

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Минцаев Магомед Шавалович

Должность: Ректор

Дата подписания: 05.06.2026 10:00:17

Уникальный программный ключ:

236bcc35c296f119d6aafdc22836b21db52dbc07971a86865a5825f9fa4304cc

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ ГРОЗНЕНСКИЙ ГОСУДАРСТВЕННЫЙ
НЕФТЯНОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени академика М.Д.
Миллионщикова**



«УТВЕРЖДАЮ»

Первый проректор

И.Г. Гайрабсков

2025 г.

РАБОЧАЯ ПРОГРАММА

дисциплины

«Серверная разработка Web-приложений»

Направление подготовки/специальность

09.03.03 Прикладная информатика

Направленность/специализация

«Разработка мобильных и Web-приложений»

Квалификация

Бакалавр

Год начала подготовки

2025

Грозный - 2025

1. Цели и задачи освоения дисциплины

Целью освоения дисциплины является подготовка обучающихся к организационно–управленческому виду деятельности по направлению подготовки 09.03.03 Прикладная информатика (профиль подготовки: Разработка мобильных и Web–приложений) посредством обеспечения этапов формирования компетенций, предусмотренных ФГОС, в части представленных ниже знаний, умений и навыков.

Задачами дисциплины является изучение понятийного аппарата дисциплины, основных теоретических положений и методов, формирование умений и привитие навыков применения теоретических знаний для решения практических и прикладных задач.

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к «Часть, формируемая участниками образовательных отношений» блока 1.

В свою очередь, данный курс, помимо самостоятельного значения, является последующей дисциплиной после курсов: «Программирование», «Языки и методы разработки Web-приложений», «Вычислительные системы, сети и телекоммуникации»

А также предшествующей дисциплиной для курсов: «Разработка Web-приложений с применением фреймворков», «Продвинутая разработка мобильных приложений: код, архитектура, алгоритмы», «Облачные сервисы и серверная разработка мобильных приложений», «Технологии разработки программных приложений»

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Компетенции дисциплины

Код по ФГОС	Индикаторы достижения	Планируемые результаты обучения по дисциплине (ЗУВ)
Профессиональные		
ПК – 5. Способен разрабатывать и адаптировать прикладное программное обеспечение	ПК – 5.1 Проектирует и разрабатывает прототипы ИС ПК – 5.2 Применяет технологии разработки и управления базами данных ИС ПК – 5.3 Обеспечивает соответствие разработанного кода и процесса кодирования на языках программирования принятым в организации или проекте стандартам и технологиям	Знает методы и технологии проектирования и разработки прототипов информационных систем, включая управление базами данных и стандарты кодирования. Осведомлён о различных языках программирования и подходах к адаптации прикладного программного обеспечения под специфические требования проекта или организации. Умеет проектировать и разрабатывать прототипы информационных систем, применять современные технологии для разработки и управления базами данных. Способен обеспечивать соответствие разработанного кода принятым стандартам и технологиям, а также адаптировать процессы кодирования для оптимизации производительности и соблюдения качества. Владет навыками эффективной разработки и адаптации прикладного программного обеспечения в соответствии с требованиями и стандартами проекта. Использует передовые методы программирования и управления данными для создания стабильных, эффективных и легко адаптируемых приложений.

4. Объем дисциплины и виды учебной работы

Таблица 2

Вид учебной работы		Всего часов/ зач.ед.	Всего часов/ зач.ед.	Всего часов/ зач.ед.	Всего часов/ зач.ед.
		ОФО 5 семестр	ОФО 6 семестр	ЗФО 5 семестр	ЗФО 6 семестр
Контактная работа (всего)		68/1.8	80/2.2	12	14
В том числе:					
Лекции		34/0.9	32/0.9	4	4
Практические занятия					
Семинары					
Лабораторные работы		34/0.9	48/1.3	8	10
Самостоятельная работа (всего)		68/1.8	72/2	118	144
В том числе:					
Курсовая работа (проект)					
Расчетно–графические работы					
Индивидуальное задание					
Рефераты		16/0.4	18/0.5	46	72
Доклады					
Презентации					
<i>И (или) другие виды самостоятельной работы:</i>					
Подготовка к лабораторным работам		36/1	36/1	36	36
Подготовка к практическим занятиям					
Подготовка к экзамену		16/0.4	18/0.5	36	36
Подготовка к зачету					
Вид отчетности		Зачет	Экзаме н	Зачет	Экзаме н
Общая трудоемкость дисциплины	ОФО 288/8	136/3.6	152/3.5		
	ЗФО			130	154

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Таблица 3

№ п/п	Наименование раздела дисциплины по семестрам	Часы лекц. занятий ОФО	Часы лаборат. занятий ОФО	Часы лекц. занятий ЗФО	Часы лаборат. занятий ЗФО
5 семестр					
1.	Тема 1: Основы интернет-технологий и HTTP	6	6	2	2
2.	Тема 2: Клиент-серверная архитектура	4	4	2	2
3.	Тема 3: HTML/CSS для веб-разработчиков	4	4		2
4.	Тема 4: JavaScript и динамическое взаимодействие	4	4		2
5.	Тема 5: Введение в Flask	4	4		
6.	Тема 6: Маршрутизация и шаблоны в Flask	4	4		
7.	Тема 7: Работа с формами и данными пользователя в Flask	4	4		
8.	Тема 8: Базы данных и Flask	4	4		
6 семестр					
9.	Тема 9: Введение в FastAPI	4	6	2	2
10.	Тема 10: Асинхронное программирование в FastAPI	4	6	2	2
11.	Тема 11: Валидация данных и схемы в FastAPI	4	6		2
12.	Тема 12: Безопасность и аутентификация в FastAPI	4	6		2
13.	Тема 13: Основы Django	4	6		2
14.	Тема 14: Модели и ORM в Django	4	6		
15.	Тема 15: Шаблоны и формы в Django	4	6		
16.	Тема 16: Расширенные возможности Django	4	6		

5.2. Лекционные занятия

Таблица 4

Раздел	Наименование раздела дисциплины	Содержание раздела
1.	Лекция 1: Основы интернет-технологий и HTTP	- Введение в интернет-технологии: История и развитие интернета. - Основы протоколов: Понятие и функции HTTP. - HTTP запросы и ответы: Структура HTTP сообщений.

