



Федеральное агентство по рыболовству  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Калининградский государственный технический университет»  
(ФГБОУ ВО «КГТУ»)

УТВЕРЖДАЮ  
И.о. директора института

Фонд оценочных средств  
(приложение к рабочей программе дисциплины)  
**«РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ВЕБ-ПРИЛОЖЕНИЙ (Backend)»**

основной профессиональной образовательной программы бакалавриата  
по направлению подготовки  
**09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА**

ИНСТИТУТ  
РАЗРАБОТЧИК

цифровых технологий  
кафедра прикладной информатики

# 1 РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ, ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ И КРИТЕРИИ ОЦЕНИВАНИЯ

## 1.1 Результаты освоения дисциплины

Таблица 1 – Планируемые результаты обучения по дисциплине, соотнесенные с установленными компетенциями

Код и наименование компетенции	Дисциплина	Результаты обучения (владения, умения и знания), соотнесенные с компетенциями
<p>ПК-1 Способен формулировать требования, проектировать и разрабатывать программное обеспечение на языках высокого уровня</p>	<p>Разработка серверной части Веб-приложений (Frontend)</p>	<p>знать:</p> <ul style="list-style-type: none"> <li>- основные принципы и технологии backend-разработки;</li> <li>- языки программирования Python, Java, PHP и их синтаксис;</li> <li>- язык структурированных запросов SQL и его команды;</li> <li>- базы данных и их структуру;</li> <li>- принципы работы с файловой системой;</li> <li>- принципы работы с сетью и протоколами HTTP/HTTPS;</li> <li>- принципы обеспечения безопасности веб-приложений;</li> <li>- принципы масштабируемости и отказоустойчивости веб-приложений;</li> <li>- принципы работы с контейнерами приложений;</li> <li>- методы оптимизации производительности веб-приложений;</li> <li>- методы автоматизированного тестирования бэкенд-компонентов;</li> <li>- стандарты и методы проектирования серверной части веб-приложений;</li> <li>- основные фреймворки и пакеты прикладных программ для разработки серверной части веб-приложений;</li> <li>- основные технологии искусственного интеллекта для разработки серверной части веб-приложений.</li> </ul> <p>уметь:</p> <ul style="list-style-type: none"> <li>- разрабатывать серверную часть веб-приложений с использованием различных языков программирования;</li> </ul>

		<ul style="list-style-type: none"><li>- работать с базами данных и языком SQL;</li><li>- взаимодействовать с файловой системой и сетью;</li><li>- обеспечивать безопасность веб-приложений;</li><li>- обеспечивать масштабируемость и отказоустойчивость веб-приложений;</li><li>- работать с контейнерами приложений;</li><li>- оптимизировать производительность веб-приложений;</li><li>- проводить автоматизированное тестирование бэкенд-компонентов;</li><li>- применять стандарты и методы проектирования серверной части веб-приложений;</li><li>- использовать основные фреймворки и пакеты прикладных программ для разработки серверной части бэкенд-компонентов;</li><li>- использовать технологии искусственного интеллекта для разработки серверной части веб-приложений.</li></ul> <p>владеть:</p> <ul style="list-style-type: none"><li>- навыками разработки серверной части веб-приложений с использованием различных языков программирования;</li><li>- навыками работы с базами данных и языком SQL;</li><li>- навыками взаимодействия с файловой системой и сетью;</li><li>- навыками обеспечения безопасности веб-приложений;</li><li>- навыками обеспечения масштабируемости и отказоустойчивости веб-приложений;</li><li>- навыками работы с контейнерами приложений;</li><li>- навыками оптимизации производительности веб-приложений;</li><li>- навыками проведения автоматизированного тестирования бэкенд-компонентов;</li><li>- навыками применения стандартов и методов проектирования серверной части веб-приложений;</li></ul>
--	--	---

		<ul style="list-style-type: none"> <li>- навыками использования основных фреймворков и пакетов прикладных программ для разработки серверной части веб-приложений;</li> <li>- технологиями искусственного интеллекта для разработки серверной части веб-приложений.</li> </ul>
--	--	---

1.2 К оценочным средствам текущего контроля успеваемости относятся:

- тестовые задания открытого и закрытого типов,
- типовые задания для контрольных работ (заочная форма обучения).

К оценочным средствам для промежуточной аттестации относятся:

- типовые задания по курсовому проекту;
- экзаменационные задания по дисциплине, представленные в виде тестовых заданий закрытого и открытого типов.

