освоения дисциплины знаний, умений и навыков на практике.

Для достижения поставленной цели в курсе «Программное обеспечение Back-End в Web разработке» решаются задачи:

- изучение технологий разработки серверной части web-приложений;
- изучение серверных языков программирования и принципов работы с MVC фреймворками;
- изучение технологии fullstack разработки web-приложения на JavaScript;
- изучение подходов в разработке REST API для web-приложений.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Программное обеспечение Back-End в Web разработке входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Компьютерные технологии в дизайне
Программное обеспечение Front-End в Web разработке
Основы разработки Web-приложений
Средства программирования Web-приложений
Программирование

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Шаблонное программирование
Управление контентом для Web-приложений
Выполнение и защита выпускной квалификационной работы

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Программное обеспечение Back-End в Web разработке» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ПК-8	Обладает способностью к настройке и контролю работы сетевых элементов инфокоммуникационной системы, управлению безопасностью сетевых устройств и программного обеспечения, диагностике отказов и ошибок сетевых устройств и программного обеспечения, контролю производительности сетевой инфраструктуры инфокоммуникационной системы, проведению регламентных работ на сетевых устройствах и программном обеспечении инфокоммуникационной системы для обеспечения работы приложений
ПК-8.1	Определяет качество настройки и контроля работы сетевых элементов инфокоммуникационной системы
ПК-8.2	Оценивает качество управления безопасностью сетевых устройств и программного обеспечения, диагностики отказов и ошибок сетевых устройств
ПК-8.3	Определяет необходимость проведения регламентных работ на сетевых устройствах и программном обеспечении

	инфокоммуникационной системы с интерфейсом
ПК-9	Обладает способностью к выполнению мониторинга событий, возникающих в процессе работы инфокоммуникационной системы, и протоколирования событий, возникающих в процессе работы инфокоммуникационной системы для обеспечения работы приложений
ПК-9.1	Оценивает результаты мониторинга событий, возникающих в процессе работы инфокоммуникационной системы
ПК-9.2	Оценивает качество протоколирования событий, возникающих в процессе работы инфокоммуникационной системы для обеспечения работы приложений

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 4 зачетных единиц 144 академических часов, в том числе:

- контактная работа – 87,7 академических часов;
- аудиторная – 84 академических часов;
- внеаудиторная – 3,7 академических часов;
- самостоятельная работа – 20,6 академических часов;
- в форме практической подготовки – 0 академических часов;
- подготовка к экзамену – 35,7 академических часов

Форма аттестации - экзамен

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в академических часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Инструменты для back-end разработки								
1.1 Технологии взаимодействия с Web-сервером. Технология CGI. Технология взаимодействия клиент-сервер AJAX. Формат данных JSON.	6	2	4		4	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2
1.2 Серверные языки программирования: PHP, Java, Node.js (JavaScript), Python, Ruby, Go, C#.		2	6		4	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2

						3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.		
1.3 Фреймворки для back-end разработки: Yii2/Laravel/Symfony, Express.js, Django, Ruby on Rails, ASP.NET MVC 5.	6	2			4	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками.	Устный опрос (собеседование).	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2
1.4 Системы баз данных (реляционные и документоориентированные): MySQL, PostgreSQL, MongoDB.		2			4	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками.	Устный опрос (собеседование).	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2
Итого по разделу		8	10		16			
2. Технология fullstack разработки web-приложения на JavaScript								
2.1 Основы технологии fullstack разработки. Связь back-end и front-end частей в единое web-приложение посредством технологию REST API.	6	2	4			1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2

						дисциплины.		
2.2 Технологии реализации серверной части.		2	4			1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2
2.3 Технологии реализации клиентской части.	6	2	4			1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2
Итого по разделу		6	12					
3. Подходы к разработке REST API. Создание простого REST API на языке PHP								
3.1 Общие понятия. Файловая структура. Настройка базы данных. Создание таблиц. Подключение к БД.	6	2	4		1,6	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2

						электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.		
3.2 Реализация функционала системы при работе с объектами типа «товар»: создание, обновление, получение, удаление, поиск, пагинация.	6	2	4		2	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2
3.3 Тестирование REST API с помощью инструмента Postman		2	4		1	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2

						работ и т.п.), предусмотренны х рабочей программой дисциплины.		
Итого по разделу		6	12		4,6			
4. Реализация RESTful API средствами PHP фреймворка								
4.1 Принципы построения RESTful API. Жизненный цикл приложения. Паттерн MVC. Модель, контроллер, представление.		2	6			1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренны х рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-8.1, ПК-8.2, ПК-8.3, ПК-9.1, ПК-9.2
4.2 Требования, установка и настройка фреймворка Yii2. Роутинг. Работа с БД. Построение сложных запросов к базе данных средствами фреймворка.	6	4	8			1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию. 4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренны х рабочей программой дисциплины.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2
4.3 Регистрация, аутентификация и авторизация. Работа с файловым хранилищем:		2	8			1. Самостоятельное изучение учебной и	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-8.1, ПК-8.3, ПК-9.1, ПК-9.2

загрузка и конвертация файлов. Работа с виджетами.					<p>научной литературы.</p> <p>2. Работа с электронными библиотеками.</p> <p>3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.</p> <p>4. Выполнение практических работ (решение задач, письменных работ и т.п.), предусмотренных рабочей программой дисциплины.</p>		
Итого по разделу	8	22					
Итого за семестр	28	56		20,6	экзамен		
Итого по дисциплине	28	56		20,6	экзамен		

5 Образовательные технологии

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающие прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

Формы учебных занятий с использованием технологий проблемного обучения:

Практическое занятие в форме практикума – организация учебной работы, направленная на решение комплексной учебно-познавательной задачи, требующей от студента применения как научно-теоретических знаний, так и практических навыков.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература:

1. Исаченко, О. В. Программное обеспечение компьютерных сетей : учебное пособие / О.В. Исаченко. — 2-е изд., испр. и доп. — Москва : ИНФРА-М, 2021. — 158 с. — ISBN 978-5-16-015447-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1189344> (дата обращения: 29.10.2020). – Режим доступа: по подписке.

2. Мартишин, С. А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для проектирования информационных систем: учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2016. — 368 с. — (Высшее образование). - ISBN 978-5-8199-0660-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/556449> (дата обращения: 29.10.2020). – Режим доступа: по подписке.

б) Дополнительная литература:

1. Введение в инфокоммуникационные технологии : учебное пособие / Л. Г. Гагарина, А. М. Баин, Г. А. Кузнецов [и др.] ; под ред. Л. Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 336 с. — (Высшее образование). - ISBN 978-5-8199-0768-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1144494> (дата обращения: 29.10.2020). – Режим доступа: по подписке.

2. Варфоломеева, А. О. Информационные системы предприятия : учеб. пособие / А.О. Варфоломеева, А.В. Коряковский, В.П. Романов. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2019. — 330 с. — (Высшее образование: Бакалавриат). — www.dx.doi.org/10.12737/21505. - ISBN 978-5-16-012274-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1002067>(дата обращения: 29.10.2020). –

Режим доступа: по подписке.

в) Методические указания:

Торчинский В. Е. Разработка интерактивных WEB-страниц [Текст] : учебное пособие / В. Е. Торчинский, Л. Л. Демиденко, Ю. А. Демиденко ; МГТУ. - Магнитогорск, 2011. - 95 с.