		• Статус-коды HTTP: Значение и использование в веб-разработке. • HTTP методы: Особенности использования GET, POST и других методов.
2.	Лекция 2: Клиент-серверная архитектура	• Основы клиент-серверной модели: Определение и ключевые принципы. • Компоненты клиент-серверной архитектуры: Роли и задачи сервера и клиента. • Примеры клиент-серверных приложений: Веб-браузеры и веб-серверы. • Преимущества и недостатки: Анализ архитектурного подхода. • Безопасность в клиент-серверных приложениях: Основные угрозы и методы защиты.
3.	Лекция 3: HTML/CSS для веб-разработчиков	• Основы HTML: Теги, атрибуты и структура документа. • Основы CSS: Селекторы, свойства и каскадность. • Создание веб-страниц: Практическое применение HTML и CSS. • Семантическая верстка: Использование семантических тегов HTML5. • Адаптивный дизайн: Медиа-запросы и принципы адаптивности.
4.	Лекция 4: JavaScript и динамическое взаимодействие	• Введение в JavaScript: Основы и история языка. • Операции с DOM: Манипуляции с элементами страницы. • События и обработчики: Реакция на действия пользователя. • AJAX и асинхронный JavaScript: Основы и примеры использования. • Современные подходы и фреймворки: Обзор React, Angular и Vue.js.
5.	Лекция 5: Введение в Flask	• Что такое Flask?: Понятие и основные особенности фреймворка. • Первое приложение на Flask: Установка и минимально возможный проект. • Структура Flask-приложения: Разбор основных компонентов и их функций. • Маршрутизация в Flask: Создание и обработка маршрутов. • Шаблоны и наследование шаблонов: Использование Jinja2 для рендеринга страниц.
6.	Лекция 6: Маршрутизация и шаблоны в Flask	• Подробное изучение маршрутизации: Параметризация маршрутов и обработчики ошибок. • Шаблоны Jinja2: Синтаксис,

		переменные, управляющие структуры. • Формирование URL и перенаправления: Программное и шаблонное создание URL. • Взаимодействие с формами: Обработка данных формы через Flask-WTF. • Улучшенная структура приложения: Применение паттернов проектирования для организации кода.
7.	Лекция 7: Работа с формами и данными пользователя в Flask	• Создание форм: Использование Flask-WTF для определения и валидации форм. • Безопасность форм: Защита от CSRF-атак. • Обработка и валидация данных: Техники и практики безопасной обработки. • Отправка и получение данных: Практические примеры с использованием запросов GET и POST. • Обратная связь пользователю: Отображение сообщений об ошибках и успешных действиях.
8.	Лекция 8: Базы данных и Flask	• Интеграция с базами данных: Конфигурация и подключение к БД. • ORM SQLAlchemy: Основы, создание моделей и сессий. • Миграции с Flask-Migrate: Управление изменениями структуры БД. • CRUD операции: Создание, чтение, обновление и удаление данных через ORM. • Оптимизация запросов: Техники улучшения производительности взаимодействия с БД.
9.	Лекция 9: Введение в FastAPI	• Что такое FastAPI?: Введение и основные преимущества использования FastAPI. • Первые шаги с FastAPI: Установка и настройка окружения. • Создание простого приложения: Основы маршрутизации и обработчиков. • Автоматическая документация: Использование Swagger и ReDoc для документации API. • Основы асинхронного программирования: Как и зачем использовать async и await в FastAPI.
10.	Лекция 10: Асинхронное программирование в FastAPI	• Глубокое погружение в async и await: Понимание асинхронности в Python. • Асинхронное взаимодействие с базами данных: Эффективное использование асинхронных ORM. • Асинхронные запросы к API: Основы использования httpx. • Преимущества асинхронности: Как асинхронность может улучшить

		производительность вашего API. • Обработка исключений в асинхронном коде: Лучшие практики.
11.	Лекция 11: Валидация данных и схемы в FastAPI	• Использование Pydantic для валидации данных: Определение и применение моделей данных. • Расширенные возможности валидации: Кастомные валидаторы и усложненные схемы. • Схемы запросов и ответов: Как определять и использовать их в FastAPI. • Ошибки валидации и обработка ошибок: Управление исключениями и пользовательскими сообщениями. • Тестирование валидации данных: Подходы и инструменты.
12.	Лекция 12: Безопасность и аутентификация в FastAPI	• Методы аутентификации: Основы и реализация токен-базированной аутентификации. • Защита маршрутов: Роли и разрешения в управлении доступом. • Использование HTTPS: Важность шифрования в обеспечении безопасности данных. • Обработка уязвимостей: XSS, CSRF и SQL инъекции. • Лучшие практики безопасности: Как обеспечивать безопасность веб-приложения на всех уровнях.
13.	Лекция 13: Основы Django	• Архитектура Django: Понимание компонентов MVT (Model-View-Template). • Установка и настройка Django: Создание первого проекта и приложения. • Маршрутизация и представления: Как Django обрабатывает входящие запросы. • Настройки и конфигурации: Оптимизация проекта для различных сред. • Введение в административную панель Django: Управление контентом через админку.
14.	Лекция 14: Модели и ORM в Django	• Определение моделей данных: Создание и использование моделей в Django. • Миграции: Управление изменениями в базе данных. • Взаимодействие с базой данных: Примеры CRUD операций через ORM. • Продвинутое использование ORM: Агрегирование, фильтрация и оптимизация запросов. • Тестирование моделей: Подходы к написанию тестов для проверки логики данных.

15.	Лекция 15: Шаблоны и формы в Django	• Система шаблонов Django: Синтаксис шаблонов и наследование. • Формы в Django: Создание форм и валидация данных. • Динамическое создание форм: Использование формсетов и модельных форм. • Стилизация форм: Подключение CSS к формам Django. • AJAX и Django: Реализация асинхронных форм без перезагрузки страницы.
16.	Лекция 16: Расширенные возможности Django	• Создание REST API с Django REST Framework: Основы DRF и создание API. • Тестирование Django-приложений: Использование pytest и factory_boy для комплексного тестирования. • Оптимизация Django-приложений: Кэширование, асинхронные задачи и использование CDN. • Интеграция с другими сервисами и API: Работа с внешними API и сервисами. • Масштабирование и развертывание: Рекомендации по развертыванию и масштабированию Django-приложений в производстве.

5.3. Практические (семинарские) занятия

Практические занятия учебным планом не предусмотрены.

