

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

УТВЕРЖДАЮ
Директор по образовательной
деятельности

_____ С.Т. Князев
« ____ » _____

РАБОЧАЯ ПРОГРАММА МОДУЛЯ

Код модуля	Модуль
1155863	Профессиональный курс. Спецкурс 4 (Node.js. Профессиональная разработка REST API)

Екатеринбург

Перечень сведений о рабочей программе модуля	Учетные данные
Образовательная программа 1. Информатика и вычислительная техника 2. Прикладная информатика 3. Программная инженерия	Код ОП 1. 09.03.01/33.01 2. 09.03.03/33.01 3. 09.03.04/33.01
Направление подготовки 1. Информатика и вычислительная техника; 2. Прикладная информатика; 3. Программная инженерия	Код направления и уровня подготовки 1. 09.03.01; 2. 09.03.03; 3. 09.03.04

Программа модуля составлена авторами:

№ п/п	Фамилия Имя Отчество	Ученая степень, ученое звание	Должность	Подразделение
1	Новиков Максим Юрьевич	к.п.н.	Доцент	Базовая кафедра «Аналитика больших данных и методы видеоанализа»

Согласовано:

Управление образовательных программ

Р.Х. Токарева

1. ОБЩАЯ ХАРАКТЕРИСТИКА МОДУЛЯ Профессиональный курс. Спецкурс 4 (Node.js. Профессиональная разработка REST API)

1.1. Аннотация содержания модуля

Модуль «Профессиональный курс. Спецкурс 4 (Node.js. Профессиональная разработка REST API)» включает разработку, тестирование на Node.js, а также практику работы с базами данных для получения студентами профессиональных навыков.

1.2. Структура и объем модуля

Таблица 1

№ п/п	Перечень дисциплин модуля в последовательности их освоения	Объем дисциплин модуля и всего модуля в зачетных единицах
1	Профессиональный курс. Спецкурс 4 (Node.js. Профессиональная разработка REST API)	3
ИТОГО по модулю:		3

1.3. Последовательность освоения модуля в образовательной программе

Пререквизиты модуля	Не предусмотрены
Постреквизиты и кореквизиты модуля	Не предусмотрены

1.4. Распределение компетенций по дисциплинам модуля, планируемые результаты обучения (индикаторы) по модулю

Таблица 2

Перечень дисциплин модуля	Код и наименование компетенции	Планируемые результаты обучения (индикаторы)
1	2	3
Профессиональный курс. Спецкурс 4	ПК-1 - Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение (модули, компоненты, продукты) и осуществлять разработку технических	З-1 - Характеризовать алгоритмы решения типовых задач, области и способы их применения З-2 - Различать синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования З-3 - Изложить основные принципы построения и виды архитектуры

	<p>документов, адресованных специалисту по информационным технологиям и пользователям</p> <p>(Информатика и вычислительная техника)</p>	<p>программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования</p> <p>У-1 - Различать особенности стандартных алгоритмов для решения задач в соответствующих областях профессиональной деятельности</p> <p>У-2 - Выбирать языки программирования для написания программного кода с учетом технического задания</p> <p>У-3 - Определять оптимальные методы и средства проектирования программного обеспечения и структур данных</p> <p>П-1 - Создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов в соответствии с требованиями технического задания</p> <p>П-2 - Создавать и оптимизировать программный код на языках программирования высокого и низкого уровня с использованием специализированных программных средств</p> <p>П-3 - Иметь практический опыт использования инструмента контроля версий Git (GitHub, GitLab и др.) и комбинированной среды управления проектами Redmine и аналогов</p> <p>П-5 - Разрабатывать архитектуру программного обеспечения</p> <p>П-6 - Осуществлять проектирование структур данных</p>
	<p>ПК-1 - Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение (модули, компоненты, продукты) и осуществлять разработку технических документов,</p>	<p>З-1 - Характеризовать алгоритмы решения типовых задач, области и способы их применения</p> <p>З-2 - Различать синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования</p> <p>З-3 - Изложить основные принципы построения и виды архитектуры программного обеспечения, методы и</p>

