

Документ подписан простой электронной подписью

Информация о документе:

ФИО: Хоружий Людмила Ивановна

Должность: Директор института экономики и управления АПК

Дата подписания: 2025.08.28 15:38:04

Уникальный идентификатор документа:

1e90b132d9b04dce67585160b015dddf2cb1e6a9



**МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ –  
МСХА имени К.А. ТИМИРЯЗЕВА»**

**(ФГБОУ ВО РГАУ - МСХА имени К.А. Тимирязева)**

Институт экономики и управления АПК  
Кафедра статистики и кибернетики

УТВЕРЖДАЮ:

Директор института  
экономики и управления АПК

Л.И. Хоружий



«28» августа 2025 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
Б1.В.14 Веб-разработка**

для подготовки бакалавров

ФГОС ВО

Направление: 09.03.02 «Информационные системы и технологии»

Направленность:

«Системная аналитика и разработка программного обеспечения»

Курс 3

Семестр 5

Форма обучения: очная

Год начала подготовки: 2025

Москва, 2025

Разработчики:

Демичев В.В., канд. экон. наук, доцент  
(ФИО, ученая степень, ученое звание)

  
(подпись)

«26» августа 2025 г.

Храмов Д.Э., ассистент

(ФИО, ученая степень, ученое звание)

  
(подпись)

«26» августа 2025 г.

Рецензент:

Вахрушева И.А., канд. пед. наук  
(ФИО, ученая степень, ученое звание)

  
(подпись)

«26» августа 2025 г.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии, профессионального стандарта и учебного плана 2025 года начала подготовки.

Программа обсуждена на заседании кафедры статистики и кибернетики. Протокол №11 от «26» августа 2025 г.

И. о. зав. кафедрой Уколова А.В., канд. экон. наук, доцент  
(ФИО, ученая степень, ученое звание)

  
(подпись)

«26» августа 2025 г.

**Согласовано:**

Председатель учебно-методической  
комиссии института экономики и управления АПК  
Гупалова Т.Н., канд. экон. наук, доцент  
(ФИО, ученая степень, ученое звание)

  
(подпись)

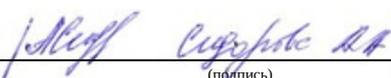
протокол №1 «28» августа 2025 г.

И. о. зав. выпускающей кафедрой  
статистики и кибернетики  
Уколова А.В., канд. экон. наук, доцент  
(ФИО, ученая степень, ученое звание)

  
(подпись)

«28» августа 2025 г.

Заведующий отделом комплектования ЦНБ

  
(подпись)

## Содержание

<b>АННОТАЦИЯ.....</b>	<b>4</b>
<b>1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....</b>	<b>5</b>
<b>2. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ.....</b>	<b>5</b>
<b>3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....</b>	<b>6</b>
<b>4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....</b>	<b>16</b>
4.1 РАСПРЕДЕЛЕНИЕ ТРУДОЁМКОСТИ ДИСЦИПЛИНЫ ПО ВИДАМ РАБОТ.....	16
ПО СЕМЕСТРАМ.....	16
4.2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	16
4.3 ПРАКТИЧЕСКИЕ ЗАНЯТИЯ.....	17
<b>5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ.....</b>	<b>22</b>
<b>6. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....</b>	<b>23</b>
6.1. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ.....	23
6.2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ КОНТРОЛЯ УСПЕВАЕМОСТИ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ.....	28
<b>7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....</b>	<b>29</b>
7.1 ОСНОВНАЯ ЛИТЕРАТУРА.....	29
7.2 ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....	29
<b>8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....</b>	<b>30</b>
<b>9. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ.....</b>	<b>31</b>
<b>10. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ.....</b>	<b>31</b>
<b>11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....</b>	<b>33</b>
Виды и формы отработки пропущенных занятий.....	33
<b>12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЯМ ПО ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....</b>	<b>33</b>

## АННОТАЦИЯ

**рабочей программы учебной дисциплины Б1.В.14 «Веб-разработка» для подготовки бакалавров по направлению 09.03.02 Информационные системы и технологии, направленности «Системная аналитика и разработка программного обеспечения»**

**Цель освоения дисциплины:** формирование у студентов профессиональных компетенций в области проектирования, реализации и развёртывания современных веб-приложений и сервисов с использованием актуальных фреймворков и архитектурных подходов на языке Python. В ходе изучения дисциплины студенты осваивают методы разработки как традиционных серверных приложений (на основе паттерна MVC/MVT), так и современных RESTful и асинхронных API, а также приобретают навыки обеспечения безопасности, аутентификации, управления состоянием и подготовки веб-проектов к промышленной эксплуатации.

**Место дисциплины в учебном плане:** дисциплина включена в часть, формируемую участниками образовательных отношений дисциплин учебного плана по направлению подготовки 09.03.02 Информационные системы и технологии.

**Требования к результатам освоения дисциплины:** в результате освоения дисциплины формируются следующие компетенции (индикаторы): ПКос-3(ПКос-3.1, ПКос-3.2, ПКос-3.3), ПКос-4(ПКос-4.1, ПКос-4.2, ПКос-4.3), ПКос-5(ПКос-5.1, ПКос-5.2), ПКос-7(ПКос-7.1, ПКос-7.2).

**Краткое содержание дисциплины:**

Архитектурные паттерны веб-приложений: MVC, MVT, клиент-сервер. Обзор современных Python-фреймворков: Django, Flask, FastAPI – сравнение, сценарии применения. Работа с маршрутизацией, параметрами URL и пространствами имён. Создание контроллеров-функций и контроллеров-классов. Шаблонизация и наследование шаблонов (Jinja2, Django Templates). Управление статическими и динамическими файлами. Формы: валидация, обработка, привязка к моделям, CSRF-защита. Аутентификация и авторизация: пользователи, группы, разграничение доступа. Сессии, cookie, всплывающие сообщения. Работа с базами данных, миграции, отношения моделей. Разработка RESTful API. Использование Django REST Framework и FastAPI для построения асинхронных сервисов. Обработка файлов, валидаторы, сигналы. Кэширование, промежуточное ПО (middleware), обработка ошибок. Тестирование веб-приложений. Подготовка к развёртыванию: настройка production-окружения, Gunicorn, Uvicorn, Nginx, управление переменными окружения. Публикация приложений в облачных средах.

