Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 03.10.2023 14:45:59
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДЕНО
Декан факультета
Информационных технологий
_____ / Д.Г. Демидов /
«16» 02 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«BackEnd-разработка»

Направление подготовки/специальность
09.03.02 Информационные системы и технологии

Профиль/специализация
Программное обеспечение игровой компьютерной индустрии

Квалификация
Бакалавр

Формы обучения
Очная

Москва, 2023 г.

Разработчик(и):
Доцент кафедры
«Информатика и информационные технологии»  / П. С. Новиков /

Согласовано:
Заведующий кафедрой
«Информатика и информационные технологии»,
к.т.н.  / Е.В. Булатников /

Содержание

1. Цели, задачи и планируемые результаты обучения по дисциплине 4
2. Место дисциплины в структуре образовательной программы 4
3. Структура и содержание дисциплины 4
   3.1. Виды учебной работы и трудоемкость 5
   3.2. Тематический план изучения дисциплины 5
   3.3. Содержание дисциплины 8
   3.4. Тематика семинарских/практических и лабораторных занятий 13
   3.5. Тематика курсовых проектов (курсовых работ) 16
4. Учебно-методическое и информационное обеспечение 17
   4.1. Нормативные документы и ГОСТы 17
   4.2. Основная литература 17
   4.3. Дополнительная литература 18
   4.4. Электронные образовательные ресурсы 18
   4.5. Лицензионное и свободно распространяемое программное обеспечение 18
   4.6. Современные профессиональные базы данных и информационные справочные системы 18
5. Материально-техническое обеспечение 18
6. Методические рекомендации 18
   6.1. Методические рекомендации для преподавателя по организации обучения 18
   6.2. Методические указания для обучающихся по освоению дисциплины 19
7. Фонд оценочных средств 19
   7.1. Методы контроля и оценивания результатов обучения 19
   7.2. Шкала и критерии оценивания результатов обучения 19
   7.3. Оценочные средства 20

1. Цели, задачи и планируемые результаты обучения по дисциплине

Целью освоения дисциплины «BackEnd-разработка» является формирование понимания идеологии и ключевых аспектов разработки внутренней и вычислительной логики веб-сайтатов или веб-приложения, а также иного программного обеспечения и информационных систем.

К основным **задачам** освоения дисциплины следует отнести:

- изучение построения внутренней структуры веб-приложений
- изучения основных методик, подходов и библиотек для построения внутренней структуры веб-приложения на языка c# и pyton

Обучение по дисциплине «BackEnd-разработка» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ПК-2. Способен выполнять работы и управлять работами по созданию (модификации) и сопровождению ИС	ИПК-2.1. Знает способы управления работами по созданию и обслуживанию ИС в игровой компьютерной индустрии ИПК-2.2. Умеет управлять работами по разработке и обслуживанию ИС в игровой компьютерной индустрии ИПК-2.3. Имеет навыки: применения программного обеспечения для управления работами по разработке ИС в игровой компьютерной индустрии
ПК-4. Способен проводить интеграцию программных модулей и компонент и верификацию выпусков программного продукта	ИПК-4.1. Знает методы и способы интеграции программных модулей и компонент и верификацию выпусков в проектах игровой компьютерной индустрии ИПК-4.2. Умеет проводить интеграцию программных модулей и компонент и верификацию выпусков проекта игровой компьютерной индустрии ИПК-4.3. Имеет навыки применения программного обеспечения для верификации версий проекта игровой компьютерной индустрии до получения готового продукта и последующей его поддержки

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к модулю «Специальное программирование» обязательной части Блока 1. Дисциплины (модули) учебного плана программы бакалавриата.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

- Проектирование и дизайн медиаприложений;
- Функциональное программирование;
- Производственная практика (преддипломная);
- Выполнение и защита выпускной квалификационной работы.

3. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 8 зачетных единиц (288 часов).

3.1. Виды учебной работы и трудоемкость (по формам обучения)

3.1.1. Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			4	5
1	**Аудиторные занятия**	**144**	**72**	**72**
	В том числе:			
1.1	Лекции	72	36	36
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	72	36	36
2	**Самостоятельная работа**	**144**	**72**	**72**
	В том числе:			
2.1	Подготовка и выполнение лабораторных работ	144	72	72
3	**Курсовое проектирование**			**КП**
3	**Промежуточная аттестация**			
	Зачет		зачет	
	Экзамен			экзамен
	Итого:	**288**	**144**	**144**

3.2. Тематический план изучения дисциплины (по формам обучения)

3.2.1. Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/практические занятия	Лабораторные занятия	Практическая подготовка	
1.1	Тема 1. Введение в ASP.NET Core	4	2				2
2.1	Тема 2. Основы в ASP.NET Core	4	2				2
2.2	Лабораторная работа № 1 «Создание приложения на основе класса WebApplication на основе ASP.NET Core»	8			4		4
3.1	Тема 3. Внедренные зависимости в ASP.NET Core	4	2				2
3.2	Лабораторная работа № 2 «Создание консольного приложения с внедренными зависимостями на основе ASP.NET Core»	4			2		2
4.1	Тема 4. Маршрутизация в ASP.NET Core	4	2				2

