МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИИСКОИ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»

УТВЕРЖДАЮ:
Директор института
энергетики и автоматизированных
систем
_____С.И.Лукьянов
«27» сентября 2017 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Практикум по разработке web-приложений
Направление подготовки (специальность)

44.03.05 Педагогическое образование (с двумя профилями подготовки)

Профиль подготовки (специализация)
Информатика и экономика

Квалификация (степень) выпускника—бакалавр

Форма обучения— очная

Институт	Институт энергетики и автоматизированных систем
Кафедра	Бизнес - информатики и информационных технологий
Курс	1-3
Семестр	1-6

Магнитогорск
2017 г.

Рабочая программа составлена на основе ФГОС ВОпо направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки), утвержденное приказом МОиН РФ от 9 февраля 2016г., №91.

Рабочая программа рассмотрена и одобрена на заседании кафедры бизнес-информатики информационных технологий «21» сентября 2017, протокол №2.

Зав. кафедрой  Г.Н. Чусавитина

Рабочая программа одобрена методической комиссией института энергетики и автоматизированных систем «27» сентября 2017, протокол №2.

Председатель  С.И. Лукьянов

Рабочая программа составлена: доцент каф. БИиИТ, к.п.н., доцент

 Е.В. Карманова

Рецензент: директор МОУ СОШ № 33, к.п.н.

 Шманева И.В.

Лист регистрации изменений и дополнений

№ п/п	Раздел программы	Краткое содержание изменения/дополнения	Дата. № протокола заседания кафедры	Подпись зав. кафедрой
1	3,4,7,8,9	Корректировка РПД в соответствии с новым макетом (распоряжение № 10-39/75 от 21.09.2018 «О формировании и актуализации образовательных программ). Актуализация информационно-методического и информационного обеспечения дисциплины. Актуализация материально-технического обеспечения дисциплины	25.09.18, протокол № 2	
2	8,9	О формировании и актуализации образовательных программ. Актуализация информационно-методического и информационного обеспечения дисциплины. Актуализация материально-технического обеспечения дисциплины	02.09.19, протокол № 1	
3	8	Актуализация учебно-методического и информационного обеспечения дисциплины	31.08.20, протокол №1	

1 Цели освоения дисциплины

Целями освоения дисциплины «Практикум по разработке Web-приложений» являются: знакомство студентов с базовыми концепциями и приемами Web-программирования, получение представления о современных Web-технологиях, о подходах к проектированию, разработке, отладке, оптимизации и развертыванию web-приложений с динамичным контентом.

2 Место дисциплины в структуре образовательной программы подготовки бакалавра

Дисциплина «Практикум по разработке Web-приложений» является факультативной в образовательной программе.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения: «Вычислительные системы, сети, телекоммуникации», «Программирование», «Информационные системы и технологии», «Технологии баз данных и СУБД».

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы при изучении: «Создание и редактирование образовательных сайтов», «Разработки интернет-приложения образовательного назначения».

3 Компетенции обучающегося, формируемые в результате освоения дисциплины и планируемые результаты обучения

В результате освоения дисциплины «Практикум по разработке Web-приложений» обучающийся должен обладать следующими компетенциями:

Структурный элемент компетенции	Планируемые результаты обучения
ОК-3 – способностью использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве	
Знать	• World Wide Web Consortium (W3C) стандарты HTML и CSS; • Технологии передачи и обмена данными в компьютерных сетях; • Принципы структурного и модульного программирования; • Принципы отладки и тестирования программных продуктов; • Принципы объектно – ориентированного программирования; • Принципы функционирования виртуального сервера; • Средства защиты программного обеспечения в компьютерных системах.
Уметь	• Выполнять отладку и тестирование программы на уровне модуля; • Создавать веб-сайты полностью соответствующие текущим стандартам W3C (http://www.w3.org); • Разрабатывать безопасное веб-приложение;
Владеть	• Принципами работы протокола HTTP(S) • Современными средствами разработки, отладки и тестирования интернет приложений • Приемами обеспечения безопасности интернет приложений (устойчивость веб-приложения к атакам и взлому);
ОК-5- способностью работать в команде, толерантно воспринимать социальные, культурные и личностные различия	

Структурный элемент компетенции	Планируемые результаты обучения
Знать	• Методы организации командной работы, знает о преимуществах командной работы
Уметь	• самостоятельно применять способы командного взаимодействия, предусматривающего толерантное восприятие социальных, культурных и личностных различий • организовывать командную работу при разработке проекта
Владеть	• навыками организации командной работы
ДПК-1 - способен использовать математический аппарат, методологию программирования и современные компьютерные технологии для решения практических задач получения, хранения, обработки и передачи информации	
Знать	• Элементы и конструкции языка JavaScript и способы их применения для построения клиентских сценариев; • Элементы и конструкции языка PHP и способы их применения для построения серверных сценариев; • Сущность, назначение и структуру объектной модели браузера и документа; • Принципы разработки веб-сервисов с применением PHP, XML (Extensible Markup Language) и JSON • Правила составления запросов SQL
Уметь	• Проектировать WEB – документ и работать с базовыми его элементами; • Создавать клиентские сценарии, осуществлять их внедрение в проект и тестирование; • Создавать html-страницы сайта на основе предоставленных графических макетов их дизайна • Разрабатывать анимацию для веб-сайта для повышения его доступности и визуальной привлекательности; • Использовать современные фреймворки и открытые библиотеки при разработке интернет приложений • Разрабатывать веб-приложения с доступом к базе данных MySQL и веб-сервисы по требованиям клиента;
Владеть	• Навыками работы с HTML/CSS на базовом уровне • Навыками работы с JavaScript, PHP на базовом уровне • Приемами интеграции существующего программного кода с API (Application Programming Interfaces), библиотеками и фреймворками • Навыками создания веб-приложения с доступом к БД.

4 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 9 зачетных единиц 324 акад. часов, в том числе:

- контактная работа – 202.3 акад. часов:
  - аудиторная – 202 акад. часов;
  - внеаудиторная – 0.3 акад. часов
- самостоятельная работа – 121.7 акад. часов;