**Общая трудоемкость дисциплины: 108 / 3 (часы/зач. ед.)**

**Промежуточный контроль: зачет с оценкой**

## **1. Цель освоения дисциплины**

Целью освоения дисциплины «Веб-разработка» является овладение студентами знаний архитектурных принципов построения веб-приложений, а также особенности клиент-серверного взаимодействия, основные подходы к проектированию RESTful и асинхронных API, возможности и сценарии применения современных Python-фреймворков – Django, Flask и FastAPI. Студент должен знать принципы маршрутизации, обработки HTTP-запросов и ответов, работы с формами, валидации данных, управления сессиями и аутентификации; понимать особенности работы с базами данных, механизмы миграций, сериализации, кэширования и защиты от уязвимостей (CSRF, XSS, SQL-инъекции); а также знать практики подготовки приложений к развёртыванию и эксплуатации в production-средах. По окончании изучения дисциплины студент должен уметь проектировать структуру веб-приложения в соответствии с поставленной задачей, создавать маршруты и контроллеры, реализовывать обработку пользовательского ввода через формы и API-запросы, интегрировать работу с базами данных, реализовывать механизмы разграничения доступа, обрабатывать загрузку файлов, управлять сессиями и cookie, а также настраивать ресурсы и готовить приложение к развёртыванию на сервере. Также по окончании изучения дисциплины студент должен владеть навыками разработки полнофункциональных веб-приложений на Django с использованием паттерна MVT, навыками создания лёгковесных API и микросервисов на Flask и асинхронных высокопроизводительных сервисов на FastAPI, методами тестирования веб-логики и API, техниками обеспечения безопасности веб-приложений, а также практиками документирования API (с использованием OpenAPI/Swagger) и оформления архитектурных решений в соответствии с профессиональными стандартами.

## **2. Место дисциплины в учебном процессе**

Дисциплина «Веб-разработка» включена в часть, формируемую участниками образовательных отношений дисциплин учебного плана. Дисциплина «Веб-разработка» реализуется в соответствии с требованиями ФГОС ВО, ОПОП ВО и Учебного плана по направлению 09.03.02 Информационные системы и технологии.

Дисциплина «Веб-разработка» изучается на третьем курсе образовательного цикла.

Предшествующими курсами, включенными в учебный план, на которых непосредственно базируются дисциплина «Веб-разработка», являются «Алгоритмизация и программирование», «Программирование на Python», «Теория информации», «Разработка профессиональных приложений».

Особенностью дисциплины является фокус на практико-ориентированном освоении современного Python-стека веб-разработки, включая фреймворки Django, Flask и FastAPI, с акцентом на проектирование архитектуры приложений, безопасность, работу с API и подготовку проектов к развёртыванию.

Рабочая программа дисциплины «Веб-разработка» для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

### **3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы**

Образовательные результаты освоения дисциплины обучающимся, представлены в таблице 1.

Таблица 1

## Требования к результатам освоения учебной дисциплины «Веб-разработка»

№ п/п	Код компетенции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
1.	ПКос-3	Способен проектировать и разрабатывать информационные ресурсы с использованием полного стека технологий	ПКос-3.1	Студент должен знать: архитектурные паттерны, лежащие в основе современных веб-приложений: клиент-сервер, MVC/MVT, REST, а также особенности синхронной и асинхронной обработки запросов; принципы проектирования RESTful API и структуры веб-проектов на Python-фреймворках; типовые решения для управления состоянием (сессии, cookie, токены), валидации данных, обработки ошибок и аутентификации; средства описания и документирования API (OpenAPI/Swagger), подходы к проектированию моделей данных, инструменты обеспечения		

			безопасности; принципы подготовки приложения к развёртыванию.	
		ПКос-3.2		Студент должен уметь: проектировать структуру веб-приложения в соответствии с поставленной задачей, выбирать подходящий фреймворк (Flask, Django, FastAPI), определять состав приложений/модулей, выделять маршруты и модели данных; разрабатывать как традиционные серверные приложения с шаблонами, так и современные API; проектировать и реализовывать модели данных, включая связи «один-ко-многим», валидацию и обработку сложных запросов; применять шаблоны проектирования; проектировать и реализовывать механизмы аутентификации, защиты от уязвимостей и логирования, а также адаптировать архитектуру под требования масштабируемости и

					развёртывания.	
			ПКос-3.3			Студент должен владеть: навыками проектирования и реализации полноценных информационных ресурсов на стеке Python от постановки задачи до развёртывания прототипа; практиками модульной архитектуры: разделение логики, изоляция зависимостей, использование конфигураций и переменных окружения; методами проектирования API; навыками проектирования структур данных с учётом будущей нагрузки и безопасности; техниками адаптации архитектуры под изменяющиеся требования: рефакторинг представлений, миграция данных.
2.	ПКос-	Способен осуществлять разработку, отладку и рефакторинг кода программного обеспечения, интеграцию программных модулей и компонент, в том числе взаимодействующих с внешней средой, средствами выбранных	ПКос-4.1	Студент должен знать: работу с контекстными менеджерами, декораторами, типизацией, асинхронными конструкциями <code>async/await</code> ); архитектурные особенности фреймворков Flask, Django и FastAPI,		

		языков программирования	<p>жизненный цикл запроса, механизмы маршрутизации, обработки форм, валидации данных и управления состоянием; структуру проекта, модульность (blueprints, приложения), разделение ответственностей; типичные источники ошибок в веб-приложениях; форматы диагностических сообщений фреймворков (трассировки Django, ошибки FastAPI, логи Flask) и принципы работы отладочных инструментов (отладчик IDE, Django Debug Toolbar, logging).</p>		
			ПКос-4.2	<p>Студент должен уметь: читать и интерпретировать код веб-приложений на Python, выявлять логические и архитектурные недочёты; писать чистый, читаемый и сопровождаемый код в соответствии с PEP 8 и лучшими практиками фреймворков; эффективно использовать среду разработки (PyCharm,</p>	

					<p>VS Code, Anaconda Nabigator) для навигации по коду, рефакторинга, запуска и отладки приложений;  локализовать и устранять ошибки;  анализировать логи и сообщения об ошибках, понимать стек-трейсы, воспроизводить дефекты в контролируемой среде;  интегрировать компоненты, подключать внешние API, обрабатывать файлы, взаимодействовать с базой данных, управлять сессиями и cookie.</p>	
			ПКос-4.3			<p>Студент должен владеть:  навыками разработки, отладки и рефакторинга веб-приложений на Python в условиях реальных проектных требований;  техниками повышения читаемости и поддерживаемости кода:  именование, документирование, выделение функций и классов, использование типов;  практиками отладки на всех уровнях;  навыками работы с инструментами анализа кода: flake8, mypy, pylint, а также отладчиком и логированием;</p>

