

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ



УТВЕРЖДАЮ
Декан ЭФ

УТВЕРЖДАЮ /Н.М. Стрельникова/
(Ф.И.О. декана (директора института))

11.03.2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б.1.1.23 Технологии программирования и создание WEB приложений

(код и наименование дисциплины по учебному плану)

Направление подготовки
(специальность)

09.03.02 Информационные системы и технологии

Квалификация выпускника

Бакалавр

(бакалавр/магистр/специалист)

Направленность

Экономическая кибернетика

Курс 3
Семестр 5

Распределение учебного времени

Трудоемкость по учебному плану	180 / 5	часов/зачетных единиц
Лекции	36	часов
Лабораторные работы	54	часов
Практические занятия	-	часов
Иная контактная работа	-	часов
Всего контактной работы (без учета экз.)	90	часов
Контактная работа по экзамену	6	часов
Курсовой проект (работа)	-	семестр
Самостоятельная работа обучающихся (без учета экз.)	54	часов
Самостоятельная работа по подготовке к экзамену	30	часов
Экзамен	5	семестр
Зачет	-	семестр
БРК, ДЗ	-	семестр

(год)

Программа составлена в соответствии с требованиями ФГОС ВО направления подготовки (специальности) 09.03.02 Информационные системы и технологии

Программу составили:

доцент с ученой степенью кандидата наук	РТиМБС	СОГЛАСОВАНО	К.О. Иванов
(должность)	(кафедра)		(И.О. Фамилия)

РАССМОТРЕНА и ОДОБРЕНА на заседании кафедры, за которой закреплена дисциплина
Кафедра информатики

22.01.2024	протокол №	6	(наименование кафедры)
(дата)			
Заведующий кафедрой	СОГЛАСОВАНО	Р.Г. Хафизов	
		(И.О. Фамилия)	

Рабочая программа СОГЛАСОВАНА с факультетом (институтом), выпускающей(ими)
кафедрой(ами).
СООТВЕТСТВУЕТ действующей ОП.

Заведующий кафедрой	СОГЛАСОВАНО	Л.В. Смоленникова
		(И.О. Фамилия)

Председатель методической комиссии факультета (института), в который входит
выпускающая кафедра

СОГЛАСОВАНО	О.Е. Иванов
	(И.О. Фамилия)

Эксперт(ы): Жубрин Алексей Анатольевич, помощник генерального директора АО «ММЗ»
по информатизации – начальник управления информационных технологий
Рабочая программа проверена и зарегистрирована в УМЦ 12.03.2024 г.
Специалист учебно-методического центра СОГЛАСОВАНО /Т.А. Смирнова/

Раздел 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является достижение планируемых результатов обучения, соответствующих установленным в ОПОП индикаторам достижения компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения
1. ОПК-6 Способен разрабатывать алгоритмы и программ, пригодные для практического применения в области информационных систем и технологий	ОПК-6.1 Знать методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий	знания: языков программирования, таких как Python, JavaScript и других, в зависимости от конкретных потребностей проекта. умения: навыки:
	ОПК-6.2 Умеет применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий	знания: умения: работать с различными структурами данных (списки, деревья, хеш-таблицы) и алгоритмами (сортировка, поиск и т. д.) для эффективного решения задач. навыки:
	ОПК-6.3 Имеет навыки программирования, отладки и тестирования прототипов программно-технических	знания: умения: навыки: работы с современными интегрированными средами разработки, отладчиками и системами контроля версий.

Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к обязательной части ОПОП.

Дисциплина является обязательной

Для продолжения формирования заявленных компетенций необходимы знания предшествующих дисциплин: Информационные технологии (включая основы программирования) (ОПК-6); практик: Учебная практика. Ознакомительная практика (ОПК-6)

Изучаемая дисциплина является основой для продолжения формирования указанных компетенций в следующих дисциплинах: Машинное обучение и анализ данных (ОПК-6), Интеллектуальные информационные технологии (ОПК-6); практиках: Преддипломная практика (ОПК-6); государственной итоговой аттестации в форме: Выполнение и защита выпускной квалификационной работы (ОПК-6)

Раздел 3. ОПИСАНИЕ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

Для формирования заявленных компетенций используются методологические технологии,

реализующие деятельностный, личностно-ориентированный, практико-ориентированный подходы.

