

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО
Заместитель директора

Ю.О. Соболев

	Рабочая программа дисциплины (модуля)
по дисциплине:	Веб-разработка
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Разработка IT-продукта
	центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
	центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
курс:	1
квалификация:	магистр

Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет

Аудиторных часов: 16 всего, в том числе:

лекции: 6 час.

семинары: 10 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 164 час.

Всего часов: 180, всего зач. ед.: 4

Программу составил: К.А. Лапин, старший методист

Программа обсуждена на заседании центра дополнительного, дополнительного профессионального и
онлайн-образования "Пуск" 01.03.2025

Аннотация

Дисциплина «Веб-разработка» посвящена изучению основ создания веб-приложений, включая разработку серверной и клиентской частей, настройку инфраструктуры, обеспечение безопасности и тестирование. Студенты получают знания и навыки, необходимые для создания современных веб-приложений с использованием современных технологий и инструментов.

1. Цели и задачи

Цель дисциплины

- формирование базовых навыков веб-разработки и деплоя веб-приложений.

Задачи дисциплины

- освоить основные этапы разработки веб-приложений;
- изучить популярные фреймворки и инструменты для веб-разработки;
- научиться создавать и настраивать веб-серверы, базы данных, API и HTTP-взаимодействия;
- сформировать понимание процесса тестирования и настройки резервного копирования веб-приложений.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.1 Анализирует проблемную ситуацию как систему, выявляя ее составляющие и связи между ними
	УК-1.2 Осуществляет поиск вариантов решения поставленной проблемной ситуации на основе доступных источников информации
	УК-1.3 Разрабатывает стратегию достижения поставленной цели как последовательность шагов, предвидя результат каждого из них и оценивая их влияние на внешнее окружение планируемой деятельности и на взаимоотношения участников этой деятельности
ОПК-4 Способен успешно реализовывать решение поставленной задачи, провести анализ результата и представить выводы, применяя знания и навыки в области математики, естественных наук и информационно-коммуникационных технологий	ОПК-4.1 Способен применять знания и навыки по использованию информационно-коммуникационных технологий для поиска и изучения научной литературы, применения прикладных программных продуктов
	ОПК-4.2 Способен применять знание информационно-коммуникационных технологий для решения поставленной задачи, формулирования выводов и оценки полученных результатов
	ОПК-4.3 Способен аргументировано выбирать способ проведения научного исследования
	ОПК-4.4 Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями
ПК-4 Способен проектировать информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-4.1 Знает основы проектирования и элементы архитектурных решений информационных систем
	ПК-4.2 Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий
	ПК-4.3 Имеет опыт оценки качества, надежности и эффективности информационной системы в конкретной профессиональной сфере

ПК-5 Способен к установке, администрированию программных систем, к реализации технического сопровождения информационных систем, к интеграции информационных систем с используемыми аппаратно-программными комплексами	ПК-5.1 Знает методику установки и администрирования программных систем
	ПК-5.2 Умеет реализовывать техническое сопровождение информационных систем

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- базовые принципы веб-разработки и ее этапы;
- фреймворки и инструменты для веб-разработки;
- принципы настройки веб-серверов, баз данных и api;
- методы обеспечения безопасности веб-приложений.

уметь:

- создавать и деплоить веб-приложения;
- настраивать веб-серверы;
- разрабатывать api и http-взаимодействия из кода;
- производить тестирование веб-приложений.

владеть:

- навыками контейнеризации с использованием docker;
- методами непрерывной интеграции и развертывания (ci/cd);
- навыками обеспечения безопасности веб-приложений.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Проектирование ТЗ		2		41
2	Верстка Frontend		2		41
3	Программирование Frontend	3	3		41
4	Программирование Backend	3	3		41
Итого часов		6	10		164
Подготовка к экзамену		0 час.			
Общая трудоёмкость		180 час., 4 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 2 (Весенний)

1. Проектирование ТЗ

Список страниц и разделов. Верхнеуровневная архитектура. Прототипы страниц. Архитектура микросервисов. Структура БД. Документация к API.

2. Верстка Frontend

Блоки, layout, расстояния. Цвета, шрифты, изображения. Стилизация.

3. Программирование Frontend

Основы JS, работа с событиями. Реакции на события. Реализация запросов к BE, async/await.

4. Программирование Backend

Программирование микросервисов. Интеграция с БД.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Занятия по учебной дисциплине проводятся с применением дистанционных образовательных технологий. Каждый обучающийся обеспечен доступом к образовательной платформе (LMS).

