

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Галунин Сергей Александрович  
Должность: проректор по учебной работе  
Дата подписания: 08.06.2023 10:34:32  
Уникальный программный ключ:  
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП  
«Разработка программно-  
информационных систем»



**СПбГЭТУ «ЛЭТИ»**  
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования  
**«Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И.Ульянова (Ленина)»  
(СПбГЭТУ «ЛЭТИ»)»**

---

**РАБОЧАЯ ПРОГРАММА**

ДИСЦИПЛИНЫ

**«WEB-ТЕХНОЛОГИИ»**

для подготовки бакалавров

по направлению

09.03.04 «Программная инженерия»

по профилю

**«Разработка программно-информационных систем»**

Санкт-Петербург

2022

## ЛИСТ СОГЛАСОВАНИЯ

Разработчики:

доцент, к.т.н., доцент Беляев С.А.

Рабочая программа рассмотрена и одобрена на заседании кафедры МОЭВМ  
15.02.2022, протокол № 2

Рабочая программа рассмотрена и одобрена учебно-методической комиссией  
ФКТИ, 24.02.2022, протокол № 2

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

## 1 СТРУКТУРА ДИСЦИПЛИНЫ

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	МОЭВМ
Общая трудоемкость (ЗЕТ)	5
Курс	3
Семестр	5
<b>Виды занятий</b>	
Лекции (академ. часов)	34
Лабораторные занятия (академ. часов)	34
Иная контактная работа (академ. часов)	3
Все контактные часы (академ. часов)	71
Самостоятельная работа, включая часы на контроль (академ. часов)	109
Всего (академ. часов)	180
<b>Вид промежуточной аттестации</b>	
Экзамен (курс)	3
Курсовой проект (курс)	3

## **2 АННОТАЦИЯ ДИСЦИПЛИНЫ**

### **«WEB-ТЕХНОЛОГИИ»**

Курс «Web-технологии» охватывает широкий спектр технологий и подходов, используемых при разработке Интернет-сайтов и web-приложений. Рассматриваются основы построения web-приложений с использованием HTML, CSS. Студенты последовательно изучают основы протокола HTTP, настройку web-сервера, основы JavaScript, TypeScript и PHP, построение статических HTML-страниц и оформления с использованием CSS, LESS и SASS, разработку сервера приложений с использованием Node.JS. Построение серверной части на основе Express и Nest, разработка клиентских приложений с использованием Angular, React и Vue. Выполнение модульного тестирования web-приложений, сборка приложений с использованием GULP и Webpack, обеспечение безопасности web-приложений.

### **SUBJECT SUMMARY**

### **«WEB TECHNOLOGIES»**

The course "Web technologies" covers a wide range of technologies and approaches used in the development of Internet sites and web applications. The basics of building web applications using HTML, CSS are considered. Students consistently study the basics of the HTTP protocol, configuring a web server, the basics of JavaScript, TypeScript and PHP, building static HTML pages and formatting using CSS, LESS and SASS. Developing an application server using Node.JS . Building a server part based on Express and Nest, developing client applications using Angular, React and Vue. Performing unit testing of web applications, building applications using GULP and Webpack, ensuring the security of web applications.

## **3 ОБЩИЕ ПОЛОЖЕНИЯ**

### **3.1 Цели и задачи дисциплины**

1. Целью дисциплины является приобретение теоретических знаний о современных web-технологиях и фреймворках и практических навыков по всем этапам разработки web-приложений от разработки web-серверов, создания и оформления web-клиентов, организации защиты от взлома.
2. Задачами изучения дисциплины является приобретение теоретических знаний по проектированию web-приложений, формирование практических умений и навыков по разработке серверной и клиентской части web-приложений, оформлению и тестированию приложений.
3. Знание основных подходов по построению клиентской и серверной части web-приложений, по оформлению web-страниц, в том числе для просмотра с мобильных устройств, по решению типовых задач тестирования, сборки и обеспечения безопасности web-приложений с использованием языков JavaScript, TypeScript, PHP.
4. Умение разрабатывать клиентскую и серверную часть web-приложений, осуществлять настройку, тестирование и автоматизированную сборку web-приложений, решать базовые вопросы кибербезопасности.
5. Формирование навыков создания статических HTML-страниц, динамических web-приложений на языках JavaScript и TypeScript, навыков применения современных фреймворков разработки клиентских и серверных частей web-приложений.