5.4 Лабораторные занятия

Таблица 5

№ п/п	Наименование работы	Содержание лабораторной работы
1.	Лабораторная работа №1: Основы интернет-технологий и HTTP	Название: Понимание интернет-протоколов и работы HTTP Цель(и): • Изучение основных интернет-протоколов и механизма работы HTTP. • Освоение методов отправки HTTP-запросов и анализа HTTP-ответов. Задания
2.	Лабораторная работа №2: Клиент-серверная архитектура	Название: Изучение клиент-серверной архитектуры Цель(и): • Понимание основных принципов и компонентов клиент-серверной архитектуры. • Разработка и тестирование простого клиент-серверного приложения. Задания

3.	Лабораторная работа №3: Основы HTML и CSS	Название: Введение в HTML и CSS Цель(и): - Освоение основ HTML и CSS для создания веб-страниц. - Разработка простой, но полноценной веб-страницы с использованием стандартных техник вёрстки. Задания
4.	Лабораторная работа №4: JavaScript и динамическое взаимодействие	Название: Основы JavaScript, AJAX и DOM Цель(и): - Изучение основ JavaScript и его использования для манипуляций с DOM. - Освоение технологии AJAX для асинхронных запросов к серверу. Задания:
5.	Лабораторная работа №5: Введение в Flask	Название: Знакомство с Flask: установка, запуск и структура приложения Цель(и): - Освоение начальных шагов работы с веб-фреймворком Flask, включая установку, конфигурацию и создание базового приложения. - Понимание структуры Flask-приложения и его компонентов. Задания:
6.	Лабораторная работа №6: Маршрутизация и шаблоны в Flask	Название: Работа с маршрутами и шаблонами в Flask Цель(и): - Изучение создания маршрутов и обработки запросов в Flask. - Освоение использования шаблонизатора Jinja2 для динамического формирования HTML-страниц. Задания:
7.	Лабораторная работа №7: Работа с формами и данными пользователя в Flask	Название: Управление данными формы в Flask Цель(и): - Изучение методов получения и обработки данных, отправленных пользователем через веб-формы в Flask. - Разработка навыков валидации данных на серверной стороне и отправки ответов пользователю. Задания:
8.	Лабораторная работа №8: Базы данных и Flask	Название: Интеграция баз данных с Flask при использовании SQLAlchemy Цель(и): - Освоение методов подключения и работы с базами данных в Flask через ORM (Object-Relational Mapping) с использованием SQLAlchemy. - Понимание принципов ORM и практика моделирования данных. Задания:
9.	Лабораторная работа №9: Введение в FastAPI	Название: Знакомство с FastAPI Цель(и): - Освоение основных особенностей и преимуществ использования веб-фреймворка FastAPI. - Разработка и запуск первого приложения на FastAPI. Задания:

10.	Лабораторная работа №10: Асинхронное программирование в FastAPI	Название: Основы асинхронного программирования в FastAPI Цель(и): - Изучение и практическое применение асинхронных операций в FastAPI с использованием синтаксиса <code>async</code> и <code>await</code>. - Разработка асинхронных функций для работы с базой данных. Задания:
11.	Лабораторная работа №11: Валидация данных и схемы в FastAPI	Название: Использование Pydantic и Swagger для валидации данных и документации в FastAPI Цель(и): - Изучение и практическое применение библиотеки Pydantic для валидации данных в FastAPI. - Освоение автоматической генерации документации API с использованием Swagger UI в FastAPI. Задания:
12.	Лабораторная работа №12: Безопасность и аутентификация в FastAPI	Название: Управление безопасностью и аутентификацией в FastAPI Цель(и): - Освоение методов реализации аутентификации и авторизации в FastAPI. - Разработка механизмов защиты маршрутов и управления доступом к ресурсам. Задания:
13.	Лабораторная работа №13: Основы Django	Название: Введение в Django: Архитектура и создание проекта Цель(и): - Понимание архитектуры Django и её компонентов. - Настройка и создание базового проекта в Django для дальнейшей разработки. Задания:
14.	Лабораторная работа №14: Модели и ORM в Django	Название: Работа с моделями и ORM в Django Цель(и): - Освоение процесса создания и использования моделей в Django для взаимодействия с базой данных. - Изучение системы миграций Django для управления изменениями в структуре базы данных. Задания:
15.	Лабораторная работа №15: Шаблоны и формы в Django	Название: Использование шаблонов и форм в Django Цель(и): - Освоение работы с системой шаблонов Django для генерации динамического HTML-контента. - Изучение создания и обработки форм с помощью Django Forms для эффективного взаимодействия с пользователем. Задания:
16.	Лабораторная работа №16: Расширенные возможности Django	Название: Создание REST API и оптимизация приложений в Django Цель(и): - Изучение создания REST API с использованием Django REST Framework. - Освоение методов тестирования и оптимизации веб-приложений на Django для улучшения производительности и надежности. Задания:

6. Самостоятельная работа студентов по дисциплине

6.1. Подготовка рефератов

В рамках самостоятельной работы студент выполняет ряд работ по предложенным темам. Студент самостоятельно собирает необходимую для выполнения работ информацию, в ряде случаев дополняя ее своими обоснованными оценками и допущениями.

Темы рефератов:

1. Сравнение производительности между Flask и FastAPI.
2. Анализ использования асинхронного программирования в веб-фреймворках на Python.
3. Введение в микросервисы с использованием Flask.
4. Разработка и развертывание контейнеризованных Python-приложений с Docker.
5. Использование WebSocket в FastAPI для создания реалтайм приложений.
6. Безопасность веб-приложений: общие угрозы и методы защиты.
7. Применение GraphQL в современных веб-приложениях на примере Django.
8. Особенности использования PostgreSQL и MongoDB с Django ORM.
9. Разработка RESTful API с Django REST Framework.
10. Тестирование и отладка веб-приложений на Flask с использованием Pytest.
11. Оптимизация запросов в Django: советы и трюки.
12. Интеграция сторонних сервисов в FastAPI.
13. Создание и управление асинхронными задачами с Celery в Django.
14. Роль Redis в кэшировании данных для Django приложений.
15. Проблемы масштабируемости веб-приложений и как их решать с помощью микросервисной архитектуры.
16. Использование современных CSS фреймворков для улучшения пользовательского интерфейса Django-приложений.
17. Автоматизация развертывания веб-приложений на Python с помощью CI/CD.
18. Использование NoSQL баз данных в проектах на FastAPI.
19. Методы защиты от CSRF и XSS атак в приложениях на Flask.
20. Применение паттернов проектирования в архитектуре веб-приложений.
21. Основы SEO и его влияние на разработку веб-приложений.
22. Влияние HTTP/2 на производительность современных веб-приложений.
23. Создание интерактивных веб-приложений с использованием Vue.js и Django.

24. Анализ преимуществ и недостатков SPA (Single Page Application) и MPA (Multi Page Application).
25. Использование JSON Web Tokens для аутентификации в FastAPI.
26. Практические аспекты использования API Gateway в архитектуре микросервисов.
27. Введение в Serverless архитектуру на примере AWS Lambda и Python.
28. Безопасное хранение и управление секретами в разработке на Flask.
29. Разработка мобильных приложений с backend на Django.
30. Архитектурные паттерны для обеспечения высокой доступности и отказоустойчивости веб-приложений.