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
Oracle My SQL Workbench Community Edition	свободно распространяемое ПО	бессрочно
Atom Editor	свободно распространяемое ПО	бессрочно
NotePad++	свободно распространяемое ПО	бессрочно
JetBrains IDEA Community Edition	свободно распространяемое ПО	бессрочно
JetBrains PyCharm Community Edition	свободно распространяемое ПО	бессрочно
MS Visual Studio Code	свободно распространяемое ПО	бессрочно
MariaDB	свободно распространяемое ПО	бессрочно
PostgreSQL	свободно распространяемое ПО	бессрочно
Git	свободно распространяемое ПО	бессрочно
Браузер Yandex	свободно распространяемое ПО	бессрочно
Браузер Mozilla Firefox	свободно распространяемое ПО	бессрочно
SCO OpenServer	свободно распространяемое ПО	бессрочно
MS Visual Studio 2017 Community Edition	свободно распространяемое ПО	бессрочно

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности»	URL: http://www1.fips.ru/

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

1. Лекционная аудитория ауд. 282. Мультимедийные средства хранения, передачи и представления информации.

2. Компьютерные классы Центра информационных технологий ФГБОУ ВО «МГТУ». Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники.

3. Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки. Все классы УИТ и АСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.

4. Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Ауд. 282 и классы УИТ и АСУ.

5. Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации. Классы УИТ и АСУ.

6. Помещения для хранения и профилактического обслуживания учебного оборудования. Центр информационных технологий – ауд. 372.

Учебно-методическое обеспечение самостоятельной работы обучающихся

По дисциплине «Программное обеспечение Back-End в Web разработке» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов предполагает выполнение работ на лабораторных занятиях. Внеаудиторная самостоятельная работа обучающихся осуществляется в виде изучения литературы по соответствующему разделу с проработкой материала при подготовке к сдаче экзамена по данной дисциплине.

Примерные задания к лабораторным занятиям:

Работа №1. Технология Common Gateway Interface

Необходимо реализовать игру “Угадай число” в виде web-приложения. Компьютер загадывает число из заданного диапазона [A; B], а пользователь пытается данное число отгадать. Компьютер при проверке введенных пользователем чисел формирует ответ пользователю в виде следующих сообщений: «загаданное число больше», «загаданное число меньше». Количество попыток для того, чтобы отгадать данное число, ограничено (не более $\log_2(B-A+1)$).

Требования:

1. При разработке интерфейса web-приложения использовать CSS-фреймворки (Bootstrap, Bulma и т.д.).
2. Логика игры должна быть реализована на стороне сервера в виде CGI-приложения (языки реализации: C/C++/C#/Java).

Работа №2. Серверные языки PHP / Python / Ruby

Необходимо реализовать игру “Угадай число” в виде web-приложения. Компьютер загадывает число из заданного диапазона [A; B], а пользователь пытается данное число отгадать. Компьютер при проверке введенных пользователем чисел формирует ответ пользователю в виде следующих сообщений: «загаданное число больше», «загаданное число меньше». Количество попыток для того, чтобы отгадать данное число, ограничено (не более $\log_2(B-A+1)$).

Требования:

1. При разработке интерфейса web-приложения использовать CSS-фреймворки (Bootstrap, Bulma и т.д.).
2. Логика игры должна быть реализована на стороне сервера (языки реализации: PHP, Python, Ruby).

Работа №3. Технология активных серверных страниц (ASP.NET).

Необходимо реализовать web-сервис для регистрации и авторизации пользователей с применением базы данных и защитой от SQL-инъекций.

Работа №4. Технология fullstack разработки web-приложений (Node.js).

Необходимо реализовать web-сервис для формирования поздравительных открыток. Пользователь вводит ФИО, указывает пол поздравляемого и выбирает праздник (например, «Новый год», «23 февраля», «8 марта», «День рождения», «День программиста» и т.д.) для автоматического формирования поздравительной открытки.

Требования:

1. Предусмотреть изменение дизайна открыток в зависимости от пола поздравляемого и праздника.
2. Предусмотреть возможность загрузки с сервера поздравительной открытки в графическом формате.

Работа №5. Создание простого REST API на языке PHP.

Реализовать функционал серверной части web-приложения при работе с объектами типа «товар» и в виде REST API.

