

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Структурное подразделение «Институт информационных технологий и анализа данных»

УТВЕРЖДЕНА:

на заседании Совета института ИТиАД им. Е.И.Попова

Протокол №8 от 24 февраля 2025 г.

Рабочая программа дисциплины

«WEB-ПРОГРАММИРОВАНИЕ»

Направление: 09.03.02 Информационные системы и технологии

Информационные системы и технологии в административном управлении

Квалификация: Бакалавр

Форма обучения: очная

Документ подписан простой электронной подписью Составитель программы: Каташевцев Михаил Дмитриевич Дата подписания: 20.06.2025

Документ подписан простой электронной подписью Утвердил: Говорков Алексей Сергеевич Дата подписания: 20.06.2025

Документ подписан простой электронной подписью Согласовал: Аршинский Вадим Леонидович Дата подписания: 20.06.2025

Год набора – 2025

Иркутск, 2025 г.

1 Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

1.1 Дисциплина «Web-программирование» обеспечивает формирование следующих компетенций с учётом индикаторов их достижения

Код, наименование компетенции	Код индикатора компетенции
ПКС-2 Способен разрабатывать программное обеспечение (ПО), включая проектирование, отладку, проверку работоспособности и модификацию ПО	ПКС-2.4, ПКС-2.5
ПКС-4 Способен выполнять проектирование и оценку юзабилити пользовательских интерфейсов	ПКС-4.1

1.2 В результате освоения дисциплины у обучающихся должны быть сформированы

Код индикатора	Содержание индикатора	Результат обучения
ПКС-2.4	Способен применять основные веб-технологии для разработки веб-приложений	Знать Основные понятия и технологии используемые при разработки веб-приложений Уметь Создавать динамические сайты без использования фреймворков. Настраивать прокси сервер. Владеть Языком разметки HTML, CSS, языком генерации веб страниц
ПКС-2.5	Способен применять современные инструменты и фреймворки при разработке веб-приложений	Знать Современные фреймворки для бэкенд и фронтенд разработки Уметь Разрабатывать сайты с использованием современных фреймворков Владеть Методами отладки веб приложений
ПКС-4.1	Способен выполнять юзабилити-аудит, разрабатывать информационную архитектуру, применять методы UX проектирования и разрабатывать макеты страниц и форм при разработке веб-приложений	Знать Способы оценки юзабилити сайта. Инструменты для макетирования приложения Уметь Настраивать метрики Владеть Интерфейсом figma, для переноса разметки в html

2 Место дисциплины в структуре ООП

Изучение дисциплины «Web-программирование» базируется на результатах освоения следующих дисциплин/практик: «Введение в веб-технологии»

Дисциплина является предшествующей для дисциплин/практик: «Проектирование информационных систем», «Производственная практика: преддипломная практика», «Проектная деятельность»

3 Объем дисциплины

Объем дисциплины составляет – 6 ЗЕТ

Вид учебной работы	Трудоемкость в академических часах (Один академический час соответствует 45 минутам астрономического часа)		
	Всего	Семес тр № 4	Семестр № 5
Общая трудоемкость дисциплины	216	108	108
Аудиторные занятия, в том числе:	96	48	48
лекции	32	16	16
лабораторные работы	64	32	32
практические/семинарские занятия	0	0	0
Самостоятельная работа (в т.ч. курсовое проектирование)	120	60	60
Трудоемкость промежуточной аттестации	0	0	0
Вид промежуточной аттестации (итогового контроля по дисциплине)	Зачет, Курсовой проект, Зачет с оценкой	Зачет с оценкой	Зачет, Курсовой проект

4 Структура и содержание дисциплины

4.1 Сводные данные по содержанию дисциплины

Семестр № 4

№ п/п	Наименование раздела и темы дисциплины	Виды контактной работы						СРС		Форма текущего контроля
		Лекции		ЛР		ПЗ(СЕМ)				
		№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	
1	2	3	4	5	6	7	8	9	10	11
1	Устройство веб сервера	1	2	6	4			1	5	Устный опрос
2	HTTP запросы, клиент-серверные протоколы взаимодействия	2	2	1	6			1	5	Устный опрос
3	Веб фреймворки на бекенде	3	2	2	4			1	10	Отчет по лабораторной работе
4	Обработка GET и POST запросов	4	2	3	8			1	10	Устный опрос
5	Взаимодействие web приложения с базой данных. Разработка REST API	5	2	5	6			1	10	Отчет по лабораторной работе
6	Обратный прокси сервер. Балансировка нагрузки	6	2					1	5	Устный опрос
7	Способы авторизация и	7	2	4	4			1	5	Устный опрос