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
Раздел 1. Основы HTML. Каскадные таблицы стилей - CSS	1							
1.1 Структура HTML-документа. Правила оформления HTML-документа. Элементы и атрибуты.	1		2		1	Изучение литературы, выполнение лабораторных и практических заданий	Контроль выполнения лабораторной работы	ОК-3-зув
1.2 Ссылки и изображения в HTML	1		2		1	Изучение литературы, выполнение лабораторных и практических заданий	Контроль выполнения лабораторной работы	ОК-3-зув
1.3 Списки: одноуровневые, многоуровневые	1		2		1	Изучение литературы, выполнение лабораторных и практических заданий	Контроль выполнения лабораторной работы	ОК-3-зув
1.4 Таблицы. Объединение ячеек. Вложенные таблицы.	1		2		1	Изучение литературы, выполнение	Контроль выполнения лабораторной работы	ОК-3-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
						лабораторных и практических заданий		
1.5 Формы и фреймы	1		2		1	Изучение литературы, выполнение лабораторных и практических заданий	Контроль выполнения лабораторной работы	ОК-3-зув
1.6 HTML5. Применение новых семантических элементов.	1		2		1	Изучение литературы, выполнение лабораторных и практических заданий	Контроль выполнения лабораторной работы	ОК-3-зув
1.7 CSS. Способы добавления. Виды селекторов. Комплексные селекторы.	1		2		1	Изучение литературы, выполнение лабораторных и практических заданий	Контроль выполнения лабораторной работы	ОК-3-зув
1.8 Позиционирование	1		2		1	Изучение литературы, выполнение лабораторных и практических заданий	Контроль выполнения лабораторной работы	ОК-3-зув
1.9 Наследование, каскадность	1		2		1	Изучение литературы, выполнение лабораторных и практических заданий	Контроль выполнения лабораторной работы	ОК-3-зув
1.10 Единицы измерения, цвет, шрифты	1		2		1	Изучение литературы, выполнение лабораторных и прак-	Контроль выполнения лабораторной работы	ОК-3-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
						тических заданий		
1.11 Возможности CSS3 (flexbox, columns, анимация, трансформация, медиа-дапросы)	1		4		2	Изучение литературы, выполнение лабораторных и практических заданий	Контроль выполнения лабораторной работы	ОК-3-зув
1.12 Использование CSS-препроцессоры. LESS	1		6		3	Изучение литературы, выполнение лабораторных и практических заданий	Контроль выполнения лабораторной работы	ОК-3-зув
1.13 Использование SVG	1		6		3	Изучение литературы, выполнение лабораторных и практических заданий	Контроль выполнения лабораторной работы	ОК-3-зув
Итого по разделу			**36**		**18**		Контрольное тестирование	
Итого за семестр			**36**		**18**		-	
Раздел 2. JavaScript - язык разработки клиентских веб-приложений	2							
2.1 Подключение js-скриптов к HTML-страницам. Работа с инструментами разработчика.	2		2			Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
2.2 Переменные. Типы данных JavaScript	2		2			Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
2.3 Реализация основных алгоритмических конструкций в JavaScript	2		2			Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
2.4 Строки. Массивы в JavaScript	2		4			Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
2.5 Функции. Замыкания	2		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
2.6 Объекты в JavaScript	2		2		4	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
2.6 BOM, DOM	2		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
2.7 Обработка форм. Регулярные выражения	2		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
2.8 События в JavaScript	2		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
2.9 Форматы JSON, XML. Сохранение данных в LocalStorage	2		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
2.10 Ajax	2		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
2.11 Работа с Canvas	2		2		2	Выполнение лабораторной работы Выполнение проектного задания	Контроль выполнения лабораторной работы	ДПК-1-зув ОК-5-зув
2.12 JS-анимация	2		2			Выполнение лабораторной работы Выполнение проект-	Контроль выполнения лабораторной работы	ДПК-1-зув ОК-5-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
						ного задания		
2.13 ООП в JavaScript	2		6		1.9	Выполнение лабораторной работы Выполнение проектного задания	Контроль выполнения лабораторной работы. Отчет по проектному заданию	ДПК-1-зув ОК-5-зув
Итого по разделу			**34**		**19.9**		**Контрольное тестирование**	
Итого за семестр			**34**		**19.9**		**зачет**	
Раздел 3. Фреймворки JavaScript	3							
3.1 jQuery. Подключение jQuery. Селекторы	3		**2**			Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
3.2 jQuery методы DOM	3		2			Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
3.3 jQuery события, эффекты	3		6		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
3.4 jQuery Ajax	3		2			Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
3.5 Возможности Vue.js/React.js	3		8		4	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
3.6 Разработка интерфейса игрового веб-приложения	3		4		4	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
3.7. Реализация логики игрового веб-приложения	3		10		6	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув ОК-5-зув
3.8. Тестирование игрового приложения	3		2		2	Выполнение лабораторной работы Выполнение проектного задания	Контроль выполнения лабораторной работы. Отчет по проектному заданию	ДПК-1-зув ОК-5-зув
Итого по разделу			**36**		**18**			
Раздел 4. PHP	4							
4.1Установка и настройка Open Server. Работа с Open Server portable	4		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
4.2 Подключение скрипта. Объявление переменных, констант. Реализация основных алгоритмических конструкций.	4		4		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы. Отчет по проектному заданию	ДПК-1-зув
4.3 Строки	4		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
4.4 Массивы	4		4		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
4.5 GET и POST	4		4		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
4.6 Работа с файлами	4		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
4.7 Функции, области видимости. Встроенные функции php	4		4		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
4.8 Использование регулярных выражений. Проверка на валидность данных	4		2		1.9	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
4.9 Куки, сессии	4		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
4.10 ООП	4		4		2	Выполнение лабораторной работы Выполнение проектного задания	Контроль выполнения лабораторной работы. Отчет по проектному заданию	ДПК-1-зув
Итого по разделу			**30**		**23.9**		**Контрольное тестирование**	
Итого за семестр			**30**		**23.9**		**Зачет**	
Раздел 5. СУБД MySQL								
5.1 Работа в IDE PHPStorm.	5		2			Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
5.2 MySQLAdmin.Настройка. Создание БД.	5		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
5.3 Создание таблиц в БД. Ключи	5		4		4	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
5.4 Язык запросов Mysql	5		8		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
5.5 Подключение к БД.	5		4		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
5.6 Расширение PDO	5		6		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
5.7 Защита информации. Шифрование данный. Проверка на SQL-инъекции	5		6		4	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
5.8 Ajax обращение к БД. Формат Json, XML	5		4		2	Выполнение лабораторной работы Выполнение проектного задания	Контроль выполнения лабораторной работы. Отчет по проектному заданию	ДПК-1-зув
Итого по разделу			**36**		**18**		**Контрольное тестирование**	
Итого за семестр			**36**		**18**		-	
Раздел 6. Фреймворк PHP- Laravel	6							
6.1 Паттерн MVC.	6		2			Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
6.2 Фреймворк. Установка. Виртуальная машина Homestead. Работа с composer. Настройка окружения.	6		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
6.3 Работа в консоли. Создание Laravel- приложения. Архитектура Laravel- приложения	6		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
6.4 Маршрутизация, представления в Laravel. Перенаправления	6		2			Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
6.5 Работа с БД. Миграции.	6		4		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
6.6 Работа с шаблонизатором. Blade	6		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
6.7 Индексный контроллер и маршрут.	6		2		5.9	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
6.8 Eloquent ORM	6		4			Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
6.9 Валидация в Laravel. Формы	6		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
6.10 Локализация дат. Carbon	6		2		2	Выполнение лабораторной работы	Контроль выполнения лабораторной работы	ДПК-1-зув
6.11 Настройка электронной почты	6		2		2	Выполнение лабораторной работы Выполнение проектного задания	Контроль выполнения лабораторной работы	ДПК-1-зув
6.12 Реализация регистрации и авторизации пользователей в Laravel-приложении	6		4		4	Выполнение лабораторной работы Выполнение проектного задания	Контроль выполнения лабораторной работы. Отчет по проектному заданию	ДПК-1-зув
Итого по разделу			**30**		**23.9**			

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
Итого за семестр			**30**		**23.9**		**Зачет с оценкой**	
Итого по дисциплине			**202**		**121.7**		**Зачет с оценкой**	

5 Образовательные и информационные технологии

В ходе проведения занятий предусматривается следующие образовательные технологии:

1. Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.
2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.
3. Практическое занятие в форме практикума – организация учебной работы, направленная на решение комплексной учебно-познавательной задачи, требующей от студента применения как научно-теоретических знаний, так и практических навыков.
4. Практическое занятие на основе кейс-метода – обучение в контексте моделируемой ситуации, воспроизводящей реальные условия научной, производственной, общественной деятельности. Обучающиеся должны проанализировать ситуацию, разобраться в сути проблем, предложить возможные решения и выбрать лучшее из них. Кейсы базируются на реальном фактическом материале или же приближены к реальной ситуации.
5. Информационно-коммуникационные образовательные технологии – организация образовательного процесса, основанная на применении специализированных программных сред и технических средств работы с информацией.
6. Формы учебных занятий с использованием информационно-коммуникационных технологий:
7. Практическое занятие в форме презентации – представление результатов проектной или исследовательской деятельности с использованием специализированных программных сред.

В рамках практических занятий предусматривается использование средств вычислительной техники при выполнении индивидуальных заданий. Используется существующий образовательный портал университета (newlms.magtu.ru) для размещения ЭУМК по дисциплине. Текущий, промежуточный и рубежный контроль проводится на образовательном портале университета.

Основной образовательной технологией данного практикума является метод проектов. Каждый студент имеет изначально заданную тематику проекта, который он разрабатывает по мере изучения новых тем курса.

6 Учебно-методическое обеспечение самостоятельной работы студентов

По дисциплине предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов предполагает поиск решений ответов на

Внеаудиторная самостоятельная работа обучающихся осуществляется в виде изучения литературы по соответствующему разделу с проработкой материала; выполнения лабораторных и проектных заданий

Тематика	Проекты по дисциплине
Раздел 1. Основы HTML. Каскадные таблицы стилей - CSS	Адаптивный сайт определенной тематики (тематику выбирает студент самостоятельно)
Раздел 2. JavaScript - язык разработки клиентских веб-приложений	Динамический сайт
Раздел 3. Фреймворки JavaScript	Игровое веб-приложение
Раздел 4. PHP	Разработанный интерфейс будущего интернет-сервиса
Раздел 5. СУБД MySQL	Работающий интернет сервис
Раздел 6. Фреймворк PHP- Laravel	Блог, созданный с использованием фреймворка

Примерная тематика проектов:

Игровое веб-приложение:

- Арканоид
- Змейка
- Космический звездолет
- Гонки
- Марио
- Мыльные пузыри
- Аквариум

Интернет-сервис по записи на услугу (кинотеатр, парикмахерская, автосервис, медицинский центр, репетиторство и т.д).