7. Оценочные средства

7.1. Вопросы к первой рубежной аттестации(5-й семестр)

1. Объясните, как работает протокол HTTP и почему он считается безсостоянием. Какие последствия это имеет для разработки веб-приложений?
2. Какие типы HTTP-запросов вы знаете, и в каких ситуациях они обычно используются при разработке веб-приложений?
3. Какие существуют стандартные HTTP-статус-коды и что они сообщают клиенту? Приведите примеры использования различных типов кодов в веб-разработке.
4. Какие меры безопасности следует рассмотреть при отправке данных через HTTP? Объясните, какие уязвимости может иметь HTTP и как их можно предотвратить.
5. Опишите принцип работы клиент-серверной архитектуры и приведите пример, как она используется в современных веб-приложениях.
6. Какие основные компоненты включает клиент-серверное взаимодействие и каковы их функции?
7. Обсудите преимущества и недостатки использования клиент-серверной архитектуры по сравнению с пиринговыми сетями.
8. Какие технологии обеспечивают масштабирование в клиент-серверных архитектурах и как они работают?
9. Какие методы лучше всего подходят для структурирования HTML-документов в больших веб-проектах? Объясните, как можно использовать HTML5 семантические элементы для улучшения доступности и SEO.
10. Обсудите, как CSS предоставляет механизмы для стилизации веб-страниц. Какие существуют методы для обеспечения согласованности стилей на больших сайтах?
11. Какие существуют методы создания адаптивных веб-дизайнов с помощью CSS? Приведите примеры использования медиа-запросов для разных типов устройств.
12. Обсудите, какие новые возможности CSS3 могут быть использованы для улучшения пользовательского интерфейса и взаимодействия на веб-сайтах.

Образец аттестационного билета

(за 1 рубежную аттестацию)

Грозненский государственный нефтяной технический университет им.акад. М.Д. Миллионщикова
Институт "цифровой экономики и технологического предпринимательства"

Группа "ПИ-24" Семестр "5"
Дисциплина "Серверная разработка Web-приложений"

Билет № 1

1. Объясните, как работает протокол HTTP и почему он считается безсостоянием. Какие последствия это имеет для разработки веб-приложений?
2. Обсудите, как CSS предоставляет механизмы для стилизации веб-страниц. Какие существуют методы для обеспечения согласованности стилей на больших сайтах?

Вопросы к второй рубежной аттестации (5-й семестр)

1. Каковы основные шаги для начала работы с Flask и что необходимо установить для запуска первого Flask-приложения?
2. Объясните, как Flask обрабатывает запросы от клиента к серверу и как данные передаются между клиентом и сервером в Flask.
3. Какие возможности предоставляет Flask для маршрутизации URL-адресов и как можно обрабатывать различные типы запросов, такие как GET и POST?
4. Опишите, как в Flask использовать шаблонизатор Jinja2 для динамической генерации HTML-страниц. Какие преимущества это предоставляет разработчикам?
5. Как можно в Flask создать и использовать формы для сбора данных от пользователя? Опишите процесс от получения данных до их обработки.
6. Какие меры предосторожности следует принимать при работе с пользовательскими данными в Flask для предотвращения распространенных уязвимостей, таких как SQL инъекции и XSS?
7. Как в Flask использовать SQLAlchemy для взаимодействия с базой данных? Объясните процесс настройки и пример простого запроса на выборку данных.
8. Как можно реализовать аутентификацию и авторизацию в приложении Flask? Приведите примеры использования расширений или встроенных функций для этого.
9. Обсудите, как в Flask управлять сессиями пользователя и какие проблемы безопасности могут возникнуть при их неправильном использовании.
10. Какие стратегии можно использовать для оптимизации производительности Flask-приложений, особенно при обращении к базам данных?
11. Как в Flask обеспечивается масштабируемость приложений, и какие инструменты или техники могут помочь в обработке большого количества пользовательских запросов?
12. Какие лучшие практики следует применять при разработке и развертывании веб-приложений на Flask в производственной среде?

Образец аттестационного билета

(за 2 рубежную аттестацию)

Билет № 1

1. Как FastAPI помогает в автоматической генерации документации API? Обсудите, как настроить и кастомизировать документацию Swagger UI.
2. Как в FastAPI использовать асинхронные базы данных? Приведите пример асинхронного взаимодействия с базой данных.

Подпись преподавателя _____ Подпись заведующего кафедрой _____

Вопросы к первой рубежной аттестации(6-й семестр)

1. Какие основные преимущества предоставляет FastAPI по сравнению с другими веб-фреймворками? Приведите примеры использования этих преимуществ в разработке веб-приложений.
2. Объясните, как в FastAPI реализована поддержка асинхронного программирования. Как это влияет на производительность веб-приложений?
3. Как FastAPI упрощает процесс валидации входящих данных? Приведите примеры использования Pydantic для валидации и сериализации данных.
4. Какие методы аутентификации поддерживает FastAPI? Обсудите, как реализовать токен-базирующую аутентификацию в FastAPI.
5. Как в FastAPI обеспечивается безопасность при обработке и хранении пользовательских данных? Обсудите методы предотвращения распространенных уязвимостей.
6. Объясните процесс настройки и развертывания первого FastAPI приложения. Какие шаги включает этот процесс?
7. Как в FastAPI использовать асинхронные базы данных? Приведите пример асинхронного взаимодействия с базой данных.
8. Какие стратегии можно использовать для оптимизации асинхронных запросов в FastAPI? Обсудите, как можно улучшить обработку запросов в условиях высокой нагрузки.
9. Как FastAPI помогает в автоматической генерации документации API? Обсудите, как настроить и кастомизировать документацию Swagger UI.
10. Как реализовать и тестировать сложные пользовательские валидации в FastAPI с использованием Pydantic? Приведите примеры таких валидаций.
11. Как можно использовать зависимости в FastAPI для повышения переиспользуемости кода и организации бизнес-логики? Приведите примеры такого использования.
12. Обсудите, как управлять доступом к различным частям приложения в FastAPI, используя систему разрешений и ролей. Как настроить эти элементы для обеспечения безопасного доступа?

Грозненский государственный нефтяной технический университет им.акад. М.Д. Миллионщикова
Институт "цифровой экономики и технологического предпринимательства"

Группа "ПИ-24" Семестр «6»

Дисциплина "Серверная разработка Web-приложений"

Билет № 1

1. Как в FastAPI обеспечивается безопасность при обработке и хранении пользовательских данных? Обсудите методы предотвращения распространенных уязвимостей.
2. Как в FastAPI использовать асинхронные базы данных? Приведите пример асинхронного взаимодействия с базой данных.