Файловая структура REST API: api/

- config/
- core.php (конфигурация: базовой URL и переменные пагинации)
- database.php (подключение к БД)
- objects/
- product.php / (объект типа «товар»)
- category.php / (объект типа «категория товара»)
- product/
- create.php/ (создание объекта типа «товар»)
- delete.php/ (удаление объекта типа «товар»)
- read.php/ (получение информации об объектах типа «товар»)
- read_paging.php/ (получение информации об объектах типа «товар» с пагинацией)
- read_one.php/ (получение информации об одном объекте типа «товар»)
- update.php/ (обновление информации об объекте типа «товар»)
- search.php/ (поиск информации об объектах типа «товар»)
- category/
- read.php () (получение информации об объектах типа «категория товаров»)
- shared/
- utilities.php (получение массива пагинации)

Работа №6. Разработка RESTful API средствами PHP фреймворка

Требуется написать RESTful API для автоматизации деятельности ресторана, реализующую сервис по добавлению описания блюд. Сервис должен стать своеобразным информационным посредником между рестораном и клиентами – существующими или потенциальными.

Функционал будет разделен на две роли:

- администратор;
- гость.

Функциональные возможности гостя:

- авторизация;
- просмотр описания блюд и комментариев;
- комментирование описания блюд;
- поиск записей по тегу.

Функциональные возможности администратора включают в себя функциональные возможности гостя, а также:

- создание, редактирование и удаление описания блюд;
- удаление комментариев к записи описания блюд.

Описание проекта и задач

API авторизации должен быть доступен POST-запросом на адрес {API}/auth, а также принимать следующие обязательные параметры:

- login: логин администратора
- password: пароль администратора

При успешной авторизации должен возвращаться ответ в следующем виде:

- status code: 200
- status text: Successful authorization
- body:
 - status: true
 - token: bearer-токен для использования API администратора

При безуспешной авторизации или ошибке в запросе должен возвращаться ответ в следующем виде:

- status code: 401
- status text: Invalid authorization data
- body:
 - status: false
 - message: Invalid authorization data

API создания блюда должен быть доступен только администратору POST-запросом на адрес {API}/posts, а также принимать следующие обязательные параметры:

- title: название блюда, не пустой, уникальный
- anons: краткое описание блюда, не пустой
- text: полное описание блюда с указанием веса ингредиентов, не пустой
- tags: тэги блюда через запятую, не обязательное поле
- image: фото блюда, не пустой, разрешенные форматы: jpg, png. Максимальный размер: 2 мегабайта

При успешном создании блюда должен возвращаться ответ в следующем виде:

- status code: 201
- status text: Successful creation
- body:
 - status: true
 - post_id: уникальный идентификатор блюда

После создания блюда изображение должно загружаться на сервер в папку {API}/post_images.

При безуспешном добавлении блюда должен возвращаться ответ в следующем виде:

- status code: 400
- status text: Creating error
- body:
 - status: false
 - message: ассоциативный массив из параметров, которые содержат ошибку. В значении должно быть описание ошибки

Пример ответа при безуспешном добавлении блюда:

```
{
  "status": false,
  "message": {
    "title": "already exists",
    "image": "invalid file format"
  }
}
```

API редактирования блюда должен быть доступен только администратору POST-запросом на адрес {API}/posts/<POST_ID>, а также принимать следующие параметры:

- title: название блюда, уникальный
- anons: краткое описание блюда
- text: полное описание блюда с указанием веса ингредиентов
- tags: тэги блюда через запятую
- image: фото блюда, разрешенные форматы: jpg, png. Максимальный размер: 2 мегабайта

При успешном редактировании блюда должен возвращаться ответ в следующем виде:

- status code: 201
- status text: Successful creation
- body:
 - status: true

- post:
 - title: название блюда
 - datetime: дата и время создания блюда в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
 - anons: краткое описание блюда
 - text: полное описание блюда с указанием веса ингредиентов
 - tags: массив из тэгов блюда ([“tag1”, “tag2”])
 - image: ссылка на фото блюда

При безуспешном добавлении блюда должен возвращаться ответ в следующем виде:

- status code: 400
- status text: Editing error
- body:
 - status: false
 - message: ассоциативный массив из параметров, которые содержат ошибку.
В значении должно быть описание ошибки