Основными стратегическими технологиями являются: лекционные занятия, практические и лабораторные занятия, процедуры самообучения

На достижение конкретных целей обучения направлены применяемые тактические технологии: классическая лекция, лекция с элементами мозгового штурма, задания, мини-проекты

Раздел 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

5 семестр

Виды и темы занятий	Количество часов	Формируемые компетенции
основы веб разработки	144	ОПК-6
Лекция. Лекция 1: Основы веб-разработки 1. Как устроен интернет и как работает веб? 2. HTTP и роли клиента и сервера в веб-архитектуре. 3. Основные технологии, используемые для создания веб-страниц.	4	
Лекция. Лекция 2: HTML и CSS: Создание структуры и стилизация 1. Основные теги HTML используемые для создания структуры веб-страницы. 2. Использование стилей CSS для изменения внешнего вида элементов страницы. 3. Что такое селекторы и как они применяются в CSS?	4	
Лекция. Лекция 3: JavaScript: Основы и взаимодействие с DOM 1. Базовые концепции программирования на JavaScript. 2. Что такое Document Object Model (DOM) и как JavaScript взаимодействует с ним? 3. Создание интерактивных элементов на веб-странице с использованием JavaScript.	4	
Лекция. Лекция 4: Responsive Web Design и мобильная адаптация 1. Что такое адаптивный дизайн и почему он важен для веб-страниц? 2. Технологии и методы, используемые для создания мобильно-адаптивных веб-сайтов. 3. Проверка и оптимизация веб-страницы для различных устройств.	4	
Лекция. Лекция 5: Введение в бэкенд-разработку 1. Что представляет собой бэкенд в веб-разработке? 2. Языки программирования, широко используемые для написания бэкенд-кода. 3. Организация сервера и обработки запросов в веб-приложении.	4	
Лекция. Лекция 6: Работа с базами данных и язык SQL 1. Как хранятся данные в веб-приложении? 2. Взаимодействие с базой данных с использованием языка SQL. 3. Обеспечение безопасности данных при работе с базой данных.	4	

<p>Лекция. Лекция 7: Аутентификация и авторизация</p> <ol style="list-style-type: none"> 1. Что такое аутентификация и почему она важна для веб-приложений? 2. Реализация системы пользовательских аккаунтов и сессий. 3. Обеспечение безопасности паролей пользователей. 	4
<p>Лекция. Лекция 8: RESTful API и взаимодействие с сервером</p> <ol style="list-style-type: none"> 1. Что такое REST и RESTful API? 2. Взаимодействие с сервером через HTTP-запросы. 3. Проектирование и разработка собственных API для веб-приложения? 	4
<p>Лекция. Лекция 9: Тестирование и оптимизация веб-приложений</p> <ol style="list-style-type: none"> 1. Почему важно тестировать веб-приложения и какие виды тестирования существуют? 2. Методы оптимизации производительности веб-приложения. 3. Инструменты для тестирования и оптимизации кода в веб-разработке. 	4
<p>Лабораторная работа. Лабораторная работа 1: Основы HTML и CSS</p> <ol style="list-style-type: none"> 1. Создание базовой HTML-страницы: <ul style="list-style-type: none"> ? Основные элементы в составе HTML. ? Использование тегов для разметки текста и изображений. 2. Применение стилей с использованием CSS: <ul style="list-style-type: none"> ? Создание и использование внутренних и внешних стилей. ? Изменение шрифтов, цвета и расположения элементов на веб-странице. 3. Разработка простого макета: <ul style="list-style-type: none"> ? Создание простого макета веб-страницы с использованием HTML и CSS. ? Обеспечение адаптивности сайта для различных устройств. 	4
<p>Лабораторная работа. Лабораторная работа 2: Основы JavaScript и взаимодействие с DOM</p> <ol style="list-style-type: none"> 1. Основы JavaScript: <ul style="list-style-type: none"> ? Создание переменных и выполнение базовых операций. ? Работа с условиями и циклами. 2. Взаимодействие с DOM: <ul style="list-style-type: none"> ? Доступ к элементам на веб-странице с помощью JavaScript. ? Динамическое изменение содержимого и структуры страницы. 3. Создание интерактивных элементов: <ul style="list-style-type: none"> ? Реализация простых интерактивных элементов, таких как кнопки и формы. ? Обработка событий пользователя. 	4
<p>Лабораторная работа. Лабораторная работа 3: Работа с веб-формами и валидация данных</p> <ol style="list-style-type: none"> 1. Создание веб-форм: <ul style="list-style-type: none"> ? Создание формы для ввода данных на веб-странице. ? Использование различных типов полей ввода. 	4