7 Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
ОК-3 – способностью использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве		
Знать	• World Wide Web Consortium (W3C) стандарты HTML и CSS; • Технологии передачи и обмена данными в компьютерных сетях; • Принципы структурного и модульного программирования; • Принципы отладки и тестирования программных продуктов; • Принципы объектно – ориентированного программирования; • Принципы функционирования виртуального сервера; • Средства защиты программного обеспечения в компьютерных системах.	• Объектная модель документов (DOM): принципы использования. • Объектная модель браузера (BOM): объекты, их свойства и методы. • Укажите основные отличия форматов XML и JSON. • Реализация ООП в JavaScript. • Обзор JavaScript фреймворков • Настройка веб-сервера. Виртуальные хосты.
Уметь	• Выполнять отладку и тестирование программы на уровне модуля; • Создавать веб-сайты полностью соответствующие текущим стандартам W3C (http://www.w3.org); • Разрабатывать безопасное веб-приложение;	**Практическое задание №5 JQuery** Цель: изучить возможности библиотеки JQuery. Задание: реализовать любые 4 эффекта из предложенных. Разрешено использовать готовые решения из библиотеки Jquery. На сайт добавить контент в виде отдельных страниц. Минимальное количество страниц сайта - 10.

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		Всплывающие подсказки 2. Карусель изображений 3. Всплывающие изображения для миниатюр 4. Эффект падающих листьев, снега, дождя и т. Д. 5. Показ текущего времени, включая день недели. 6. Отсчёт времени до какой либо знаменательный даты, праздника. 7. Кнопка наверх. 8. Меню сайта. 9. Голосование на сайте. 10. Всплывающее окно. 11. Подсветка текста при наведении. 12. Увеличение текста при наведении. 13. Анимированный логотип 14. Виджет интерактивный календарь для поля с форматом дата. 15. Аккордеон для текста. Дополнительно: продемонстрировать работу скрипта с помощью инструмента разработчиков Google Chrome **Практическое задание №6 Установка и настройка Openserver** Задача: Научиться работать в OpenServer. Изучить правила объявления php скрипта. Задание: 1 этап Для выполнения последующих работ нам необходимо сделать небольшой веб-проект, в котором и будем размещать выполненные лабораторные работы. Размещать этот проект

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		будем на локальном компьютере с помощью «OpenServer». В каталоге «domains» (для разных версий может отличаться) установленного «OpenServer» создадим папку «StudyKarmanova». Вместо Karmanova вы должны указать свою фамилию, написанную английскими буквами. В этом каталоге мы будем размещать все файлы и подкаталоги, созданные во время выполнения лабораторных работ. После создания каталога необходимо перезапустить OpenServer, как это делается – читайте в документации к программе. https://ospanel.io/ 2 этап Создать файл *.php демонстрирующий интеграцию кода html с php 3 этап Продемонстрировать работу операторов if, for, while, foreach. Составить программу, которая печатает таблицу умножения.
Владеть	• Принципами работы протокола HTTP(S) • Современными средствами разработки, отладки и тестирования интернет приложений • Приемами обеспечения безопасности интернет приложений (устойчивость веб-приложения к атакам и взлому);	**Практическое задание №7 GET и POST. Валидация данных.** Задачи: • Изучить правила работы с файлами в php • Методы GET и POST • Познакомиться с методами проверки корректности данных отправленных на сервер. Задание: Напишите скрипт, который будет сохранять на сервере в виде текстового файла данные, которые пользователь вводит в форму на сайте. Замечание: реализуйте проверку на корректность ввода данных на стороне сервера.

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		Практическое задание № 8 Работа с MySQL. Язык запросов. Задачи: • Изучить графический интерфейс СУБД MYSQL • Познакомиться с основными запросами SQL Задание: • Спроектируйте БД для вашего веб-ресурса. • В БД должно быть не менее 3-х таблиц, связанных между собой. • Заполните таблицы записями, не менее 10 записей в каждой таблице. • Попробуйте реализовать запросы к своим таблицам SELECT, INSERT, DELETE. **Практическое задание №9 Работа с БД** Задачи: • Изучить правила подключения к БД с помощью PHP • Основные команды работы с БД • Команды защиты данных. Задание: Реализовать сохранение данных в БД, отправленных с сайта. Реализовать вывод данных из БД на страницы сайта. Реализовать шифрование данных в БД. Реализовать технологию ajax для одного из элементов сайта.
ОК-5- способностью работать в команде, толерантно воспринимать социальные, культурные и личностные различия		
Знать	• Методы организации командной работы, знает о преимуществах командной работы	Контрольные вопросы: • Основные роли участников команды при разработке web-приложений • Методы управления командой

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		• Способы регулирования проблемных ситуаций в команде при разработке web-приложений • Жизненный цикл web-приложений
Уметь	• самостоятельно применять способы командного взаимодействия, предусматривающего толерантное восприятие социальных, культурных и личностных различий • организовывать командную работу при разработке проекта	Анкетирование (Проводится в начале проекта, все студенты отвечают на вопросы анкетирования, далее обсуждают полученные ответы): 1. Готовы ли вы стать лидером команды при разработке проекта? 2. Какой тип управления командой для вас является эффективным? 3. С каким количеством участников в команде вам комфортно работать? 4. Какими чертами должен обладать, на ваш взгляд, лидер команды?
Владеть	• навыками организации командной работы	**Проектное задание:** Необходимо команде студентов из 3-х человек разработать интернет-сервис по одной из ниже предложенных тематик: 1. Необходимо создать службу, предоставляющую доступ к информации о курсах валют, которая собирается нашим приложением, и накапливается в базе данных. Далее посредством веб-сервиса, данная информация передается сторонним приложениям для отображения в удобном для них виде. 2. Необходимо разработать сервис поиска арендных мест в торговых центрах города, представление информации об условиях аренды, наличие фильтров поиска (стоимость, район, площадь, минимальное время аренды), реализовать возможность подачи заявки на аренду. 3. Необходимо реализовать сервис поиска лучшего мастера красоты (визаж, ногтевой сервис и т.д.), мастера по ремонту

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		оргтехники (авто) с графиком работы мастеров, расписанием свободных дат, стоимости услуг, он-лайн запись к мастеру, также реализовать рейтинг и возможность оставить отзыв.
ДПК-1 - способен использовать математический аппарат, методологию программирования и современные компьютерные технологии для решения практических задач получения, хранения, обработки и передачи информации		
Знать	• Элементы и конструкции языка JavaScript и способы их применения для построения клиентских сценариев; • Элементы и конструкции языка PHP и способы их применения для построения серверных сценариев; • Сущность, назначение и структуру объектной модели браузера и документа; • Принципы разработки веб-сервисов с применением PHP, XML (Extensible Markup Language) и JSON • Правила составления запросов SQL	1. Основные элементы HTML. 2. Приведите базовую структуру HTML-документа. 3. Перечислите основные способы включения каскадных таблиц стилей в HTML-документ. Приведите примеры. 4. Семантические теги в HTML5. 5. Укажите основные типы селекторов каскадных таблиц стилей. 6. Перечислите основные свойства каскадных таблиц стилей. 7. Возможности CSS3 8. Работа с SVG графикой. 9. Препроцессор Less. 10. Приведите синтаксис SSI-директив. 11. Перечислите основные способы включения скриптов JavaScript в HTML-документ. Приведите примеры. 12. Синтаксис JavaScript. 13. Элементы form. Обработка данных формы на валидность. 14. Правило объявления самовызывающийся (анонимной) функции в JavaScript. 15. Замыкания в JavaScript. 16. Правила использования jQuery. 17. Реализация Ajax с использованием jQuery 18. Скриптовые языки. Структура файла со скриптами на

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		языке PHP. 19. PHP. Синтаксис языка 20. Оператор условия. Переключатель. 21. Операторы цикла с предусловием и с постусловием. 22. Оператор цикла с заданным числом повторений for. Операторы управления циклом. 23. Массивы. Оператор foreach. 24. Операции с массивами. 25. Функции для добавления и удаления элементов массива. 26. Многомерные, ассоциативные массивы. 27. Строки. Строковые функции. 28. Работа с файлами в PHP. 29. Функции для работы с каталогами: getcwd(), opendir(), readdir(), scandir(), chdir(). 30. Функции. Фактические и формальные параметры. Вызов функции с переменным числом параметров. 31. Пространство имён. Локальные, глобальные и суперглобальные переменные. 32. Работа с регулярными выражениями. 33. Приём параметров из браузера. Суперглобальные массивы $_GET, $_POST и $_REQUEST. 34. Работа с куки и сессими.
Уметь	• Проектировать WEB – документ и работать с базовыми его элементами; • Создавать клиентские сценарии, осуществлять их внедрение в проект и тестирование; • Создавать html-страницы сайта на основе предоставленных графических макетов их дизайна • Разрабатывать анимацию для веб-сайта для повышения его доступности и визуальной привлека-	**Практическое задание №1 HTML** Разработайте сайт рассказывающий о вашем хобби. Сайт должен состоять из 5-7 страниц связанных между собой гиперссылками и включать работу с графикой, таблицами, элементами форматирования текста, якорями, списками и ссылками. Постарайтесь применить как можно больше изученных атрибутов для используемых элементов. **Практическое задание №2 CSS**