### **3.2 Место дисциплины в структуре ОПОП**

Дисциплина изучается на основе ранее освоенных дисциплин учебного плана:

1. «Объектно-ориентированное программирование»
2. «Алгоритмы и структуры данных»
3. «Введение в информационные технологии»
4. «Программирование»

и обеспечивает изучение последующих дисциплин:

1. «Компьютерная графика»
2. «Введение в нереляционные системы управления базами данных»
3. «Проектирование человеко-машинного интерфейса»
4. «Разработка приложений для мобильных платформ»
5. «Конструирование программного обеспечения»

### 3.3 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения образовательной программы обучающийся должен достичь следующие результаты обучения по дисциплине:

<b>Код компетенции/ индикатора компетенции</b>	<b>Наименование компетенции/индикатора компетенции</b>
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;
<i>ОПК-6.2</i>	<i>Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ</i>

## 4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1 Содержание разделов дисциплины

#### 4.1.1 Наименование тем и часы на все виды нагрузки

№ п/п	Наименование темы дисциплины	Лек, ач	Лаб, ач	ИКР, ач	СР, ач
1	Тема 1. Введение	2	2		6
2	Тема 2. Основы языка JavaScript	2	3	1	10
3	Тема 3. Применение Node.js. Основы Deno	2	2		8
4	Тема 4. Основы HTML	2	2	1	6
5	Тема 5. Основы настройки стилей CSS, LESS, SASS	2	2	1	6
6	Тема 6. Применение Express, NestJS, RESTful, Ajax	2	3		8
7	Тема 7. Основы использования библиотек jQuery и jQuery UI	2	2		6
8	Тема 8. Основы языка TypeScript. Статический контроль с использованием Flow	2	2		6
9	Тема 9. Основы работы с фреймворком Angular	3	4		8
10	Тема 10. Использование библиотеки React	3	4		8
11	Тема 11. Основы работы с фреймворком Vue	3	2		8
12	Тема 12. Тестирование web-приложений	3	2		8
13	Тема 13. Сборка приложений, новые возможности языка JavaScript	2	2		8
14	Тема 14. Применение языка PHP	2	1		7
15	Тема 15. Взлом и безопасность web-приложений	2	1		6
	Итого, ач	34	34	3	109
	Из них ач на контроль	0	0	0	35
	Общая трудоемкость освоения, ач/зе	180/5			

#### 4.1.2 Содержание

№ п/п	Наименование темы дисциплины	Содержание
1	Тема 1. Введение	Введение. Описание трехуровневой архитектуры. Описание протокола HTTP: методы, параметры, запрос, ответ, заголовки, общие вопросы безопасности. Сервер nginx: установка, конфигурации, статическое содержимое, использование в качестве прокси. Apache-сервер.



№ п/п	Наименование темы дисциплины	Содержание
2	Тема 2. Основы языка JavaScript	Стандарт ECMAScript. Установка и основы использования Node.js. Язык JavaScript: базовые конструкции языка, строки, функции, объекты, классы, массивы, JSON, промисы, генераторы, Map, Set, прокси. Строгий режим JavaScript.
3	Тема 3. Применение Node.js. Основы Deno	Работа с модулями. Использование npm и yarn, готовые модули. Событийная модель в Node.js. Создание простейшего сервера на Node.js. Эхо-сервер, параметры запроса, заголовки. Журналирование. Внешние сервера для журналирования: Rollbar – попытка отловить ошибки раньше пользователей, Sentry – отслеживание JavaScript. Работа с файлами и потоками. «Асинхронный try-catch». Использование npm. Использование Deno: установка, настройка среды, разрешения и примеры.
4	Тема 4. Основы HTML	Структура HTML, понятие DOM-модели. Базовые элементы и поддерживаемые ими атрибуты. Обработка событий в HTML. Элемент form, его атрибуты, соответствующие элементы: input, select, fieldset, textarea, button. Валидация введенных значений на JavaScript.
5	Тема 5. Основы настройки стилей CSS, LESS, SASS	Оформление HTML с использованием CSS. Селекторы, свойства и их значения, псевдоклассы. Оформление границ, отступов, размеров, ссылок, списков, таблиц, текста, настройка шрифта, использование иконок. Позиционирование элементов. Адаптивная верстка, использование grid, flexbox. Использование шаблонов web-страниц, переменных в CSS. Работа с CSS с использованием JavaScript. Использование CSS-фильтров. Библиотека w3.css. Знакомство с фреймворками LESS и SASS.
6	Тема 6. Применение Express, NestJS, RESTful, Ajax	Использование Express и Koa. Использование шаблонов на основе PUG и EJS. Cookies на сервере. Реализация сессии. Фреймворк NestJS: архитектура приложения, базовый пример на JavaScript, основные возможности, подключение шаблонов. Реализация RESTful на Node.js. Применение Ajax: XMLHttpRequest, fetch. Ограничения CORS.
7	Тема 7. Основы использования библиотек jQuery и jQuery UI	Применение jQuery: селекторы, манипуляции с DOM, обработка событий использование свойств и классов CSS, эффекты, Ajax-запросы. Применение jQuery UI: взаимодействие, перетаскивание, изменение размера, выбираемые и сортируемые элементы, виджеты, аккордеон, автозаполнение, кнопки, переключатели, controlgroup, выбор даты, диалоговое окно, меню, отображение прогресса, selectmenu, бегунок, спиннер, вкладки, подсказки, темы. Знакомство с jQuery mobile.