<p>2. Валидация данных на стороне клиента:</p> <ul style="list-style-type: none"> ? Реализация простой валидации данных в форме с использованием JavaScript. ? Предотвращение отправки некорректных данных на сервер. <p>3. Более сложная валидация:</p> <ul style="list-style-type: none"> ? Добавление более сложной валидации, такой как проверка формата электронной почты или пароля? 	
<p>Лабораторная работа. Лабораторная работа 4: Работа с AJAX и взаимодействие с сервером</p> <p>1. Загрузка данных асинхронно:</p> <ul style="list-style-type: none"> ? Использование AJAX для отправки и получения данных без перезагрузки страницы. ? Обработка асинхронных запросов с использованием промисов. <p>2. Взаимодействие с API:</p> <ul style="list-style-type: none"> ? Получение данных с внешнего API. ? Обработка ответов и обновление содержимого страницы. 	4
<p>Лабораторная работа. Лабораторная работа 5: Responsive Web Design</p> <p>1. Медиазапросы и адаптивный дизайн:</p> <ul style="list-style-type: none"> ? Использование медиазапросов для адаптации дизайна под различные устройства. ? Изменение стилей в зависимости от размера экрана. <p>2. Флексбокс и сетки:</p> <ul style="list-style-type: none"> ? Использование флексбоксов и сеток для создания гибкого макета. ? Управление порядком и выравниванием элементов на странице. 	4
<p>Лабораторная работа. Лабораторная работа 6: Работа с фреймворками и библиотеками</p> <p>1. Выбор и установка фреймворка:</p> <ul style="list-style-type: none"> ? Выбор фреймворков или библиотек для разработки фронтенда. ? Установка и настройка выбранных средств. <p>2. Создание компонентов:</p> <ul style="list-style-type: none"> ? Организация кода с использованием компонентов. ? Передача данных между компонентами. 	4
<p>Лабораторная работа. Лабораторная работа 7: Тестирование фронтенда и оптимизация</p> <p>1. Тестирование пользовательского интерфейса:</p> <ul style="list-style-type: none"> ? Создание и запуск тестов для пользовательского интерфейса. ? Фреймворки для тестирования пользовательского интерфейса. 	4
<p>Лабораторная работа. Лабораторная работа 8: Настройка окружения и базовые операции для создания бекенда</p> <p>1. Настройка серверного окружения:</p> <ul style="list-style-type: none"> ? Установка и настройка сервера (Node.js, Django, Flask). ? Создание простого сервера и проверка его работоспособности. ? Базовые операции с файлами и базой данных на сервере. 	4

Лабораторная работа. Лабораторная работа 9: Работа с базой данных и SQL 1. Интеграция с базой данных: ? Подключение к базе данных с использованием выбранного языка (например, SQL, MongoDB). ? Выполнение CRUD-операции (Create, Read, Update, Delete) в базе данных. ? Обеспечение безопасности при работе с базой данных.	4
Лабораторная работа. Лабораторная работа 10: Разработка API 1. Проектирование и разработка API: ? Определение структуры и конечных элементов точки вашего API. ? Реализация обработки HTTP-запросов для различных действий. ? Выбор методов аутентификации и авторизации.	4
Лабораторная работа. Лабораторная работа 11: Работа с сессиями и пользователями 1. Система пользовательских аккаунтов: ? Создание системы регистрации и аутентификации пользователей. ? Управление данными сессии и куки. ? Обеспечение безопасности и конфиденциальности данных пользователей.	4
Лабораторная работа. Лабораторная работа 12: Обработка ошибок и логирование 1. Обработка ошибок и логирование: ? Обработка и вывод ошибок для клиента. ? Ведение логирования для отслеживания действий на сервере. ? Обеспечение безопасности и целостности данных при возникновении ошибок.	4
Лабораторная работа. Лабораторная работа 13: Работа с внешними сервисами и API 1. Взаимодействие с внешними ресурсами: ? Отправка запросов к внешним API. ? Обработка ответов от внешних сервисов. ? Реализация асинхронных операций в бэкенд-приложении.	4
Лабораторная работа. Лабораторная работа 14: Тестирование и документирование 1. Тестирование и документирование кода: ? Создание и запуск тестов для бэкенд-приложения. ? Документация API для удобства использования. ? Обеспечение надежности и стабильности бэкенд-приложения.	2
Задания для самостоятельной работы, в том числе выполнение КР синтаксис SQL запросов, современные фреймворки для создания интерфейса десктоп-приложений, синтаксис HTML и CSS.	54
Иная контактная работа:	0
Подготовка к экзамену	30