	аутентификации									
8	Применение фреймворков all-in-one на бекенде	8	2					1	10	Отчет по лабораторной работе
	Промежуточная аттестация									Зачет с оценкой
	Всего		16		32				60	

Семестр № 5

№ п/п	Наименование раздела и темы дисциплины	Виды контактной работы						СРС		Форма текущего контроля
		Лекции		ЛР		ПЗ(СЕМ)				
		№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	
1	2	3	4	5	6	7	8	9	10	11
1	Основы javascript	1	2	1	6			1	10	Устный опрос
2	Реактивные фреймворки	2	4	2, 5	8			1	10	Отчет по лаборатор ной работе
3	Разработка API	3	4	3	8			1	10	Отчет по лаборатор ной работе
4	Состояние SPA приложений	4	2	4	6			1	10	Отчет по лаборатор ной работе
5	Тестирование WEB приложений	5	2					1	10	Отчет по лаборатор ной работе
6	Деплой фулл-стек приложения	6	2	6	4			1	10	Устный опрос
	Промежуточная аттестация									Зачет, Курсовой проект
	Всего		16		32				60	

4.2 Краткое содержание разделов и тем занятий

Семестр № 4

№	Тема	Краткое содержание
1	Устройство веб сервера	архитектура, основные компоненты и функции, обеспечивающие обработку и доставку веб-контента
2	HTTP запросы, клиент-серверные протоколы взаимодействия	Структура HTTP-запросов и ответов. Основные коды состояния и их значения. Принципы клиент-серверной архитектуры. Вопросы безопасности и производительности в HTTP.
3	Веб фреймворки на бекенде	Устройство фреймворков. Роутинг. Мидлвэер. Обработчики запросов. Сервисы
4	Обработка GET и POST запросов	Отличие GET и POST запросов. Работа с формами
5	Взаимодействие web приложения с базой данных. Разработка REST API	Использование ORM для работы БД. REST протокол.
6	Обратный прокси	Назначение прокси серверов. Влияние на скорость

	сервер. Балансировка нагрузки	обработки запросов. Работа с нагрузкой
7	Способы авторизация и аутентификации	Отличие авторизации от аутентификации. Простая авторизация. Работа с сессией. Работа JWT-токенами
8	Применение фреймворков all-in-one на бекенде	Структура современного фреймворка. Утилиты управления проектами.

Семестр № 5

№	Тема	Краткое содержание
1	Основы javascript	Особенности обработки данных в браузере. Базовый синтаксис. Работа с пакетами
2	Реактивные фреймворки	Понятие реактивности. Структура реактивного приложения. Роутинг.
3	Разработка API	Особенности обработки http запросов. Передача файлов. Ajax запросы
4	Состояние SPA приложений	Работа с реактивными событиями. Проброс данных между компонентами. Работа с состоянием
5	Тестирование WEB приложений	Назначение тестовой БД. Особенности создания интеграционных тестов. Тестирование фронтенда
6	Деплой фулл-стек приложения	Особенности публикации web приложений. Настройка прокси сервера. Настройка web сервера. Настройка супервизора

4.3 Перечень лабораторных работ

Семестр № 4

№	Наименование лабораторной работы	Кол-во академических часов
1	Создание многостраничного сайта	6
2	Генерация страниц на основе данных БД	4
3	Реализация CRUD операций	8
4	Реализация ролевой модели доступа к сайту	4
5	Реализация REST Api	6
6	Деплой веб-приложения на сервер	4

Семестр № 5

№	Наименование лабораторной работы	Кол-во академических часов
1	Работа с DOM деревом	6
2	Создание многостраничного приложения с использованием реактивного фреймворка	4
3	Взаимодействие фронтенда с сервером	8
4	Использование общего хранилища данных	6
5	Реализация ролевой модели на фронтенде	4
6	Деплой фулл-стэк приложения	4

4.4 Перечень практических занятий

Практических занятий не предусмотрено

4.5 Самостоятельная работа

Семестр № 4

№	Вид СРС	Кол-во академических часов
1	Подготовка к практическим занятиям (лабораторным работам)	60

Семестр № 5

№	Вид СРС	Кол-во академических часов
1	Подготовка к практическим занятиям (лабораторным работам)	60