### 1.3 Критерии оценки результатов освоения дисциплины

Универсальная система оценивания результатов обучения включает в себя системы оценок: 1) «отлично», «хорошо», «удовлетворительно», «неудовлетворительно»; 2) «зачтено», «не зачтено»; 3) 100 – балльную/процентную систему и правило перевода оценок в пятибалльную систему (табл. 2).

Таблица 2 – Система оценок и критерии выставления оценки

Система оценок	2	3	4	5
	0-40%	41-60%	61-80 %	81-100 %
Критерий	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
	«не зачтено»	«зачтено»		
<b>1 Системность и полнота знаний в отношении изучаемых объектов</b>	Обладает частичными и разрозненными знаниями, которые не может научно-корректно связывать между собой (только некоторые из которых может связывать между собой)	Обладает минимальным набором знаний, необходимым для системного взгляда на изучаемый объект	Обладает набором знаний, достаточным для системного взгляда на изучаемый объект	Обладает полной знаний и системным взглядом на изучаемый объект
<b>2 Работа с информацией</b>	Не в состоянии находить необходимую информацию, либо в состоянии находить отдельные	Может найти необходимую информацию в рамках поставленной задачи	Может найти, интерпретировать и систематизировать необходимую информацию в рамках	Может найти, систематизировать необходимую информацию, а также выявить новые, дополнительные

Система оценок  Критерий	2	3	4	5
	0-40%	41-60%	61-80 %	81-100 %
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
	«не зачтено»	«зачтено»		
	фрагменты информации в рамках поставленной задачи		поставленной задачи	ные источники информации в рамках поставленной задачи
<b>3 Научное осмысление изучаемого явления, процесса, объекта</b>	Не может делать научно корректных выводов из имеющихся у него сведений, в состоянии проанализировать только некоторые из имеющихся у него сведений	В состоянии осуществлять научно корректный анализ предоставленной информации	В состоянии осуществлять систематический и научно корректный анализ предоставленной информации, вовлекает в исследование новые релевантные задачи данные	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые релевантные поставленной задаче данные, предлагает новые ракурсы поставленной задачи
<b>4 Освоение стандартных алгоритмов решения профессиональных задач</b>	В состоянии решать только фрагменты поставленной задачи в соответствии с заданным алгоритмом, не освоил предложенный алгоритм, допускает ошибки	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом, понимает основы предложенного алгоритма	Не только владеет алгоритмом и понимает его основы, но и предлагает новые решения в рамках поставленной задачи

## 2 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

ПК-1 Способен формулировать требования, проектировать и разрабатывать программное обеспечение на языках высокого уровня

### Тестовые задания открытого типа

1. \_\_\_\_\_ отвечает за обработку данных, взаимодействие с базами данных, аутентификацию пользователей, управление бизнес-логикой и предоставление данных для фронтенда

**Ответ: Серверная часть (backend)**

2. Архитектурный стиль взаимодействия между клиентом и сервером, при котором используются стандартные HTTP методы (GET, POST, PUT, DELETE) для обмена данными называется \_\_\_\_\_

**Ответ: REST API (Representational State Transfer)**

3. Технология, которая позволяет разработчикам работать с базой данных на уровне объектов, что упрощает написание SQL-запросов и уменьшает вероятность ошибок, называется \_\_\_\_\_

**Ответ: ORM (Object-Relational Mapping)**

4. \_\_\_\_\_ проверяет подлинность пользователя (аутентификация) и определяет его права доступа к различным ресурсам (авторизация)

**Ответ: Серверная часть**

5. Промежуточное ПО, которое обрабатывает запросы и ответы, добавляя к ним дополнительную функциональность, например, логирование, обработку ошибок или аутентификацию, называется \_\_\_\_\_

**Ответ: Middleware.**

6. \_\_\_\_\_ определяет в веб-приложениях, какой обработчик должен быть вызван в зависимости от URL и HTTP метода, который использует клиент

**Ответ: Маршрутизация**

7. \_\_\_\_\_ обрабатывает HTTP-запросы от клиентов и передает их в серверную часть, которая выполняет бизнес-логику и возвращает ответ обратно клиенту

**Ответ: Веб-сервер**

8. \_\_\_\_\_ хранит и управляет данными в серверной части, с которыми работает веб-приложение

**Ответ: База данных**

9. Уязвимость, при которой злоумышленник может вставить вредоносный SQL-код в запрос, защита от которой осуществляется за счет использования параметризованных запросов и ORM, называется \_\_\_\_\_

**Ответ: SQL-инъекция**

10. Программирование в серверной части, при котором задачи выполняются последовательно, блокируя поток выполнения, называется \_\_\_\_\_