Проведение экзамена	6	
---------------------	---	--

Раздел 5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Изучение дисциплины рекомендуется начать с ознакомления с рабочей программой, ее структурой и содержанием разделов. Учебный материал структурирован, изучение дисциплины осуществляется в тематической последовательности.

Занятия лекционного типа дают систематизированные знания по дисциплине, концентрируют внимание на наиболее сложных и важных вопросах. Во время лекционных занятий рекомендуется вести конспектирование учебного материала; обращать внимание на формулировки и категории, раскрывающие суть проблемы, явления или процесса; зафиксировать выводы и практические рекомендации. Подготовка к **занятиям семинарского типа** включает ознакомление с планом практического занятия; работу с конспектом лекций, выполнение домашнего задания, работу с учебной и учебно-методической литературой, научными изданиями и электронными образовательными ресурсами, рекомендованными рабочей программой дисциплины.

Содержание **самостоятельной работы** определяется рабочей программой дисциплины, оценочными и методическими материалами, заданиями и указаниями преподавателя. Самостоятельная работа может осуществляться в аудиторной и внеаудиторной формах. Эффективным средством осуществления самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к образовательной программе, рабочей программе дисциплины, к электронным библиотечным системам, профессиональным базам данных и информационным справочным системам. Изучение дисциплины включает выполнение контрольной работы и лабораторных работ. Периодичность проведения, формы текущего контроля успеваемости, система оценивания хода освоения дисциплин представлены в рабочей программе.

Раздел 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Учебно-методическое обеспечение

№№ п/п	Список используемой литературы	Количество экземпляров печатных изданий, имеющих в библиотеке, или электронный адрес издания (ресурса) в сети Интернет
УЧЕБНЫЕ, УЧЕБНО-МЕТОДИЧЕСКИЕ И НАУЧНЫЕ ИЗДАНИЯ		
1.	Рябов, В. А. Современные веб-технологии [Электронный ресурс] / Рябов В. А., Несвижский А. И. 2-е изд. Москва: ИНТУИТ, 2016. - 1080 с.	https://e.lanbook.com/book/100499
2.	Кузнецова, Л. В. Лекции по современным веб-технологиям [Электронный ресурс] / Кузнецова Л. В. 2-е изд. Москва: ИНТУИТ, 2016. - 187 с.	https://e.lanbook.com/book/100629
3.	Кудряшев, А. В. Введение в современные веб-технологии [Электронный ресурс] / Кудряшев А. В., Светашков П. А. 2-е изд. Москва: ИНТУИТ, 2016. - 360 с.	https://e.lanbook.com/book/100711
4.	Введение в стандарты Web [Электронный ресурс]. 2-е изд. Москва: ИНТУИТ, 2016. - 800 с.	https://e.lanbook.com/book/100712
5.	Янцев, В. В. Web-программирование на Python [Электронный ресурс] : учебное пособие для вузов /	https://e.lanbook.com/book/3