4.2	Лабораторная работа № 3 «Добавление в веб-приложение на основе ASP.NET Core маршрутизации»	4			2		2
5.1	Тема 5. Статические файлы в ASP.NET Core	4	2				2
5.2	Лабораторная работа № 5 «Добавление в веб-приложение на основе ASP.NET Core статических файлов»	4			2		2
6.1	Тема 6. Конфигурация в ASP.NET Core	4	2				2
6.2	Лабораторная работа № 6 «Конфигурация веб-приложение на основе ASP.NET Core»	4			2		2
7.1	Тема 7. Логгирование в ASP.NET Core	4	2				2
7.2	Лабораторная работа № 7 «Логирование веб-приложение на основе ASP.NET Core»	4			2		2
8.1	Тема 8. Состояние приложения в ASP.NET Core	4	2				2
8.2	Лабораторная работа № 8 «Изучение состояний в веб-приложение на основе ASP.NET Core»	4			2		2
9.1	Тема 9. Обработка ошибок в ASP.NET Core	4	2				2
9.2	Лабораторная работа № 9 «Обработка ошибок в веб-приложении на основе ASP.NET Core»	4			2		2
10.1	Тема 10. Results API в ASP.NET Core	4	2				2
10.2	Лабораторная работа № 10 «Изучение выдачи ответов в веб-приложении на основе ASP.NET Core»	4			2		2
11.1	Тема 11. Web API в ASP.NET Core	4	2				2
11.2	Лабораторная работа № 11 «Создание веб-API приложения на основе ASP.NET Core»	4			2		2
12.1	Тема 12. Работа с базой данных и Entity Framework в ASP.NET Core	4	2				2
12.2	Лабораторная работа № 12 «Работа с базами данных в приложении на основе ASP.NET Core»	4			2		2
13.1	Тема 13. Аутентификация и авторизация в ASP.NET Core	4	2				2
13.2	Лабораторная работа № 13 «Добавление авторизации и аутентификации в веб-приложение на основе ASP.NET Core»	4			2		2
14.1	Тема 14. CORS и кросс-доменные запросы в ASP.NET Core	4	2				2
14.2	Лабораторная работа № 14 «Работа с кросс-доменными запросами веб-	4			2		2

	приложение на основе ASP.NET Core»						
15.1	Тема 15. URL Rewriting в ASP.NET Core	4	2				2
15.2	Лабораторная работа № 15 «Контроль доступа к адресам в веб-приложение на основе ASP.NET Core»	4			2		2
16.1	Тема 16. Клиентская разработка в ASP.NET Core	4	2				2
16.2	Лабораторная работа № 16 «Добавление скриптов JS и стилей CSS в веб-приложение, для использование на стороне фронтенда, на основе ASP.NET Core»	4			2		2
17.1	Тема 17. Кэширование в ASP.NET Core	4	2				2
17.2	Лабораторная работа № 17 «Изучение кеширования в веб-приложении на основе ASP.NET Core»	4			2		2
18.1	Тема 18. Мониторинг работоспособности приложения в ASP.NET Core	4	2				2
18.2	Лабораторная работа № 18 «Мониторинг веб-приложения на основе ASP.NET Core»	4			2		2
19.1	Тема 19. Введение в Python	4	2				2
19.2	Лабораторная работа № 19 «Введение в язык Python»	4			2		2
20.1	Тема 20. Основы Python	4	2				2
20.2	Лабораторная работа № 20 «Основы языка Python»	4			2		2
21.1	Тема 21. Объектно-ориентированное программирование в Python	4	2				2
21.2	Лабораторная работа № 21 «Объектно-ориентированное программирование на языке Python»	4			2		2
22.1	Тема 22. Обработка ошибок и исключений в Python	4	2				2
23.1	Тема 23. Списки, кортежи и словари в Python	4	2				2
24.1	Тема 24. Модули в Python	4	2				2
25.1	Тема 25. Строки в Python	4	2				2
26.1	Тема 26. Pattern matching в Python	4	2				2
27.1	Тема 27. Работа с файлами в Python	4	2				2
28.1	Тема 28. Работа с датами и временем в Python	4	2				2
29.1	Тема 29. Основы FastAPI	4	2				2
29.2	Лабораторная работа № 22 «Введение в библиотеку FastAPI»	4			2		2
29.3	Лабораторная работа № 23 «Создание веб-приложения с использованием FastAPI»	4			2		2