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
	тельности; • Использовать современные фреймворки и открытые библиотеки при разработке интернет приложений • Разрабатывать веб-приложения с доступом к базе данных MySQL и веб-сервисы по требованиям клиента;	Изучить практическое пособие по CSS. Модифицировать разработанный в Задании 1 сайт таким образом, чтобы все элементы оформления сайта были вынесены в отдельный .css файл. Используйте как можно больше селекторов CSS. Требование: наличие меню, шапки, подвала, блока с дополнительной информацией, блока с контентом; наличие блоков параллельно расположенных друг относительно друга. 1. Название HTML и CSS файлов ТОЛЬКО латинскими буквами, рекомендуется - index, home, pageN, news, style ....) 2. Наличие файла css с внешним представлением сайта. 3. Все графические изображения должны лежать в отдельной папке Images 4. Структура страницы сайта обязательно содержит следующие разделы: шапка, меню, контент, футер. Разрешается также дополнительно размещать требуемые блоки. 5. В блоке head - наличие фавикона, мета данных (ключевые слова, описание, кодировка), title. 6. Реализация 5 (на выбор) эффектов с помощью CSS: • Градиент • Тень • Скругленные уголки элементов (блоков/изображений) • Анимация • Декоративные рамки • Полупрозрачный фон с картинкой • Спрайт • Декорирование текста • Изменение внешнего вида курсора • Декорирование списков • Текст в несколько колонок 7. Реализация резиновой верстки.

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		8. Структура кода с табуляцией - елочка. 9. Применить все способы объявления css. Полезный Интернет-ресурс - https://html5book.ru/css-css3/ **Практическое задание №3. Адаптивная верстка. Знакомство с JS** Задачи: реализация адаптивности сайта, добавление формы. В качестве ответа загрузить на портал архив, содержащий файл HTML, CSS, папку с используемыми изображениями. Требования: 1. Используя медиа запросы реализовать адаптивность сайта для следующих размеров и видов ориентации: • desctop (свыше 1280px) • планшет (800px - портретная) • смартфон (320px - портретная, 540px - альбомная) • в медиа запросах реализовать исчезновение логотипа в футоре (вид - смартфон) • в медиа запросах реализовать изменения расположения вертикального меню на горизонтальное меню (планшет, смартфон) • в медиа запросах уменьшить размеры шрифтов, картинок на страницах. Информация о медиа-запросах: https://html5book.ru/css3-mediazaprosy/ 2. Создать форму с контактами, используя следующие элементы: • поле ввода однострочного текста, • поле ввода электронного адреса, реализуйте подсказку для ввода

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		• поле ввода многострочного текста, • элементы переключатели, • элемент выпадающий список, • кнопка Оформите подписи к полям (label). Сгруппируйте элементы формы в отдельные разделы (не менее 2-х разделов) - fielset. Оформите вид формы в стилях вашего сайта. Информация о формах: https://html5book.ru/html5-forms/ 3. Реализовать следующий функционал для формы: • Чтение данных с формы и вывод на страницу HTML без перезагрузки. • Каждое следующее сообщение должно добавляться ниже. • Реализовать стили для ленты сообщений (придумать самим) • После обновления страницы лента сообщений пустая • Обработать поля ввода для формы – если пользователь не ввел текст в отдельное поле, то выводится сообщение об ошибке. (Разрешается Alert) • Скрипт js разместить в отдельном файле. **Практическое задание №4. Слайдер на JS** Задание: создать слайдер на JS для своего сайта. Слайдер – это специальный элемент веб-дизайна, представляющий собой блок определенной ширины чаще всего в шапке веб-страницы. Главная его особенность в изменяющихся в ручном или автоматическом режиме элементах – картинок, текстов и ссылок. Требования к слайдеру:

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		• Экран (с изображением) • Средства навигации (возможность ручного пролистывания слайдов) • Маркеры с общим количеством слайдов и текущим состоянием • Слайды сопровождаются дополнительными текстовыми блоками с информацией, ссылками или таблицами. • Автоматическая смена слайдов Дополнительные функции: 1. Миниатюры остальных слайдов 2. Таймер со временем до смены слайда 3. Пауза при наведении на слайд Код js должен сопровождаться комментариями. А также быть валидным. *В приложении 1 представлены тренировочные задания по JS*
Владеть	• Навыками работы с HTML/CSS на базовом уровне • Навыками работы с JavaScript, PHP на базовом уровне • Приемами интеграции существующего программного кода с API (Application Programming Interfaces), библиотеками и фреймворками • Навыками создания веб-приложения с доступом к БД.	***Проектное задание:*** Разработать браузерное игровое приложение.

Варианты проектного задания

1. Игра «Подводный мир»

Игровое поле представляет собой подводный мир с плавающими рыбками. Игрок должен набрать максимальное количество баллов, кликая

на них. После клика по рыбке она исчезает, а игрок получает очки. Каждая рыбка движется со случайными скоростью и траекторией. Рыбка не может находиться на игровом поле постоянно, со временем она должна уплывать, если по ней так и не кликнули. Одновременно на одном игровом поле может находиться не более 10 рыбок. Существуют рыбки 3 размеров. При клике на маленькую начисляется 30 очков, на среднюю – 20 очков, на большую – 10 очков.

Реализация логики, в состав которой должны быть включены следующие функции:

● Пауза игрового процесса - останавливается время на таймере, запрещается воздействие на игровое поле. Режим паузы также может быть инициирован по нажатию на клавишу пробел. Возобновление игры так же возможно по нажатию на клавишу “пробел” или по нажатию на кнопку “Пауза” на игровом поле

● Таймер обратного отсчета - начинает обратный отсчет с началом игры, как только доходит до значения 00:00 игра заканчивается

● Режима теста – режим, в котором таймер обратного отсчета не запускается, и игра не останавливается при взаимодействиях, которые подразумевают проигрыш или конец игры

2. Сервис «Интерьер-онлайн»

Симулятор сервиса – «ИНТЕРЬЕР ОНЛАЙН» - небольшое веб-приложение, позволяющее расставить мебель в помещении. Основная идея этого задания заключается в том, чтобы сделать приложение, позволяющее загрузить в него фотографию плана помещения и наложить на него различные предметы мебели, сантехнику, бытовую технику.

● При загрузке приложения пользователю необходимо загрузить фотографию плана помещения в специальную область.

● Пользователь может перетащить файл на специальную область веб-приложения. При помещении файла в эту область, изменяется цвет фона области.

● Необходимо отобразить сообщение об успешной загрузке изображения.

● После загрузки фотография должна появится в области приложения и пользователь может применить доступные рамки.

● Пользователь должен иметь возможность наложить предмет мебели на изображение (фото помещения) при помощи мыши. Если предмет мебели находится вне блока фотографии, то он должен вернуться в исходное положение.

● При нажатии на предмет мебели, размещенное в блоке фотографии, вокруг него должна появится граница из точек. После этого пользователь может изменять размер активного предмета мебели.

● Пользователь может изменять позицию предмета мебели при помощи мыши.

● Для изменения размера активного элемента предмета мебели на плане помещения нужно использовать кнопки приложения + и – для изменения размера.

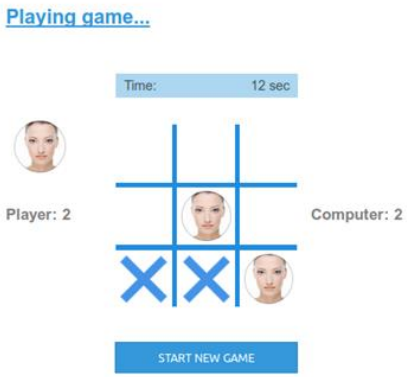
● На блоке фотографии не может располагаться более одного предмета мебели из каждой категории. Если пользователь поместил на блок фотографии более одного элемента типа мебели одной категории (например несколько диванов или умывальников), то он должен вернуться в исходное положение.