При попытке редактирования несуществующего блюда должен возвращаться ответ в следующем виде:

- status code: 404
- status text: Post not found
- body:
 - message: Post not found

API удаления блюда должен быть доступен только администратору DELETE-запросом на адрес {API}/posts/<POST_ID>. При успешном удалении блюда должен возвращаться ответ в следующем виде:

- status code: 201
- status text: Successful delete
- body:
 - status: true

При попытке удаления несуществующего блюда должен возвращаться ответ в следующем виде:

- status code: 404
- status text: Post not found
- body:
 - message: Post not found

API просмотра записей блюда должен быть доступен GET-запросом на адрес {API}/posts. Ответ должен содержать массив объектов блюда, содержащие следующие параметры:

- title: название блюда
- datetime: дата и время создания блюда в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
- anons: краткое описание блюда
- text: полное описание блюда с указанием веса ингредиентов
- tags: массив из тэгов блюда ([“tag1”, “tag2”])
- image: ссылка на фото блюда

Ответ должен иметь status code “200” и status text “List posts”.

API просмотра одной записи блюда должен быть доступен GET-запросом на адрес {API}/posts/<POST_ID>.

Ответ должен содержать следующие параметры:

- title: название блюда
- datetime: дата и время создания блюда в формате чч:мм дд.мм.гггг (12:35 06.08.2018)

- anon: краткое описание блюда
- text: полное описание блюда с указанием веса ингредиентов
- tags: массив из тэгов блюда (["tag1", "tag2"])
- image: ссылка на фото блюда
- comments: массив из объектов комментариев содержащие следующие параметры:
 - comment_id: уникальный идентификатор комментария
 - datetime: дата и время в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
 - author: имя автора, если автором комментария является администратор, то параметр содержит текст "admin"
 - comment: текст комментария

Ответ должен иметь status code "200" и status text "View post".

При попытке просмотра несуществующего блюда должен возвращаться ответ в следующем виде:

- status code: 404
- status text: Post not found
- body:
 - message: Post not found

API добавления комментария к блюду должен быть доступен POST-запросом на адрес {API}/posts/<POST_ID>/comments и содержать следующие обязательные параметры:

- author: имя комментатора, обязательное только для гостя
- comment: обязательное, максимум 255 символов

При успешном добавлении комментария должен возвращаться ответ в следующем виде:

- status code: 201
- status text: Successful creation
- body:
 - status: true

При безуспешном добавлении комментария должен возвращаться ответ в следующем виде:

- status code: 400
- status text: Creating error
- body:
 - status: false
 - message: ассоциативный массив из параметров, которые содержат ошибку. В значении должно быть описание ошибки

При попытке добавить комментарий к несуществующему блюду должен возвращаться ответ в следующем виде:

- status code: 404
- status text: Post not found
- body:
 - message: Post not found

API удаления комментария к блюду должен быть доступен только администратору DELETE-запросом на адрес {API}/posts/<POST_ID>/comments/<COMMENT_ID>. При успешном удалении комментария должен возвращаться ответ в следующем виде:

- status code: 201
- status text: Successful delete
- body:
 - status: true

При попытке удаления комментария к несуществующему блюду должен возвращаться ответ в следующем виде:

- status code: 404
- status text: Post not found
- body:
 - message: Post not found

При попытке удаления несуществующего комментария должен возвращаться ответ в следующем виде:

- status code: 404
- status text: Comment not found
- body:
 - message: Comment not found

API поиска блюд по тегу должен быть доступен GET-запросом на адрес {API}/posts/tag/<TAG_NAME>.

В ответ должен возвращаться ответ в следующем виде:

- status code: 200
- status text: Found posts
- body: массив из объектов блюд, содержащие данный тэг:
 - title: название блюда
 - datetime: дата и время создания блюда в формате чч:мм дд.мм.гггг (12:35 06.08.2018)
 - anons: краткое описание блюда
 - text: полное описание блюда с указанием веса ингредиентов
 - tags: массив из тэгов блюда (["tag1", "tag2"])
 - image: ссылка на фото блюда

Авторизация пользователя при использовании административных функциональных возможностей будет проверяться через bearer-token авторизацию. Т.е. к каждому такому запросу в headers будет дописываться bearer-token: Authorization: Bearer xxxxxxxx.