30.1	Тема 30. Работа с базой данных через SQLAlchemy	4	2				2
30.2	Лабораторная работа № 24 «Работа с базами данных с использованием библиотеки SQLAlchemy»	4			2		2
31.1	Тема 31. Введение в Django	8	4				4
31.2	Лабораторная работа № 25 «Введение в библиотеку Django»	4			2		2
31.3	Лабораторная работа № 26 «Создание веб-приложения на основе библиотеки Django»	4			2		2
31.4	Лабораторная работа № 27 «Добавления представлений в веб-приложение на основе библиотеки Django»	4			2		2
32.1	Тема 32. Представления и маршрутизация в Django	8	4				4
32.2	Лабораторная работа № 28 «Добавления маршрутизации на в веб-приложение основе библиотеки Django»	8			4		4
32.3	Лабораторная работа № 29 «Авторизация в веб-приложении основе библиотеки Django»	8			4		4
32.4	абораторная работа № 30 «Использование шаблонов в веб-приложении на основе библиотеки Django»	8			4		4
34.1	Тема 34. Модели в Django	6	4				2
34.2	Лабораторная работа № 31 «Работа с базами данных в веб-приложении на основе библиотеки Django»	8			4		4
34.3	Лабораторная работа № 32 «Создание CRUD API в веб-приложении на основе библиотеки Django»	6			2		4
Итого		**288**	**72**		**72**		**144**

3.3. Содержание дисциплины

Тема 1. Введение в ASP.NET Core

- Введение в ASP.NET Core
- Способы создания приложений
- Среду разработки и их настройка

Тема 2. Основы в ASP.NET Core

- Создание и запуск приложения. WebApplication и WebApplicationBuilder
- Конвейер обработки запроса и middleware
- Метод Run и определение терминального middleware
- HttpResponse. Отправка ответа
- HttpRequest. Получение данных запроса
- Отправка файлов
- Отправка форм
- Переадресация
- Отправка и получение json

- Создание простейшего API
- Загрузка файлов на сервер
- Метод Use
- Создание ветки конвейера. UseWhen и MapWhen
- Метод Map
- Классы middleware
- Построение конвейера обработки запроса
- IWebHostEnvironment и окружение

Тема 3. Внедренные зависимости в ASP.NET Core

- Внедрение зависимостей и IServiceCollection
- Создание сервисов
- Получение зависимостей
- Жизненный цикл зависимостей
- Применение сервисов в классах middleware
- Scoped-сервисы в singleton-объектах
- Множественная регистрация сервисов

Тема 4. Маршрутизация в ASP.NET Core

- Конечные точки. Метод Map
- Параметры маршрута
- Ограничения маршрутов
- Создание ограничений маршрутов
- Передача зависимостей в конечные точки
- Сопоставление запроса с конечной точкой
- Сочетание конечных точек с другими middleware
- Получение параметров строки запроса

Тема 5. Статические файлы в ASP.NET Core

- Установка каталога статических файлов. UseStaticFiles
- Работа со статическими файлами

Тема 6. Конфигурация в ASP.NET Core

- Основы конфигурации
- Нефайловые провайдеры конфигурации
- Конфигурация в файлах JSON, XML и Ini
- Конфигурация по умолчанию и объединение конфигураций
- Анализ конфигурации
- Создание провайдера конфгурации
- Проекция конфигурации на классы
- Передача конфигурации через IOptions

Тема 7. Логгирование в ASP.NET Core

- Ведение лога и ILogger
- Фабрика логгера и провайдеры логгирования
- Конфигурация и фильтрация логгирования
- Создание провайдера логгирования

Тема 8. Состояние приложения в ASP.NET Core

- HttpContext.Items
- Куки
- Сессии

Тема 9. Обработка ошибок в ASP.NET Core

- Обработка исключений
- Обработка ошибок HTTP

Тема 10. Results API в ASP.NET Core

- Введение в Results API
- Отправка текста и json в Results API
- Переадресация в Results API
- Отправка статусных кодов в Results API
- Отправка файлов в Results API
- Определение своего типа IResult

Тема 11. Web API в ASP.NET Core

- Пример приложения Web API

Тема 12. Работа с базой данных и Entity Framework в ASP.NET Core

- Подключение Entity Framework
- Основные операции с данными в Entity Framework Core

Тема 13. Аутентификация и авторизация в ASP.NET Core

- Введение в аутентификацию и авторизацию
- Аутентификация с помощью JWT-токенов
- Авторизация с помощью JWT-токенов в клиенте JavaScript
- Аутентификация с помощью куки
- HttpContext.User, ClaimPrincipal и ClaimsIdentity
- ClaimPrincipal и объекты Claim
- Авторизация по ролям
- Авторизация на основе Claims
- Создание ограничений для авторизации

Тема 14. CORS и кросс-доменные запросы в ASP.NET Core

- Подключение CORS в приложении
- Конфигурация CORS
- Политики CORS
- Глобальная и локальная настройка CORS

Тема 15. URL Rewriting в ASP.NET Core

- Введение в URL Rewriting
- Правила IIS для URL Rewriting
- Применение правил Apache для URL Rewriting
- Создание правил URL Rewriting