Подпись преподавателя _____ Подпись заведующего кафедрой _____

Вопросы к второй рубежной аттестации (6-й семестр)

1. Какие архитектурные компоненты составляют Django и как они взаимодействуют между собой для обработки веб-запроса?
2. Объясните процесс создания нового проекта в Django. Какие команды и шаги необходимы для начала работы?
3. Как Django использует модель-шаблон-представление (MVT) в своей архитектуре? Обсудите каждый компонент и его роль в создании веб-приложений.
4. Каким образом Django управляет базами данных через свою ORM? Приведите пример создания модели и выполнения запроса к базе данных.
5. Объясните, как в Django создать и использовать формы для сбора данных от пользователя. Какие инструменты предоставляет Django для валидации и обработки форм?
6. Как настроить и использовать шаблоны в Django для генерации HTML? Обсудите, как шаблоны упрощают создание динамических веб-страниц.
7. Какие методы оптимизации производительности доступны в Django? Обсудите использование кэширования и других техник для улучшения отклика приложения.
8. Как реализовать аутентификацию и авторизацию в Django? Обсудите использование встроенных средств, таких как система пользователей и групп.
9. Как можно расширить возможности Django с помощью сторонних приложений? Приведите примеры популярных расширений и их применения.
10. Как использовать Django REST Framework для создания REST API? Обсудите настройку сериализаторов и маршрутизацию запросов.
11. Какие стратегии тестирования рекомендуются для Django-приложений? Обсудите, как настроить и выполнить тесты для проверки функциональности и безопасности приложения.

12. Как Django поддерживает международизацию и локализацию? Обсудите, как настроить приложение для поддержки множества языков и региональных настроек.

Образец аттестационного билета

(за 2 рубежную аттестацию)

Грозненский государственный нефтяной технический университет им.акад. М.Д. Миллионщикова
Институт "цифровой экономики и технологического предпринимательства"

Группа "ПИ-24" Семестр «6»

Дисциплина "Серверная разработка Web-приложений"

Билет № 1

1. Как использовать Django REST Framework для создания REST API? Обсудите настройку сериализаторов и маршрутизацию запросов.
2. Как можно расширить возможности Django с помощью сторонних приложений? Приведите примеры популярных расширений и их применения.

Подпись преподавателя_____ **Подпись заведующего кафедрой**_____

7.2. Вопросы на зачет(5-й семестр)

1. Объясните, как работает протокол HTTP и почему он считается безсостоянием. Какие последствия это имеет для разработки веб-приложений?
2. Какие типы HTTP-запросов вы знаете, и в каких ситуациях они обычно используются при разработке веб-приложений?
3. Какие существуют стандартные HTTP-статус-коды и что они сообщают клиенту? Приведите примеры использования различных типов кодов в веб-разработке.
4. Какие меры безопасности следует рассмотреть при отправке данных через HTTP? Объясните, какие уязвимости может иметь HTTP и как их можно предотвратить.
5. Опишите принцип работы клиент-серверной архитектуры и приведите пример, как она используется в современных веб-приложениях.
6. Какие основные компоненты включает клиент-серверное взаимодействие и каковы их функции?
7. Обсудите преимущества и недостатки использования клиент-серверной архитектуры по сравнению с пиринговыми сетями.
8. Какие технологии обеспечивают масштабирование в клиент-серверных архитектурах и как они работают?
9. Какие методы лучше всего подходят для структурирования HTML-документов в больших веб-проектах? Объясните, как можно использовать HTML5 семантические элементы для улучшения доступности и SEO.
10. Обсудите, как CSS предоставляет механизмы для стилизации веб-страниц. Какие существуют методы для обеспечения согласованности стилей на больших сайтах?
11. Какие существуют методы создания адаптивных веб-дизайнов с помощью CSS? Приведите примеры использования медиа-запросов для разных типов устройств.
12. Обсудите, какие новые возможности CSS3 могут быть использованы для улучшения пользовательского интерфейса и взаимодействия на веб-сайтах.
13. Каковы основные шаги для начала работы с Flask и что необходимо установить для запуска первого Flask-приложения?
14. Объясните, как Flask обрабатывает запросы от клиента к серверу и как данные передаются между клиентом и сервером в Flask.
15. Какие возможности предоставляет Flask для маршрутизации URL-адресов и как можно обрабатывать различные типы запросов, такие как GET и POST?
16. Опишите, как в Flask использовать шаблонизатор Jinja2 для динамической генерации HTML-страниц. Какие преимущества это предоставляет разработчикам?
17. Как можно в Flask создать и использовать формы для сбора данных от пользователя? Опишите процесс от получения данных до их обработки.

18. Какие меры предосторожности следует принимать при работе с пользовательскими данными в Flask для предотвращения распространенных уязвимостей, таких как SQL инъекции и XSS?
19. Как в Flask использовать SQLAlchemy для взаимодействия с базой данных? Объясните процесс настройки и пример простого запроса на выборку данных.
20. Как можно реализовать аутентификацию и авторизацию в приложении Flask? Приведите примеры использования расширений или встроенных функций для этого.
21. Обсудите, как в Flask управлять сессиями пользователя и какие проблемы безопасности могут возникнуть при их неправильном использовании.
22. Какие стратегии можно использовать для оптимизации производительности Flask-приложений, особенно при обращении к базам данных?
23. Как в Flask обеспечивается масштабируемость приложений, и какие инструменты или техники могут помочь в обработке большого количества пользовательских запросов?
24. Какие лучшие практики следует применять при разработке и развертывании веб-приложений на Flask в производственной среде?

Образец билета на зачет

Грозненский государственный нефтяной технический университет им.акад. М.Д. Миллионщикова
Институт "цифровой экономики и технологического предпринимательства"

Группа "ПИ-24" Семестр "5"
Дисциплина "Серверная разработка Web-приложений"

Билет № 1

1. Как FastAPI помогает в автоматической генерации документации API? Обсудите, как настроить и кастомизировать документацию Swagger UI.
2. Обсудите, как управлять доступом к различным частям приложения в FastAPI, используя систему разрешений и ролей. Как настроить эти элементы для обеспечения безопасного доступа?

Подпись преподавателя _____ Подпись заведующего кафедрой _____

Вопросы на экзамен(6-й семестр)

1. Какие основные преимущества предоставляет FastAPI по сравнению с другими веб-фреймворками? Приведите примеры использования этих преимуществ в разработке веб-приложений.
2. Объясните, как в FastAPI реализована поддержка асинхронного программирования. Как это влияет на производительность веб-приложений?
3. Как FastAPI упрощает процесс валидации входящих данных? Приведите примеры использования Pydantic для валидации и сериализации данных.