В ходе проведения занятий по дисциплине используются следующие интерактивные методы обучения: Командная работа

5 Перечень учебно-методического обеспечения дисциплины

5.1 Методические указания для обучающихся по освоению дисциплины

5.1.1 Методические указания для обучающихся по курсовому проектированию/работе:

<https://el.istu.edu/course/view.php?id=6275>

5.1.2 Методические указания для обучающихся по лабораторным работам:

<https://el.istu.edu/course/view.php?id=6275>

5.1.3 Методические указания для обучающихся по самостоятельной работе:

<https://el.istu.edu/course/view.php?id=6275>

6 Фонд оценочных средств для контроля текущей успеваемости и проведения промежуточной аттестации по дисциплине

6.1 Оценочные средства для проведения текущего контроля

6.1.1 семестр 4 | Устный опрос

Описание процедуры.

Устройство веб-сервера

Что такое веб-сервер? Назовите примеры (Apache, Nginx, IIS).

Какие основные функции выполняет веб-сервер?

Как веб-сервер обрабатывает статические и динамические страницы?

Углубленные:

Как веб-сервер взаимодействует с сервером приложений (например, PHP-FPM + Nginx)?

Что такое виртуальный хостинг (virtual hosts) и как он настраивается?

HTTP-запросы, клиент-серверные протоколы
Базовые вопросы:

Из чего состоит HTTP-запрос? Назовите методы (GET, POST, PUT, DELETE).
Чем отличаются HTTP и HTTPS? Что такое SSL/TLS?

Что такое заголовки (headers) в HTTP? Примеры (User-Agent, Content-Type).

Углубленные:

Как работает механизм keep-alive в HTTP?
Что такое WebSockets и чем они отличаются от HTTP?

4-4. Обработка GET и POST запросов
Базовые вопросы:

В чем разница между GET и POST? Когда какой метод использовать?
Как передаются данные в GET (через URL) и POST (через тело запроса)?

Что такое URL-параметры и как их обрабатывает сервер?

Углубленные:

Как защититься от SQL-инъекций при обработке GET/POST?
Что такое CSRF и как POST-запросы могут быть защищены (токены, SameSite cookies)?

4-6. Обратный прокси-сервер. Балансировка нагрузки
Базовые вопросы:

Что такое обратный прокси (reverse proxy)? Примеры (Nginx, HAProxy).
Зачем нужна балансировка нагрузки (load balancing)?

Какие алгоритмы балансировки вы знаете (round-robin, least connections)?

Углубленные:

Как Nginx может работать как кэширующий прокси?

4-7. Способы авторизации и аутентификации
Базовые вопросы:

Чем отличается аутентификация от авторизации?
Какие методы аутентификации вы знаете (логин/пароль, OAuth, JWT)?
Что такое cookies и как они используются для сессий?

Углубленные:

Как работает OAuth 2.0 (роли: клиент, ресурс, авторизационный сервер)?
В чем уязвимости базовой аутентификации (Basic Auth) и как её защитить?

Критерии оценивания.

Ясность ответа, точность терминов, примеры.

6.1.2 семестр 4 | Отчет по лабораторной работе

Описание процедуры.

Отчет о сделанной работе фиксируется в README.md файле в git-репозитории

Критерии оценивания.

Отчет в README.md должен быть структурированным, информативным и отражать проделанную работу. Оценка зависит от полноты, ясности и технической корректности.

6.1.3 семестр 5 | Устный опрос

Описание процедуры.

Цель: Проверить понимание синтаксиса, типов данных, функций, DOM и асинхронности.

Базовые вопросы:

Переменные и типы данных:

Чем отличаются let, const и var?

Какие примитивные типы данных есть в JS?

Что такое NaN и как проверить, что переменная — NaN?

Функции:

Как объявить функцию (function declaration vs arrow function)?

Что такое callback-функция? Приведите пример.

Работа с DOM:

Как выбрать элемент по id и по классу?

Как добавить обработчик события на кнопку?

Асинхронность:

Что такое Promise? Как обработать ошибку в Promise?

Чем async/await отличается от .then()?

Деплой full-stack приложения

Цель: Оценить понимание процесса развертывания фронтенда + бэкенда + БД.

Базовые вопросы (для всех):

Подготовка к деплою:

Какие файлы нужно добавить в .gitignore для full-stack проекта?

Зачем нужен package.json (для фронтенда) и requirements.txt (для бэкенда)?