	<p>адресованных специалисту по информационным технологиям и пользователям</p> <p>(Прикладная информатика)</p>	<p>средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования</p> <p>У-1 - Различать особенности стандартных алгоритмов для решения задач в соответствующих областях профессиональной деятельности</p> <p>У-2 - Выбирать языки программирования для написания программного кода с учетом технического задания</p> <p>У-3 - Определять оптимальные методы и средства проектирования программного обеспечения и структур данных</p> <p>П-1 - Создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов в соответствии с требованиями технического задания</p> <p>П-2 - Создавать и оптимизировать программный код на языках программирования высокого и низкого уровня с использованием специализированных программных средств</p> <p>П-3 - Иметь практический опыт использования инструмента контроля версий Git (GitHub, GitLab и др.) и комбинированной среды управления проектами Redmine и аналогов</p> <p>П-5 - Разрабатывать архитектуру программного обеспечения</p> <p>П-6 - Осуществлять проектирование структур данных</p>
	<p>ПК-1 - Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение (модули, компоненты, продукты) и осуществлять разработку технических документов, адресованных</p>	<p>З-1 - Характеризовать алгоритмы решения типовых задач, области и способы их применения</p> <p>З-2 - Различать синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования</p> <p>З-3 - Изложить основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного</p>

	<p>специалисту по информационным технологиям и пользователям</p> <p>(Программная инженерия)</p>	<p>обеспечения, методологии разработки программного обеспечения и технологии программирования</p> <p>У-1 - Различать особенности стандартных алгоритмов для решения задач в соответствующих областях профессиональной деятельности</p> <p>У-2 - Выбирать языки программирования для написания программного кода с учетом технического задания</p> <p>У-3 - Определять оптимальные методы и средства проектирования программного обеспечения и структур данных</p> <p>П-1 - Создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов в соответствии с требованиями технического задания</p> <p>П-2 - Создавать и оптимизировать программный код на языках программирования высокого и низкого уровня с использованием специализированных программных средств</p> <p>П-3 - Иметь практический опыт использования инструмента контроля версий Git (GitHub, GitLab и др.) и комбинированной среды управления проектами Redmine и аналогов</p> <p>П-5 - Разрабатывать архитектуру программного обеспечения</p> <p>П-6 - Осуществлять проектирование структур данных</p>
--	--	--

1.5. Форма обучения

Обучение по дисциплинам модуля может осуществляться в очной, очно-заочной и заочной формах.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Профессиональный курс. Спецкурс 4
(Node.js. Профессиональная разработка REST API)

Рабочая программа дисциплины составлена авторами:

№ п/п	Фамилия Имя Отчество	Ученая степень, ученое звание	Должность	Подразделение
1	Новиков Максим Юрьевич	к.п.н.	Доцент	Базовая кафедра «Аналитика больших данных и методы видеоанализа»

1. СОДЕРЖАНИЕ И ОСОБЕННОСТИ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ

Авторы:

- **Новиков Максим Юрьевич, к.п.н., доцент, Базовая кафедра «Аналитика больших данных и методы видеоанализа»**

1.1. Технологии реализации, используемые при изучении дисциплины модуля

- Традиционная (репродуктивная) технология
- Разноуровневое (дифференцированное) обучение
 - Базовый уровень

**Базовый I уровень – сохраняет логику самой науки и позволяет получить упрощенное, но верное и полное представление о предмете дисциплины, требует знание системы понятий, умение решать проблемные ситуации. Освоение данного уровня результатов обучения должно обеспечить формирование запланированных компетенций и позволит обучающемуся на минимальном уровне самостоятельности и ответственности выполнять задания; Продвинутый II уровень – углубляет и обогащает базовый уровень как по содержанию, так и по глубине проработки материала дисциплины. Это происходит за счет включения дополнительной информации. Данный уровень требует умения решать проблемы в рамках курса и смежных курсов посредством самостоятельной постановки цели и выбора программы действий. Освоение данного уровня результатов обучения позволит обучающемуся повысить уровень самостоятельности и ответственности до творческого применения знаний и умений.*

1.2. Содержание дисциплины

Таблица 1.1

Код раздела, темы	Раздел, тема дисциплины*	Содержание
1	Введение в Node.js. CLI	TypeScript и Node.js. Классы в TypeScript. Подготовка окружения. Разработка CLI. Формат данных TSV.
2	Цикл событий. Потoki (Streams)	Цикл событий в Node.js. Микро- и макрозадачи. Порядок асинхронного кода.
3	Архитектура приложения	Проектирование архитектуры основного приложения. Подходы к построению архитектуры.
4	Введение в базы данных. MongoDB	Базы данных. Хранение данных на сервере. Запуск с помощью Docker.
5	Базы данных на практике. Проектирование REST	Описание модели. Валидаторы. Сервисы для доступа к данным.
6	Express.js	Имплементация REST API. Фреймворк Express.js.
7	Middleware. Валидация. Загрузка файлов	Концепция middleware. Валидация.
8	Ограничение доступа. Авторизация	Авторизация и аутентификация. JWT.
9	Бэкенд и фронтенд	DTO. CORS. Адаптер.