3. Крестики-нолики

Крестики-нолики запускаются в режиме одного пользователя. Логика должна отрабатывать следующие ситуации:

- При обновлении страницы, состояние процесса не должно быть нарушено, т.е. все измененные элементы должны остаться на месте. Отсчёт времени продолжается.
- В ничейной ситуации победителя нет и пользователю просто предлагается начать игру заново.
- Поле игры состоит из квадрата с 9 полями. Они пронумерованы следующим образом:

1	2	3
4	5	6
7	8	9



- По умолчанию всегда используется фото “default-user”, но как только игрок загрузит свое изображение хотя бы раз перед игрой, оно будет использоваться далее для всех игровых сессий (в текущей сессии браузера). (т.е. при рестарте не надо просить фото снова)
- При старте новой игры, все поля очищаются. Игрок ходит первым. Выбранное им поле маркируется его фотографией по умолчанию (или той что загружена однажды).
- Компьютер делает следующий ход. Он случайным образом выбирает одно из свободных полей и маркирует его большим X. Компьютер всегда думает 2 секунды и осуществляет ход на игровом поле.
- Количество совершенных шагов и время игры постоянно отображается и обновляется над игровым полем.
- Сообщение отображается для информирования игрока об его победе или проигрыше.

4. «Забавные лица»

Приложение – «Забавные лица» - небольшое веб-приложение, добавляющее элементы на лицо. Основная идея этого задания заключается в том, чтобы сделать приложение, позволяющее загрузить в него фотографию и наложить на неё различные аксессуары.

- При загрузке приложения пользователю необходимо загрузить фотографию лица в специальное поле.
- После загрузки фотография должна появится в области приложения и пользователь может применить доступные аксессуары.
- Пользователь может изменять размеры фотографии при нажатии на неё. Когда изображение активно вокруг него должна появляться видимая рамка из точек. Над фотографией можно делать следующие операции: изменение пропорций, увеличение, уменьшение. Операции выполняются кнопками приложения + и -. Изображение с изменённым масштабом должно выравниваться по центру блока изображения. Граница из точек должна исчезать, когда выбран аксессуар.

- Пользователь должен иметь возможность наложить аксессуары на изображение при помощи мыши. Если аксессуар находится вне блока фотографии, то он должен вернуться в исходное положение.
- При нажатии на аксессуар размещённый в блоке фотографии вокруг него должна появится граница из точек. После этого пользователь может изменять размер активного аксессуара
- Пользователь может изменять позицию аксессуаров при помощи мыши, а так же при помощи стрелок на клавиатуре (←,→,↑,↓).
- Аксессуар на блоке фотографии можно повернуть при помощи клавиш L и R.
- На блоке фотографии не может располагаться более одного аксессуара из каждой категории. Если пользователь поместил на блок фотографии более одного аксессуара из одной категории, то он должен вернуться в исходное положение

5. Игра «StarBattle»

Игра запускается после нажатия кнопки «Начать игру» на начальном экране. Планеты должны быть анимированы и двигаться справа налево для того, чтобы создавалось впечатление движения космического корабля в космосе. Таймер начинается с нуля и показывает, сколько космический корабль находится в движении в секундах. Счетчик топлива снижается на одно деление в секунду. Когда игра начинается, у игрока есть 15 единиц топлива (15 секунд). Максимальный запас топлива – на 30 секунд (30 секунд полета). Отсчет очков начинается с нуля.

- Космический корабль стреляет, когда игрок нажимает пробел, но он не может постоянно держать клавишу нажатой для производства множества выстрелов подряд. Один выстрел – одно нажатие на пробел.
- Выстрел одного космического корабля может разрушить только одну мишень. Выстрел не может повредить более одного элемента за раз.

Пользователь может передвигать космический корабль на экране.

Во время полета космический корабль должен уничтожить вражеские космические корабли и астероиды, находящиеся в космосе. Если космический корабль сталкивается с астероидом или другим кораблем, этот элемент разрушается и топливо должно падать на 15 очков.

- Космические корабли и астероиды должны располагаться на экране в случайном порядке и быть анимированными. Они должны передвигаться справа налево.
- Вражеские корабли можно уничтожить одним выстрелом. Каждый разрушенный вражеский корабль добавляет по 5 очков игроку.
- Игра допускает отрицательный рейтинг.
- Астероиды разрушаются двумя выстрелами. За каждый сбитый астероид игрок получает 10 очков.
- Во время полета главный космический корабль должен собирать иконки топлива, сталкиваясь с ними. Иконки должны сбрасываться сверху в виде анимации в случайном порядке. За каждую собранную иконку топлива счетчик топлива увеличивается на 15 пунктов. Одно очко добавляет одну секунду к полету.
- Планеты на заднем фоне должны быть анимированы для создания эффекта движения. Они должны двигаться с разной скоростью, большие планеты должны быть быстрее маленьких, создавая эффект параллакса. На заднем фоне должны быть планеты по меньшей мере 5 разных размеров.
- Если топливо кончается и счетчик топлива находится на нуле, игра кончается.

6. Игра «Змейка»

Игра запускается после нажатия кнопки «Начать игру» на начальном экране. Игровое поле реализовано в виде сетки из квадратных клеток. На клетке может располагаться часть змейки, препятствие, продукт (еда для змейки), либо клетка пустая.

- Начальная длина змеи всегда 1 клетка – ее голова, далее добавляется клетка в виде хвоста.
- С течением времени скорость змейки увеличивается.
- Для увеличения длины змейки она должна питаться. Продукты увеличивают змейку на одну клетку – добавляется в конец хвоста.
- При увеличении длины змейки добавляются баллы игроку. Баллы увеличиваются с увеличением скорости змейки.
- Если змейка натолкнулась на препятствие, либо на свой хвост – игра прекращается.
- Игрок управляет направлением движения змейки с помощью клавиатуры.
- Реализована возможность паузы в игре. И возобновление игры с того же места.
- Змейка может двигаться в любом направлении при этом игровое поле достраивается.
- При увеличении скорости движения змейки, змейка также меняет свой цвет.

7. Игра Симулятор вируса

Игра начинается с распространения вируса. Вирус случайным образом атакует 5 компьютеров в городе. Если вирус атаковал обычный домашний компьютер, то после 2 секунд он заражает еще два компьютера. Вирус также может заразить компьютер – сервер, в этом случае, сервер через 1 секунду заразит 5 компьютеров.

- Игра прекращается: если вирус побежден и все компьютеры города не заражены; либо закончилось время игры – 90 секунд.
- Все компьютеры города заражены вирусом – поражение игрока.
- Вирус уничтожается установкой антивирусной системы.
- Пользователь может устанавливать антивирусную систему и на незараженные компьютеры, в этом случае, данные компьютеры не может поразить вирус.
- При поражении вирусом компьютера, данный компьютер подсвечивается красным цветом.
- Внешний вид серверов и клиентских компьютеров отличаются.
- После «лечения» компьютера он не может быть заражен повторно.
- Установка антивирусной системы реализуется кликом мыши на компьютер.

Если компьютер был заражен, что красная подсветка исчезает и появляется зеленая подсветка, означающая, что компьютер в безопасности.

8. Игра Пожарник

При напуске игры демонстрируется игровое поле в виде леса, реки, пожарной станции. В некоторой области вспыхивает огонь, начинает распространяться пожар. Пожар может быть трех степеней: слабый, средний, сильный. На каждый из видов пожара необходимо определен-

ной количество воды, которое нужно набрать пожарным для тушения. Набор воды занимает время. Пожар за определенный промежуток времени переходит с первой на третью степень (если его не тушить). Если пожар третий степени его не тушат, он распространяется на соседние участки леса. Задача игрока определять столько ему нужно воды для тушения пожара, потушить за определенный интервал времени все очаги пожара.

- Игра прекращается: если все очаги пожара потушены, и это считается победой игрока; закончилось время игры – 70 секунд (если остался огонь на игровом поле – это поражение игрока).
- Игрок проигрывает, если горят все участки леса.
- Отдельный очаг пожара заливается определенным количеством воды, при этом учитывается место возгорания (время прибытия пожарной машины, время загрузки заданного объема воды для тушения пожара).
- Время распространения пожара закладывается разработчиком, аналогично -параметры тушения пожара.
- Для тушения пожара пользователь указателем мыши щелкает по объекту, который нужно потушить. Далее указывает сколькими литрами воды нужно заправить пожарную машину, чтобы потушить огонь.