№ п/п	Наименование темы дисциплины	Содержание
8	Тема 8. Основы языка TypeScript. Статический контроль с использованием Flow	Основы языка TypeScript: типы переменных, интерфейсы, классы, функции, шаблоны, перечисления, импорт-экспорт, пространства имён, конфигурации, JSX. Статический контроль в JavaScript с использованием Flow: настройка среды исполнения, использование из командной строки, проверка типов переменных, проверка параметров функций, generics, перечислимые типы.
9	Тема 9. Основы работы с фреймворком Angular	Основы фреймворка Angular: структура приложения, компоненты, сервисы, иерархия сервисов. Создание директив, привязка, задание маршрутов, параметры маршрутизации. Использование шаблонных переменных, работа с дочерними компонентами, валидация данных, взаимодействие с сервером.
10	Тема 10. Использование библиотеки React	Библиотека Immutable.JS. Применение JSX. Компоненты и их иерархия в библиотеке React, свойства, состояние переменных. Жизненный цикл компонентов, события. Работа с формами в React. Маршрутизация, хуки. Библиотека Redux для создания единого состояния приложения: действия, редьюсеры, хранилище.
11	Тема 11. Основы работы с фреймворком Vue	Декларативная отрисовка в фреймворке Vue. Работа с вводом пользователя. Условия и циклы. Композиция приложения из компонентов. Синтаксис шаблонов. Вычисляемые свойства и наблюдатели. Работа с классами и стилями. Маршрутизация. Применение Vuex для обеспечения единого состояния приложения.
12	Тема 12. Тестирование web-приложений	Понятие TDD-разработки. Модульные тесты: Assert – встроенный в Node.js модуль для тестирования, применение принципов TDD; Should – библиотека утверждений; Chai – библиотека утверждений. Mocha – фреймворк для тестирования. Jest – фреймворк для тестирования. Тестирование с использованием Selenium, Selenium IDE, Protractor. Использование headless браузеров. Использование Postman для тестирования web-сервера.
13	Тема 13. Сборка приложений, новые возможности языка JavaScript	Автоматизация задач с помощью GULP, применение src, dest, обработка стилей, скриптов. Сборка модулей (bundle) с помощью Webpack: простая конфигурация (entry, output), использование загрузчиков и плагинов, контроль изменений кода при разработке, отладка web-приложений. Новые возможности JavaScript: web-общения, web-worker, web-сокеты (ws, socket.io). Возможности history, sessionStorage, localStorage, cookie. Авторизация с Passportjs. Работа с реляционными базами данных с использованием Sequelize ORM. Применение WebAssembly.

№ п/п	Наименование темы дисциплины	Содержание
14	Тема 14. Применение языка PHP	Настройка nginx для работы с php. Знакомство с языком PHP: область применения, основные элементы языка (переменные, условные операторы, циклы, строки, массивы, функции, элементы ООП), работа с файлами (в т.ч. xml), обработка запросов с клиента (get, post), обработка исключительных ситуаций и ошибок. Знакомство с CMS.
15	Тема 15. Взлом и безопасность web-приложений	Возможные уязвимости web-приложений. Обеспечение безопасности web-приложения от взлома: SQL-уязвимость, Shell-код, Upload-уязвимость, Include-уязвимость, XSS-уязвимость. Инструменты проверки на наличие уязвимостей.