1.3. Направление, виды воспитательной деятельности и используемые технологии

Направление воспитательной деятельности	Вид воспитательной деятельности	Технология воспитательной деятельности	Компетенция	Результаты обучения
Профессиональное воспитание	профориентационная деятельность	Технология формирования уверенности и готовности к самостоятельной успешной профессиональной деятельности	ПК-1 - Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение (модули, компоненты, продукты) и осуществлять разработку технических документов, адресованных специалисту по информационным технологиям и пользователям	З-3 - Изложить основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования

1.4. Программа дисциплины реализуется на государственном языке Российской Федерации.

2. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Профессиональный курс. Спецкурс 4 (Node.js. Профессиональная разработка REST API)

Электронные ресурсы (издания)

1. Каскиаро М., Маммино Л. / Шаблоны проектирования Node.js / пер. с англ. А. Н. Киселева. – М.: ДМК Пресс, 2017. – 396 с.: ил.
2. Янг А., Мек Б., Кантелон М. / Node.js в действии. 2-е изд. — СПб.: Питер, 2018. — 432 с.: ил. — (Серия «Для профессионалов»).
3. Крокфорд Дуглас / Как устроен JavaScript. — СПб.: Питер, 2019. — 304 с. — (Серия «Для профессионалов»).
4. Флэнаган Дэвид / JavaScript. Полное руководство, 7-е изд. : Пер. с англ. — СПб. : ООО «Диалектика», 2021. — 720 с
5. Хавербеке Марейн / Выразительный JavaScript. Современное веб-программирование / пер. с англ. Е. Сандицкая. – СПб.: Питер, 2020. – 480 с.: ил
6. Резиг Джон, Бибо Безр, Марас Иосип / Секреты JavaScript ниндзя, 2-е изд. : Пер. с англ. – СПб.: ООО «Альфа-книга», 2017. — 544 с.: ил.

7. Закас Н. / JavaScript для профессиональных веб-разработчиков : Пер. с англ. А. Лютича. — СПб.: Питер, 2015. — 960 с.: ил. — (Серия «Для профессионалов»)
8. Фримен Э., Робсон Э. / Изучаем программирование на JavaScript. — СПб.: Питер, 2015. — 640 с.: ил. — (Серия «Head First O'Reilly»).
9. Херман Д. / Сила JavaScript. 68 способов эффективного использования JS. — СПб.: Питер, 2013. — 288 с.: ил. — (Серия «Библиотека специалиста»).
10. Н. А. Прохоренок, В. А. Дронов. / HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. — 4-е изд., — СПб.: БХВ-Петербург, 2015. — 768 с.: ил. — (Профессиональное программирование)
11. Крокфорд Д. / JavaScript: сильные стороны. — СПб.: Питер, 2012. — 176 с.: ил.

Профессиональные базы данных, информационно-справочные системы

1. Единое окно доступа к образовательным ресурсам. Раздел Информатика и информационные технологии <http://window.edu.ru/catalog>
2. Интернет-Университет Информационных Технологий <http://www.intuit.ru/>
3. Федеральный центр информационно-образовательных ресурсов <http://eor.edu.ru/>

Материалы для лиц с ОВЗ

Весь контент ЭБС представлен в виде файлов специального формата для воспроизведения синтезатором речи, а также в тестовом виде, пригодном для прочтения с использованием экранной лупы и настройкой контрастности.