Тема 16. Клиентская разработка в ASP.NET Core

- Бандлинг и минификация
- Пакетный менеджер Libman
- Пакетный менеджер NPM

Тема 17. Кэширование в ASP.NET Core

- Кэширование с помощью MemoryCache
- Распределенное кэширование. Redis
- Сжатие ответа
- Кэширование статических файлов
- Кэширование ответа и OutputCache

Тема 18. Мониторинг работоспособности приложения в ASP.NET Core

- Health Check Middleware

Тема 19. Введение в Python

- Язык программирования Python
- Установка Python и первая программа на Windows
- Установка Python и первая программа на MacOS
- Установка Python и первая программа на Linux

- Управление версиями Python на Windows, MacOS и Linux
- Первая программа в PyCharm
- Python в Visual Studio

Тема 20. Основы Python

- Введение в написание программ
- Переменные и типы данных
- Консольный ввод и вывод
- Арифметические операции с числами
- Поразрядные операции с числами
- Условные выражения
- Условная конструкция if
- Циклы
- Функции
- Параметры функции
- Оператор return и возвращение результата из функции
- Функция как тип, параметр и результат другой функции
- Лямбда-выражения
- Преобразование типов
- Область видимости переменных
- Замыкания
- Декораторы

Тема 21. Объектно-ориентированное программирование в Python

- Классы и объекты
- Инкапсуляция, атрибуты и свойства
- Наследование
- Переопределение функционала базового класса
- Атрибуты классов и статические методы
- Класс object. Строковое представление объекта

Тема 22. Обработка ошибок и исключений в Python

- Конструкция try...except...finally
- except и обработка разных типов исключений
- Генерация исключений и создание своих типов исключений

Тема 23. Списки, кортежи и словари в Python

- Списки
- Кортежи
- Диапазоны
- Словари
- Множества
- List comprehension
- Упаковка и распаковка
- Упаковка и распаковка в параметрах функций

Тема 24. Модули в Python

- Определение и подключение модулей
- Генерация байткода модулей
- Модуль random
- Модуль math
- Модуль locale
- Модуль decimal
- Модуль dataclasses. Data-классы

Тема 25. Строки в Python

- Работа со строками
- Основные методы строк
- Форматирование

Тема 26. Pattern matching в Python

- Конструкция match
- Кортежи в pattern matching
- Массивы в pattern matching
- Словари в pattern matching
- Классы в pattern matching
- guards или ограничения шаблонов
- Установка псевдонимов и паттерн AS

Тема 27. Работа с файлами в Python

- Открытие и закрытие файлов
- Текстовые файлы
- Файлы CSV
- Бинарные файлы
- Модуль shelve
- Модуль OS и работа с файловой системой
- Программа подсчета слов
- Запись и чтение архивных zip-файлов

Тема 28. Работа с датами и временем в Python

- Модуль datetime
- Операции с датами

Тема 29. Основы FastAPI

- Что такое FastAPI
- Класс FastAPI и обработка запроса
- Отправка ответа
- Отправка файлов с сервера
- Параметры пути
- Параметры строки запроса
- Отправка статусных кодов
- Переадресация
- Статические файлы
- Получение данных запроса
- Получение данных запроса в виде объекта класса
- Создание простейшего API
- Отправка и получение заголовков
- Куки
- Отправка форм

Тема 30. Работа с базой данных через SQLAlchemy

- Подключение к базе данных и создание таблиц
- Взаимодействие с базой данных
- Базовые операции с базой данных в веб-приложении

Тема 31. Введение в Django

- Что такое Django
- Установка и настройка Django
- Создание первого проекта
- Создание первого приложения

Тема 32. Представления и маршрутизация в Django

- Обработка запроса
- Определение маршрутов и функции path и re_path
- Получение данных запроса. HttpRequest
- HttpResponse и отправка ответа
- Параметры представлений
- Вложенные маршруты и функция include
- Параметры строки запроса
- Переадресация и отправка статусных кодов
- Отправка json
- Отправка и получение кук
- Тема 3. Шаблоны
- Создание и использование шаблонов
- Передача данных в шаблоны
- Встроенные теги шаблонов
- Фильтры шаблонов
- Статические файлы
- TemplateView
- Конфигурация шаблонов
- Расширение шаблонов и фильтр extends
- Вложенные шаблоны и фильтр include

Тема 33. Работа с формами в Django

- Отправка форм
- Определение форм Django
- Типы полей формы
- Настройка формы и ее полей
- Валидация данных
- Детальная настройка полей формы
- Стилизация полей форм

Тема 34. Модели в Django

- Подключение к базе данных
- Создание моделей
- Типы полей моделей
- QuerySet API
- Создание и получение объектов модели
- Редактирование и удаление объектов модели
- Фильтрация
- values и values_list и сортировка
- Операции с множествами
- Получение отдельных объектов и проверка их наличия
- Агрегатные операции
- Выполнение SQL-выражений
- CRUD. Все базовые операции с моделями в веб-приложении
- Отношение один ко многим (One to Many)
- Практический пример связи один ко многим
- Отношение многие ко многим (Many to Many)
- Отношение многие ко многим (Many to Many)
- Отношение один к одному (One to one)

3.4. Тематика семинарских/практических и лабораторных занятий

3.4.1. Семинарские/практические занятия

Семинарские и практические занятия не предусмотрены.