9. Игра Шашки

Основное поле игры представляет собой 2 D-доску, разделенную на 25 (5*5) одинаковых квадратов. В начале игры на нижней горизонтальной линии выстраиваются фигуры первого игрока, на верхней – второго. Игра оканчивается победой того игрока, фигур которого остается на доске больше за отведенное время. При этом реализуются следующие правила:

- Изначально набранные каждой фигурой очки равны 1.
- Ходы игроков осуществляются последовательно, первым ходит игрок с фигурами на нижней лини.
- В каждый ход обязательно необходимо сдвинуть, используя захват мышью, одну свою фигуру на одну клетку (вниз, влево, вправо, вверх).
- Фигура не может ходить за пределы игрового поля.
- Фигура не может быть сдвинута на соседнюю клетку, если она занята другой фигурой.
- Если фигура пытается занять клетку с фигурой противника, они вступают в противоборство.
- В единоборстве побеждает фигура с большим количеством набранных очков. При этом очки победившей фигуры увеличиваются на 1.

Если у фигур одинаковое количество очков – победа определяется генератором случайных чисел.

- Побежденная фигура удаляется с доски, победившая – занимает ее место.
- Игра заканчивается если фигуры одного игрока удалены с доски, либо закончилось время игры
- Интерфейс представляет собой три экрана. Стартовый, игровой и результаты. Игра должна сопровождаться интерактивными сообщениями.

10. Игра Судоку

Игра состоит из трех уровней: новичок, средний и продвинутый. На новичке строится поле 3*3, средний: 6*6, продвинутый: 9*9.

Игровое поле представляет собой квадрат, разделённый на меньшие квадраты со стороной в 3 клетки. В них уже в начале игры стоят некоторые числа, называемые подсказками.

От игрока требуется заполнить свободные клетки цифрами в зависимости от уровня, чтобы в каждой строке, в каждом столбце и в каждом малом квадрате 3×3 каждая цифра встречалась бы только один раз.

- Интерфейс представляет собой три экрана. Стартовый, игровой и результаты. Игра должна сопровождаться интерактивными сообщениями.
- На стартовом экране предлагается выбрать уровень игры, ввести имя игрока, а также указать время игры.
- Игра может идти на время, если игрок указал это в настройках. Если время закончилось, то открываются результаты игры.
- Переходы по ячейкам осуществляется с помощью клавиатуры (->, <-, вниз, вверх).
- Ввод чисел осуществляется клавиатурой и подтверждением (Enter).
- Игрок может удалить введенной число.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

11. Электронный друг

Игра электронный друг реализует симуляцию выращивания животного (кота, собаки). Игроку предоставляется возможность выбрать животного на первом экране. На игровом поле появляется объект-животное. Данный объект необходимо кормить, гладить, водить гулять, укладывать спать.

- У объекта все время работают биологические часы. Если объект похудеет до 500 грамм, он умирает. Если его не гладить каждую минуту, он умирает.
- Необходимо вырастить объект до 3-летнегго возраста и отпустить на природу.
- Реализация работы с объектом осуществляется манипулятором мышь
- Все биологические характеристики все время доступны для просмотра игроком.

- Присутствует таймер на экране.
- Разработчик имеет право настроить параметры игры на меньшие временные отрезки.
- Объект-животное должен быть анимирован (движение глаз, ушей, хвоста).

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Практикум по разработке Web-приложений» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме зачета (2, 4 семестр) и зачета с оценкой (6 семестр)

Критерии оценки (в соответствии с формируемыми компетенциями и планируемыми результатами обучения):

– *на оценку* ***«зачтено»*** **–** обучающийся показывает высокий уровень сформированности компетенций, т.е. выполняет тренировочные, практические и лабораторные работы в установленные сроки, ориентируется в программном коде; разрабатывает проектные задания по дисциплине с учетом заявленных требований к веб-приложениям, владеет терминологическим аппаратом, демонстрирует глубокое теоретическое знание вопроса в области разработки интернет приложений, грамотно определяет логико-структурные связи, обосновывает свое решение и формулирует необходимые выводы.

– *на оценку* ***«не зачтено»*** – результат обучения не достигнут, обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач в области веб-разработки.

Критерии оценки

– *на оценку* ***«отлично»*** – полностью выполнен объем работ за семестр, также разработано проектное задание, учащийся чётко и правильно дает определения и раскрывает содержание материала; ответ самостоятельный, при ответе использованы знания, приобретённые ранее;

– *на оценку* ***«хорошо»*** – задания семестра выполнены на 85-90% от всего объема работ за семестр, также разработано проектное задание, учащийся в основном правильно дает определения, понятия; при ответе допускает неточности, практические навыки нетвёрдые;

– *на оценку* ***«удовлетворительно»*** – задания семестра выполнены на 60-80% от всего объема работ за семестр, не разработано проектное задание, усвоено основное содержание материала, но изложено фрагментарно, не всегда последовательно; определения и понятия даны не чётко; практические навыки слабые;

– *на оценку* ***«неудовлетворительно»*** – задания семестра не выполнены, основное содержание учебного материала не раскрыто; не даны ответы на дополнительные вопросы преподавателя

8 Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература:

1. Лисьев, Г.А. Программное обеспечение компьютерных сетей и web-серверов : учебное пособие / Г. А. Лисьев, П. Ю. Романов, Ю. И. Аскерко. — Москва : ИНФРА-М, 2020. — 145 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-013565-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1068576

2. *Тузовский, А. Ф.* Проектирование и разработка web-приложений : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2020. — 218 с. — (Высшее образование). — ISBN 978-5-534-00515-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/451207

б) Дополнительная литература:

1. Романова М. В. Разработка Web-страниц и презентаций [Электронный ресурс] : практикум / М. В. Романова, Е. В. Чернова. - Магнитогорск : МГТУ, 2017. - 70 с. : ил., табл. - Режим доступа: https://magtu.informsystema.ru/uploader/fileUpload?name=2704.pdf&show=dcatalogues/1/1131734/2704.pdf&view=true. - Макрообъект.
2. Маркин, А. В. Программирование на SQL в 2 ч. Часть 1 : учебник и практикум для вузов / А. В. Маркин. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 403 с. — (Высшее образование). — ISBN 978-5-534-12256-5. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/452357 (дата обращения: 25.09.2020).
3. Лавлинский, В. В. WEB-инжиниринг: Учебное пособие / Лавлинский В.В., Табаков Ю.Г. - Воронеж:ВГЛТУ им. Г.Ф. Морозова, 2013. - 268 с. - Текст : электронный. - URL: https://znanium.com/catalog/product/858312 (дата обращения: 27.09.2020). – Режим доступа: по подписке
4. Журнал «Программные продукты и системы» [Электронный ресурс]. Научно-исследовательский институт «Центрпрограммсистем» — Режим доступа: https://e.lanbook.com/journal/2276#journal_name — Загл. с экрана.
5. Каталог межгосударственных стандартов. [Электронный ресурс]. Росстандарт. - Режим доступа: https://www.gost.ru/portal/gost/home/standarts/cataloginter

в) Методические указания:

Методические указания по выполнению практических заданий представлены в приложении.

г) Программное обеспечение и Интернет-ресурсы:

Наименование ПО	№ договора	Срок действия лицензии
MS Windows 7	Д-1227 от 08.10.2018	11.10.2021
MS Office 2007	№ 135 от 17.09.2007	бессрочно
7Zip	свободно распространяемое	бессрочно
SCO OpenServer	свободно распространяемое	бессрочно
Google Chrome	свободно распространяемое ПО	бессрочно

Профессиональные базы данных и информационные справочные системы:

1. Официальный Интернет-ресурс Федерального агентства по техническому регулированиюи метрологии . – http://www.gost.ru
2. Справочник по ГОСТам и стандартам. Информационные технологии. [Электронный ресурс]. Информационное агентство MetalTorg.Ru. — Режим доступа: http://gostbank.metaltorg.ru/oks/629/

Интернет-ресурсы:

1. On-line учебник HTML5BOOK — Режим доступа: https://html5book.ru/
2. PHP: справочник языка - Режим доступа: http://php.net/manual/ru/langref.php

9 Материально-техническое обеспечение дисциплины

Материально-техническое обеспечение дисциплины включает:

Тип и название аудитории	Оснащение аудитории
Учебные аудитории для проведения лабораторных занятий	Персональные компьютеры с выходом в Интернет и с доступом в электронную информационно-образовательную среду университета. Браузер Google Chrome.
Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки	Персональные компьютеры с выходом в Интернет и с доступом в электронную информационно-образовательную среду университета. Браузер Google Chrome.
Аудитории для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Персональные компьютеры с выходом в Интернет и с доступом в электронную информационно-образовательную среду университета Браузер Google Chrome.
Аудитория для хранения и профилактического обслуживания учебного оборудования № 086	Мебель для хранения и обслуживания оборудования (шкафы, столы), учебно-методические материалы, компьютеры, ноутбуки, принтеры.