Базы данных, информационно-справочные и поисковые системы

1. Издательство "Лань" <http://e.lanbook.com/>
2. ЭБС Университетская библиотека онлайн «Директ-Медиа» <http://www.biblioclub.ru/>
3. ООО Научная электронная библиотека <http://elibrary.ru>

3. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Профессиональный курс. Спецкурс 4 (Node.js. Профессиональная разработка REST API)

Сведения об оснащённости дисциплины специализированным и лабораторным оборудованием и программным обеспечением

Таблица 3.1

№ п/п	Виды занятий	Оснащённость специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения
1	Лекции	Мебель аудиторная с количеством рабочих мест в соответствии с количеством студентов Рабочее место преподавателя Доска аудиторная	Microsoft Windows 8.1 Pro 64-bit RUS OLP NL Acdmc Office Professional 2003 Win32 Russian CD-ROM

		Подключение к сети Интернет	
2	Лабораторные занятия	Мебель аудиторная с количеством рабочих мест в соответствии с количеством студентов Рабочее место преподавателя Доска аудиторная Персональные компьютеры по количеству обучающихся Подключение к сети Интернет	Microsoft Windows 8.1 Pro 64-bit RUS OLP NL Acdmc Office Professional 2003 Win32 Russian CD-ROM
3	Самостоятельная работа студентов	Персональные компьютеры по количеству обучающихся Подключение к сети Интернет	Microsoft Windows 8.1 Pro 64-bit RUS OLP NL Acdmc Office Professional 2003 Win32 Russian CD-ROM
4	Текущий контроль и промежуточная аттестация	Мебель аудиторная с количеством рабочих мест в соответствии с количеством студентов Рабочее место преподавателя Доска аудиторная Персональные компьютеры по количеству обучающихся Подключение к сети Интернет	Microsoft Windows 8.1 Pro 64-bit RUS OLP NL Acdmc Office Professional 2003 Win32 Russian CD-ROM

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ПО ДИСЦИПЛИНЕ**

Профессиональный курс. Спецкурс 4
(Node.js. Профессиональная разработка REST API)

Код модуля
1155863(2)

Модуль
Профессиональный курс. Спецкурс 4
(Node.js. Профессиональная разработка REST
API)

Екатеринбург

Оценочные материалы составлены автором(ами):

№ п/п	Фамилия, имя, отчество	Ученая степень, ученое звание	Должность	Подразделение
1	Новиков Максим Юрьевич	к.п.н.	Доцент	Базовая кафедра «Аналитика больших данных и методы видеоанализа»

Согласовано:

Управление образовательных программ

Т.Г. Комарова

Авторы:

- Новиков Максим Юрьевич, к.п.н., доцент, Базовая кафедра «Аналитика больших данных и методы видеоанализа»

1. СТРУКТУРА И ОБЪЕМ ДИСЦИПЛИНЫ Профессиональный курс. Спецкурс 4 (Node.js. Профессиональная разработка REST API)

1.	Объем дисциплины в зачетных единицах	3	
2.	Виды аудиторных занятий	Лекции Лабораторные занятия	
3.	Промежуточная аттестация	Зачет	
4.	Текущая аттестация	Контрольная работа	1
		Домашняя работа	1

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ (ИНДИКАТОРЫ) ПО ДИСЦИПЛИНЕ МОДУЛЯ Профессиональный курс. Спецкурс 4 (Node.js. Профессиональная разработка REST API)

Индикатор – это признак / сигнал/ маркер, который показывает, на каком уровне обучающийся должен освоить результаты обучения и их предъявление должно подтвердить факт освоения предметного содержания данной дисциплины, указанного в табл. 1.3 РПМ-РПД.

Таблица 1

Код и наименование компетенции	Планируемые результаты обучения (индикаторы)	Контрольно-оценочные средства для оценивания достижения результата обучения по дисциплине
1	2	3
ПК-1 -Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение (модули, компоненты, продукты) и осуществлять разработку технических документов, адресованных специалисту по информационным технологиям и	3-1 - Характеризовать алгоритмы решения типовых задач, области и способы их применения 3-2 - Различать синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования 3-3 - Изложить основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного	Домашняя работа Зачет Лабораторные занятия Контрольная работа Лекции

<p>пользователям (Информатика и вычислительная техника)</p>	<p>обеспечения и технологии программирования П-1 - Создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов в соответствии с требованиями технического задания П-2 - Создавать и оптимизировать программный код на языках программирования высокого и низкого уровня с использованием специализированных программных средств П-3 - Иметь практический опыт использования инструмента контроля версий Git (GitHub, GitLab и др.) и комбинированной среды управления проектами Redmine и аналогов П-5 - Разрабатывать архитектуру программного обеспечения П-6 - Осуществлять проектирование структур данных У-1 - Различать особенности стандартных алгоритмов для решения задач в соответствующих областях профессиональной деятельности У-2 - Выбирать языки программирования для написания программного кода с учетом технического задания У-3 - Определять оптимальные методы и средства проектирования программного обеспечения и структур данных</p>	
<p>ПК-1 -Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное</p>	<p>З-1 - Характеризовать алгоритмы решения типовых задач, области и способы их применения З-2 - Различать синтаксис языков программирования, особенности программирования</p>	<p>Домашняя работа Зачет Лабораторные занятия Контрольная работа Лекции</p>