3.4.2. Лабораторные занятия

Лабораторная работа № 1 «Создание приложения на основе класса WebApplication на основе ASP.NET Core»

В данной работе рассмотрено, создание веб-приложение на основе классов WebApplication и WebApplicationBuilder, конвейер обработки запросов, принятие и получение запроса, отправка файлов, форм, создание простейшего API, создание middleware.

Лабораторная работа № 2 «Создание консольного приложения с внедренными зависимостями на основе ASP.NET Core»

В данной работе рассмотрена работа с внедренными зависимости и механизм взаимодействия в приложении слабосвязанных объектов.

Лабораторная работа № 3 «Добавление в веб-приложение на основе ASP.NET Core маршрутизации»

В работе рассмотрена маршрутизация внутри веб-приложения и способы добавления и управления маршрутами.

Лабораторная работа № 5 «Добавление в веб-приложение на основе ASP.NET Core статических файлов»

В работе рассмотрена отправка из веб-приложения статических файлов.

Лабораторная работа № 6 «Конфигурация веб-приложение на основе ASP.NET Core»

В работе рассмотрены способы конфигурирования веб-приложения

Лабораторная работа № 7 «Логирование веб-приложение на основе ASP.NET Core»

В работе рассмотрены способы логирования веб-приложения

Лабораторная работа № 8 «Изучение состояний в веб-приложение на основе ASP.NET Core»

В работе рассмотрены способы конфигурирования веб-приложения

Лабораторная работа № 9 «Обработка ошибок в веб-приложении на основе ASP.NET Core»

В работе рассмотрены состояния веб-приложения, работа с сессиями и хуками

Лабораторная работа № 10 «Изучение выдачи ответов в веб-приложении на основе ASP.NET Core»

В работе рассмотрена работа с Result Api

Лабораторная работа № 11 «Создание веб-API приложения на основе ASP.NET Core»

В работе рассмотрено создания Rest API на основе веб-приложения.

Лабораторная работа № 12 «Работа с базами данных в приложении на основе ASP.NET Core»

В работе рассмотрена работа с базами данными и основы Entity Framework

Лабораторная работа № 13 «Добавление авторизации и аутентификации в веб-приложение на основе ASP.NET Core»

В работе рассмотрена авторизации и аутентификации в веб-приложении на основе ASP.NET Identity.

Лабораторная работа № 14 «Работа с кросс-доменными запросами веб-приложение на основе ASP.NET Core»

В работе рассмотрена работа с CORS и принципы его конфигурирования

Лабораторная работа № 15 «Контроль доступа к адресам в веб-приложение на основе ASP.NET Core»

Лабораторная работа № 16 «Добавление скриптов JS и стилей CSS в веб-приложение, для использование на стороне фронтенда, на основе ASP.NET Core»

Лабораторная работа № 17 «Изучение кеширования в веб-приложении на основе ASP.NET Core»

В работе рассмотрено кеширование веб-приложения

Лабораторная работа № 18 «Мониторинг веб-приложения на основе ASP.NET Core»

В работе рассмотрен мониторинг веб-приложения

Лабораторная работа № 19 «Введение в язык Python»

В работе рассмотрено создание приложения на языке Pyton, настрока среды для его работы, подключение библиотек.

Лабораторная работа № 20 «Основы языка Python»

В работе рассмотрены основные принципа написания программ на языке Pyton, а так же их запуск и отладка.

Лабораторная работа № 21 «Объектно-ориентированное программирование на языке Python»

В работе рассмотрено написание приложения на языке Pyton на основе ООП принципов.

Лабораторная работа № 22 «Введение в библиотеку FastAPI»

В работе рассмотрены основные особенности фреймворка FastAPI, его установка, настройка и возможные сценарии применения.

Лабораторная работа № 23 «Создание веб-приложения с использованием FastAPI»

В работе рассмотрено создание веб-приложения на языке Pyton с применением фреймворка FastAPI.

Лабораторная работа № 24 «Работа с базами данных с использованием библиотеки SQLAlchemy»

В работе рассмотрена работа с базами данных из веб-приложения на Pyton с применением фреймворка SQLAlchemy.

Лабораторная работа № 25 «Введение в библиотеку Django»

В работе рассмотрены основные возможности фреймворка Django и особенности построения веб-приложения на его основе.

Лабораторная работа № 26 «Создание веб-приложения на основе библиотеки Django»

В работе рассмотрено создание веб-приложения на фреймворке Django, его запуск и отладка.