## 4.2 Перечень лабораторных работ

Наименование лабораторной работы	Количество ауд. часов
1. Тетрис на JavaScript.	4
2. REST-приложение управления библиотекой.	6
3. Модуль администрирования приложения «Социальная сеть».	6
4. Модуль пользователя приложения «Социальная сеть».	6
5. Модуль администрирования приложения «Биржа акций».	6
6. Модуль приложения «Покупка и продажа акций».	6
Итого	34

## 4.3 Перечень практических занятий

Практические занятия не предусмотрены.

## 4.4 Курсовое проектирование

Цель работы (проекта): Выработка навыков использования различных технологий разработки программного обеспечения, готовности обосновывать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности, способность оценивать временную и емкостную сложность программного обеспечения, способность создавать программные интерфейсы.

Содержание работы (проекта): Курсовая работа включает в себя разработку

тайловой однопользовательской игры и состоит из следующих этапов:

1. Анализ и проектирование состава действий и механизмов взаимодействия участников игры;
2. Проектирование не менее двух уровней с использованием тайлового менеджера;
3. Реализация всех менеджеров игры в соответствии с учебным пособием;
4. Реализация форм авторизации и рейтинга участников;
5. Тестирование игры.

Студенту на первой неделе семестра выдается задание на курсовое проектирование.

Результаты выполнения курсовой работы представляются в виде отчета. Отчёт по курсовой работе оформляется в соответствии с шаблоном оформления курсовой работы, размещённым на сайте вуза. Отчет должен содержать разделы в соответствии с шаблоном: титульный, лист задания, аннотация, содержание, введение, выводы. Стиль оформления (шрифт, размер шрифта и т.п.) должны соответствовать шаблону, размещённому на сайте вуза. Ограничений на количество использованных источников литературы нет. Формат оформления - документ в формате doc, docx, odt или pdf. Отчет размещается в Moodle вуза.

Примерные темы:

1. Игра "Марио".
2. Игра "Танчики".
3. Игра "Пакман".
4. Игра "Мир коробок".
5. Игра "Прохождение лабиринта"..

Темы:

№ п/п	Название темы	Перевод темы
1	Двумерная однопользовательская игра на JavaScript	Two dimensions single user play on JavaScript

#### **4.5 Реферат**

Реферат не предусмотрен.

#### **4.6 Индивидуальное домашнее задание**

Индивидуальное домашнее задание не предусмотрено.

#### **4.7 Доклад**

Доклад не предусмотрен.

#### **4.8 Кейс**

Кейс не предусмотрен.

#### **4.9 Организация и учебно-методическое обеспечение самостоятельной работы**

Изучение дисциплины сопровождается самостоятельной работой студентов с рекомендованными преподавателем литературными источниками и информационными ресурсами сети Интернет.

Планирование времени для изучения дисциплины осуществляется на весь период обучения, предусматривая при этом регулярное повторение пройденного материала. Обучающимся, в рамках внеаудиторной самостоятельной работы, необходимо регулярно дополнять сведениями из литературных источников материал, законспектированный на лекциях. При этом на основе изучения рекомендованной литературы целесообразно составить конспект основных положений, терминов и определений, необходимых для освоения разделов учебной дисциплины.

<b>Текущая СРС</b>	<b>Примерная трудоемкость, ач</b>
Работа с лекционным материалом, с учебной литературой	23

<b>Текущая СРС</b>	<b>Примерная трудоемкость, ач</b>
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	0
Самостоятельное изучение разделов дисциплины	15
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	10
Подготовка к контрольным работам, коллоквиумам	6
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	20
Поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
Работа над междисциплинарным проектом	0
Анализ данных по заданной теме, выполнение расчетов, составление схем и моделей, на основе собранных данных	0
Подготовка к зачету, дифференцированному зачету, экзамену	35
<b>ИТОГО СРС</b>	<b>109</b>