						оптимизировать код с учётом производительности и безопасности.
3.	ПКос-	Способен осуществлять методическое сопровождение испытаний системы	ПКос-5.1	Студент должен знать: виды и уровни тестирования, применяемые в веб-разработке: unit-тестирование компонентов (моделей, сервисов), интеграционное тестирование (представлений, API-эндпоинтов), функциональное тестирование пользовательских сценариев; особенности тестирования в контексте разных фреймворков (TestCase и Client в Django, pytest и test_client во Flask, TestClient в FastAPI); принципы проектирования тестовых сценариев; основы управления изменениями в требованиях.		
			ПКос-5.2		Студент должен уметь: разрабатывать и выполнять	

					<p>ручные тестовые сценарии для веб-приложений;  проводить функциональное тестирование REST API с использованием инструментов и программных клиентов;  анализировать влияние изменений в коде на работоспособность функционала и определять, какие тестовые сценарии требуют перепроверки;  демонстрировать работоспособность приложения: запускать его в тестовом окружении, воспроизводить ключевые пользовательские сценарии, пояснять логику работы и показывать обработку ошибок.</p>	
4.	ПКос-	Способен осуществлять концептуально-логическое проектирование системы, разрабатывать техническое задание	ПКос-7.1	<p>Студент должен знать:  подходы к анализу предметной области веб-проекта;  методы декомпозиции задачи на модули в архитектуре веб-приложения, выделение маршрутов, моделей данных, сервисов, контроллеров и слоёв ответственности;</p>		

				<p>принципы оценки качества веб-систем, критерии производительности, надёжности, безопасности и сопровождаемости; назначение и структуру технического задания на веб-приложение; общепринятые практики документирования API (OpenAPI/Swagger) и архитектурных решений.</p>	
			ПКос-7.2	<p>Студент должен уметь:  формулировать цели веб-проекта на основе анализа реальной проблемы и выявлять потребности пользователей;  разрабатывать техническое задание на веб-приложение: описывать функциональные требования, нефункциональные, а также ограничения;  декомпозировать функции системы на подфункции;  алгоритмизировать бизнес-процессы в виде последовательности HTTP-взаимодействий, валидаций и обработки данных, а также обосновывать выбор архитектурного стека (Flask</p>	

					vs Django vs FastAPI) исходя из требований проекта.	
--	--	--	--	--	---	--

## 4. Структура и содержание дисциплины

### 4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 3 зач.ед. (108 часов), их распределение по видам работ и семестрам представлено в таблице 2.

Таблица 2

#### Распределение трудоёмкости дисциплины по видам работ

Вид учебной работы	Трудоёмкость
	час. всего
<b>Общая трудоёмкость</b> дисциплины по учебному плану	<b>108</b>
<b>1. Контактная работа:</b>	<b>50,35</b>
<b>Аудиторная работа</b>	<b>50</b>
лекции (Л)	16
практические занятия (ПЗ)	34
контактная работа на промежуточном контроле (КРА)	0,35
<b>2. Самостоятельная работа (СРС)</b>	<b>57,65</b>
самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к практическим занятиям)	57,65
Вид промежуточного контроля:	Зачет с оценкой

### 4.2 Содержание дисциплины

Таблица 3

#### Тематический план учебной дисциплины

Наименование разделов и тем дисциплин (укрупнённо)	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ всего	ПКР	
Тема 1. «Введение в веб-разработку и архитектура современных приложений»	16	2	4		10
Тема 2 «Разработка на Flask: от простого приложения до продвинутых паттернов»	23	3	8		12
Тема 3 «Продвинутое приложение на Django»	26	4	10		12
Тема 4 «Современные API с FastAPI и асинхронностью»	24	4	8		12
Тема 5 «Тестирование, безопасность и развёртывание веб-приложений»	18,65	3	4		11,65
контактная работа на промежуточном контроле (КРА)	0,35	-	-	0,35	-
<b>Итого по дисциплине</b>	<b>108</b>	<b>16</b>	<b>34</b>	<b>0,35</b>	<b>57,65</b>

#### Введение в веб-разработку и архитектура современных приложений.

Основы клиент-серверного взаимодействия, протокол HTTP, принципы REST, сравнение архитектурных подходов (MVC, MVT), обзор современных Python-фреймворков – Django, Flask, FastAPI – их назначение, сильные стороны

и типичные сценарии применения, выбор стека под задачу, структура веб-проекта, организация работы с зависимостями и виртуальными окружениями.

### **Разработка на Flask: от простого приложения до продвинутых паттернов.**

Создание маршрутов и обработка HTTP-запросов, шаблонизация с использованием Jinja2, работа с формами и валидацией, управление сессиями и cookie, организация кода с помощью blueprints, middleware, обработка ошибок, базовая аутентификация, подготовка приложения к тестированию и развёртыванию.

### **Продвинутая разработка на Django.**

Углублённое использование паттерна MTV: классы-контроллеры (CBV), кастомные формы и валидаторы, работа с ORM – отношения, агрегации, аннотации, сигналы, middleware, расширенная настройка админки, управление правами доступа и группами пользователей, работа с файлами, кэширование, обработка исключений и логирование.

### **Создание современных API с FastAPI.**

Разработка асинхронных RESTful-сервисов, валидация данных с помощью Pydantic, автоматическая генерация документации (Swagger/OpenAPI), аутентификация по токенам и OAuth2, интеграция с базами данных (SQLAlchemy, TortoiseORM), обработка фоновых задач, обработка ошибок и кастомные ответы, тестирование API.

### **Тестирование, безопасность и развёртывание веб-приложений.**

Методы тестирования: unit- и интеграционные тесты для вьюх, форм и API, использование pytest и встроенных средств фреймворков, защита от уязвимостей (XSS, CSRF, SQL-инъекции, CORS), настройка production-окружения – Gunicorn, Uvicorn, Nginx, управление переменными окружения, базовое использование Docker, развёртывание на облачных платформах (Render, Railway), мониторинг и логирование в production.