<p>обеспечение (модули, компоненты, продукты) и осуществлять разработку технических документов, адресованных специалисту по информационным технологиям и пользователям (Прикладная информатика)</p>	<p>на этих языках, стандартные библиотеки языков программирования</p> <p>З-3 - Изложить основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования</p> <p>П-1 - Создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов в соответствии с требованиями технического задания</p> <p>П-2 - Создавать и оптимизировать программный код на языках программирования высокого и низкого уровня с использованием специализированных программных средств</p> <p>П-3 - Иметь практический опыт использования инструмента контроля версий Git (GitHub, GitLab и др.) и комбинированной среды управления проектами Redmine и аналогов</p> <p>П-5 - Разрабатывать архитектуру программного обеспечения</p> <p>П-6 - Осуществлять проектирование структур данных</p> <p>У-1 - Различать особенности стандартных алгоритмов для решения задач в соответствующих областях профессиональной деятельности</p> <p>У-2 - Выбирать языки программирования для написания программного кода с учетом технического задания</p>	
---	---	--

	У-3 - Определять оптимальные методы и средства проектирования программного обеспечения и структур данных	
ПК-1 -Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение (модули, компоненты, продукты) и осуществлять разработку технических документов, адресованных специалисту по информационным технологиям и пользователям (Программная инженерия)	<p>З-1 - Характеризовать алгоритмы решения типовых задач, области и способы их применения</p> <p>З-2 - Различать синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования</p> <p>З-3 - Изложить основные принципы построения и виды архитектуры программного обеспечения, методы и средства проектирования программного обеспечения, методологии разработки программного обеспечения и технологии программирования</p> <p>П-1 - Создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов в соответствии с требованиями технического задания</p> <p>П-2 - Создавать и оптимизировать программный код на языках программирования высокого и низкого уровня с использованием специализированных программных средств</p> <p>П-3 - Иметь практический опыт использования инструмента контроля версий Git (GitHub, GitLab и др.) и комбинированной среды управления проектами Redmine и аналогов</p> <p>П-5 - Разрабатывать архитектуру программного обеспечения</p> <p>П-6 - Осуществлять проектирование структур данных</p>	<p>Домашняя работа</p> <p>Зачет</p> <p>Лабораторные занятия</p> <p>Контрольная работа</p> <p>Лекции</p>

	<p>У-1 - Различать особенности стандартных алгоритмов для решения задач в соответствующих областях профессиональной деятельности</p> <p>У-2 - Выбирать языки программирования для написания программного кода с учетом технического задания</p> <p>У-3 - Определять оптимальные методы и средства проектирования программного обеспечения и структур данных</p>	
--	---	--

3. ПРОЦЕДУРЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ В РАМКАХ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ МОДУЛЯ В БАЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ (ТЕХНОЛОГИЧЕСКАЯ КАРТА БРС)

3.1. Процедуры текущей и промежуточной аттестации по дисциплине

1. Лекции: коэффициент значимости совокупных результатов лекционных занятий – 0.6		
Текущая аттестация на лекциях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>контрольная работа</i>	6, 4	60
<i>домашняя работа</i>	6, 13	40
Весовой коэффициент значимости результатов текущей аттестации по лекциям – 0.4		
Промежуточная аттестация по лекциям – зачет		
Весовой коэффициент значимости результатов промежуточной аттестации по лекциям – 0.6		
2. Практические/семинарские занятия: коэффициент значимости совокупных результатов практических/семинарских занятий – не предусмотрено		
Текущая аттестация на практических/семинарских занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по практическим/семинарским занятиям – не предусмотрено		
Промежуточная аттестация по практическим/семинарским занятиям – нет		
Весовой коэффициент значимости результатов промежуточной аттестации по практическим/семинарским занятиям – не предусмотрено		
3. Лабораторные занятия: коэффициент значимости совокупных результатов лабораторных занятий – 0.4		
Текущая аттестация на лабораторных занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах

<i>выполнение и защита лабораторных работ</i>	6,16	100
Весовой коэффициент значимости результатов текущей аттестации по лабораторным занятиям -1		
Промежуточная аттестация по лабораторным занятиям –нет		
Весовой коэффициент значимости результатов промежуточной аттестации по лабораторным занятиям – не предусмотрено		
4. Онлайн-занятия: коэффициент значимости совокупных результатов онлайн-занятий –не предусмотрено		
Текущая аттестация на онлайн-занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент значимости результатов текущей аттестации по онлайн-занятиям -не предусмотрено		
Промежуточная аттестация по онлайн-занятиям –нет		
Весовой коэффициент значимости результатов промежуточной аттестации по онлайн-занятиям – не предусмотрено		

3.2. Процедуры текущей и промежуточной аттестации курсовой работы/проекта

Текущая аттестация выполнения курсовой работы/проекта	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
Весовой коэффициент текущей аттестации выполнения курсовой работы/проекта– не предусмотрено		
Весовой коэффициент промежуточной аттестации выполнения курсовой работы/проекта– защиты – не предусмотрено		

4. КРИТЕРИИ И УРОВНИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ МОДУЛЯ

4.1. В рамках БРС применяются утвержденные на кафедре/институте критерии (признаки) оценивания достижений студентов по дисциплине модуля (табл. 4) в рамках контрольно-оценочных мероприятий на соответствие указанным в табл.1 результатам обучения (индикаторам).

Таблица 4

Критерии оценивания учебных достижений обучающихся

Результаты обучения	Критерии оценивания учебных достижений, обучающихся на соответствие результатам обучения/индикаторам
Знания	Студент демонстрирует знания и понимание в области изучения на уровне указанных индикаторов и необходимые для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Умения	Студент может применять свои знания и понимание в контекстах, представленных в оценочных заданиях, демонстрирует освоение умений на уровне указанных индикаторов и необходимых для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.

Опыт /владение	Студент демонстрирует опыт в области изучения на уровне указанных индикаторов.
Другие результаты	Студент демонстрирует ответственность в освоении результатов обучения на уровне запланированных индикаторов. Студент способен выносить суждения, делать оценки и формулировать выводы в области изучения. Студент может сообщать преподавателю и коллегам своего уровня собственное понимание и умения в области изучения.

4.2 Для оценивания уровня выполнения критериев (уровня достижений обучающихся при проведении контрольно-оценочных мероприятий по дисциплине модуля) используется универсальная шкала (табл. 5).

Таблица 5

Шкала оценивания достижения результатов обучения (индикаторов) по уровням

Характеристика уровней достижения результатов обучения (индикаторов)				
№ п/п	Содержание уровня выполнения критерия оценивания результатов обучения (выполненное оценочное задание)	Шкала оценивания		
		Традиционная характеристика уровня		Качественная характеристика уровня
1.	Результаты обучения (индикаторы) достигнуты в полном объеме, замечаний нет	Отлично (80-100 баллов)	Зачтено	Высокий (В)
2.	Результаты обучения (индикаторы) в целом достигнуты, имеются замечания, которые не требуют обязательного устранения	Хорошо (60-79 баллов)		Средний (С)
3.	Результаты обучения (индикаторы) достигнуты не в полной мере, есть замечания	Удовлетворительно (40-59 баллов)		Пороговый (П)
4.	Освоение результатов обучения не соответствует индикаторам, имеются существенные ошибки и замечания, требуется доработка	Неудовлетворительно (менее 40 баллов)	Не зачтено	Недостаточный (Н)
5.	Результат обучения не достигнут, задание не выполнено	Недостаточно свидетельств для оценивания		Нет результата

5. СОДЕРЖАНИЕ КОНТРОЛЬНО-ОЦЕНОЧНЫХ МЕРОПРИЯТИЙ ПО ДИСЦИПЛИНЕ МОДУЛЯ

5.1. Описание аудиторных контрольно-оценочных мероприятий по дисциплине модуля

5.1.1. Лекции

Самостоятельное изучение теоретического материала по темам/разделам лекций в соответствии с содержанием дисциплины (п. 1.2. РПД)

5.1.2. Лабораторные занятия

Примерный перечень тем

1. Введение в Node.js. CLI
2. Цикл событий. Потoki (Streams)
3. Архитектура приложения
4. Введение в базы данных. MongoDB
5. Базы данных на практике. Проектирование REST
6. Express.js
7. Middleware. Валидация. Загрузка файлов
8. Ограничение доступа. Авторизация
9. Бэкенд и фронтенд

LMS-платформа – не предусмотрена

5.2. Описание внеаудиторных контрольно-оценочных мероприятий и средств текущего контроля по дисциплине модуля

Разноуровневое (дифференцированное) обучение.