Введение. Для выполнения практических работ рекомендуем просмотреть структуру HTML-документа, основные теги HTML, а также работу с CSS.

После формулировки задания вам предоставлен пример реализации, который вы можете взять в основу выполнения данного задания. Обращаем внимание, некоторые пункты задания не реализованы, вам необходимо будет их реализовать самостоятельно.

Теоретический материал, а примеры реализации JS скриптов можно найти на следующем ресурсе: https://learn.javascript.ru

Практическая работа №1. Переменные JavaScript

Задачи:

1. Изучите способы добавления JS на страницу HTML.
2. Изучите правила объявления переменных.
3. Основные типы данных: число, строка, логический тип, null, undefined
4. Правила преобразования типов.
5. Команды alert, prompt, write.

Задание:

Напишите скрипт, который запрашивает у пользователя два числа и выводит их сумму.

```
<script type="text/javascript">
        var num1 = prompt("Введите первое число");
        var num2 = prompt ("Введите второе число");
        alert (Number(num1)+Number(num2));
</script>
```

Ответьте на вопросы преподавателя. Будьте готовы выполнить дополнительное задание по данной теме.

Практическая работа №2. Логические операции. Регулярные выражения.

Задачи:

1. Изучите правила объявления логических выражений, операций.
2. Познакомьтесь в функциями обработки типов данных.

Задание:

Добавить в скрипт проверку на валидацию данных.

```
<script type="text/javascript">
var num1 = prompt("Введите первое число");
if (isFinite(num1)) {
var num2 = prompt ("Введите второе число");
if (isFinite(num2)) document.write (Number(num1)+Number(num2));
else alert ('второе значение не числовое');
} else alert ('первое значение не числовое')
</script>
```

Ответьте на вопросы преподавателя. Будьте готовы выполнить дополнительное задание по данной теме.

Практическая работа №3. Регулярные выражения.

Задачи:

1. Изучите правила использования регулярных выражений.
2. Познакомьтесь с оператором switch. Метод eval. Какие варианты алгоритма они позволяют реализовать?
3. Изучите арифметические операции в JS

Задание:

Измените скрипт – помимо чисел, скрипт запрашивает, какую арифметическую операцию нужно выполнить (+,*,-,/). Выводит результат арифметической операции.

Перечень арифметических операций: +,*,-,/, возведение в степень, наименьшее число, наибольшее число.

```
<script type="text/javascript">
var stand_oper = /[*+-/]/;
var num1 = prompt("Введите первое число");
if (isFinite(num1)) {
                    var num2 = prompt ("Введите второе число");
                    if (isFinite(num2))
{
  var operacia = prompt ("введите арифметическую операцию");
  document.write (operacia);
  if (stand_oper.test(operacia)) { var res = num1 + operacia +num2;

  document.write (eval(res));}
  else alert ('С такой операцией мы не работаем');
}

  else alert ('первое значение не числовое');
  } else alert ('первое значение не числовое')
</script>
```

Ответьте на вопросы преподавателя. Будьте готовы выполнить дополнительное задание по данной теме.

Практическая работа №4. Циклы

Задачи: Изучите виды циклов, из реализацию в JS.

Задание: Реализуйте возможность ввода арифметических операций пользователем до тех пор, пока он не введет слово ВЫХОД.

```
<script type="text/javascript">
var stand_oper = /[*+-/]/;
do {
    var num1 = prompt("Введите первое число");
  if (isFinite(num1)) {
       var num2 = prompt ("Введите второе число");
  if (isFinite(num2))
                    {
```

```
                var operacia = prompt ("введите арифметическую операцию");
            document.write (operacia);
            if (stand_oper.test(operacia))
                    { var res = num1 + operacia +num2;
                     alert (eval(res));
                    }
                    else alert ('С такой операцией мы не работаем');
        }
          else alert ('второе значение не числовое');
  } else alert ('первое значение не числовое');
}while ((num1!="ВЫХОД")&&(num2!="ВЫХОД")&&(operacia!="ВЫХОД"))
</script>
```

Ответьте на вопросы преподавателя. Будьте готовы выполнить дополнительное задание по данной теме.

Практическая работа №5. Функции. Функциональные выражения

Задачи:

1. Познакомьтесь с правилами объявления функций.
2. Изучите правила объявления функциональных выражений. В чем разница между функцией и функциональным выражением?
3. Рассмотрите теорию по области видимости переменных.

Задача:

Реализуйте две функции: первая функция проверяет корректность ввода числовых значений и операции; вторая функция реализует арифметическую операцию и возвращает ее результат в основную часть скрипта.

```
<script type="text/javascript">
function checking (data,data2,op)
  {
var stand_oper = /[*+-/]/;
if (isFinite(data) && isFinite(data2) && stand_oper.test(operacia))  return true;
                    else return false;
}
function calcul (data,data2,op)
{
switch (op){
  case '+': return Number(data) + Number(data2); break;
  case '-': return Number(data) - Number(data2); break;
  case '*': return Number(data) * Number(data2); break;
  case '/': return Number(data) / Number(data2); break;
  }
}

do {
  var num1 = prompt("Введите первое число");
  var num2 = prompt ("Введите второе число");
  var operacia = prompt ("введите арифметическую операцию");
  if (!checking (num1,num2,operacia)) {
                                        alert ("ERROR!");
                                        }
                                        else alert (calcul(num1,num2,operacia));
```

```
  }while ((num1!="ВЫХОД")&&(num2!="ВЫХОД")&&(operacia!="ВЫХОД"))
</script>
```

Ответьте на вопросы преподавателя. Будьте готовы выполнить дополнительное задание по данной теме.

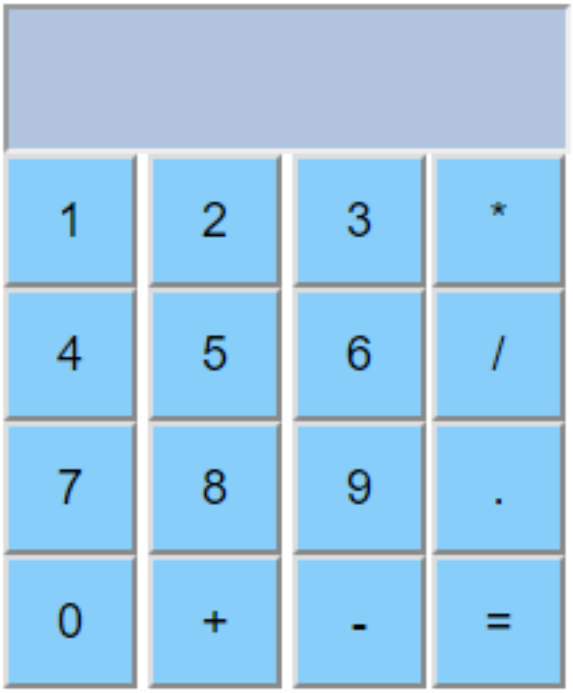
Практическая работа №6. Работа с DOM

Задачи:

1. Изучите объектную модель документа.
2. Основные методы работы с DOM

Задание:

Создайте интерфейс калькулятора, см. примерный вид:



Создайте логику работы калькулятора.

Нельзя использовать alert и prompt.

Логику работы калькулятора вынесите в отдельный файл JS.

В готовом калькуляторе должны быть все объявленные ранее арифметические операции. Проверка на корректность ввода данных. А также кнопка удалить введенные символы с соответствующей реализацией.