## **4.3 Практические занятия**

Таблица 4

### **Содержание практических занятий и контрольные мероприятия**

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
Тема 1. «Введение в веб-разработку и архитектура современных приложений»	Лекция 1 «Основы веб-разработки и выбор стека технологий на Python»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	-	2

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
	Практическая работа 1 «Настройка среды разработки и создание первого веб-приложения»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	Устный опрос	2
	Практическая работа 2 «Сравнительный анализ фреймворков: создание аналогичного API на Flask и FastAPI»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	Устный опрос	2
Тема 2 «Разработка на Flask: от простого приложения до продвинутых паттернов»	Лекция 2 «Flask: архитектура, маршрутизация, шаблоны и управление состоянием»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	-	3
	Практическая работа 3 «Создание многостраничного веб-приложения на Flask»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	Устный опрос	4
	Практическая работа 4 «Реализация персонального блога с авторизацией»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2,	Устный опрос	4

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		ПКос-7.1, ПКос-7.2		
Тема 3 «Продвинутая разработка на Django»	Лекция 3 «Продвинутые контроллеры и формы в Django»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	-	2
	Лекция 4 «Архитектура, расширяемость и производительность Django-приложений»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	-	2
	Практическая работа 5 «Разработка системы управления задачами с расширенной формой и ограничениями доступа»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	Устный опрос -	2
	Практическая работа 6 «Добавление аватаров пользователей и обработка загрузки файлов»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	Устный опрос -	2
	Практическая работа 7 «Реализация системы уведомлений с использованием сигналов»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3,	Устный опрос -	3

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2		
	Практическая работа 8 «Оптимизация и кэширование веб-приложения»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	Устный опрос	3
Тема 4 «Современные API с FastAPI и асинхронностью»	Лекция 5 «Основы FastAPI: асинхронность, Pydantic и автоматическая документация»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	-	2
	Лекция 6 «Продвинутые возможности FastAPI: безопасность, базы данных и интеграция»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	-	2
	Практическая работа 9 «Разработка RESTful API для управления задачами»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	Устный опрос	4
	Практическая работа 10 «Добавление аутентификации и подключение базы данных»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1,	Устный опрос	4

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2		
Тема 5 «Тестирование, безопасность и развёртывание веб-приложений»	Лекция 7 «Обеспечение качества, безопасности и подготовка веб-приложений к production»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	-	3
	Практическая работа 11 «Тестирование и безопасность веб-приложения»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2	Устный опрос	4

Таблица 5

**Перечень вопросов для самостоятельного изучения дисциплины**

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
1.	Тема 1. «Введение в веб-разработку и архитектура современных приложений»	Сравнение архитектур Django, Flask и FastAPI: когда использовать каждый фреймворк. Принципы работы WSGI и ASGI. Роль виртуальных окружений (venv, conda) и управления зависимостями (pip, poetry). (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2).
2.	Тема 2 «Разработка на Flask: от простого приложения до продвинутых паттернов»	Разница между функциональными и классовыми представлениями в Flask. Использование blueprints для модульной архитектуры. Принципы работы сессий и механизмы защиты от подделки (CSRF) в микрофреймворках. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2).
3.	Тема 3 «Продвинутое приложение на Django»	Сравнение функциональных (FBV) и классовых (CBV) представлений в Django. Принцип работы сигналов и их влияние на производительность. Отличие middleware от декораторов и случаи их применения.

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
		(ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2).
4.	Тема 4 «Современные API с FastAPI и асинхронностью»	Преимущества асинхронной обработки запросов в FastAPI по сравнению с синхронными фреймворками. Роль Pydantic в валидации и сериализации данных. Сравнение SQLAlchemy Core и TortoiseORM для асинхронной работы с БД. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2).
5.	Тема 5 «Тестирование, безопасность и развёртывание веб-приложений»	Отличие unit-тестов от интеграционных в контексте веб-приложений. Механизмы защиты от XSS и SQL-инъекций в Django и FastAPI. Принципы настройки production-окружения: почему нельзя использовать встроенный сервер (gunicorn, uvicorn --reload). (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3, ПКос-5.1, ПКос-5.2, ПКос-7.1, ПКос-7.2).

## 5. Образовательные технологии

Таблица 6

### Применение активных и интерактивных образовательных технологий

№ п/п	Тема и форма занятия		Наименование используемых активных и интерактивных образовательных технологий (форм обучения)
1.	Тема 2 «Разработка на Flask: от простого приложения до продвинутых паттернов»	ПЗ	Разбор конкретных ситуаций
2.	Тема 2 «Разработка на Flask: от простого приложения до продвинутых паттернов»	ПЗ	Мозговой штурм
3.	Тема 3 «Продвинутое приложение на Django»	ПЗ	Разбор конкретных ситуаций
4.	Тема 4 «Современные API с FastAPI и асинхронностью»	ПЗ	Разбор конкретных ситуаций

## **6. Текущий контроль успеваемости и промежуточная аттестация по итогам освоения дисциплины**

### **6.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности**

#### **Вопросы для подготовки к устным опросам**

##### **Тема 1. Введение в веб-разработку и архитектура современных приложений**

1. В чём разница между WSGI и ASGI? Какой из них используется в FastAPI и почему?
2. Почему для веб-разработки на Python рекомендуется использовать виртуальные окружения?
3. Какие ключевые отличия между Django, Flask и FastAPI с точки зрения архитектуры и типичного применения?
4. Что такое REST и какие принципы он предполагает?
5. Какую роль играет файл requirements.txt в проекте на Python?

##### **Тема 2. Разработка на Flask: от простого приложения до продвинутых паттернов**

1. Как в Flask обрабатываются HTTP-запросы разных методов (GET, POST)?
2. Зачем нужны blueprints и как они улучшают структуру проекта?
3. Как работает механизм сессий в Flask и где хранятся данные по умолчанию?
4. Что такое CSRF и как Flask защищает формы от этой уязвимости?
5. Как настроить обработку 404-ошибки в приложении на Flask?

##### **Тема 3. Продвинутое развитие разработки на Django**

1. В чём преимущество классовых представлений (CBV) перед функциональными (FBV)?
2. Как в Django реализовать ограничение доступа к странице только для авторизованных пользователей?
3. Что такое сигналы в Django и в каких случаях их использование оправдано?
4. Какие инструменты Django помогают избежать проблемы N+1 при работе с ORM?
5. Зачем нужны middleware и как создать собственный middleware?

##### **Тема 4. Создание современных API с FastAPI**

1. Как Pydantic обеспечивает валидацию входных данных в FastAPI?
2. Почему FastAPI генерирует документацию автоматически и как к ней обратиться?
3. Как реализовать простую аутентификацию по токену в FastAPI?
4. В чём преимущество асинхронных эндпоинтов в FastAPI по сравнению с синхронными?
5. Как организовать структуру проекта с использованием APIRouter?