Базовый

5.2.1. Контрольная работа

Примерный перечень тем

1. Основы асинхронного программирования в Node.js
2. Создание сервера с использованием Express.js
3. Маршрутизация запросов в Express.js
4. Обработка запросов и ответов в Express.js
5. Middleware в Express.js для REST API
6. Работа с базами данных в Node.js при разработке REST API
7. Использование MongoDB с Node.js для REST API
8. Аутентификация пользователей в REST API
9. Авторизация пользователей в REST API
10. Использование JSON Web Tokens (JWT) для безопасности REST API
11. Хэширование паролей пользователей в Node.js
12. Защита REST API от атак: CSRF, XSS, SQL Injection
13. Тестирование REST API с использованием Mocha и Chai
14. Документирование REST API с помощью Swagger
15. Работа с файлами и изображениями в REST API
16. Оптимизация производительности REST API на Node.js
17. Логирование и мониторинг в Node.js приложениях
18. Работа с WebSocket для реального времени в REST API
19. Работа с кэшем для улучшения производительности REST API
20. Работа с очередями сообщений для обработки задач в REST API

LMS-платформа – не предусмотрена

5.2.2. Домашняя работа

Примерный перечень тем

1. Сайт резюме и вакансий.
2. Доставка домашней выпечки и сладостей.
3. Сайт советов по уходу за животными.
4. Практические идеи и советы.
5. Рецензии и книги.
6. История родного края.
7. Обзоры фильмов.
8. Сайт аренды недвижимости.
9. Новостной сайт.
10. Туристический сайт.
11. Кулинарный сайт.
12. Сайт историй и рассказов.
13. Рецензии на книги.
14. Сайт по изучению языка программирования.
15. Сайт туристической фирмы.
16. Ремонт.
17. Компьютерная техника.
18. Программное обеспечение

Примерные задания

В домашней работе требуется:

- выбрать тему для создания сайта или предложить свою;
- самостоятельно создать многостраничный сайт по выбранной теме;
- создать горизонтальное или вертикальное выпадающее меню;
- создать скрипт, который сворачивает и разворачивает текст статьи;
- выполнить проверку вводимых данных в поля форм на соответствие заданному шаблону;
- создать произвольные 5 скриптов и применить их на своем сайте;
- загрузить проект на GitHub

LMS-платформа – не предусмотрена

5.3. Описание контрольно-оценочных мероприятий промежуточного контроля по дисциплине модуля

5.3.1. Зачет

Список примерных вопросов

1. Основы JavaScript для Node.js
2. Асинхронное программирование в Node.js
3. Модули и пакеты в Node.js
4. Express.js: основы и установка
5. Создание REST API с использованием Express.js
6. Маршрутизация в Express.js
7. Обработка запросов и ответов в Express.js
8. Middleware в Express.js
9. Работа с базами данных в Node.js
10. Использование MongoDB с Node.js
11. Использование Mongoose для работы с MongoDB
12. Аутентификация и авторизация в REST API

13. JSON Web Tokens (JWT) для аутентификации
 14. Хэширование паролей в Node.js
 15. Работа с сессиями в Express.js
 16. Защита REST API от атак
 17. Тестирование REST API в Node.js
 18. Использование Postman для тестирования API
 19. Документирование REST API с помощью Swagger
 20. Работа с файлами и изображениями в REST API
 21. Оптимизация производительности REST API
 22. Логирование и мониторинг в Node.js
 23. Работа с WebSocket в Node.js
 24. Работа с кэшем в Node.js приложениях
 25. Работа с очередями сообщений в Node.js
 26. Развертывание Node.js приложений на сервере
 27. Continuous Integration и Continuous Deployment для Node.js
 28. Масштабирование Node.js приложений
- LMS-платформа – не предусмотрена

5.4 Содержание контрольно-оценочных мероприятий по направлениям воспитательной деятельности

Направление воспитательной деятельности	Вид воспитательной деятельности	Технология воспитательной деятельности	Компетенция	Результаты обучения	Контрольно-оценочные мероприятия
Профессиональное воспитание	профориентационная деятельность	Технология формирования уверенности и готовности к самостоятельной успешной профессиональной деятельности	ПК-1	3-3	Домашняя работа Зачет Лабораторные занятия Контрольная работа Лекции