Index. Html

```
<!doctype html>
<!DOCTYPE html>
<html>
<head>
  <title>JS</title>
  <meta charset="UTF-8">
  <style>
  form {
  margin-left: auto;
  margin-right: auto;
            width: 220px;

  }
```

```
        #wrapper {
                display:flex;
                flex-direction: row;
                justify-content: center;
                align-content: space-between;

        }
        .output {
                width: 208px;
                height: 50px;
                font-size: 20px;
                background: #B0C4DE;
        }
        #num {
                width: 160px;
        }
        #oper
        {
                width: 60px;
        }
                .number {
                        width: 50px;
                        height: 50px;
                        background: #87CEFA;
                        font-size: 18px;
                }
                </style>

      </head>
      <body>
      <script type="text/javascript" src ="calculate.js"></script>
        <form>
                <input type="text" class="output">
                <div id="wrapper">
                        <div id="num">
                                <input type="button" name="one" value = "1"
class="number" OnClick="show_enter(value);">
                                <input type="button" name="two" value = "2" class="number"
OnClick="show_enter(value);">
                                <input type="button" name="three" value = "3"
class="number" OnClick="show_enter(value);">
                                <input type="button" name="four" value = "4"
class="number" OnClick="show_enter(value);">
                                <input type="button" name="five" value = "5" class="number"
OnClick="show_enter(value);">
                                <input type="button" name="six" value = "6" class="number"
OnClick="show_enter(value);">
                                <input type="button" name="seven" value = "7"
class="number" OnClick="show_enter(value);">
                                <input type="button" name="eigte" value = "8"
class="number" OnClick="show_enter(value);">
                                <input type="button" name="nine" value = "9"
```

```
class="number" OnClick="show_enter(value);">
                                <input     type="button"     name="zero"     value     =     "0"
class="number" OnClick="show_enter(value);">
                                <input     type="button"     name="plus"     value     =     "+"
class="number" OnClick="show_enter(value);">
                                <input     type="button"     name="minus"     value     =     "-"
class="number" OnClick="show_enter(value);">
                        </div>
                            <div id="oper">
                                <input     type="button"     name="con"     value     =     "*"
class="number"OnClick="show_enter(value);">
                                <input type="button" name="div" value = "/" class="number"
OnClick="show_enter(value);">
                                <input type="button" name="real" value = "." class="number"
OnClick="show_enter(value);">
                                <input     type="button"     name="enter"     value     =     "="
class="number" OnClick="calculate();">
                            </div>
                    <div>

      </form>
    </body>
    </html>
```

Calculate.js

```
function show_enter (num) {
  var screen=document.querySelector('.output');
  screen.value +=num;
}
function calculate () {
  var screen=document.querySelector('.output');
  var res = eval (screen.value);
  screen.value=res;
}
```

Ответьте на вопросы преподавателя. Будьте готовы выполнить дополнительное задание по данной теме.

Во многих современных веб-приложениях для реализации обновляющегося на лету пользовательского интерфейса, работающего в режиме реального времени, используются WebSockets. Когда какая-либо информация изменяется на сервере, обычно посылается сообщение через WebSocket-подключение для обработки на клиенте. Это обеспечивает более надёжную и эффективную альтернативу постоянному опросу вашего приложения о наличии изменений.

Для помощи в создании таких приложений Laravel обеспечивает простую настройку «вещания» ваших событий через WebSocket-подключение. Вещание Laravel-событий позволяет использовать одинаковые названия событий в коде на стороне клиента и в клиентском JavaScript-приложении.

Первые шаги

5. Откройте терминал и создайте новый каталог для кода проекта.

```
cd ~/Code
mkdir links
cd links
```

6. Затем установите программу установки Laravel :

```
composer global require "laravel/installer"
```

7. После завершения процесса, можно создать новый проект, выполнив:

```
laravel new links
```

Эта команда создаст новый каталог и установит пустой проект Laravel.

8. Теперь нужно создать виртуальный хост, ссылающийся на ~/Code/links/public, чтобы можно было загрузить сайт в браузере. К примеру, links.dev.

Вы должны увидеть страницу-заглушку Laravel

9. Теперь сгенерируем систему аутентификации использую следующую команду:

```
php artisan make:auth
```

Создаем список ссылок

Несмотря на то, что список ссылок кажется небольшой задачей она требует базу данных, таблицу в базе данных, данные в этой таблице, запрос к базе данных, и файл представления.

1. Первым шагом будет создание миграции. В этом нам поможет консольный инструмент artisan.

```
php artisan make:migration create_links_table --create=links
```

2. Теперь откройте файл, который создала команда. Он находится по следующему пути database/migrations/{{ДатаВремя}}_create_links_table.php

Внутри метода up добавьте столбцы таблицы

```
Schema::create('links', function (Blueprint $table) {
    $table->increments('id');
    $table->string('title');
    $table->string('url')->unique();
    $table->text('description');
    $table->timestamps();
});
```

3. Сохраните файл и запустите миграцию

```
php artisan migrate
```

4. Теперь надо добавить данные. В Laravel для этой цели есть две функции: заполнение начальными данными (seeding) и фабрики моделей.
5. Перед тем как продолжить процесс напишем первый юнит-тест. Тесты можно разделить на две категории: функционал (features) и юниты (unit).

6. Тест будет юнит-тестом, поэтому создаем файл ./tests/unit/SeederTest.php и добавляем тестирующий метод :

```
public function testLinksTable()
{
 $this->seeInDatabase('links', ['title' => 'dotdev.co']);
}
```

7. Этот метод должен выдать ошибку. Запускаем - так и есть

… matches PCRE pattern "/dotdev.co/i" …

8. НО мы не это вообще-то тестируем. При установке в Laravel уже есть один тестовый файл ExampleTest.php . Удалите его и запустите phpunit снова.
9. Теперь мы видим нашу настоящую ошибку

There was 1 failure:

1) SeederTest::testLinksTable Unable to find row in database table [links] that matched attributes [{"ti-tle":"dotdev.co"}]. Failed asserting that 0 is greater than 0.

10. Для того, чтобы этот тест прошел удачно, необходимо создать файл, заполняющий базу данных.

php artisan make:seeder LinksTableSeeder

11. Откройте DatabaseSeeder.php и добавьте следующее в метод run:

```
$this->call(LinksTableSeeder::class);
```

12. Чтобы тест прошел удачно мы могли бы просто добавить новую запись в LinksTableSeeder. Но, мы создадим фабрику модели сейчас, чтобы сэкономить время будущем, т.к. наши будущие тесты будут на нее завязаны. До того как создать фабрику, необходимо добавить саму модель.

php artisan make:model Link

13.Отредактируем созданную модель app/Link.php добавив имя таблицы в бд:

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Link extends Model
{
   protected $table = 'links';
}
```

13. Теперь вернемся к ModelFactory.php и добавим наши ссылки:

```
$factory->define(App\Link::class, function (Faker\Generator $faker) {
   return [
      'title' => $faker->name,
      'url' => $faker->url,
      'description' => $faker->paragraph,
   ];
});
```

Если мы снова запустим сейчас phpunit, ошибка все еще будет, т.к. ссылки с названием “dotdev.co” в базе еще нет. В нашем SeederTest мы можем создать эту запись в бд с помощью той же фабрики моделей:

```
<?php

use Illuminate\Foundation\Testing\WithoutMiddleware;
use Illuminate\Foundation\Testing\DatabaseMigrations;
use Illuminate\Foundation\Testing\DatabaseTransactions;
```

```
class SeederTest extends TestCase
{
    public function testLinksTable()
    {
        factory(App\Link::class)->create([
            'title' => 'dotdev.co',
        ]);
        $this->seeInDatabase('links', ['title' => 'dotdev.co']);
    }
}
```

14. Теперь, если выполнить phpunit все должно завершиться успешно.

Следующий шаг – вывод самого списка ссылок. Напишем новый тест.

15. Создаем новый файл ./tests/features/LinkTest.php. Спросим себя: что мы ожидаем увидеть? На странице списка ссылок мы можем найти известную нам ссылку, как мы делали выше, и подтвердить, что мы реально ее видим. Вот код такой проверки:

```
<?php

use Illuminate\Foundation\Testing\WithoutMiddleware;
use Illuminate\Foundation\Testing\DatabaseMigrations;
use Illuminate\Foundation\Testing\DatabaseTransactions;

class LinkTest extends TestCase
{
    public function testWeSeeAListOfLinks()
    {
        factory(App\Link::class)->create([
            'title' => 'dotdev.co',
        ]);
        $this->visit('/')
             ->see('dotdev.co');
    }
}
```

Phpunit снова выдает ошибку, что искомая строка не найдена. Откроем карту маршрутов ./app/Http/routes.php и посмотрим куда указывает маршрут «/» .

```
Route::get('/', function () {
    return view('welcome');
});
```

Давайте сделаем выборку наших ссылок и отправим их в представление.

```
Route::get('/', function () {
    $links = \App\Link::all();
    return view('welcome', compact('links'));
});
```

Отредактируем welcome.blade.php добавив простой цикл foreach:

```
@foreach ($links as $link)
  <li>{{ $link->title }}</li>
@endforeach
```

Теперь phpunit позеленел. Тесты прошли успешно.