## **Тема 5. Тестирование, безопасность и развёртывание веб-приложений**

1. Чем unit-тесты отличаются от интеграционных в контексте веб-приложений?
2. Какие меры защиты от XSS и CSRF предусмотрены в Django «из коробки»?
3. Почему нельзя использовать `python manage.py runserver` в production?
4. Как проверить, что ваше приложение корректно обрабатывает 404 и 500 ошибки?
5. Какие шаги необходимо выполнить, чтобы развернуть Django-приложение на платформе Render или Railway?

**Критерии оценки:** опрос осуществляет из всего списка студентов. За каждый вопрос, студент может получить максимум 10 баллов. 1 студент, в течении устного опроса отвечает только на 1 вопрос. Таким образом, за каждый опрос студент может получить максимум 10 баллов.

### **Перечень вопросов, выносимых на промежуточную аттестацию (зачет с оценкой)**

1. В чём разница между WSGI и ASGI? Какой из них используется в FastAPI?
2. Какие основные отличия между Flask, Django и FastAPI? В каких случаях предпочтителен каждый из них?
3. Как в Flask организовать проект с использованием blueprints?
4. Как работает механизм сессий в Flask и как обеспечить их безопасность?
5. Что такое CSRF и как Flask защищает формы от этой уязвимости?
6. Как в Django реализовать классовое представление для отображения списка объектов?
7. В чём преимущество CBV (Class-Based Views) перед FBV (Function-Based Views) в Django?
8. Как в Django реализовать ограничение доступа к странице только для авторизованных пользователей?
9. Что такое сигналы в Django и в каких случаях их использование оправдано?
10. Как избежать проблемы N+1 при работе с ORM в Django?
11. Что такое middleware в Django и как создать собственный middleware?
12. Как Pydantic обеспечивает валидацию данных в FastAPI?
13. Почему FastAPI автоматически генерирует документацию и как к ней обратиться?
14. Как реализовать простую аутентификацию по токену в FastAPI?
15. В чём преимущество асинхронных эндпоинтов в FastAPI?
16. Как организовать структуру проекта в FastAPI с использованием APIRouter?
17. Как работает валидация входных данных в Pydantic? Можно ли добавить кастомную логику?
18. Как в FastAPI обрабатывать загрузку файлов?

19. Чем unit-тесты отличаются от интеграционных в контексте веб-приложений?
20. Как в Django написать тест для представления, которое возвращает JSON?
21. Как проверить, что ваше приложение корректно обрабатывает ошибку 404?
22. Какие встроенные средства защиты от XSS и CSRF есть в Django?
23. Почему нельзя использовать `python manage.py runserver` в production?
24. Как настроить CORS в FastAPI для взаимодействия с фронтендом?
25. Какие шаги необходимо выполнить, чтобы развернуть Django-приложение на Render?
26. Что такое `.env`-файл и зачем он нужен при развёртывании?
27. Как измерить покрытие кода тестами в Python-проекте?
28. Как в Flask настроить обработку ошибок 404 и 500?
29. Как работает аутентификация по OAuth2 в FastAPI (Password Flow)?
30. Какие инструменты помогают отладить SQL-запросы в Django (например, избежать N+1)?

### **Практическое задание № 1**

#### **«Создание многостраничного веб-приложения на Flask»**

Цель: изучить основы разработки серверных веб-приложений с использованием фреймворка Flask, включая маршрутизацию, шаблонизацию и обработку пользовательского ввода.

Условие: требуется разработать веб-приложение для демонстрации сферы деятельности агрохолдинга.

Требуется, используя Flask:

1. Создать базовую структуру проекта:
  - a. Настроить виртуальное окружение и установить Flask.
  - b. Создать файл `app.py` с инициализацией приложения.
2. Реализовать маршрутизацию:
  - a. Главная страница (`/`) – краткое описание компании.
  - b. Страница «О нас» (`/about`) – информация о миссии и ценностях.
  - c. Страница «Контакты» (`/contacts`) – адрес, телефон, email.
3. Настроить шаблоны:
  - a. Создать базовый шаблон `base.html` с общим заголовком и навигацией.
  - b. Использовать наследование шаблонов для дочерних страниц.
  - c. Добавить стили через тег `<style>` или подключить CSS-файл.
4. Добавить обработку формы:
  - a. На странице «Контакты» разместить форму обратной связи (имя, email, сообщение).
  - b. При отправке формы отображать flash-сообщение «Спасибо за обращение!».
  - c. Валидировать email с помощью простой проверки на наличие `@`.

## Практическое задание № 2

### «Разработка RESTful API для управления данными»

Цель: освоить разработку простого REST API с использованием Flask и научиться обмениваться данными в формате JSON.

Требуется:

1. Создать API для управления списком сельхозкультур:
  - a. Эндпоинт GET /api/crops – возвращает список всех культур в формате JSON.
  - b. Эндпоинт GET /api/crops/<id> – возвращает данные о культуре по ID.
  - c. Эндпоинт POST /api/crops – принимает JSON с названием и урожайностью, добавляет новую запись.
  - d. Эндпоинт DELETE /api/crops/<id> – удаляет запись по ID.
2. Хранить данные во временном списке (в памяти):
  - a. Использовать глобальный список словарей: crops = [{"id": 1, "name": "Пшеница", "yield": 3.5}, ...]
  - b. Автоматически генерировать ID при добавлении.
3. Обеспечить валидацию входных данных:
  - a. Проверять наличие обязательных полей в POST-запросе.
  - b. Возвращать ошибку 400 Bad Request при некорректных данных.
4. Протестировать API:
  - a. Использовать встроенный Swagger или Postman для отправки запросов.
  - b. Продемонстрировать успешное создание, чтение, удаление записей.

## Комплект разноуровневых заданий

### Тема 1 «Введение в веб-разработку и архитектура современных приложений»

1. Задание репродуктивного уровня

Опишите разницу между WSGI и ASGI. В каких фреймворках они используются и почему FastAPI требует ASGI, а Flask – WSGI?

2. Задание реконструктивного уровня

Приведите примеры типичных ошибок при настройке виртуального окружения и управления зависимостями в Python-проекте (например, конфликт версий, отсутствие requirements.txt). Как их избежать?

### Тема 2 «Разработка на Flask: от простого приложения до продвинутых паттернов»

1. Задание репродуктивного уровня

Опишите, как реализовать:

- o обработку POST-запроса с формой в Flask,
- o валидацию email-адреса в форме,
- o отображение flash-сообщения после успешной отправки.

2. Задание реконструктивного уровня

Дан фрагмент кода на Flask с ошибкой в обработке сессий (например, данные не сохраняются между запросами). Найдите ошибку и предложите исправленную версию с пояснением.