## 5 Учебно-методическое обеспечение дисциплины

### 5.1 Перечень основной и дополнительной литературы, необходимой для освоения дисциплины

№ п/п	Название, библиографическое описание	К-во экз. в библ.
Основная литература		
1	Беляев, Сергей Алексеевич. Разработка игр на языке JavaScript [Текст] : учеб. пособие / С. А. Беляев, 2020. -125, [2] с.	35
2	Программирование интернет -приложений на языке Java [Текст] : метод. указания к лаб. работам / Санкт-Петербургский государственный электротехнический университет им. В.И. Ульянова (Ленина) ”ЛЭТИ”, 2014. -63, [1] с.	30
3	Беляев, Сергей Алексеевич. WEB-технологии. Современные фреймворки [Текст] : лаб. практикум / С. А. Беляев, 2022. -78, [1] с.	38
4	Беляев, Сергей Алексеевич. Web-технологии [Текст] : лаб. практикум / С. А. Беляев, 2019. -75 с.	50
Дополнительная литература		
1	Гоше, Хуан Диего. HTML5 для профессионалов [Текст] / Х. Д. Гоше, 2015. -559 с.	14
2	Уоллс, Крейг. Spring в действии [Текст] / К. Уоллс, 2015. -751 с.	8

### 5.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», используемых при освоении дисциплины

№ п/п	Электронный адрес
1	<a href="http://nginx.org/ru/">nginxhttp://nginx.org/ru/</a>
2	<a href="https://httpd.apache.org/">Apache HTTP Server Projecthttps://httpd.apache.org/</a>
3	<a href="https://developer.mozilla.org/ru/docs/Web/JavaScript">JavaScript Учебные материалыhttps://developer.mozilla.org/ru/docs/Web/JavaScript</a>
4	<a href="https://nodejs.org/">Node.jshttps://nodejs.org/</a>
5	<a href="https://www.npmjs.com/">npmhttps://www.npmjs.com/</a>
6	<a href="https://deno.land/">Denohttps://deno.land/</a>
7	<a href="https://www.w3schools.com/">HTML. The language for building web pageshttps://www.w3schools.com/</a>
8	<a href="https://webref.ru/css">Справочник CSShttps://webref.ru/css</a>
9	<a href="https://lesscss.org/">\textbraceleftless\textbraceright. It’s CSS, with just a little morehttps://lesscss.org/</a>
10	<a href="https://sass-scss.ru/">CSS с суперсилойhttps://sass-scss.ru/</a>
11	<a href="https://www.tutorialspoint.com/expressjs/">ExpressJS Tutorialhttps://www.tutorialspoint.com/expressjs/</a>
12	<a href="https://expressjs.com/ru/">Express. Быстрый, гибкий, минималистичный веб-фреймворк для приложений Node.jshttps://expressjs.com/ru/</a>

№ п/п	Электронный адрес
13	PUG. Getting started <a href="https://pugjs.org/">https://pugjs.org/</a>
14	EJS <a href="https://ejs.co/">https://ejs.co/</a>
15	NestJS <a href="https://nestjs.com/">https://nestjs.com/</a>
16	jQuery <a href="http://jquery.com/">http://jquery.com/</a>
17	TypeScript is JavaScript with syntax for types <a href="https://www.typescriptlang.org/">https://www.typescriptlang.org/</a>
18	Flow is a static type checker for JavaScript <a href="https://flow.org/">https://flow.org/</a>
19	Angular. The modern web developer's platform <a href="https://angular.io/">https://angular.io/</a>
20	React. JavaScript-библиотека для создания пользовательских интерфейсов <a href="https://ru.reactjs.org/">https://ru.reactjs.org/</a>
21	Redux. A Predictable State Container for JS Apps <a href="https://redux.js.org/">https://redux.js.org/</a>
22	Vue.js. Прогрессивный JavaScript-фреймворк <a href="https://v3.ru.vuejs.org/ru/">https://v3.ru.vuejs.org/ru/</a>
23	Should.js <a href="https://shouldjs.github.io/">https://shouldjs.github.io/</a>
24	Mocha. Simple, flexible, fun <a href="https://mochajs.org/">https://mochajs.org/</a>
25	Jest -это фреймворк для тестирования JavaScript с акцентом на простоту <a href="https://jest.js.io/ru/">https://jest.js.io/ru/</a>
26	Selenium automates browsers. That's it! <a href="https://www.selenium.dev/">https://www.selenium.dev/</a>
27	GULP. A toolkit to automate & enhance your workflow <a href="https://gulpjs.com/">https://gulpjs.com/</a>
28	Babel is a JavaScript compiler <a href="https://babeljs.io/">https://babeljs.io/</a>
29	Webpack <a href="https://webpack.js.org/">https://webpack.js.org/</a>
30	Socket.IO. Bidirectional and low-latency communication for every platform <a href="https://socket.io/">https://socket.io/</a>
31	Passport. Simple, unobtrusive authentication for Node.js <a href="http://www.passportjs.org/">http://www.passportjs.org/</a>
32	Sequelize ORM <a href="https://sequelize.org/">https://sequelize.org/</a>
33	WebAssembly <a href="https://webassembly.org/">https://webassembly.org/</a>
34	PHP Tutorial <a href="http://www.tutorialspoint.com/php/">http://www.tutorialspoint.com/php/</a>
35	Основы веб-хакинга. Более 30 примеров уязвимостей <a href="https://leanpub.com/white-hat-hacking-ru">https://leanpub.com/white-hat-hacking-ru</a>