**Ответ: Синхронным**

11. Программирование в серверной части, при котором задачи могут выполняться параллельно, что позволяет не блокировать основной поток и обрабатывать большее количество запросов, называется \_\_\_\_\_

**Ответ: Асинхронным**

12. \_\_\_\_\_ обеспечивают постоянное двустороннее соединение между клиентом и сервером, что позволяет передавать данные в реальном времени, например, для чатов или обновления информации на странице без перезагрузки

**Ответ: Веб-сокеты**

13. Архитектурный шаблон, разделяющий логику приложения на три компонента: Модель (управление данными), Представление (отображение данных) и Контроллер (обработка запросов), называется \_\_\_\_\_

**Ответ: MVC (Model-View-Controller)**

14. \_\_\_\_\_ позволяют разбить приложение на независимые сервисы, что облегчает их масштабирование, обновление и поддержку

**Ответ: Микросервисная архитектура**

15. Протокол авторизации, который позволяет сторонним приложениям получать доступ к ресурсам пользователя без передачи пароля, называется \_\_\_\_\_

**Ответ: OAuth**

16. Популярный инструмент, который изолирует приложение и его зависимости в контейнере, что упрощает развёртывание и управление средами, называется \_\_\_\_\_

**Ответ: Docker**

17. Механизм, который позволяет контролировать доступ к ресурсам на сервере из другого домена, называется \_\_\_\_\_

**Ответ: CORS (Cross-Origin Resource Sharing)**

18. Протокол \_\_\_\_\_ обеспечивает шифрование данных между клиентом и сервером, защищая их от перехвата и атак типа "человек посередине"

**Ответ: HTTPS**

19. Компактный токен, который используется для передачи данных между клиентом и сервером, часто для аутентификации. Токен содержит зашифрованную информацию и подписывается сервером. Он называется \_\_\_\_\_

**Ответ: JWT (JSON Web Token)**

20. Метод \_\_\_\_\_ распределяет входящие запросы между несколькими серверами, что помогает избежать перегрузки, улучшить доступность и обеспечить отказоустойчивость системы \_\_\_\_\_

**Ответ: Load balancing**

21. Высокопроизводительный RPC-фреймворк от Google, который использует HTTP/2 и протоколы буфера (protobuf) для сериализации данных, обеспечивает более высокую производительность и поддержку стриминга по сравнению с REST, называется \_\_\_\_\_

**Ответ: gRPC**

22. Методы \_\_\_\_\_ позволяют серверу отправлять уведомления о событиях на другие серверы в реальном времени, что полезно для интеграции с внешними системами \_\_\_\_\_

**Ответ: Веб-хуки (webhooks)**

23. \_\_\_\_\_ предоставляют готовую архитектуру и набор инструментов, что ускоряет разработку, улучшает структурированность кода и обеспечивает поддержку общих задач, таких как маршрутизация и аутентификация

**Ответ: Фреймворки.**

### **Тестовые задания закрытого типа**

1. Используется для передачи данных между клиентом и сервером в REST API протокол
  - а. FTP
  - б. SMTP
  - в. HTTP**
  - г. POP3
2. Наиболее популярен для разработки серверной части на платформе Node.js следующий язык программирования
  - а. Python
  - б. JavaScript**
  - в. Ruby
  - г. PHP
3. Метод HTTP, который используется для создания нового ресурса на сервере
  - а. GET
  - б. POST
  - в. PUT**
  - г. DELETE
4. Преимуществом микросервисной архитектуры является
  - а. Упрощение масштабирования отдельных компонентов**
  - б. Облегчение отладки и тестирования
  - в. Уменьшение количества серверов
  - г. Упрощение разработки UI
5. Для разработки серверной части на Python предназначен фреймворк
  - а. React
  - б. Angular
  - в. Django**

г. Laravel

б. Наиболее подходит для обработки большого числа запросов в реальном времени

а. Синхронное программирование

**б. Асинхронное программирование**

в. Использование файловых баз данных

г. Использование статических файлов

7. Для создания нового контейнера в Docker используется команда

а. docker start

**б. docker run**

в. docker build

г. docker commit

## **2 ТИПОВЫЕ ЗАДАНИЯ НА КОНТРОЛЬНУЮ РАБОТУ, КУРСОВУЮ РАБОТУ/ КУРСОВОЙ ПРОЕКТ, РАСЧЕТНО-ГРАФИЧЕСКУЮ РАБОТУ**

- учебным планом предусмотрено выполнение курсового проекта,
- учебным планом предусмотрено выполнение контрольной работы (заочная форма обучения).