### **Тема 3 «Продвинутая разработка на Django»**

1. Задание реконструктивного уровня  
Опишите, в чём разница между функциональными (FBV) и классовыми (CBV) представлениями в Django. В каких случаях предпочтительнее использовать CBV?
2. Задание реконструктивного уровня  
Опишите поэтапно, как реализовать защищённый от N+1-проблемы список задач с отображением автора каждой задачи:
  - o с использованием `select_related`,
  - o с добавлением кэширования на уровне представления.

### **Тема 4 «Создание современных API с FastAPI»**

1. Задание репродуктивного уровня  
Составьте таблицу, сравнивающую Flask и FastAPI по следующим параметрам:
  - o поддержка асинхронности,
  - o валидация данных,
  - o автоматическая документация,
  - o типизация кода,
  - o производительность.
2. Задание реконструктивного уровня  
Разработайте структуру проекта FastAPI для RESTful-сервиса «Управление задачами». Объясните, почему выбраны:
  - o Pydantic-модели для валидации,
  - o APIRouter для модульности,
  - o асинхронные эндпоинты,
  - o автоматическая документация.

### **Тема 5 «Тестирование, безопасность и развёртывание веб-приложений»**

1. Задание репродуктивного уровня  
Опишите различия между unit-тестами и интеграционными тестами в контексте веб-приложений. Приведите пример каждого для Django или FastAPI.
2. Задание реконструктивного уровня  
Дано: при развёртывании Django-приложения на Render возникает ошибка 500.  
Предложите пошаговый план диагностики:
  - o проверка логов,
  - o настройка `DEBUG=False`,
  - o управление статикой,
  - o использование `.env`-файла.

- о объясните, почему нельзя оставлять DEBUG=True в production.

**Критерии оценки:** принимаются преподавателем и оцениваются вместе с устными опросами.

## **6.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания**

Для оценки знаний, умений, навыков и формирования компетенции по дисциплине применяется **балльно-рейтинговая** система контроля и оценки успеваемости студентов.

В основу балльно-рейтинговой системы (БРС) положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущей работы в семестре.

Работы должны быть выполнены по своему варианту, оформлены в соответствии с требованиями стандартов по оформлению текстовых документов в текстовом редакторе MS Word. Работы сдаются в электронном виде.

По результатам защиты могут быть получены следующие баллы:

9-10 баллов – расчеты (если имеются) проведены корректно, результаты правильно интерпретированы. Полностью выполнены все пункты выданного задания. Работа оформлена в соответствии с требованиями стандартов по оформлению текстовых документов. Студент развернуто и свободно ответил на все вопросы при защите работы.

7-8 баллов – работа выполнена, выполнены все пункты выданного задания, но не полностью, либо с несущественными ошибками, имеются незначительные ошибки в интерпретации результатов и/или оформлении. Студент в целом ответил на все поставленные вопросы, ориентируется в работе.

4-6 баллов – работа в целом выполнена, выполнены основные, но не все пункты выданного задания, либо с существенными ошибками, имеются значительные ошибки в интерпретации полученных результатов и представления данных, оформления работы. Некоторые вопросы по работе вызывают затруднения.

1-3 балла – имеются грубые ошибки в методике выполнения, интерпретации полученных результатов и представления данных, оформления работы, большая часть пунктов выданного задания не выполнена. Студент не отвечает на вопросы при защите.

В течение периода обучения по дисциплине студент должен выполнить и защитить 11 практических заданий (индивидуальных или групповых проектов), каждое из которых оценивается максимум на 10 баллов. За посещение занятий добавляется 0,15 балла за каждый час ( $68 \cdot 0,15$ ), участие в конференции с докладом по теме, связанной с возможностями практического применения

языка Python – 10 баллов. Таким образом, максимально возможная сумма баллов равна:  $11 \cdot 10 + 68 \cdot 0,15 + 10 = 110 + 10 + 10 = 130$ .

Зачет по дисциплине получают студенты, набравшие не менее 60% от максимального количества баллов, т.е. 78 баллов и более.

Итоговая оценка по дисциплине выставляется преподавателем в соответствии со шкалой:

Текущий рейтинг	Оценка			
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
в процентах	0-59	60-69	70-84	85-100
в баллах	0-77	78-90	91-110	111-130

Студенты, набравшие в течение семестра менее 78 баллов, пишут итоговую зачетную работу. К написанию итоговой зачетной работы допускаются студенты, **в случае выполнения всех практических работ.**

## 7. Учебно-методическое и информационное обеспечение дисциплины

### 7.1 Основная литература

1. Полупанов, Д. В. Программирование в Python 3 : учебное пособие / Д. В. Полупанов, С. Р. Абдюшева, А. М. Ефимов. – Уфа : БашГУ, 2020. – 164 с. – ISBN 978-5-7477-5230-6. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/179915>

2. Решение задач вычислительной математики на языке Python: лабораторный практикум : учебное пособие / Е. А. Демчинова, М. С. Красавина, И. Г. Панин, А. С. Чувиляева. – Кострома : КГУ им. Н.А. Некрасова, 2021. – 103 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/177618>

3. Винокуров, И. В. Разработка веб-приложений с использованием фреймворка Flask : учебное пособие для вузов / И. В. Винокуров. – Санкт-Петербург : Лань, 2025. – 64 с. – ISBN 978-5-507-52020-6. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/469016> – Режим доступа: для авториз. пользователей.

### 7.2 Дополнительная литература

1. Янцев, В. В. Web-программирование на Python : учебное пособие для вузов / В. В. Янцев. – Санкт-Петербург : Лань, 2022. – 180 с. – ISBN 978-5-8114-9461-3. — Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/233264>

2. Программные системы статистического анализа. Обнаружение закономерностей в данных с использованием системы R и языка Python : учебное пособие / В. М. Волкова, М. А. Семёнова, Е. С. Четвертакова, С. С. Вожов. – Новосибирск : НГТУ, 2017. – 74 с. – ISBN 978-5-7782-3183-2. – Текст :

электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/118287>

3. Борзунов, С. В. Алгебра и геометрия с примерами на Python / С. В. Борзунов, С. Д. Кургалин. – 3-е изд., стер. – Санкт-Петербург : Лань, 2022. – 444 с. – ISBN 978-5-8114-9980-9. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/202154>

4. Елисеев, А. И. Разработка программных интерфейсов веб-приложений с использованием фреймворка FastAPI : учебное пособие / А. И. Елисеев, Ю. В. Минин. – Тамбов : ТГТУ, 2024. – 81 с. – ISBN 978-5-8265-2821-1. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/472310> – Режим доступа: для авториз. пользователей.