### 5.3 Адрес сайта курса

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=10689>



## 6 Критерии оценивания и оценочные материалы

### 6.1 Критерии оценивания

Для дисциплины «Web-технологии» предусмотрены следующие формы промежуточной аттестации: экзамен.

#### Экзамен

<b>Оценка</b>	<b>Описание</b>
Неудовлетворительно	Курс не освоен. Студент испытывает серьезные трудности при ответе на ключевые вопросы дисциплины
Удовлетворительно	Студент в целом овладел курсом, но некоторые разделы освоены на уровне определений и формулировок теорем
Хорошо	Студент овладел курсом, но в отдельных вопросах испытывает затруднения. Умеет решать задачи
Отлично	Студент демонстрирует полное овладение курсом, способен применять полученные знания при решении конкретных задач.

## Особенности допуска

Допуск к экзамену осуществляется после сдачи курсовой работы, лабораторных работ и тестов текущего контроля успеваемости.

## 6.2 Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине

### Вопросы к экзамену

№ п/п	Описание
1	Что такое протокол HTTP? Какие он поддерживает методы, как передаёт параметры? Для чего предназначены заголовки?
2	Для чего предназначен nginx? Ключевые особенности конфигурирования и использования в качестве прокси. В чём заключается настройка для работы с PHP?
3	Чем ECMAScript отличается от JavaScript? В чём ключевые отличия ECMAScript 5 от более новых версий? Расскажите особенности работы с массивами.
4	Опишите ключевые особенности обещаний, генераторов, Map, WeakMap, прокси, async/await.
5	Для чего нужны npm и yarn? Перечислите основные команды ими поддерживаемые. Приведите примеры стандартных пакетов и особенностей их применения. Чем отличается Deno?
6	Что такое ORM? Какие ORM вы знаете для JavaScript? Приведите примеры использования.
7	Приведите пример создания эхо-сервера на Node.js. Приведите способы и примеры журналирования в Node.js?
8	В чём отличие асинхронного выполнения кода от синхронного? Приведите соответствующие примеры при работе с файлами. Как выполнить «асинхронный try-catch» (обработку ошибок в асинхронном режиме)?
9	Опишите структуру HTML, понятие DOM-модели. Как осуществляется обработка событий в HTML?
10	Опишите, как использовать CSS. Что такое псевдоклассы, селекторы? Приведите примеры. Что такое динамическая вёрстка и как она выполняется?
11	Зачем нужны фреймворки LESS, SASS? Опишите нотацию и особенности применения по назначению.
12	Какие элементы HTML используются в <FORM> для взаимодействия с пользователем? Как осуществляется валидация введённых значений?
13	С помощью чего можно создавать шаблоны web-страниц? Как использовать средства шаблонизации совместно с модулем Express?
14	Как можно выполнить авторизацию и аутентификацию? Что такое Cookies, как они работают и используются? Опишите применение Ajax.
15	Опишите реализацию REST на Node.js.
16	Как можно работать с историей в браузере? В чём отличия sessionStorage, localStorage и Cookies? Приведите примеры.