4. Какие методы аутентификации поддерживает FastAPI? Обсудите, как реализовать токен-базирующую аутентификацию в FastAPI.
5. Как в FastAPI обеспечивается безопасность при обработке и хранении пользовательских данных? Обсудите методы предотвращения распространенных уязвимостей.
6. Объясните процесс настройки и развертывания первого FastAPI приложения. Какие шаги включает этот процесс?
7. Как в FastAPI использовать асинхронные базы данных? Приведите пример асинхронного взаимодействия с базой данных.
8. Какие стратегии можно использовать для оптимизации асинхронных запросов в FastAPI? Обсудите, как можно улучшить обработку запросов в условиях высокой нагрузки.
9. Как FastAPI помогает в автоматической генерации документации API? Обсудите, как настроить и кастомизировать документацию Swagger UI.
10. Как реализовать и тестировать сложные пользовательские валидации в FastAPI с использованием Pydantic? Приведите примеры таких валидаций.
11. Как можно использовать зависимости в FastAPI для повышения переиспользуемости кода и организации бизнес-логики? Приведите примеры такого использования.
12. Обсудите, как управлять доступом к различным частям приложения в FastAPI, используя систему разрешений и ролей. Как настроить эти элементы для обеспечения безопасного доступа?
13. Какие архитектурные компоненты составляют Django и как они взаимодействуют между собой для обработки веб-запроса?
14. Объясните процесс создания нового проекта в Django. Какие команды и шаги необходимы для начала работы?
15. Как Django использует модель-шаблон-представление (MVT) в своей архитектуре? Обсудите каждый компонент и его роль в создании веб-приложений.
16. Каким образом Django управляет базами данных через свою ORM? Приведите пример создания модели и выполнения запроса к базе данных.
17. Объясните, как в Django создать и использовать формы для сбора данных от пользователя. Какие инструменты предоставляет Django для валидации и обработки форм?
18. Как настроить и использовать шаблоны в Django для генерации HTML? Обсудите, как шаблоны упрощают создание динамических веб-страниц.

19. Какие методы оптимизации производительности доступны в Django? Обсудите использование кэширования и других техник для улучшения отклика приложения.
20. Как реализовать аутентификацию и авторизацию в Django? Обсудите использование встроенных средств, таких как система пользователей и групп.
21. Как можно расширить возможности Django с помощью сторонних приложений? Приведите примеры популярных расширений и их применения.
22. Как использовать Django REST Framework для создания REST API? Обсудите настройку сериализаторов и маршрутизацию запросов.
23. Какие стратегии тестирования рекомендуются для Django-приложений? Обсудите, как настроить и выполнить тесты для проверки функциональности и безопасности приложения.
24. Как Django поддерживает международизацию и локализацию? Обсудите, как настроить приложение для поддержки множества языков и региональных настроек.

Образец билета на экзамен

Грозненский государственный нефтяной технический университет им.акад. М.Д. Миллионщикова
Институт "цифровой экономики и технологического предпринимательства"

Группа "ПИ-24" Семестр "6"

Дисциплина "Серверная разработка Web-приложений"

Билет № 1

1. Какие методы оптимизации производительности доступны в Django? Обсудите использование кэширования и других техник для улучшения отклика приложения.
2. Объясните процесс создания нового проекта в Django. Какие команды и шаги необходимы для начала работы?

Подпись преподавателя _____ Подпись заведующего кафедрой _____

7.3. Текущий контроль

В качестве оценочных средств текущего контроля используются средства контроля выполнения и защиты лабораторных работ по дисциплине. Защита лабораторной работы – ответ на контрольные вопросы после выполнения лабораторной работы.

Лабораторная работа №1: Основы интернет-технологий и HTTP

Лабораторная работа №2: Клиент-серверная архитектура

Лабораторная работа №3: Основы HTML и CSS

Лабораторная работа №4: JavaScript и динамическое взаимодействие

Лабораторная работа №5: Введение в Flask

Лабораторная работа №6: Маршрутизация и шаблоны в Flask

Лабораторная работа №7: Работа с формами и данными пользователя в Flask

Лабораторная работа №8: Базы данных и Flask

Лабораторная работа №9: Введение в FastAPI

Лабораторная работа №10: Асинхронное программирование в FastAPI

Лабораторная работа №11: Валидация данных и схемы в FastAPI

Лабораторная работа №12: Безопасность и аутентификация в FastAPI

Лабораторная работа №13: Основы Django

Лабораторная работа №14: Модели и ORM в Django

Лабораторная работа №15: Шаблоны и формы в Django

Лабораторная работа №16: Расширенные возможности Django

Лабораторная работа №1: Основы интернет-технологий и HTTP

Название: Понимание интернет-протоколов и работы HTTP

Цель(и):

- Изучение основных интернет-протоколов и механизма работы HTTP.
- Освоение методов отправки HTTP-запросов и анализа HTTP-ответов.

Задания:

1. Используйте инструменты разработчика в браузере для изучения HTTP-запросов и ответов на примере популярных веб-сайтов.
2. Создайте простую веб-страницу с формой, которая отправляет данные на сервер. Используйте любой онлайн HTTP сервер, который может принять и показать возвращаемые данные (например, httpbin.org).
3. С помощью Python и библиотеки requests напишите скрипт, который отправляет HTTP-запросы различного типа (GET, POST) и выводит ответы сервера на экран.
4. Используйте Postman для создания и отправки запросов к различным API и анализа ответов.

Контрольные вопросы:

1. Что такое HTTP и каковы его основные функции?
2. Какие типы HTTP-запросов вы знаете и в каких случаях их используют?
3. Что такое статус-код в HTTP-ответе и что он обозначает?
4. Какие инструменты можно использовать для тестирования и отладки HTTP-запросов и ответов?

7.4. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания.

Таблица 7

Планируемые результаты освоения компетенции	Критерии оценивания результатов обучения				Наименование оценочного средства
	менее 41 баллов (неудовлетворительно)	41–60 баллов (удовлетворительно)	61–80 баллов (хорошо)	81–100 баллов (отлично)	
ПК – 5. Способен разрабатывать и адаптировать прикладное программное обеспечение					
Знать методы и технологии проектирования и разработки прототипов информационных систем, включая управление базами данных и стандарты кодирования. Осведомлён о различных языках программирования и подходах к адаптации прикладного программного обеспечения под специфические требования проекта или организации.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	задания для лабораторной работы, билеты рубежных аттестаций, темы докладов

Уметь проектировать и разрабатывать прототипы информационных систем, применять современные технологии для разработки и управления базами данных. Способен обеспечивать соответствие разработанного кода принятым стандартам и технологиям, а также адаптировать процессы кодирования для оптимизации производительности и соблюдения качества.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	<i>задания для лабораторной работы, билеты рубежных аттестаций, темы докладов</i>
Владеть навыками эффективной разработки и адаптации прикладного программного обеспечения в соответствии с требованиями и стандартами проекта. Использует передовые методы программирования и управления данными для создания стабильных, эффективных и легко адаптируемых приложений.	Фрагментарные знания	Неполные знания	Сформированные, но содержащие отдельные пробелы знания	Сформированные систематические знания	<i>задания для лабораторной работы, билеты рубежных аттестаций, темы докладов</i>

8. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Для осуществления процедур текущего контроля успеваемости и промежуточной аттестации обучающихся созданы фонды оценочных средств, адаптированные для инвалидов и лиц с ограниченными возможностями здоровья и позволяющие оценить достижение ими запланированных в основной образовательной программе результатов обучения и уровень сформированности всех компетенций, заявленных в образовательной программе. Форма проведения текущей аттестации для студентов–инвалидов устанавливается с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.). При тестировании для слабовидящих студентов используются фонды оценочных средств с укрупненным шрифтом.