5. Баланов, А. Н. Бэкенд-разработка веб-приложений: архитектура, проектирование и управление проектами : учебное пособие для вузов / А. Н. Баланов. – 2-е изд., стер. – Санкт-Петербург : Лань, 2025. – 312 с. – ISBN 978-5-507-52472-3. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/451820> – Режим доступа: для авториз. пользователей.

## **8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

1. Python 3.11.1 documentation. – URL: <https://docs.python.org/3/>
2. Google's Python Class. – URL: <https://developers.google.com/edu/python>
3. Machine Learning Crash Course. – URL: <https://developers.google.com/machine-learning/crash-course>
4. Введение в Python. – URL: <https://steps.2035.university/collections/c4706f68-0aa9-419b-8d8a-c9a968a108fc>
5. Цифровые профессии: Искусственный интеллект. – URL: <https://steps.2035.university/collections/f6361b9a-ea2e-41b1-a18f-9a2f84a9fcd4>
6. Kaggle. – URL: <https://www.kaggle.com/>
7. Django – URL: <https://www.djangoproject.com/>
8. Flask – URL: <https://flask.palletsprojects.com/en/stable/>
9. FastAPI – URL: <https://fastapi.tiangolo.com/>

## 9. Перечень программного обеспечения и информационных справочных систем

Таблица 9

### Перечень программного обеспечения

№ п/п	Наименование раздела учебной дисциплины	Наименование программы	Тип программы	Автор	Год разработки
1	Тема 1. «Введение в веб-разработку и архитектура современных приложений»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
2	Тема 2 «Разработка на Flask: от простого приложения до продвинутых паттернов»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
3	Тема 3 «Продвинутая разработка на Django»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
4	Тема 4 «Современные API с FastAPI и асинхронностью»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
5	Тема 5 «Тестирование, безопасность и развёртывание веб-приложений»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012

## 10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Таблица 10

### Сведения об обеспеченности специализированными аудиториями, кабинетами, лабораториями

Наименование специальных* помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	Оснащенность специальных помещений и помещений для самостоятельной работы**
1	2
учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных	1. Компьютер – 29 шт.; 2. Стенд «Сергеев Сергей Степанович 1910-1999» (Инв.№591013/25) – 1 шт.; 3. Огнетушитель порошковый (Инв. №559527) – 1 шт.; 4. Подвесное крепление к огнетушителю (Инв. № 559528) – 1 шт.; 5. Жалюзи (Инв. №1107-221225, Инв. №1107-

консультаций, учебная аудитория для текущего контроля и промежуточной аттестации (2й учебный корпус, 102 ауд.)	221225) – 2шт.; 6. Стул – 29 шт.; 7. Стол компьютерный – 28 шт.; 8. Стол для преподавателя – 1 шт.; 9. Доска маркерная (Инв. № 558762/5) – 1 шт.; 10. Трибуна напольная (без инв. №) – 1 шт.
учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации, помещение для самостоятельной работы (2й учебный корпус, 106 ауд.)	1. Рабочая станция FORSITE TH1516G512G, Российская Федерация A4Tech Fstyler F1512 – 16 шт.; 2. Стол наборный (Инв. №410136000010828) – 1 шт. 3. Стол компьютерный (Инв. № 410136000010813-410136000010827) – 15 шт.; 4. Стул (Инв. № 410136000010829-410136000010853) – 25 шт.; 5. Интерактивная панель (Инв. № 410124000603715) – 1 шт.
учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации, помещение для самостоятельной работы (2й учебный корпус, 302 ауд.)	1. Компьютер – 16 шт. 2. Телевизор – 1 шт. 3. Стол для преподавателя – 1 шт. 4. Стол компьютерный – 16 шт. 5. Стул офисный – 17 шт. 6. Компьютер: PRO-3159209 Intel Core i5-10400 2900МГц, Intel B460, 16Гб DDR4, Intel UHD Graphics 630 (встроенная), SSD 240Гб, 500Вт, Mini-Tower – 1 шт. 7. Кондиционер HAIER HSU -24HPL03/R3 (Инв. № 210134000062198) – 1 шт. 8. Вешалка напольная (Инв.№1107-333144, Инв.№1107-333144) – 2 шт.
учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации, помещение для самостоятельной работы (2й учебный корпус, 303 ауд.)	1. Трибуна напольная (Инв.№ 599206) – 1 шт.; 2. Жалюзи (Инв.№591110) – 1 шт.; 3. Доска маркетинговая (Инв.№ 35643/4) – 1 шт.; 4. Стол – 15 шт.; 5. Скамейка – 14 шт.; 6. Стол эрго – 1 шт.; 7. Стул – 16 шт.
Центральная научная библиотека имени Н.И. Железнова	Читальные залы библиотеки
Студенческое общежитие	Комната для самоподготовки

## **11. Методические рекомендации студентам по освоению дисциплины**

Приступая к изучению дисциплины «Веб-разработка», студенты должны ознакомиться с учебной программой, учебной, научной и методической литературой, имеющейся в библиотеке РГАУ-МСХА им. К.А. Тимирязева, получить в библиотеке рекомендованные учебники и учебно-методические пособия, завести новую тетрадь для работы с первоисточниками.

В ходе занятий вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

В ходе подготовки к практическим занятиям изучить основную литературу, ознакомиться с дополнительной литературой в соответствии с поставленной задачей. При этом учесть рекомендации преподавателя и требования учебной программы. Необходимо дорабатывать свой конспект, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой.

При подготовке к экзамену (в конце семестра) повторять пройденный материал в строгом соответствии с учебной программой. Использовать конспекты и литературу, рекомендованную преподавателем. Обратит особое внимание на темы учебных занятий, пропущенных студентом по разным причинам. При необходимости обратиться за консультацией и методической помощью к преподавателю.

### **Виды и формы отработки пропущенных занятий**

Студент, пропустивший занятия обязан самостоятельно подготовиться к теме устного опроса, которые состоялись на практическом занятии. В рамках часов консультаций студент может ответить на вопросы пропущенного устного опроса, которые были пропущены.

## **12. Методические рекомендации преподавателям по организации обучения по дисциплине**

Курс «Веб-разработка» должен давать не абстрактно-формальные, а прикладные знания. Данная цель может быть реализована только при условии соблюдения в учебных планах преемственности учебных дисциплин. Базовые знания для изучения Разработки профессиональных приложений дают такие дисциплины, как программирование на Python, Алгоритмизация и программирование. Освоение основных тем данной дисциплины позволит студентам сформировать представление о таком сложном предмете как тестирование программного обеспечения, понять всю ширину науки и получить необходимые знания для последующего профессионального развития в этой области.