	Янцев В. В. 2-е изд., стер. Санкт-Петербург: Лань, 2023. - 180 с. ISBN 978-5-507-46546-0.	10289
6.	Диков, А. В. Клиентские технологии веб-дизайна. HTML5 и CSS3 [Электронный ресурс] : учебное пособие для вузов / Диков А. В. 2-е изд., стер. Санкт-Петербург: Лань, 2023. - 188 с. ISBN 978-5-507-46740-2.	https://e.lanbook.com/book/318443
7.	Ипатов, Юрий Аркадьевич. Современные технологии коллективной работы в глобальных сетях [Текст] : компьютерный практикум / Ю. А. Ипатов, А. В. Кревецкий; М-во образования и науки РФ, ФГБОУ ВПО "Поволж. гос. технол. ун-т". Йошкар-Ола: ПГТУ, 2013. - 223 с. ISBN 978-5-8158-1256-7. Экземпляры: всего 151.	151 / https://portal.volgatech.net/books/Ipatov_sovremennie_tehnologii_kollektivnoi_raboti_globalnix_setiax_2013.pdf
8.	Малашкевич, Василий Борисович. Интернет-программирование [Текст] : лабораторный практикум / В. Б. Малашкевич; М-во образования и науки Рос. Федерации, ФГБОУ ВО "Поволж. гос. технол. ун-т". Йошкар-Ола: ПГТУ, 2017. - 96 с. ISBN 978-5-8158-1854-5. Экземпляры: всего 26.	26 / https://portal.volgatech.net/books/Malashkevich_internet_programmirovanie_2017.pdf
ЭЛЕКТРОННЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ		
1.	Научная электронная библиотека eLIBRARY.RU	http://elibrary.ru
2.	Научная электронная библиотека «Киберленинка»	http://cyberleninka.ru
ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ		
1.	Справочно-правовая система Консультант+	http://www.consultant.ru
2.	Информационно-правовой портал Гарант	http://www.garant.ru
3.	Профессиональные справочные системы Техэксперт	http://www.cntd.ru

6.2. Материально-техническая база и программное обеспечение

№№ п/п	Аудитории для проведения учебных занятий, самостоятельной работы и проведения государственной итоговой аттестации	Перечень основного оборудования	Программное обеспечение
1.	531 (III)	ПК B112,2 420W/Intel Celeron Dual-Core E3300/кл,мышь,фильт,мон. VA1931 (5)	Microsoft Windows Enterprise, Справочная правовая система "Консультант Плюс", Microsoft Office Standard, Агент Dr.Web, Комплект ГАРАНТ-Мастер, Microsoft Access, Microsoft Visio Professional, Microsoft Project Professional, Microsoft Visual Studio Enterprise, Комплект ПО для решения основных пользовательских задач

Раздел 7. ФОРМЫ КОНТРОЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ/ ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Критерии оценивания индикаторов достижения компетенций направлены на:

- усвоение теоретического материала (объем знаний, глубина усвоения), предусмотренного рабочей программой;
 - умение излагать материал (четкость, грамотность изложения материала, точность и полнота воспроизведения учебного материала);
 - умение применять теоретические знания при решении практических заданий.
- Шкала оценивания представлена ниже.

Уровень сформированности элементов компетенции	Критерии оценивания	Шкала оценивания
Пороговый уровень	Обучающийся имеет знания основного материала, проявляет умение логично его излагать, но может допускать неточности в изложении материала, недостаточно правильные формулировки, испытывает затруднения в выполнении практических заданий.	удовлетворительно
Продвинутый уровень	Обучающийся твердо знает программный материал, излагает его грамотно и по существу, не допускает существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения	хорошо
Высокий уровень	Обучающийся глубоко и прочно усвоил программный материал, грамотно и логически стройно его излагает, дает исчерпывающие ответы на поставленные вопросы. В ответе тесно увязывается теория с практикой, при этом обучающийся не затрудняется с ответом при видоизменении задания, свободно справляется с задачами, вопросами и другими видами применения знаний, показывает знакомство с монографической литературой, периодическими изданиями, правильно обосновывает принятые решения, свободно владеет разносторонними навыками, приемами выполнения практических работ	отлично

7.1. Текущий контроль успеваемости

Текущий контроль успеваемости обеспечивает оценивание хода освоения дисциплины (модуля) и производится с применением технологии рейтингового контроля в соответствии с технологической картой дисциплины. Порядок составления технологической карты и алгоритм проведения процедуры оценивания видов деятельности обучающихся, направленных на освоение знаний, умений, навыков и/ или опыта деятельности, по накопительной системе в баллах устанавливается положением о системе РИТМ в ФГБОУ ВО «ПГТУ»

7.2. Промежуточная аттестация обучающихся

Промежуточная аттестация обучающихся направлена на оценивание результатов обучения по дисциплине (модулю) и проводится с использованием фондов оценочных средств.