На экзамен приглашается сопровождающий, который обеспечивает техническое сопровождение студенту. При необходимости студенту–инвалиду предоставляется дополнительное время для подготовки ответа на экзамене (или экзамене). Обучающиеся с ограниченными возможностями здоровья и обучающиеся инвалиды обеспечиваются печатными и электронными образовательными ресурсами (программы, учебные пособия для самостоятельной работы и т.д.) в формах, адаптированных к ограничениям их здоровья и восприятия информации:

для инвалидов и лиц с ограниченными возможностями здоровья по зрению:

– для слепых: задания для выполнения на семинарах и практических занятиях оформляются рельефно–точечным шрифтом Брайля или в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением для слепых, либо зачитываются ассистентом; письменные задания выполняются на бумаге рельефно–точечным шрифтом Брайля или на компьютере со специализированным программным обеспечением для слепых либо надиктовываются ассистенту;

обучающимся для выполнения задания при необходимости предоставляется комплект письменных принадлежностей и бумага для письма рельефно–точечным шрифтом Брайля, компьютер со специализированным программным обеспечением для слепых;

– для слабовидящих: обеспечивается индивидуальное равномерное освещение не менее 300 люкс; обучающимся для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; задания для выполнения заданий оформляются увеличенным шрифтом;

для инвалидов и лиц с ограниченными возможностями здоровья по слуху:

– для глухих и слабослышащих: обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования; предоставляются услуги сурдопереводчика;

– для слепоглухих допускается присутствие ассистента, оказывающего услуги тифлосурдопереводчика (помимо требований, выполняемых соответственно для слепых и глухих);

для лиц с тяжелыми нарушениями речи, глухих, слабослышащих лекции и семинары, проводимые в устной форме, проводятся в письменной форме;

для инвалидов и лиц с ограниченными возможностями здоровья, имеющих нарушения опорно–двигательного аппарата:

– для лиц с нарушениями опорно–двигательного аппарата, нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей: письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту; выполнение заданий (тестов, контрольных работ), проводимые

в письменной форме, проводятся в устной форме путем опроса, беседы с обучающимися.

9. Учебно–методическое и информационное обеспечение дисциплины

9.1. Литература

1. Гринберг М. "Flask Web Development: Developing Web Applications with Python", 2nd edition. – O'Reilly Media, 2018. – 316 с.
2. Макграт М. "Web Development with Django". – Packt Publishing, 2020. – 798 с.
3. Саммерфильд М. "Программирование на Python 3. Подробное руководство". – СПб.: Питер, 2016. – 496 с.
4. Лутц М. "Изучаем Python, том 1", 5-е изд. – М.: ООО "И.Д. Вильямс", 2019. – 688 с.
5. Лутц М. "Изучаем Python, том 2", 5-е изд. – М.: ООО "И.Д. Вильямс", 2019. – 720 с.
6. Ульянов В.В. "Разработка веб-приложений в среде Python". – М.: ДМК Пресс, 2021. – 384 с.
7. Мозер М. "Программирование на Django". – М.: Эксмо, 2018. – 512 с.
8. Шой З. "Learn Python the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code", 3rd Edition. – Addison-Wesley, 2017. – 320 с.
9. Винсент Т. "Build APIs You Won't Hate: Everyone and their dog wants an API, so you should probably learn how to build them". – CreateSpace Independent Publishing Platform, 2014. – 150 с.
10. Перкинс Б. "Python 3 Web Development Beginner's Guide". – Packt Publishing, 2011. – 336 с.
11. Ричардсон Л., Амундсен М., Руби С. "RESTful веб-сервисы". – СПб.: Символ-Плюс, 2011. – 448 с.
12. Флиган Д. "Python. Подробный справочник". – СПб.: ООО "Альфа-книга", 2016. – 544 с.

9.2. Методические указания для обучающихся по освоению дисциплины (приложение)

10. Материально–техническое обеспечение дисциплины

10.1. Материально–техническая база

Лекционная аудитория, оснащенная компьютером, видеопроекционным оборудованием, в том числе для презентаций, средствами звуковоспроизведения, экраном.

Мультимедийные средства и другая техника для презентаций учебного материала, офисный пакет программ MSWindows (MS Excel, MSWord) для оформления расчетов экономической эффективности информационных систем, OpenOfficeGoogleChrome.

10.2. Помещения для самостоятельной работы

Помещение для самостоятельной работы (Главный учебный корпус ФГБОУ ВО «Грозненский государственный нефтяной технический университет» 364902, Чеченская республика, г. Грозный, проспект им. Х.А. Исаева, 100. Аудитория оснащена необходимой компьютерной техникой, в наличии есть необходимое ПО: WinPro 10 RUS Upgrd OLP NL Acdmc; OfficeStd RUS OLP NL Acdmc (право на использование согласно Контракту № 267–ЭА/19 от 15.09.2019 г.).

Система ГАРАНТ (проприетарная лицензия) Visual Studio–(Freemium) 1С Предприятие договор от 02.12.2020 регистрационные номера продуктов (9334859; 9334952) Sublime Text– (открытый доступ) Notepad++ (открытый доступ).

Методические указания по освоению дисциплины «Серверная разработка Web-приложений»

1. Методические указания для обучающихся по планированию и организации времени, необходимого для освоения дисциплины.

Изучение рекомендуется начать с ознакомления с рабочей программой дисциплины, ее структурой и содержанием разделов (модулей), фондом оценочных средств, ознакомиться с учебно–методическим и информационным обеспечением дисциплины.

Дисциплина «Серверная разработка Web-приложений» состоит из 9 связанных между собою тем, обеспечивающих последовательное изучение материала.

Обучение по дисциплине «Серверная разработка Web-приложений» осуществляется в следующих формах:

1. Аудиторные занятия (лекции, практические занятия).
2. Самостоятельная работа студента (подготовка к лекциям, лабораторным занятиям, рефератам и иным формам письменных работ, индивидуальная консультация с преподавателем).
3. Интерактивные формы проведения занятий (лекция–дискуссия, групповое решение кейса и др. формы).