Студент может подготовить доклад по теме, представляющей его научный интерес, представить результаты в виде презентации. В случае надлежащего качества, его работа может быть заслушана на научном кружке

кафедры или на студенческой научной конференции. По решению кафедры, студенты, занявшие призовые места на научных студенческих конференциях, могут освободиться от сдачи экзамена по этой дисциплине.

Преподаватель должен указывать, в какой последовательности следует изучать материал дисциплины, обращать внимание на особенности изучения отдельных тем и разделов, помогать отбирать наиболее важные и необходимые сведения из учебных пособий, а также давать объяснения вопросам программы курса, которые обычно вызывают затруднения. При этом преподавателю необходимо учитывать следующие моменты:

1. Не следует перегружать студентов творческими заданиями.
2. Чередовать творческую работу на занятиях с заданиями во внеаудиторное время.
3. Давать студентам четкий инструктаж по выполнению самостоятельных заданий: цель задания; условия выполнения; объем; сроки; требования к оформлению.
4. Осуществлять текущий учет и контроль за самостоятельной работой.
5. Давать оценку и обобщать уровень усвоения навыков самостоятельной, творческой работы.

**Программу разработал:**

Демичев В.В., канд. экон. наук, доцент

(ФИО, ученая степень, ученое звание)

Храмов Д.Э., ассистент

(ФИО, ученая степень, ученое звание)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(подпись)



## РЕЦЕНЗИЯ

на рабочую программу дисциплины Б1.В.14 «Веб-разработка» ОПОП ВО по направлению 09.03.02 Информационные системы и технологии, направленность «Системная аналитика и разработка программного обеспечения» (квалификация выпускника – бакалавр)

Вахрушевой Инны Алексеевны, доцентом кафедры высшей математики, кандидатом педагогических наук (далее по тексту рецензент), проведено рецензирование рабочей программы дисциплины «Веб-разработка» ОПОП ВО по направлению 09.03.02 Информационные системы и технологии, направленность «Системная аналитика и разработка программного обеспечения» (бакалавриат) разработанной в ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», на кафедре статистики и кибернетики (разработчики – Демичев Вадим Владимирович, доцент, кандидат экономических наук, Храмов Дмитрий Эдуардович, ассистент кафедры статистики и кибернетики).

Рассмотрев представленные на рецензирование материалы, рецензент пришел к следующим выводам:

1. Предъявленная рабочая программа дисциплины «Веб-разработка» (далее по тексту Программа) соответствует требованиям ФГОС ВО по направлению 09.03.02 Информационные системы и технологии. Программа содержит все основные разделы, соответствует требованиям к нормативно-методическим документам.

2. Представленная в Программе **актуальность** учебной дисциплины в рамках реализации ОПОП ВО не подлежит сомнению – дисциплина относится к дисциплинам части, формируемой участниками образовательных отношений – Б1.В

3. Представленные в Программе **цели** дисциплины соответствуют требованиям ФГОС ВО направления 09.03.02 Информационные системы и технологии.

4. В соответствии с Программой за дисциплиной «Веб-разработка» закреплены **4 профессиональные компетенции (ПКос)**. Дисциплина «Веб-разработка» и представленная Программа способна реализовать их в объявленных требованиях.

5. **Результаты обучения**, представленные в Программе в категориях знать, уметь, владеть соответствуют специфике и содержанию дисциплины и демонстрируют возможность получения заявленных результатов.

6. Общая трудоёмкость дисциплины «Веб-разработка» составляет 3 зачётные единицы (108 часов).

7. Информация о взаимосвязи изучаемых дисциплин и вопросам исключения дублирования в содержании дисциплин соответствует действительности. Дисциплина «Веб-разработка» взаимосвязана с другими дисциплинами ОПОП ВО и Учебного плана по направлению 09.03.02 Информационные системы и технологии и возможность дублирования в содержании отсутствует.

8. Представленная Программа предполагает использование современных образовательных технологий, используемые при реализации различных видов учебной работы. Формы образовательных технологий соответствуют специфике дисциплины.

9. Программа дисциплины «Веб-разработка» предполагает проведение занятий в интерактивной форме.

10. Виды, содержание и трудоёмкость самостоятельной работы студентов, представленные в Программе, соответствуют требованиям к подготовке выпускников, содержащимся во ФГОС ВО направления 09.03.02 Информационные системы и технологии.

11. Представленные и описанные в Программе формы *текущей* оценки знаний (устный опрос), соответствуют специфике дисциплины и требованиям к выпускникам.

Форма промежуточного контроля знаний студентов, предусмотренная Программой, осуществляется в форме экзамена в восьмом семестре, что соответствует статусу дисциплины, как дисциплины части, формируемой участниками образовательных

отношений дисциплин учебного плана – Б1.В ФГОС ВО направления 09.03.02. Информационные системы и технологии.

12. Формы оценки знаний, представленные в Программе, соответствуют специфике дисциплины и требованиям к выпускникам.

13. Учебно-методическое обеспечение дисциплины представлено: основной литературой – 3 источника (базовые учебники), дополнительной литературой – 5 наименования, Интернет-ресурсы – 9 источника и соответствует требованиям ФГОС ВО направления 09.03.02 Информационные системы и технологии.

14. Материально-техническое обеспечение дисциплины соответствует специфике дисциплины «Веб-разработка» и обеспечивает использование современных образовательных, в том числе интерактивных методов обучения.

15. Методические рекомендации студентам и методические рекомендации преподавателям по организации обучения по дисциплине дают представление о специфике обучения по дисциплине «Веб-разработка».

#### **ОБЩИЕ ВЫВОДЫ**

На основании проведенного рецензирования можно сделать заключение, что характер, структура и содержание рабочей программы дисциплины «Веб-разработка» ОПОП ВО по направлению 09.03.02 Информационные системы и технологии, направленность «Системная аналитика и разработка программного обеспечения» (квалификация выпускника – бакалавр), разработанная Демичевым Вадимом Владимировичем, доцентом, кандидатом экономических наук, Храмовым Дмитрием Эдуардовичем, ассистентом кафедры статистики и кибернетики соответствует требованиям ФГОС ВО, современным требованиям экономики, рынка труда и позволит при её реализации успешно обеспечить формирование заявленных компетенций.

Рецензент: Вахрушева Инна Алексеевна, доцент кафедры высшей математики ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», кандидат педагогических наук

(подпись)

«26» августа 2025 г.