### **Типовые задания для курсового проекта**

1. Разработать серверную часть веб-приложения, которая позволяет пользователям регистрироваться, авторизоваться и управлять своими профилями.
2. Создать систему управления контентом (CMS), которая позволяет администраторам добавлять, редактировать и удалять контент на сайте.
3. Разработать систему управления заказами, которая позволяет пользователям размещать заказы, отслеживать их статус и получать уведомления о доставке.
4. Создать систему обработки платежей, которая позволяет пользователям оплачивать товары или услуги через различные платежные системы.
5. Разработать систему управления пользователями, которая позволяет администраторам создавать группы пользователей с различными правами доступа.
6. Создать систему резервного копирования данных, которая автоматически создает резервные копии базы данных и файлов веб-приложения.
7. Разработать систему мониторинга производительности веб-приложения, которая позволяет отслеживать нагрузку на сервер, время ответа и другие показатели.
8. Создать систему уведомлений, которая позволяет отправлять пользователям уведомления о новых событиях или обновлениях через электронную почту или push-уведомления.

9. Разработать систему безопасности, которая защищает веб-приложение от хакерских атак, таких как SQL-инъекции, XSS-атаки и другие.
10. Создать систему аналитики, которая собирает данные о поведении пользователей на сайте и предоставляет информацию о наиболее популярных страницах, источниках трафика и других показателях.
11. Разработать систему управления базой данных, которая позволяет администраторам создавать, изменять и удалять таблицы, поля и записи.
12. Создать систему обработки файлов, которая позволяет пользователям загружать, скачивать и обрабатывать файлы на сервере.
13. Разработать систему интеграции с внешними сервисами, которая позволяет веб-приложению взаимодействовать с другими системами через API.
14. Создать систему кеширования данных, которая ускоряет работу веб-приложения, сохраняя часто используемые данные в памяти сервера.
15. Разработать систему безопасности, которая защищает веб-приложение от хакерских атак, таких как SQL-инъекции, XSS-атаки и другие.
16. Создать систему логирования, которая записывает все важные события, происходящие в системе, для последующего анализа и устранения ошибок.
17. Разработать систему мониторинга производительности, которая позволяет отслеживать нагрузку на сервер, время ответа и другие показатели.
18. Создать систему масштабирования, которая позволяет увеличивать мощность сервера при увеличении нагрузки на веб-приложение.
19. Разработать систему резервного копирования, которая автоматически создает резервные копии базы данных и файлов веб-приложения.
20. Создать систему уведомлений, которая позволяет администраторам получать уведомления о важных событиях в системе через электронную почту или SMS.

### **Типовые задания для контрольных работ (заочная форма обучения)**

1. Введение в RESTful API и его принципы.
2. Использование Python для разработки серверной части.
3. Работа с фреймворком Flask для создания API.
4. Разработка RESTful API с использованием Python и Flask
5. Создание и маршрутизация HTTP-запросов.
6. Обработка ошибок и исключений.
7. Обеспечение безопасности API с помощью аутентификации и авторизации.

8. Тестирование и отладка серверной части API.
9. Развертывание API на сервере.
10. Введение в Node.js и его особенности для серверной разработки.
11. Работаты фреймворком Express для создания веб-сервера.
12. Параллельная обработка запросов и оптимизация производительности.
13. Кэширование и работа с базами данных.
14. Настройка балансировки нагрузки и отказоустойчивости.
15. Мониторинг и управление сервером.
16. Безопасность веб-сервера и защита от атак.
17. Разработка высоконагруженного веб-сервера с использованием Node.js и Express
18. Примеры реальных высоконагруженных веб-серверов.

#### **4 СВЕДЕНИЯ О ФОНДЕ ОЦЕНОЧНЫХ СРЕДСТВ И ЕГО СОГЛАСОВАНИИ**

Фонд оценочных средств для аттестации по дисциплине «Разработка серверной части Веб-приложений (Backend)» представляет собой компонент основной профессиональной образовательной программы бакалавриата по направлению подготовки 09.03.03 Прикладная информатика.

Преподаватель-разработчик – к.т.н., доцент В.А. Петрикин

Фонд оценочных средств рассмотрен и одобрен заведующим кафедрой прикладной информатики.

Заведующий кафедрой



М.В. Соловей

Фонд оценочных средств рассмотрен и одобрен методической комиссией института цифровых технологий (протокол №5 от 29.08.2024 г).

Председатель методической комиссии



О.С. Витренко