Учебный материал структурирован и изучение дисциплины производится в тематической последовательности. Каждому практическому занятию и самостоятельному изучению материала предшествует лекция по данной теме. Обучающиеся самостоятельно проводят предварительную подготовку к занятию, принимают активное и творческое участие в обсуждении теоретических вопросов, разборе проблемных ситуаций и поисков путей их решения. Многие проблемы,

изучаемые в курсе, носят дискуссионный характер, что предполагает интерактивный характер проведения занятий на конкретных примерах.

Описание последовательности действий обучающегося:

При изучении курса следует внимательно слушать и конспектировать материал, излагаемый на аудиторных занятиях. Для его понимания и качественного усвоения рекомендуется следующая последовательность действий:

1. После окончания учебных занятий для закрепления материала просмотреть и обдумать текст лекции, прослушанной сегодня, разобрать рассмотренные примеры (10 – 15 минут).

2. При подготовке к лекции следующего дня повторить текст предыдущей лекции, подумать о том, какая может быть следующая тема (10 – 15 минут).

3. В течение недели выбрать время для работы с литературой в библиотеке (по 1 часу).

4. При подготовке к практическому занятию повторить основные понятия по теме, изучить примеры. Решая конкретную ситуацию, – предварительно понять, какой теоретический материал нужно использовать. Наметить план решения, попробовать на его основе решить лабораторные задания.

2. Методические указания по работе обучающихся во время проведения лекций.

Лекции дают обучающимся систематизированные знания по дисциплине, концентрируют их внимание на наиболее сложных и важных вопросах. Лекции обычно излагаются в традиционном или в проблемном стиле. Для студентов в большинстве случаев в проблемном стиле. Проблемный стиль позволяет стимулировать активную познавательную

деятельность обучающихся и их интерес к дисциплине, формировать творческое мышление, прибегать к противопоставлениям и сравнениям, делать обобщения, активизировать внимание обучающихся путем постановки проблемных вопросов, поощрять дискуссию.

Во время лекционных занятий рекомендуется вести конспектирование учебного материала, обращать внимание на формулировки и категории, раскрывающие суть того или иного явления, или процессов, выводы и практические рекомендации.

Конспект лекции лучше подразделять на пункты, соблюдая красную строку. Этому в большой степени будут способствовать вопросы плана лекции, предложенные преподавателям. Следует обращать внимание на акценты, выводы, которые делает преподаватель, отмечая наиболее важные моменты в лекционном материале замечаниями

«важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек, подчеркивая термины и определения.

Целесообразно разработать собственную систему сокращений, аббревиатур и символов. Однако при дальнейшей работе с конспектом символы лучше заменить обычными словами для быстрого зрительного восприятия текста.

Работая над конспектом лекций, необходимо использовать не только основную литературу, но и ту литературу, которую дополнительно рекомендовал преподаватель. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть теоретическим материалом.

Тематика лекций дается в рабочей программе дисциплины.

3. Методические указания обучающимся по подготовке к лабораторным/семинарским занятиям.

На практических занятиях приветствуется активное участие в обсуждении конкретных ситуаций, способность на основе полученных знаний находить

наиболее эффективные решения поставленных проблем, уметь находить полезный дополнительный материал по тематике семинарских занятий.

Студенту рекомендуется следующая схема подготовки к семинарскому занятию:

1. Ознакомление с планом практического занятия, который отражает содержание предложенной темы;
2. Проработать конспект лекций;
3. Прочитать основную и дополнительную литературу.

В процессе подготовки к практическим занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал из-за лимита аудиторных часов. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов отношение к конкретной проблеме. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса;

1. Ответить на вопросы плана практического занятия;
2. Выполнить домашнее задание;
3. Проработать тестовые задания и задачи;
4. При затруднениях сформулировать вопросы к преподавателю.

Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы практикума, выступать и участвовать в коллективном обсуждении вопросов изучаемой темы, правильно выполнять практические задания и иные задания, которые даются в фонде оценочных средств дисциплины.

4. Методические указания обучающимся по организации самостоятельной работы.

Цель организации самостоятельной работы по дисциплине «Серверная разработка Web-приложений» — это углубление и расширение знаний в области гуманитарных наук; формирование навыка и интереса к самостоятельной познавательной деятельности.

Самостоятельная работа обучающихся является важнейшим видом освоения содержания дисциплины, подготовки к практическим занятиям и к контрольной работе. Сюда же относятся и самостоятельное углубленное изучение тем дисциплины. Самостоятельная работа представляет собой постоянно действующую систему, основу образовательного процесса и носит исследовательский характер, что послужит в будущем основанием для написания выпускной квалификационной работы, практического применения полученных знаний.

Организация самостоятельной работы обучающихся ориентируется на активные методы овладения знаниями, развитие творческих способностей, переход от поточного к индивидуализированному обучению, с учетом потребностей и возможностей личности.

Правильная организация самостоятельных учебных занятий, их систематичность, целесообразное планирование рабочего времени позволяет студентам развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивать высокий уровень успеваемости в период обучения, получить навыки повышения профессионального уровня.

Подготовка к лабораторному занятию включает, кроме проработки конспекта и презентации лекции, поиск литературы (по рекомендованным спискам и самостоятельно), решение задач из перечня лабораторных работ (текущий контроль). Лабораторное занятие – выполнение поставленных перед

студентом задач с использованием ПК и специального программного обеспечения.

При подготовке к контрольной работе обучающийся должен повторять пройденный материал в строгом соответствии с учебной программой, используя конспект лекций и литературу, рекомендованную преподавателем. При необходимости можно обратиться за консультацией и методической помощью к преподавателю.

Самостоятельная работа реализуется:

- непосредственно в процессе аудиторных занятий – на лекциях, практических занятиях;
- в контакте с преподавателем вне рамок расписания – на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.
- в библиотеке, дома, на кафедре при выполнении обучающимся учебных и практических задач.

Виды СРС и критерии оценок (по балльно–рейтинговой системе ГГНТУ, СРС оценивается в 15 баллов)

1. Реферат/Доклад/Презентация
2. Индивидуальное задание

Темы для самостоятельной работы прописаны в рабочей программе дисциплины. Эффективным средством осуществления обучающимся самостоятельной работы является электронная информационно–образовательная среда университета, которая обеспечивает доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем.

Составитель:

Ассистент каф. «ИСЭ»



/Ахматсулатнов А.Ю./

СОГЛАСОВАНО:

Зав. выпускающей каф. «ИСЭ»



/Магомаева Л.Р./

Директор ДУМР



/Магомаева М.А./