17	Опишите для чего нужны и как используются web-сообщения, web-worker, web-сокеты.
18	Опишите конфигурирование проекта с использованием GULP.
19	Опишите конфигурирование проекта с использованием Webpack.
20	Опишите применение и приведите примеры использования jQuery (селекторы, манипуляция с DOM, обработка событий, CSS, эффекты).
21	Опишите применение фреймворка Angular (структура приложения, компоненты, сервисы, директивы, задание маршрутов).
22	Опишите применение React. Что такое JSX? Как осуществляется рендеринг в React? Приведите примеры.
23	Опишите применение фреймворка Vue (структура приложения, компоненты, задание маршрутов).
24	Опишите особенности и приведите примеры использования TypeScript (типы переменных, интерфейсы, классы, функции, шаблоны, перечисления, импорт-экспорт, пространства имён).
25	Опишите применение и приведите примеры использования Flow.
26	Опишите особенности и приведите примеры использования Immutable.js и Redux.
27	Опишите применение React. Опишите компоненты и свойства в React, обработку событий. Приведите примеры.
28	Опишите применение React. Опишите работу с формами в React. Приведите примеры.
29	Опишите область применения PHP, основные элементы языка, особенности обработки запросов.
30	Приведите описание понятия TDD-разработки. Опишите применение и приведите примеры Assert, Should и Chai.
31	Приведите описание понятия TDD-разработки. Опишите фреймворк Mocha.
32	Приведите описание понятия TDD-разработки. Опишите фреймворк Jest.
33	Опишите применение фреймворка NestJS (структура приложения, компоненты, сервисы, директивы, задание маршрутов).
34	Опишите как обеспечивается безопасность web-приложений от взлома.
35	Что такое WebAssembly? Как это работает? Приведите примеры использования.

## Форма билета

Министерство науки и высшего образования Российской Федерации  
 ФГАОУ ВО «Санкт-Петербургский государственный электротехнический  
 университет «ЛЭТИ» имени В.И. Ульянова (Ленина)»

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

Дисциплина **Web-технологии** ФКТИ

1. Что такое протокол HTTP? Какие он поддерживает методы, как передаёт параметры? Для чего предназначены заголовки?

2. Опишите структуру HTML, понятие DOM-модели. Как осуществляется обработка событий в HTML?

УТВЕРЖДАЮ

Заведующий кафедрой

К. В. Кринкин

### **Образцы задач (заданий) для контрольных (проверочных) работ**

Тесты размещены в курсе Moodle, примеры вопросов:

1. Сопоставьте коды состояний HTTP их описаниям
2. В чём преимущество HTTP/2 по сравнению с предыдущими версиями протокола?
3. Какой HTTP метод используется для описания параметров соединения с ресурсом?
4. Какими функциями можно управлять с помощью HTTP?
5. Выберите соответствие команд для управления модулями с помощью `pm` и `uapn`.

Весь комплект контрольно-измерительных материалов для проверки сформированности компетенции (индикатора компетенции) размещен в закрытой части по адресу, указанному в п. 5.3

### 6.3 График текущего контроля успеваемости

Неделя	Темы занятий	Вид контроля
2	Тема 1. Введение	
3	Тема 2. Основы языка JavaScript	Отчет по лаб. работе
4	Тема 3. Применение Node.js. Основы Deno	
5	Тема 4. Основы HTML	Отчет по лаб. работе
6	Тема 1. Введение Тема 2. Основы JavaScript Тема 3. Применение Node.js. Основы Deno Тема 4. Основы HTML Тема 5. Основы CSS, LESS, SASS	Тест
7	Тема 5. Основы настройки стилей CSS, LESS, SASS Тема 6. Применение Express, NestJS, RESTful, Ajax Тема 7. Основы использования библиотек jQuery и jQuery UI	Отчет по лаб. работе
8	Тема 2. Основы языка JavaScript Тема 4. Основы HTML Тема 3. Применение Node.js. Основы Deno	Защита КР / КП
9	Тема 8. Основы языка TypeScript. Статический контроль с использованием Flow Тема 9. Основы работы с фреймворком Angular Тема 6. Применение Express, NestJS, RESTful, Ajax Тема 5. Основы настройки стилей CSS, LESS, SASS	Отчет по лаб. работе
10	Тема 10. Использование библиотеки React Тема 6. Применение Express, NestJS, RESTful, Ajax Тема 5. Основы настройки стилей CSS, LESS, SASS	Отчет по лаб. работе
11	Тема 11. Основы работы с фреймворком Vue Тема 8. Основы языка TypeScript. Статический контроль с использованием Flow Тема 6. Применение Express, NestJS, RESTful, Ajax	Отчет по лаб. работе
12	Тема 6. Применение Express, NestJS, RESTful, Ajax Тема 7. Основы использования jQuery и jQuery UI Тема 8. Основы TypeScript. Статический контроль с Flow Тема 9. Основы работы с Angular	Тест
14	Тема 10. Использование библиотеки React	
15	Тема 11. Основы работы с Vue Тема 12. Тестирование web-приложений Тема 13. Сборка приложений, новые возможности JavaScript	Тест

### 6.4 Методика текущего контроля

Текущий контроль на лекционных занятиях включает в себя выполнение тестов по темам лекций. Тесты размещены в Moodle вуза и оцениваются автоматически, максимальная оценка за тест - 5 баллов ("отлично"). Успешное

прохождение тестирования - не менее 3 баллов ("удовлетворительно").