При безуспешной попытке пройти авторизацию по токenu на любой из запросов, который

должен быть доступен только администратору, должен возвращаться ответ в следующем виде:

- status code: 401
- status text: Unauthorized
- body:
 - message: Unauthorized

Под {API} подразумевается URI-адрес. Логин администратора admin, пароль - admin. Все ответы должны приходить в JSON формате.

Во время выполнения задания для тестирования RESTful API использовать Postman.

Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
<p>ПК-8: Обладает способностью к настройке и контролю работы сетевых элементов инфокоммуникационной системы, управлению безопасностью сетевых устройств и программного обеспечения, диагностике отказов и ошибок сетевых устройств и программного обеспечения, контролю производительности сетевой инфраструктуры инфокоммуникационной системы, проведению регламентных работ на сетевых устройствах и программном обеспечении инфокоммуникационной системы для обеспечения работы Web-приложений</p>		
ПК-8.1	<p>Определяет качество настройки и контроля работы сетевых элементов инфокоммуникационной системы</p>	<p><i>Перечень теоретических вопросов</i></p> <ol style="list-style-type: none"> 1. Технологии взаимодействия с Web-сервером. 2. Технология Common Gateway Interface. 3. Технология взаимодействия клиент-сервер AJAX. Формат данных JSON. 4. Серверный язык программирования PHP (Yii2/Laravel/Symfony). 5. Серверный язык программирования Node.js (JavaScript). 6. Серверный язык программирования Python (Django). 7. Серверный язык программирования Ruby (Ruby on Rails). 8. Серверный язык программирования C# (ASP.NET MVC 5). 9. Применение реляционных и документоориентированных СУБД в разработке web-приложений. 10. Технология fullstack разработки. Инструменты реализации. 11. Технологии реализации серверной части при fullstack разработке web-приложений. 12. Технологии реализации клиентской части при fullstack разработке web-приложений. <p><i>Практические задания</i></p> <p>Необходимо реализовать front-end web-приложения игры “Угадай число”</p> <p>Компьютер загадывает число из заданного диапазона [A; B], а пользователь пытается данное число отгадать. Компьютер при проверке введенных пользователем чисел формирует ответ пользователю в виде следующих сообщений: «загаданное число больше», «загаданное число меньше».</p>
ПК-8.2	<p>Оценивает качество управления безопасностью сетевых устройств и программного обеспечения, диагностики отказов и ошибок сетевых устройств</p>	
ПК-8.3	<p>Определяет необходимость проведения регламентных работ на сетевых устройствах и программном обеспечении инфокоммуникационной системы с Web</p>	

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>Количество попыток для того, чтобы отгадать данное число, ограничено (не более $\log_2(B-A+1)$).</p> <p><i>Задания на решение задач из профессиональной области, комплексные задания</i></p> <p>Необходимо реализовать back-end web-приложения игры “Угадай число” используя технологию CGI (язык реализации CGI-программы: C/C++/C#/Java). Компьютер загадывает число из заданного диапазона [A; B], а пользователь пытается данное число отгадать. Компьютер при проверке введенных пользователем чисел формирует ответ пользователю в виде следующих сообщений: «загаданное число больше», «загаданное число меньше». Количество попыток для того, чтобы отгадать данное число, ограничено (не более $\log_2(B-A+1)$).</p> <p><i>Тестовые задания:</i></p> <ol style="list-style-type: none"> Какой код для пустой web-страницы правильный? <ol style="list-style-type: none"> <code><html> <head> <title> </head> <body> </body> </html></code> <code><html> <head> <title> </title> </head> <body> </body> </html></code> <code><html> <head> <title> <body> </body> </html></code> <code><html> <head> <title> </title> </head> <body> </body></code> <code><html> <head> <title> </title> </head> <body> </body></code> Требуется написать химическую формулу тетрасульфида димышьяка (As₂S₄). Каким образом это можно сделать? <ol style="list-style-type: none"> As²S⁴ As^{pow}2S^{pow}4 As₂S₄ As²S⁴ нет правильного ответа Какие методы можно применять для отправки формы? <ol style="list-style-type: none"> POST TRY PUT HEAD MAILTO
<p>ПК-9: Обладает способностью к выполнению мониторинга событий, возникающих в процессе работы инфокоммуникационной системы, и протоколирования событий, возникающих в процессе работы инфокоммуникационной системы для обеспечения работы Web-приложений</p>		