Примеры типовых контрольных заданий из базы фонда оценочных средств по образовательной программе.

Пример экзаменационного билета:

1. Какие виды тестирования чаще всего применяются в веб-разработке?
2. Какие инструменты для совместной работы и управления задачами используются при разработке веб-приложений?
3. Какие декораторы используются для определения маршрутов в Flask?

Краткие ответы к экзаменационному билету:

1. Какие виды тестирования чаще всего применяются в веб-разработке?

В веб-разработке применяют различные виды тестирования для обеспечения качества веб-приложений. Наиболее распространенные виды тестирования включают:

1. Модульное тестирование (Unit Testing): Проверка отдельных компонентов (модулей) кода на корректность работы. Обычно проводится на уровне функций или классов.
2. Интеграционное тестирование (Integration Testing): Проверка взаимодействия между различными компонентами приложения. Гарантирует, что все части приложения работают корректно в совокупности.
3. Функциональное тестирование (Functional Testing): Проверка функциональности приложения с учетом его спецификации. Тестирование основных функций и возможностей.
4. Приемочное тестирование (Acceptance Testing): Тестирование, направленное на подтверждение того, что приложение соответствует требованиям заказчика и может быть принято им.
5. Системное тестирование (System Testing): Проверка всей системы в целом, включая взаимодействие всех ее компонентов. Обеспечивает соответствие системы целевым требованиям.
6. Нагрузочное тестирование (Load Testing): Исследование поведения системы при нагрузке для оценки ее производительности и устойчивости.
7. Интерфейсное тестирование (UI Testing): Проверка интерфейса приложения на соответствие дизайну и корректность взаимодействия с пользователем.
8. Безопасность веб-приложения (Security Testing): Анализ приложения на наличие уязвимостей и проверка защиты от возможных атак.
9. Тестирование совместимости (Compatibility Testing): Проверка работы приложения на различных платформах, браузерах и устройствах.

Эти виды тестирования выполняются в различных фазах разработки, начиная от ранних этапов и до постоянного мониторинга в процессе эксплуатации веб-приложения.

2. Какие инструменты для совместной работы и управления задачами используются при разработке веб-приложений?

При разработке веб-приложений используется множество инструментов для совместной работы и управления задачами. Вот несколько популярных инструментов в этой области:

1. Jira: Универсальный инструмент управления задачами и проектами, который обеспечивает возможность создания, отслеживания и управления задачами, а также позволяет формировать отчеты.
2. Trello: Простой и интуитивно понятный инструмент визуального управления задачами в виде карточек на доске. Позволяет легко отслеживать ход выполнения задач.
3. Asana: Платформа для управления проектами и задачами удобным интерфейсом. Позволяет создавать проекты, задачи, отслеживать сроки и общаться внутри команды.
4. Git и GitHub/GitLab/Bitbucket: Для управления версиями кода и совместной разработки. Git предоставляет систему контроля версий, а GitHub, GitLab и Bitbucket — хостинг для удаленного хранения репозитория и совместной работы.
5. Confluence: Инструмент для создания и совместного редактирования документации, вики-страницы технических спецификаций.

6. Slack: Платформа для коммуникации и обмена сообщениями в реальном времени внутри команды. Позволяет создавать каналы для обсуждения конкретных тем.

7. Notion: Инструмент для организации заметок, задач и документации. Обладает гибкими возможностями структурирования информации.

8. MS Teams: Платформа Microsoft для совместной работы, включающая в себя чат, видеоконференции, совместную работу с файлами и другие возможности.

3. Какие декораторы используются для определения маршрутов в Flask?