6. Перечень рекомендуемой литературы

Основная литература

1. Win32 API : Эффективная разработка приложений [Текст]/Ю. А. Щупак, -СПб., Питер, 2007
2. Django 2 web development cookbook, 100 practical recipes on building scalable Python web apps with Django 2/J. Kronika, A. Bendoraitis, -Birmingham ; Mumbai, Packt, 2018

Дополнительная литература

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Современный подход к версионированию для CI/CD:

<https://www.atlassian.com/continuous-delivery/continuous-integration/trunk-based-development>

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Студенту для занятий потребуются:

1. Google Drive / Yandex disk для доступа к материалам курса
2. Zoom
3. LMS МФТИ
4. Приложение для коммуникации с преподавателями
5. Ноутбук для участия в интерактивных занятиях

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий дисциплину, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике. В результате изучения дисциплины студент должен знать основные определения дисциплины, уметь применять полученные знания для решения прикладных задач.

Успешное освоение курса требует:

- посещения всех онлайн-занятий, предусмотренных учебным планом по дисциплине;
- ведения конспекта занятий;
- напряжённой самостоятельной работы студента.

Самостоятельная работа включает в себя:

- чтение материалов на платформе дистанционного обучения и рекомендованной литературы;
- проработку учебного материала, подготовку ответов на вопросы, предназначенных для самостоятельного изучения;

- решение задач, предлагаемых студентам на занятиях;
- подготовку к выполнению заданий текущей и промежуточной аттестации.

Показателем владения материалом служит умение без вспомогательных материалов и конспектов отвечать на вопросы по темам дисциплины.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к преподавателю.

Возможен текущий контроль знаний студентов в виде решения задач в соответствии с тематикой занятий.

ПРИЛОЖЕНИЕ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Информатика и вычислительная техника
профиль подготовки:	Разработка IT-продукта центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск" центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
курс:	<u>1</u>
квалификация:	магистр
Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет	
Разработчик:	К.А. Лапин, старший методист

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.1 Анализирует проблемную ситуацию как систему, выявляя ее составляющие и связи между ними
	УК-1.2 Осуществляет поиск вариантов решения поставленной проблемной ситуации на основе доступных источников информации
	УК-1.3 Разрабатывает стратегию достижения поставленной цели как последовательность шагов, предвидя результат каждого из них и оценивая их влияние на внешнее окружение планируемой деятельности и на взаимоотношения участников этой деятельности
ОПК-4 Способен успешно реализовывать решение поставленной задачи, провести анализ результата и представить выводы, применяя знания и навыки в области математики, естественных наук и информационно-коммуникационных технологий	ОПК-4.1 Способен применять знания и навыки по использованию информационно-коммуникационных технологий для поиска и изучения научной литературы, применения прикладных программных продуктов
	ОПК-4.2 Способен применять знание информационно-коммуникационных технологий для решения поставленной задачи, формулирования выводов и оценки полученных результатов
	ОПК-4.3 Способен аргументировано выбирать способ проведения научного исследования
	ОПК-4.4 Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями
ПК-4 Способен проектировать информационные системы, их компоненты и протоколы их взаимодействия, собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-4.1 Знает основы проектирования и элементы архитектурных решений информационных систем
	ПК-4.2 Умеет применять в практической деятельности профессиональные стандарты в области информационных технологий
	ПК-4.3 Имеет опыт оценки качества, надежности и эффективности информационной системы в конкретной профессиональной сфере
ПК-5 Способен к установке, администрированию программных систем, к реализации технического сопровождения информационных систем, к интеграции информационных систем с используемыми аппаратно-программными комплексами	ПК-5.1 Знает методику установки и администрирования программных систем
	ПК-5.2 Умеет реализовывать техническое сопровождение информационных систем

2. Показатели оценивания компетенций

В результате изучения дисциплины «Веб-разработка» обучающийся должен:

знать:

- базовые принципы веб-разработки и ее этапы;
- фреймворки и инструменты для веб-разработки;
- принципы настройки веб-серверов, баз данных и api;
- методы обеспечения безопасности веб-приложений.

уметь:

- создавать и деплоить веб-приложения;
- настраивать веб-серверы;
- разрабатывать api и http-взаимодействия из кода;
- производить тестирование веб-приложений.

владеть:

- навыками контейнеризации с использованием docker;
- методами непрерывной интеграции и развертывания (ci/cd);
- навыками обеспечения безопасности веб-приложений.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

В начале занятия проводится опрос по материалам предыдущего занятия и даются задачи для совместного решения.