В процессе обучения по дисциплине «Web-технологии» студент обязан выполнить 8 лабораторных работ. Под выполнением лабораторных работ подразумевается подготовка к работе, разработка web-приложения, подготовка отчета и его защита. Выполнение лабораторных работ студентами осуществляется индивидуально. Оформление отчета студентами осуществляется индивидуально в соответствии с принятыми в СПбГЭТУ правилами оформления студенческих работ. Отчет оформляется после выполнения лабораторной работы и представляется преподавателю на проверку. После проверки отчет либо возвращается (при наличии замечаний) на доработку, либо подписывается к защите.

Лабораторные работы защищаются студентами индивидуально. Каждый студент получает вопрос по теоретической части, или по разработанному web-приложению после чего ему предоставляется время для подготовки ответа. При обсуждении ответа преподаватель может задать несколько уточняющих вопросов. В случае если студент демонстрирует достаточное знание вопроса, работа считается защищенной.

На защите лабораторной работы студент должен показать: понимание использованных web-технологий, особенности их применения, понимание и умение объяснять особенности применяемых методов, возможные области их применения и т.д., умение вносить изменения в web-приложения в соответствии с изменением требований заказчика.

Текущий контроль включает в себя выполнение, сдачу в срок отчетов и их защиту по всем лабораторным работам, по результатам которой студент получает допуск на экзамен. При защите лабораторной работы проверяется выполнение всех требований к лабораторным работам в соответствии с методическими указаниями к лабораторным работам. Работы, выполненные не в полном объеме, не принимаются.

Защита лабораторной работы в первые 12 недель курса - оценка "отлично" за лабораторную работу.

Защита лабораторной работы в с 13 по 14 неделю курса - оценка "хорошо" за лабораторную работу.

Защита лабораторной работы в после 14 недели курса - оценка "удовлетворительно" за лабораторную работу.

Контроль самостоятельной работы студентов осуществляется на лекционных и лабораторных занятиях студентов по методикам, описанным выше.

Текущий контроль при выполнении курсового проекта (работы) осуществляется в соответствии с методическими указаниями по курсовом проектированию и заданием на курсовой проект (работу). При защите курсовой работы проверяется выполнение всех требований в соответствии с методическими указаниями и выполнение требований, приведённых в п.4.4.

Защита курсовой работы в первые 12 недель курса - оценка "отлично" за курсовую работу.

Защита курсовой работы в с 13 по 14 неделю курса - оценка "хорошо" за курсовую работу.

Защита курсовой работы после 14 недели курса - оценка "удовлетворительно" за курсовую работу.

Оформление пояснительной записки на курсовой проект (работу) выполняется в соответствии с требованиями к студенческим работам принятым в СПбГЭТУ.

Защита курсового проекта (работы) осуществляется в соответствии с требованиями «Положения о промежуточной аттестации».

## 7 Описание информационных технологий и материально-технической базы

Тип занятий	Тип помещения	Требования к помещению	Требования к программному обеспечению
Лекция	Лекционная аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, компьютер или ноутбук с возможностью подключения к сети "Интернет", проектор, экран, меловая или маркерная доска.	1) Windows 10 и выше; 2) Microsoft Office 2013 и выше.
Лабораторные работы	Лаборатория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, компьютер или ноутбук с возможностью подключения к сети "Интернет".	1) Ubuntu 16.04 LTS и выше; 2) LibreOffice или OpenOffice; 3) JetBrains WebStorm; 4) Node.JS 16 и выше; 5) TypeScript.
Самостоятельная работа	Помещение для самостоятельной работы	Оснащено компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета.	1) Ubuntu 16.04 LTS и выше; 2) LibreOffice или OpenOffice; 3) JetBrains WebStorm; 4) Node.JS 16 и выше; 5) TypeScript.



## **8 Адаптация рабочей программы для лиц с ОВЗ**

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

<b>№ п/п</b>	<b>Дата</b>	<b>Изменение</b>	<b>Дата и номер протокола заседания УМК</b>	<b>Автор</b>	<b>Начальник ОМОЛА</b>