Лабораторная работа № 27 «Добавления представлений в веб-приложение на основе библиотеки Django»

В работе рассмотрено добавление представлений в веб-приложения на фреймворке Django.

Лабораторная работа № 28 «Добавления маршрутизации на в веб-приложение основе библиотеки Django»

В работе рассмотрено добавление маршрутизации в веб-приложения на фреймворке Django.

Лабораторная работа № 29 «Авторизация в веб-приложении основе библиотеки Django»

В работе рассмотрено добавление авторизации в веб-приложения на фреймворке Django.

Лабораторная работа № 30 «Использование шаблонов в веб-приложении на основе библиотеки Django»

В работе рассмотрено использование шаблонов в веб-приложения на фреймворке Django.

Лабораторная работа № 31 «Работа с базами данных в веб-приложении на основе библиотеки Django»

В работе рассмотрена работа с базами данных в веб-приложения на фреймворке Django.

Лабораторная работа № 32 «Создание CRUD API в веб-приложении на основе библиотеки Django»

В работе рассмотрено создание RestAPI в веб-приложения на фреймворке Django.

3.5. Тематика курсовых проектов (курсовых работ)

Курсовой проект для дисциплины «BackEnd-разработка» предусматривает написание следующих заданий по тематикам:

1. Разработка RESTful API для онлайн-магазина. API должно обрабатывать заявки от клиентов, включая операции по созданию, чтению, обновлению и удалению данных из базы данных. Взаимодействие с базой данных должно осуществляться через ORM (Object-Relational Mapping) или SQL-запросы.

2. Создание веб-приложения для социальной сети. Приложение должно поддерживать функционал регистрации и авторизации пользователей, возможность создания, просмотра и комментирования постов, а также подписки на других пользователей. Веб-приложение должно запускаться на сервере и обеспечивать обмен данных между клиентом и сервером.

3. Разработка системы управления задачами (Task Management System). Система должна включать функционал создания, назначения, отслеживания и завершения задач. Пользователи должны иметь возможность создавать задачи, присваивать их другим пользователям, отслеживать прогресс выполнения и отмечать их как выполненные.

4. Создание веб-приложения для онлайн-бронирования номеров в отеле. Приложение должно предоставлять возможность просмотра свободных номеров, бронирования, отмены бронирования, а также отображения информации о занятости номеров. Реализация должна быть масштабируемой и безопасной, обеспечивая защиту данных пользователей.

5. Разработка сервиса для загрузки, хранения и обработки файлов. Сервис должен поддерживать возможность загрузки файлов на сервер, их хранение, а также предоставлять API для обработки и представления файлов пользователю (например, конвертация файлов между различными форматами или генерация превью-изображений).

6. Создание системы учета инвентаря для компании. Система должна поддерживать функционал добавления, удаления и редактирования записей об инвентаре, а также отслеживание его наличия и использования. Система также должна предоставлять отчеты о состоянии инвентаря и возможность автоматического уведомления о нехватке материалов.

7. Разработка платформы для онлайн-курсов. Платформа должна предоставлять функционал создания, управления и прохождения курсов, а также включать модуль оценки успеваемости студентов. Взаимодействие с базой данных должно быть реализовано с использованием ORM или SQL-запросов.

8. Создание API для мобильного приложения доставки пиццы. API должно поддерживать функционал создания и обновления заказов, отслеживать их статус, а также предоставлять информацию о меню и доступные способы оплаты. Реализация должна быть безопасной и масштабируемой, чтобы обеспечивать быструю и надежную работу приложения.

9. Разработка системы обмена сообщениями. Система должна позволять пользователям обмениваться сообщениями в режиме реального времени, поддерживать создание чатов, отправку и прием сообщений, а также возможность поиска и фильтрации сообщений. Система должна быть безопасной и масштабируемой, чтобы обеспечить эффективное общение между пользователями.

10. Создание веб-приложения для управления задачами и проектами в организации. Приложение должно поддерживать функционал создания и управления задачами, распределение ролей и прав доступа для пользователей, а также отслеживание прогресса выполнения проектов и задач.

4. Учебно-методическое и информационное обеспечение

4.1. Нормативные документы и ГОСТы

1. Федеральный закон от 29 декабря 2012 года No 273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);
2. Федеральный государственный образовательный стандарт высшего образования - бака-лавриат по направлению подготовки 09.03.02 Информационные системы и технологии, утвер-жденный Приказом Министерства образования и науки РФ от 19 сентября 2017 г. N 929 "Об утверждении федерального... Редакция с изменениями N 1456 от 26.11.2020;
3. Приказ Министерства образования и науки РФ от 05 апреля 2017 г. No 301 «Об утвер-ждении Порядка организации и осуществления образовательной деятельности по образователь-ным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры.