Код индикатора	Индикатор достижения компетенции	Оценочные средства
ПК-9.1	Оценивает результаты мониторинга событий, возникающих в процессе работы инфокоммуникационной системы	<p><i>Перечень теоретических вопросов</i></p> <ol style="list-style-type: none"> 1. Подходы к разработке REST API. Файловая структура REST API. 2. Описание функционала системы при работе с объектами типа «товар» и пример фрагмента ее реализации в виде REST API.
ПК-9.2	Оценивает качество протоколирования событий, возникающих в процессе работы инфокоммуникационной системы для обеспечения работы Web-приложений	<ol style="list-style-type: none"> 3. Инструменты тестирования REST API. 4. Подходы к разработке RESTful API средствами фреймворка. Паттерн MVC. Модель, контроллер, представление. 5. Последовательность работы приложения Yii2. Роутинг. Работа с БД. Построение сложных запросов к базе данных средствами фреймворка. 6. Последовательность работы приложения Yii2. Работа с формами. 7. Последовательность работы приложения Yii2. Работа с виджетами. 8. Технологии подключения серверной части web-приложений к базам данных. <p><i>Практические задания</i></p> <p>Необходимо реализовать front-end web-приложения игры “Угадай число” используя CSS-фреймворк (Bootstrap, Bulma и т.п.). Компьютер загадывает число из заданного диапазона [A; B], а пользователь пытается данное число отгадать. Компьютер при проверке введенных пользователем чисел формирует ответ пользователю в виде следующих сообщений: «загаданное число больше», «загаданное число меньше». Количество попыток для того, чтобы отгадать данное число, ограничено (не более $\log_2(B-A+1)$).</p> <p><i>Задания на решение задач из профессиональной области, комплексные задания</i></p> <p>Необходимо реализовать back-end web-приложения игры “Угадай число” используя серверный язык программирования (PHP/Python/Ruby). Компьютер загадывает число из заданного диапазона [A; B], а пользователь пытается данное число отгадать. Компьютер при проверке введенных пользователем чисел формирует ответ пользователю в виде следующих сообщений: «загаданное число больше», «загаданное число меньше». Количество попыток для того, чтобы отгадать данное число, ограничено (не более</p>

Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>$\log_2(B-A+1)$.</p> <p><i>Тестовые задания:</i></p> <p>1. Дан программный код. Что будет выведено после запуска скрипта?</p> <pre data-bbox="818 371 1374 595" style="background-color: #f0f0f0; padding: 5px;"> <?php for (\$i = 0; \$i < 5; \$i++) { if (\$i % 2 == 0) continue; echo \$i; } ?> </pre> <p>а) 024 б) 24 в) 13 г) 013</p> <p>2. Какая из функций осуществляет подключение к СУБД MySQL:</p> <p>а) connect() б) db_connect() в) mysql_bd_connect() г) mysql_connect()</p> <p>3. Что делает функция include в php?</p> <p>а) включает и выполняет указанный файл б) записывает данные в файл в) подключает файл как новую страницу</p>

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Программное обеспечение Back-End в Web разработке» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме экзамена.

Экзамен по данной дисциплине проводится в устной форме по экзаменационным билетам, каждый из которых включает 2 теоретических вопроса и одно практическое задание.

Показатели и критерии оценивания экзамена:

– на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.

– на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

– на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.