Деплой фронтенда:

Как собрать React/Vue-приложение для продакшена (npm run build)?
Куда можно задеплоить статический фронтенд (Netlify, Vercel, GitHub Pages)?

Деплой бэкенда:

Как запустить Node.js/Express-сервер на хостинге?
Что такое process.env.PORT и зачем он нужен?

База данных и окружение:

Как подключить облачную БД (MongoDB Atlas, PostgreSQL на Heroku)?
Где хранить секреты (API-ключи, пароли) при деплое?

Углубленные вопросы (для сильных студентов):

Как настроить CI/CD для автоматического деплоя? (GitHub Actions, GitLab CI)
Что такое reverse proxy (Nginx) и зачем он нужен при деплое?
Как развернуть приложение в Docker-контейнере?

Критерии оценивания.

Ясность ответа, точность терминов, примеры.

6.1.4 семестр 5 | Отчет по лабораторной работе

Описание процедуры.

Отчет о сделанной работе фиксируется в README.md файле в git-репозитории

Критерии оценивания.

Отчет в README.md должен быть структурированным, информативным и отражать проделанную работу. Оценка зависит от полноты, ясности и технической корректности.

6.2 Оценочные средства для проведения промежуточной аттестации

6.2.1 Критерии и средства (методы) оценивания индикаторов достижения компетенции в рамках промежуточной аттестации

Индикатор достижения компетенции	Критерии оценивания	Средства (методы) оценивания промежуточной аттестации
ПКС-2.4	Способен разработать динамический многостраничный сайт с использованием базовых веб-технологий и выложить его на сервер	Практическое задание и устный опрос
ПКС-2.5	Способен разработать сайт с использованием современных технологий с изолированной бэкенд и фронтенд частью, связать их вместе и выложить на сервер	Практическое задание и устный опрос
ПКС-4.1	Способен оценить качество figma	Практическое

	шаблона и перенести его в реальную разметку	задание и устный опрос
--	---	------------------------

6.2.2 Типовые оценочные средства промежуточной аттестации

6.2.2.1 Семестр 4, Типовые оценочные средства для проведения дифференцированного зачета по дисциплине

6.2.2.1.1 Описание процедуры

1. Что такое клиент?
2. Виды клиентов
3. Клиенты в повседневной жизни.
4. Что такое сервер.
5. Виды серверов
6. Что такое веб сервис.
7. В чем отличие сервера от веб сервиса.
8. Схема взаимодействия клиента с веб сервисом.
9. Что такое статика.
10. Что такое медиа файлы.
11. Что такое шаблоны.
12. Что такое проху сервер
13. Протокол HTTP 1.1.
14. Метод GET.
15. Метод POST.
16. Устройство URL.
17. Устройство http запроса.
18. Что такое параметры GET запроса.
19. Что такое заголовки запроса.
20. Что такое тело запроса.
21. Устройство http ответа.
22. Что такое заголовки ответа.
23. Что такое тело ответа.
24. Устройство html документа.
25. Назначение тега head.
26. Какие теги помещаются в head.
27. Подключение сторонней библиотеки.
28. Назначение тега body.
29. Какие теги помещаются в body.
30. Тег ul.
31. Теги создания заголовков.
32. Тег для создания ссылок.
33. Тег для вставки картинок.
34. Общие теги для блочной и строчной верстки.
35. Что такое CSS.
36. Основные стили.
37. Стили для работы со шрифтами.
38. Стили для верстки на базе flex.
39. Использование атрибута style.
40. Что такое классы.
41. Подключение сторонних стилевых библиотек.

42. Что такое горячие клавиши.
43. Что такое автоформатирование.
44. Назначение плагина emmet.
45. Синтаксис сокращений emmet.
46. Назначение logem.
47. Что такое плагины.
48. Плагины для автообновления страницы браузера.
49. Структура многостраничного сайта.
50. Связывание страниц через ссылки.
51. Вынос стилей в отдельный файл.
52. Подключение изображения.
53. Публикация сайта в интернет.
54. Веб-сервис на python.
55. Привязка сервиса к порту.
56. Что такое обработчик запроса.
57. Что такое шаблон.
58. Что такое базовые шаблоны.
59. Разбиение шаблона на элементы.
60. Склейка шаблонов.
61. Передача данных в шаблон.
62. Что такое SPA приложение
63. Что такое интеграционные тесты
64. Что такое ајах запрос
65. Что такое реактивность?
66. Что такое хранилище состояния?
67. Что такое вебсокет?
68. Что такое кеш?
69. Что такое ORM?
70. Как устроена авторизация через OAuth
71. Что такое REST?
72. Как реализовать схему разграничения прав
73. Как работает реактивность на уровне бэкенда?
74. Как отправить ајах запрос
75. Как отправить файл через ајах запрос
76. Как обрабатывать ошибки HTTP запросов
77. Структура бэкенда-фреймворка
78. Структура реактивного фронтенд-фреймворка