4.2. Основная литература

1. Документация по C#
2. C# Sharp 10 и .NET 6. Современная кросс-платформенная разработка : 16+ / Марк Прайс ; [перевел с английского С. Черников]. - 6-е изд. - Санкт-Петербург [и др.] : Питер, 2023. - 844, [1] с. : ил., портр., табл.; 23 см. - (Серия "Для профессионалов").; ISBN 978-5-4461-2249-3 : 500 экз.

3. Язык программирования C# 9 и платформа .NET 5: основные принципы и практики программирования / Эндрю Троелсен, Филипп Джепикс; перевод с английского и редакция Ю. Н. Артеменко. - 10-е изд. - Москва : Диалектика ; Санкт-Петербург : Диалектика, 2022. - 1391 с. : ил., табл.; 24 см. - (Профессионалам от профессионалов).; ISBN 978-5-907458-67-3 : 500 экз.
4. C# 9.0. Справочник : полное описание языка / Албахари Джозеф; перевод с английского и редакция Ю. Н. Артеменко. - Москва : Диалектика ; Санкт-Петербург : Диалектика, 2021. - 1056 с. : ил., табл.; 25 см.; ISBN 978-5-907365-81-0 : 500 экз.
5. Разработка веб-приложений с использованием Flask на языке Python [Текст] : Мигель-Гринберг ; пер. с анг. А. Н. Киселева. - Москва : ДМК Пресс, 2014. - 272 с. : ил.; ISBN 978-5-97060-138-9
6. Django 4. Практика создания веб-сайтов на Python / Владимир Дронов. - Санкт-Петербург : БХВ-Петербург, 2023. - 798 с. : ил.; 24 см. - (Профессиональное программирование (PRO)).; ISBN 978-5-9775-1774-4 : 1000 экз.

4.3. Дополнительная литература

1. Программирование на C#. Основные сведения [Текст] : [12+] / Васильев А. Н. - Москва : Эксмо, 2018. - 582, [1] с. : табл.; 24 см. - (Российский компьютерный бестселлер).; ISBN 978-5-04-092519-3 : 2000 экз.
2. C#. Программирование [Текст] : учебное пособие : [в 3 ч.] / Н. А. Тюкачев, В. Г. Хлебостроев ; М-во образования и науки Российской Федерации, Федеральное гос. бюджетное образовательное учреждение высш. проф. образования "Воронежский гос. ун-т". - Воронеж : Изд.-полиграфический центр Воронежского гос. ун-та, 2013. - 21 см. - (Учебник Воронежского государственного университета).

4.4. Электронные образовательные ресурсы

ЭОР не предусмотрен.

4.5. Лицензионное и свободно распространяемое программное обеспечение

1. Visual Studio
2. Visual Studio Code

4.6. Современные профессиональные базы данных и информационные справочные системы

Не предусмотрены.

5. Материально-техническое обеспечение

Компьютерные классы со следующей оснащенностью: столы, стулья, аудиторная доска, использование переносного мультимедийного комплекса (переносной проектор, персональный ноутбук). Персональные компьютеры, мониторы, мышки, клавиатуры. Рабочее место преподавателя: стол, стул.

Программное обеспечение: Microsoft Windows или Linux на основе deb-пакетов (Debian, Ubuntu, Astra и т.д.), сервер с системой контроля версий GIT (GitLab)

6. Методические рекомендации

6.1. Методические рекомендации для преподавателя по организации обучения

Методика преподавания дисциплины «BackEnd-разработка» предусматривает использование онлайн-курса в системе дистанционного обучения Университета, групповых и индивидуальных консультаций обучающихся, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Лабораторные работы по дисциплине «BackEnd-разработка» осуществляются в форме самостоятельной проработки теоретического материала обучающимися; выполнения практического задания; защиты преподавателю лабораторной работы (знание теоретического материала и выполнение практического задания по теме лабораторной работы).

6.2. Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в соответствии с учебным планом.

На занятиях осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на умение применять полученные знания на практике, в том числе при решении реальных задач, отличающихся от проработанных.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, самостоятельно знакомятся с теоретическим материалом, выполняют лабораторные работы, готовятся к текущему контролю и промежуточной аттестации.

Выполнение и сдача лабораторных работ проводится с применением системы контроля версий GIT.

Текущий контроль осуществляется на аудиторных занятиях в виде защиты лабораторных работ. Критериями оценки результатов являются:

– уровень освоения теоретического материала;

– уровень владения практическими навыками (в виде вопросов по процессу выполнения лабораторных работ);

– умения обучающегося использовать теоретические знания при выполнении практических задач (в виде дополнительных заданий);

– сформированность компетенций;

– оформление материала в соответствии с требованиями.

Промежуточный контроль осуществляется на зачете в форме тестирования в системе дистанционного обучения Университета, включающего вопросы на знание практической части построения внутренней структуры веб-приложений на основе языков C# и Pyton.

7. Фонд оценочных средств

7.1. Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций: **лабораторные работы**, **зачёт, экзамен и курсовой проект.**

7.2. Шкала и критерии оценивания результатов обучения

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «BackEnd-разработка».