В Flask, маршруты определяются с использованием декораторов. Вот несколько основных декораторов, используемых для определения маршрутов:

`@app.route('/')`: Определяет маршрут для корневой страницы. Пример:

python

Copy code

```
@app.route('/')
def index():
```

```
    return 'Привет, мир!'
```

`@app.route('/about')`: Определяет маршрут для страницы "О нас". Пример:

python

Copy code

```
@app.route('/about')
def about():
```

```
    return 'Страница "О нас"'
```

`@app.route('/user/<username>')`: Позволяет передавать переменные в маршрут. Пример:

python

Copy code

```
@app.route('/user/<username>')
def show_user(username):
```

```
    return f'Профиль пользователя {username}'
```

`@app.route('/login', methods=['GET', 'POST'])`: Указывает поддерживаемые методы запроса для данного маршрута. Пример:

python

Copy code

```
@app.route('/login', methods=['GET', 'POST'])
def login():
```

```
    if request.method == 'POST':
```

```
        # Обработка данных из формы
```

```
        pass
```

```
    else:
```

```
        # Отображение страницы входа
```

```
        pass
```

`@app.route('/logout')`: Декоратор для маршрута выхода из системы. Пример:

python

Copy code

```
@app.route('/logout')
def logout():
```

```
    # Логика выхода из системы
```

```
    pass
```

Эти декораторы позволяют определить обработчики для различных маршрутов в Flask-

приложении, обеспечивая эффективное управление навигацией и отображением контента на веб-страницах.

Пример контрольной работы:

Пример контрольной работы:

Пример контрольной работы:

Пример контрольной работы:

Пример контрольной работы:

Вопрос 1:

Какоерасширение обычно используется для файлов стилей в веб-разработке?

- a) .js
- b) .html
- c) .css
- d) .php

Вопрос 2:

Что означает аббревиатура HTML?

- a) Hyperlink and Text Markup Language
- b) Hyper Text Markup Language
- c) High-level Text Markup Language
- d) Hyper Transfer Markup Language

Вопрос 3:

Какой тег используется для создания гиперссылок в HTML?

- a) <link>
- b) <a>
- c) <href>
- d) <url>

Вопрос 4:

Какие из перечисленных языков программирования обычно используются для создания динамических веб-страниц?

- a) HTML
- b) CSS
- c) JavaScript
- d) SQL

Вопрос 5:

Какая функция выполняется на стороне клиента в веб-разработке?

- a) Серверная
- b) Фронтенд
- c) Бэкенд
- d) Клиентская

Вопрос 6:

Какой HTTP-метод обычно используется для отправки данных на сервер?

- a) GET
- b) POST
- c) PUT
- d) DELETE

Вопрос 7:

Какая из следующих технологий используется для структурирования документов и представления данных на веб-странице?

- a) XML
- b) JSON
- c) YAML
- d) HTML

Вопрос 8:

В основе фреймворков для веб-разработки на Python лежит шаблонизатор?

- a) Jinja2
- b) Jinjis
- c) IBM Watson
- d) EasyEDA

Правильные ответы:

- 1. c
- 2. b

- 3. b
- 4. c
- 5. b
- 6. b
- 7. d
- 8. d

Перечень вопросов для проведения промежуточной аттестации

1. Каковы базовые принципы объектно-ориентированного программирования (ООП)?
2. Что такое переменная и какие типы данных вы знаете? Приведите примеры.
3. Какова роль HTML, CSS и JavaScript в веб-разработке?
4. В чем разница между фреймворками React, Angular и Vue.js?
5. Какие типы баз данных вы знаете и в чем их отличия?
6. Какие SQL-операторы используются для выборки данных из таблицы?
7. Что такое SQL-инъекции и как их предотвратить?
8. Какие меры безопасности следует принимать для защиты от межсайтовой подделки запроса (CSRF)?
9. Какие виды тестирования чаще всего применяются в веб-разработке?
10. Какие преимущества предоставляет автоматизированное тестирование?
11. Как работает система контроля версий Git? Расскажите о базовых командах.
12. Зачем нужны ветки (branches) в Git?
13. Какие инструменты для совместной работы и управления задачами вы использовали в проектах?
14. Как можно обеспечить эффективную коммуникацию в команде разработки веб-приложений?
15. Что такое Flask и каковы его основные характеристики?
16. Как установить Flask и создать базовое веб-приложение?
17. Объясните структуру каталогов стандартного Flask-приложения.
18. Какие декораторы используются для определения маршрутов в Flask?
19. Что такое представления (views) в контексте Flask?
20. Как использовать шаблоны Jinja2 в Flask-приложении?
21. Как передать переменные из представления (view) в шаблон Jinja2?
22. Как интегрировать базу данных в Flask-приложение?
23. Как выполнять запросы к базе данных с использованием Flask-SQLAlchemy?
24. Поясните синтаксис декларативного языка программирования UML.