6.2.2.1.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Обучающийся показал глубокое понимание принципов работы разработанного приложения, ответил на все вопросы	Обучающийся показал хорошее понимание принципов работы разработанного приложения, ответил на 80% вопросов	Обучающийся показал удовлетворительное понимание принципов работы разработанного приложения, ответил на 60% вопросов	Обучающийся показал низкое понимание принципов работы разработанного приложения, ответил менее чем на 60% вопросов

6.2.2.2 Семестр 5, Типовые оценочные средства для курсовой работы/курсового проектирования по дисциплине

6.2.2.2.1 Описание процедуры

В ходе работы над проектом, студент должен выбрать себе индивидуальную тему. И разработать веб приложение, включающее в себя следующие обязательные элементы:

1. Не менее 5 таблиц и не менее 5 связей между ними
2. Реализовать все REST операции для каждой модели
3. Реализовать SPA фронтенд
4. Интеграционные тесты для бекенда

И набор дополнительных компонент на выбор:

1. Добавить элемент асинхронности в сервис (чат, система оповещения и т. п)
2. Система ограничения прав
3. Генерация документов excel/word/pdf
4. Загрузка данных из внешнего файла
5. Кеширование
6. Авторизация через внешний сервис
7. Отправка сообщений через почту

6.2.2.2.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительн о	Неудовлетворительно
Реализованы все обязательные компоненты и не менее пяти дополнительных	Реализованы все обязательные компоненты и не менее трех дополнительных	Реализованы все обязательные компоненты	Обязательные компоненты не реализованы

6.2.2.3 Семестр 5, Типовые оценочные средства для проведения зачета по дисциплине

6.2.2.3.1 Описание процедуры

Зачет принимается по результатам сданных лабораторных работ

6.2.2.3.2 Критерии оценивания

Зачтено	Не зачтено
Обучающийся выполнил все лабораторные, загрузил результат работы в систему электронного обучения	Обучающийся не выполнил все поставленные задачи либо не загрузил результат работы в систему электронного обучения

7 Основная учебная литература

1. Гоше, Хуан Диего. HTML5. Для профессионалов : учебный курс / Х. Д. Гоше ; пер. с англ. Е. Шикарева, А. Крупник. - 2-е изд. - Санкт-Петербург : Питер, 2015. - 559 с. : ил

2. Робсон, Элизабет. Изучаем HTML, XHTML и CSS : к изучению дисциплины / Э. Робсон, Э. Фримен ; пер. с англ. В. Черник. - 2-е изд. - Санкт-Петербург : Питер, 2015. - 718 с. : ил. - (Head First O'Reilly)

3. Каташевцев М. Д. Web-программирование (09.03.02) : электронный курс / М. Д. Каташевцев, С. В. Бахвалов, 2023

8 Дополнительная учебная литература и справочная

1. Начало работы с Bootstrap. <https://bootstrap-4.ru/docs/5.2/getting-started/introduction/> [Электронный ресурс]

9 Ресурсы сети Интернет

1. <http://django.tealeaf.su/>
2. <http://library.istu.edu/>
3. <https://stackoverflow.com>
4. <https://habr.com/>
5. <https://getbootstrap.com>
6. <https://html5book.ru>
7. <https://www.djangoproject.com>
8. <https://vuejs.org>

10 Профессиональные базы данных

1. <http://new.fips.ru/>
2. <http://www1.fips.ru/>

11 Перечень информационных технологий, лицензионных и свободно распространяемых специализированных программных средств, информационных справочных систем

1. Свободно распространяемое программное обеспечение Visual Studio Code
2. Свободно распространяемое программное обеспечение miniconda
3. Microsoft Windows Professional 8 Russian
4. Microsoft Office Professional Plus 2013

12 Материально-техническое обеспечение дисциплины

1. Проектор Epson EB-460i LCD или аналогичный по техническим характеристикам.
2. Компьютер "i5-4440(3.1)/8Gb/500Gb/VGA/23"" или аналогичный по техническим характеристикам: не менее 16 шт