7.2.1. Критерии оценки ответа на зачёте

(формирование компетенций — ПК-2, ПК-4)

«Зачтено»:

Выполнены все виды учебной работы, предусмотренные учебным планом. Обучающийся демонстрирует прочные теоретические знания, практические навыки, владеет терминами, делает аргументированные выводы и обобщения, приводит примеры, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации, которые обучающийся может исправить при коррекции преподавателем.

«Не зачтено»:

Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Обучающийся демонстрирует незнание теоретических основ предмета, отсутствие практических навыков, не умеет делать аргументированные выводы и приводить примеры, не владеет терминами, проявляет отсутствие логичности и последовательности изложения, делает ошибки, которые не может исправить даже при коррекции преподавателем, отказывается отвечать на дополнительные вопросы, допускает значительные ошибки, испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7.2.2. Критерии оценки работы обучающегося на лабораторных занятиях:

(формирование компетенций — ПК-2, ПК-4)

«5» (отлично): выполнены все практические задания, предусмотренные лабораторными работами, обучающийся четко и без ошибок ответил на все контрольные вопросы, проявил творческий подход при выполнении заданий, смог выполнить дополнительные задания.

«4» (хорошо): выполнены все практические задания, предусмотренные лабораторными работами, обучающийся с корректирующими замечаниями преподавателя ответил на все контрольные вопросы, проявил творческий подход при выполнении заданий, смог частично выполнить дополнительные задания.

«3» (удовлетворительно): выполнены все практические задания, предусмотренные лабораторными работами, с замечаниями преподавателя; обучающийся ответил на все контрольные вопросы с замечаниями, дополнительные задания выполнены с замечаниями.

«2» (неудовлетворительно): обучающийся не выполнил или выполнил неправильно практические задания, предусмотренные лабораторными работами, обучающийся ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы, дополнительные задания выполнены неверно или не выполнены.

1.1.1. Критерии оценки ответа на экзамене:

(формирование компетенций — ПК-2, ПК-4)

«5» (отлично): выполнены все практические задания, обучающийся четко и без ошибок ответил на все контрольные вопросы, проявил творческий подход при выполнении заданий, смог выполнить дополнительные задания.

«4» (хорошо): выполнены все практические задания,обучающийся с корректирующими замечаниями преподавателя ответил на все контрольные вопросы, проявил творческий подход при выполнении заданий, смог частично выполнить дополнительные задания.

«3» (удовлетворительно): выполнены все практические задания, с замечаниями преподавателя; обучающийся ответил на все контрольные вопросы с замечаниями, дополнительные задания выполнены с замечаниями.

«2» (неудовлетворительно): обучающийся не выполнил или выполнил неправильно практические задания, обучающийся ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы, дополнительные задания выполнены неверно или не выполнены.

7.3. Оценочные средства

1.1.2. Текущий контроль

Текущий контроль осуществляется на аудиторных занятиях в виде защиты лабораторных работ. Лабораторная работа – средство контроля усвоения учебного материала темы, раздела или разделов дисциплины, организованное как учебное занятие в виде демонстрации полученных навыков при решении поставленных практических задач.

Примеры вопросов к защите лабораторных работ, сдаче зачета, сдачи экзамена (оцениваемые компетенции — ПК-2, ПК-4).

1. Что такое внедренные зависимости? Как они применяются? Покажите на примере.
2. Как осуществляется конфигурирование приложения ASP.NET Core? Приведите несколько вариантов
3. Какие реализации DI контейнеров вы знаете на C#. Какой DI контейнер является лучшим?
4. LINQ lazy loading, eager loading в чем разница?
5. С помощью какого ключевого слова осуществляется блокировка одновременного выполнения определенных участков кода несколькими потоками?
6. Как осуществляется логирование в ASP.NET Core? Что такое фабрика логирования?
7. Для чего применяется Results API? Какие его возможности? Приведите примеры.
8. Что такое аутентификация с помощью JWT-токенов? Расскажите методику применения и реализацию в ASP.NET Core?
9. Расскажите для чего применяются HttpContext.User, ClaimPrincipal и ClaimsIdentity? Как их использовать? Приведите примеры.
10. Что такое CORS и кросс-доменные запросы? Как их обработка настраиваться в ASP.NET Core?
11. Что такое FASTAPI и чем он отличается от других веб-фреймворков Python?
12. Опишите процесс создания RESTful API с использованием FASTAPI.
13. Что такое внедрение зависимостей в FASTAPI и как это реализовано?
14. Как FASTAPI обрабатывает проверку запроса и сериализацию данных?
15. Каковы различные способы интеграции базы данных с FASTAPI?
16. Опишите архитектуру Django.
17. Что такое модели в Django?
18. Чем повторное использование кода Django лучше, чем у других фреймворков?
19. Что происходит, когда веб-сайт Django получает запрос?
20. Объясните структуру Django REST