Какие основные этапы включает в себя процесс разработки веб-приложений? Опишите каждый этап.

Какие фреймворки для веб-разработки вы знаете? Укажите их особенности и сферы применения.

Какие инструменты вы используете для настройки веб-серверов? Приведите примеры конфигураций для Apache и Nginx.

Что такое контейнеризация и какой инструмент вы используете для её реализации? Опишите основные принципы работы с Docker.

Какие типы тестирования вы используете при разработке веб-приложений? Дайте определения и примеры каждого типа тестирования.

Чем отличаются аутентификация и авторизация пользователей в контексте веб-разработки? Какие методы вы используете для их реализации?

Какие преимущества непрерывной интеграции и развертывания (CI/CD) вы видите для проектов веб-разработки? Приведите примеры сценариев использования.

Какие уязвимости веб-приложений вы считаете наиболее опасными? Опишите, какие меры предосторожности можно принять для защиты от них.

Какие инструменты вы используете для мониторинга и управления вашими веб-приложениями в производственной среде?

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Промежуточный контроль проходит в форме дифференцированного зачета.

Примеры практических заданий на дифференцированный зачет:

Задание 1: Разработка и деплой простого веб-приложения

Цель: Создать простейшее веб-приложение с использованием выбранного фреймворка (Django, Flask, или Express.js), которое отображает статическую страницу с приветственным сообщением. Затем развернуть это приложение на локальном веб-сервере.

Шаги выполнения:

1. Выберите фреймворк для разработки веб-приложения.
2. Создайте веб-приложение, которое отображает страницу с текстом «Добро пожаловать в моё веб-приложение!».
3. Настройте локальный веб-сервер (Apache или Nginx) для обслуживания вашего приложения.
4. Продемонстрируйте работающее веб-приложение и его доступность через веб-браузер.

Ожидаемый результат: Студент сможет настроить веб-сервер, разработать и развернуть простое веб-приложение.

Задание 2: Работа с базой данных через ORM

Цель: Создать небольшую базу данных и выполнить операции CRUD (создание, чтение, обновление и удаление данных) с использованием объектно-реляционного отображения (ORM) выбранного веб-фреймворка.

Шаги выполнения:

1. Определите сущность (например, пользователь с полями: имя, email) для вашей базы данных.
2. Используя механизмы ORM вашего фреймворка, создайте модель данных и миграции для создания соответствующей таблицы в базе данных.
3. Реализуйте функционал для выполнения операций CRUD через веб-интерфейс или API.
4. Продемонстрируйте добавление, чтение, обновление и удаление данных через ваше веб-приложение.

Ожидаемый результат: Студент продемонстрирует умение работать с базой данных через ORM, реализовав базовые операции с данными.

Задание 3: Введение в Docker и развертывание приложения с использованием контейнеров

Цель: Создать Dockerfile для вашего веб-приложения и развернуть его используя Docker.

Шаги выполнения:

1. Создайте Dockerfile, который описывает среду, необходимую для работы вашего веб-приложения (включая все зависимости).
2. Соберите Docker-образ вашего приложения, используя созданный Dockerfile.
3. Запустите контейнер из собранного образа на вашем локальном компьютере.
4. Проверьте доступность и работоспособность веб-приложения через веб-браузер, обращаясь к контейнеру.

Ожидаемый результат: Студент научится создавать Dockerfile, собирать образы и запускать веб-приложения в Docker-контейнерах, демонстрируя понимание основ контейнеризации.

Вопросы на дифференцированный зачет:

Определение веб-разработки и её этапы.

Обзор популярных фреймворков для веб-разработки.

Базы данных и фреймворки для работы с ними.

Настройка веб-серверов.

Создание баз данных и настройка доступов к ним.

Разработка API и HTTP-взаимодействия из кода.

Введение в Docker и контейнеризацию.

Настройка зависимостей и библиотек, конфигурации приложений.

Непрерывная интеграция и развертывание (CI/CD).

Критерии оценивания

Дифференцированный зачет по дисциплине проводится в форме выполнения итогового задания.

Оценка отлично (10 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (9 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (8 баллов) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо (7 баллов) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо (6 баллов) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо (5 баллов) - выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно (4 балла) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно (3 балла) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно (2 балла) - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно (1 балл) - выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Дифференцированный зачет проходит в письменном формате, на lms платформе.

Время проведения письменного дифференцированного зачета оставляет 2 академических часа.

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины.