

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кузбасский государственный технический университет
имени Т.Ф. Горбачева»

Составитель
А. А. Тайлакова

АРХИТЕКТУРА ВЕБ-ПРИЛОЖЕНИЙ

**Методические указания к лабораторным занятиям
и самостоятельной работе**

Рекомендовано учебно-методической комиссией
направления подготовки бакалавров 09.03.03
«Прикладная информатика» в качестве электронного издания
для использования в учебном процессе

Кемерово 2016

Рецензенты:

Сарапулова Татьяна Викторовна – кандидат технических наук, доцент кафедры прикладных информационных технологий

Соколов Игорь Александрович – кандидат технических наук, доцент, заведующий кафедрой прикладных информационных технологий, председатель учебно-методической комиссии направления 09.03.03 «Прикладная информатика»

Тайлакова Анна Александровна

Архитектура веб-приложений: метод. указания к лабораторным занятиям и самостоятельной работе по дисциплине «Интернет-программирование» [Электронный ресурс]: для студентов направления подготовки бакалавров 09.03.03 «Прикладная информатика», профиль «Экономика», очной формы обучения / сост. А. А. Тайлакова; КузГТУ. – Электрон. дан. – Кемерово, 2016. – Систем. требования : ОЗУ 64 Мб ; Windows XP/Vista/7. – Загл. с экрана.

Методические указания предназначены для изучения технологий разработки веб-приложений. В состав указаний входят теоретическая база и задания для выполнения лабораторным занятиям и выполнения самостоятельной работы.

© КузГТУ, 216
© Тайлакова А. А.,
составление, 2016

ОГЛАВЛЕНИЕ

Лабораторная работа №1 РАСЧЕТ ПЛАТЕЖЕЙ ПО КРЕДИТУ СРЕДСТВАМИ PHP	4
Задание для выполнения лабораторной работы	4
Контрольные задания	8
Лабораторная работа №2 ТЕХНОЛОГИИ СОOKIE И СЕССИЙ	12
Задание для выполнения лабораторной работы	12
Контрольные задания	14
Лабораторная работа №3 РАБОТА С ФАЙЛАМИ... Ошибка! Закладка не определена.	
Задание для выполнения лабораторной работы Ошибка! Закладка не определена.	
Контрольные вопросы	Ошибка! Закладка не определена.
Лабораторная работа №4 ЯЗЫК ЗАПРОСОВ SQL.....	16
Задание для выполнения лабораторной работы	17
Контрольные задания	20
Лабораторная работа №5 ТЕХНОЛОГИЯ AJAX.....	24
Задание для выполнения лабораторной работы	24
Добавление комментариев без перезагрузки страницы	17
Контрольные вопросы	20
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	27
ПРИЛОЖЕНИЕ А.....	31

Лабораторная работа №1

ОСНОВЫ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ

С ПРИМЕНЕНИЕМ ЯЗЫКА PHP

Целью данной лабораторной работы является изучение основ создания серверных сценариев с использованием языка программирования PHP, получение практических навыков разработки веб-приложений.

PHP – это широко используемый язык сценариев общего назначения с открытым исходным кодом, специально разработанный для написания веб-приложений (сценариев), исполняющихся на веб-сервере.

Преимуществом PHP является предоставление веб-разработчикам возможности быстрого создания динамически генерируемых веб-страниц. Языка PHP позволяет создавать HTML документы с внедренными командами PHP.

Значительным отличием PHP от какого-либо кода, выполняющегося на стороне клиента, например, JavaScript, является то, что PHP-скрипты выполняются на стороне сервера. Вы можете сконфигурировать свой сервер таким образом, чтобы HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта.

PHP позволяет создавать качественные веб-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем [1].

Задание для выполнения лабораторной работы

При подготовке к выполнению лабораторной работы следует установить виртуальный сервер или зарегистрировать хостинг для размещения лабораторных работ; изучить основы синтаксиса языка PHP, методы передачи данных и способы работы с пользовательскими формами.

Задание к лабораторной работе: создать веб-приложение на языке PHP, которое позволит выполнять расчет платежей по кредитам, согласно варианту. Приложение должно включать следующие файлы:

index.html – содержит форму для ввода исходных данных, созданную средствами языка гипертекстовой разметки HTML.

script.php – содержит программный код на языке PHP. Необходимо создать функции проверки введенных данных, табулирования функции и вывода результатов.

style.css – файл со стилями.

Варианты заданий

№	Вариант	Исходные данные	Методика расчета
1	Калькулятор для расчета ежемесячных платежей по ипотеке [15].	S – размер ссуды; p – месячная процентная ставка; m – срок выплат, выраженный в месяцах.	<p>Осуществить расчет ежемесячных платежей по ипотеке можно с помощью формулы:</p> $X = (S \cdot p) / (1 - (1+p)^{-m}),$ <p>где X – размер платежа</p>
2	Калькулятор для расчета процентов по вкладам (депозитам). Формула простых процентов [16].	I – годовая процентная ставка; t – количество дней начисления процентов по привлеченному вкладу; K – количество дней в календарном году (365 или 366); P – первоначальная сумма привлеченных в депозит денежных средств.	<p>Формула простых процентов применяется, если начисляемые на вклад проценты причисляются к вкладу только в конце срока депозита или вообще не причисляются, а переводятся на отдельный счет, т.е. расчет простых процентов не предусматривает капитализации процентов.</p> <p>Осуществить расчет можно с помощью формулы:</p> $S = P + (P \cdot I \cdot t) / (K \cdot 100),$ <p>где S – сумма денежных средств, причитающихся к возврату вкладчику по окончании срока депозита. Она состоит из первоначальной суммы размещенных денежных средств, плюс начисленные проценты.</p>
3	Калькулятор для расчета процентов по вкладам (депозитам). Формула сложных процентов [16].	I – годовая процентная ставка; j – количество календарных дней в периоде, по итогам которого банк производит капитализацию начисленных процентов; K – количество дней в календарном году (365 или 366); P – первоначальная сумма привлеченных в депозит денежных средств; n – количество операций по капитализации начисленных процентов в течение общего срока привлечения денежных средств.	<p>Формула сложных процентов применяется, если начисление процентов по вкладу, осуществляется через равные промежутки времени (ежедневно, ежемесячно, ежеквартально) а начисленные проценты причисляются к вкладу, т. е. расчет сложных процентов предусматривает капитализацию процентов (начисление процентов на проценты).</p> <p>Осуществить расчет можно с помощью формулы:</p> $S = P \cdot (1 + (I \cdot j) / (100 \cdot K))^n,$ <p>где S – сумма денежных средств, причитающихся к возврату вкладчику по окончании срока депозита. Она состоит из суммы вклада (депозита) с процентами.</p>
4	Калькулятор для расчета платежей по кредиту. Ежемесячный возврат части кредита	pV – сумма кредита; n – срок кредита, выраженный в месяцах; rate – месячная процентная ставка.	<p>Ежемесячный возврат части кредита с уплатой процентов, это метод расчета кредита предусматривающий ежемесячный возврат заранее оговоренной, одной и той же части кредита и ежемесячную уплату процентов.</p>

	с уплатой процентов [16].		<p>Величина очередного платежа по кредиту определяется по формуле</p> $V = pV / n$ <p>Величина очередного платежа по процентам определяется по формуле</p> $I = pV * rate$
5	Калькулятор для расчета платежей по кредиту Аннуитетный платеж [16].	<p>pV – сумма кредита; n – срок кредита, выраженный в месяцах; $rate$ – месячная процентная ставка.</p>	<p>Аннуитетный платеж, это метод расчета процентов по кредиту, который предусматривает погашение кредита ежемесячными равновеликими (при условии, что ставка кредита неизменна) платежами, содержащими в себе платеж по уплате процентов и платеж по возврату кредита. Величина аннуитетного платежа по кредиту определяется по формуле</p> $Pmt = pV * rate / [1 - (1 / (1 + rate))n]$
6	Калькулятор для расчета платежей по кредиту Единовременный возврат кредита с периодической уплатой процентов[16].	<p>pV – сумма кредита; n – срок кредита, выраженный в месяцах; $rate$ – месячная процентная ставка.</p>	<p>Единовременный возврат кредита с периодической уплатой процентов, это расчет по кредиту, который предусматривает возврат кредита в конце срока и периодическое (как правило, ежемесячное) перечисление банку процентов за кредит. Этот метод применяется банками крайне редко, в виде исключения. Причина - повышенный риск невозвратности кредита одной суммой в конце срока не гарантирует своевременного погашения, особенно если заемщик получает доходы не равномерно. Платежи по кредиту определяются по формуле</p> $i = pV * rate$

Данная работа создается в двух экземплярах. В первом случае для передачи данных между функциями используются параметры, во втором файле используйте глобальные переменные.

После выполнения лабораторной работы необходимо представить два веб-приложения, позволяющие выполнять табулирование функции, и отчет о проделанной работе (в электронном и печатном виде).

Отчет должен содержать:

1. **Титульный лист** (приложение А).
2. В **содержании** указываются названия всех частей отчета (кроме титульного листа и самого содержания) со ссылками на соответствующие страницы. Содержание должно быть сформировано и вставлено в отчет средствами MS Word.
3. **Введение** должно содержать цель и задачи лабораторной работы, задание, анализ предметной области.
4. **Основная часть** должна содержать листинг кода страницы.
5. **Заключение** должно содержать краткие выводы по результатам проделанной работы.
6. **Список использованных источников.** Отчет должен содержать полный список использованных литературных источников с соблюдением ссылок при цитировании. При составлении списка следует соблюдать требования ГОСТов [2, 3].

Отчет должен быть выполнен *на русском языке* с соблюдением его грамматических и синтаксических правил. Оформление текстовых и графических материалов курсовой работы [3] производится в соответствии с требованиями комплекса стандартов Единой системы конструкторской документации (ЕСКД). Электронный вариант отчета о курсовой работе готовится с помощью текстового процессора MS Word для односторонней распечатки на стандартных листах бумаги формата А4. Размеры полей должны быть следующими: левое, верхнее и нижнее – по 20 мм, правое – 15 мм. Основной текст набирается шрифтом Times New Roman размером 12 pt, кроме титульного листа, содержимое которого набирается шрифтом 14 pt. Настройки абзаца: выравнивание – по ширине, первая строка – красная (отступ 1,25 см), интервал (отступ) перед абзацем – 6 pt, междустрочный интервал – одинарный. Фрагменты исходных текстов набирать шрифтом Courier, Courier new.

Названия структурных элементов отчета, заголовки разделов и подразделов основной части работы выделяются жирным шрифтом размера 14 pt с использованием стилей заголовков различного уровня

для последующей вставки содержания. Отдельные слова для усиления смысловой нагрузки можно выделять жирным или курсивом, не злоупотребляя эффектами начертания.

Каждый структурный элемент отчета и каждый новый раздел основной части начинается с новой страницы. Все страницы отчета должны иметь сквозную нумерацию. Номер страницы пишется арабскими цифрами размера 12 pt и располагается снизу по центру. Титульный лист первым включается в общую нумерацию, но при этом номер на нем не проставляется.

Контрольные задания

Вариант 1

При помощи языка программирования PHP разработать алгоритм программы, которая рассчитывает значение x :

$$\begin{cases} \text{при } x < 0, \text{ возвести } x \text{ в квадрат } 10 \text{ раз} \\ \text{при } x = 0, x = \sin(x) \\ \text{при } x > 0, \quad x = x + 1 \text{ пока } x > 100 \end{cases}$$

Пользователь вводит значение x в текстовое поле формы. Передача данных осуществляется методом POST. Для вычисления создать пользовательскую функцию *calculation_x*, параметр x передавать по ссылке. Вывод результата при помощи конструкции *echo*.

Вариант 2

При помощи языка программирования PHP разработать алгоритм программы, которая рассчитывает значение x :

$$\begin{cases} \text{при } x < 0, \text{ возвести } x \text{ в квадрат } 10 \text{ раз} \\ \text{при } x = 0, x = \sin(x) \\ \text{при } x > 0, \quad x = x + 1 \text{ пока } x > 100 \end{cases}$$

Пользователь вводит значение x в текстовое поле формы. Передача данных осуществляется методом GET. Для вычисления создать пользовательскую функцию *calculation_x*, переменная x является глобальной. Вывод результата при помощи конструкции *echo*.

Вариант 3

При помощи языка программирования PHP разработать алгоритм программы, которая рассчитывает значение x :

$$\left\{ \begin{array}{l} \text{при } x < 0, \text{ возвести } x \text{ в квадрат } 10 \text{ раз} \\ \text{при } x = 0, x = \sin(x) \\ \text{при } x > 0, \quad \left\{ \begin{array}{ll} \text{при } x < 10, & x = x * 2 \\ \text{при } x \geq 10, & x = x + 5 \end{array} \right. \end{array} \right.$$

Пользователь вводит значение x в текстовое поле формы. Передача данных осуществляется методом POST. Для вычисления создать пользовательскую функцию *calculation_x*, параметр x передавать по ссылке. Вывод результата при помощи конструкции *echo*.

Вариант 4

При помощи языка программирования PHP разработать алгоритм программы, которая рассчитывает значение x :

$$\left\{ \begin{array}{l} \text{при } x < 0, \text{ возвести } x \text{ в квадрат } 10 \text{ раз} \\ \text{при } x = 0, x = \sin(x) \\ \text{при } x > 0, \quad x = x + 1 \text{ пока } x > 100 \end{array} \right.$$

Пользователь вводит значение x в текстовое поле формы. Передача данных осуществляется методом POST. Для вычисления создать пользовательскую функцию *calculation_x*, параметр x передавать по ссылке. Вывод результата при помощи конструкции *echo*.

Вариант 5

При помощи языка программирования PHP разработать алгоритм программы, которая рассчитывает значение x :

$$\left\{ \begin{array}{l} \text{при } x < 0, \text{ возвести } x \text{ в квадрат } 10 \text{ раз} \\ \text{при } x = 2, x = \sin(x) \\ \text{при } x = 10, x = \sqrt{x} \\ \text{при } x = 15, x = x + 3 \\ \text{иначе } x = 0 \\ \text{при } x > 15, \quad x = x + 1 \text{ пока } x > 100 \end{array} \right.$$

Пользователь вводит значение x в текстовое поле формы. Передача данных осуществляется методом POST. Для вычисления создать пользовательскую функцию *calculation_x*, параметр x передавать по ссылке. Вывод результата при помощи конструкции *echo*.

Вариант 6

При помощи языка программирования PHP разработать алгоритм программы, которая рассчитывает значение x :

$$\left\{ \begin{array}{l} \text{при } x < 0, \text{ возвести } x \text{ в квадрат } 10 \text{ раз} \\ \text{при } x = 2, x = \sin(x) \\ \text{при } x = 10, x = \sqrt{x} \\ \text{при } x = 15, x = x + 3 \\ \text{иначе } x = 0 \\ \text{при } x > 15, \quad x = x + 1 \text{ пока } x > 100 \end{array} \right.$$

Пользователь вводит значение x в текстовое поле формы. Передача данных осуществляется методом GET. Для вычисления создать пользовательскую функцию *calculation_x*, переменная x является глобальной. Вывод результата при помощи конструкции *echo*.

Вариант 7

При помощи языка программирования PHP разработать алгоритм программы, которая находит сумму положительных чисел от x до x^4 .

Пользователь вводит значение x в текстовое поле формы. Передача данных осуществляется методом POST. Для вычисления создать пользовательскую функцию *calculation_x*, параметр x передавать по ссылке. Вывод результата при помощи конструкции *echo*.

Вариант 8

При помощи языка программирования PHP разработать алгоритм программы, которая находит сумму положительных чисел от x до x^4 .

Пользователь вводит значение x в текстовое поле формы. Передача данных осуществляется методом GET. Для вычисления создать пользовательскую функцию *calculation_x*, переменная x является глобальной. Вывод результата при помощи конструкции *echo*.

Вариант 9

При помощи языка программирования PHP разработать алгоритм программы, которая для заданного ряда чисел x_1, x_2, \dots, x_n подсчитывает количество отрицательных чисел.

Пользователь вводит значение n в текстовое поле формы. Передача данных осуществляется методом GET. Массив значений x генерируется случайным образом. Для вычисления создать пользовательскую функцию *calculation_x*, параметры n и x передавать по ссылке. Вывод результата при помощи конструкции *echo*.

Вариант 10

При помощи языка программирования PHP разработать алгоритм программы, которая для заданного ряда чисел x_1, x_2, \dots, x_n подсчитывает количество отрицательных чисел.

Пользователь вводит значение n в текстовое поле формы. Передача данных осуществляется методом POST. Массив значений x генерируется случайным образом. Для вычисления создать пользовательскую функцию *calculation_x*, переменные x и n являются глобальными. Вывод результата при помощи конструкции *echo*.

Вариант 11

При помощи языка программирования PHP разработать алгоритм программы, которая для заданного ряда чисел x_1, x_2, \dots, x_n подсчитывает сумму положительных чисел.

Пользователь вводит значение n в текстовое поле формы. Передача данных осуществляется методом GET. Массив значений x генерируется случайным образом. Для вычисления создать пользовательскую функцию *calculation_x*, параметры n и x передавать по ссылке. Вывод результата при помощи конструкции *echo*.

Лабораторная работа №2 ТЕХНОЛОГИИ СООКИЕ И СЕССИЙ

Целью данной лабораторной работы является изучение возможностей суперглобальных массивов PHP для работы с данными веб-приложения.

Протокол HTTP является протоколом "без сохранения состояния". Это означает, что данный протокол не имеет встроенного способа сохранения состояния между двумя транзакциями. То есть, когда пользователь открывает сначала одну страницу сайта, а затем переходит на другую страницу этого же сайта, то основываясь только на средствах, предоставляемых протоколом HTTP невозможно установить, что оба запроса относятся к одному пользователю. Т. о. необходим метод, при помощи которого было бы отслеживать информацию о пользователе в течение одного сеанса связи с Web-сайтов. Одним из таких методов является управление сеансами при помощи предназначенных для этого функций. Для нас важно то, что сеанс по сути, представляет собой группу переменных, которые, в отличие от обычных переменных, сохраняются и после завершения выполнения PHP-сценария.

Сессии и cookies предназначены для хранения сведений о пользователях при переходах между несколькими страницами. При использовании сессий данные сохраняются во временных файлах на сервере. Файлы с cookies хранятся на компьютере пользователя, и по запросу отсылаются браузером серверу.

Использование сессий и cookies очень удобно и оправдано в таких приложениях как Интернет-магазины, форумы, доски объявлений, когда, во-первых, необходимо сохранять информацию о пользователях на протяжении нескольких страниц, а, во-вторых, своевременно предоставлять пользователю новую информацию [4].

Задание для выполнения лабораторной работы

При подготовке к выполнению лабораторной работы следует изучить возможности хранения и передачи данных при помощи технологий cookie и сессий.

Задание к лабораторной работе:

Данная работа создается на основе предыдущей лабораторной работы.

1. При первом обращении затребовать ввод имени пользователя;

2. Используя технологию cookie, при последующих обращениях приветствовать пользователя по введенному имени, а также выдавать дату последнего посещения, заполнять поля формы ввода последними удачно введенными значениями;
3. Создать механизм позволяющий сменить имя пользователя, используя технологию cookie.
4. Реализовать аналогичные задачи при помощи технологии сессий.

После выполнения лабораторной работы необходимо представить два веб-приложения, а также отчет о проделанной работе (в электронном и печатном виде).

Отчет должен содержать:

1. **Титульный лист** (приложение А).
2. В **содержании** указываются названия всех частей отчета (кроме титульного листа и самого содержания) со ссылками на соответствующие страницы. Содержание должно быть сформировано и вставлено в отчет средствами MS Word.
3. **Введение** должно содержать цель и задачи лабораторной работы, задание, анализ предметной области.
4. **Основная часть** должна содержать листинг кода страницы.
5. **Заключение** должно содержать краткие выводы по результатам проделанной работы.
6. **Список использованных источников.** Отчет должен содержать полный список использованных литературных источников с соблюдением ссылок при цитировании. При составлении списка следует соблюдать требования ГОСТов [2, 3].

Отчет должен быть выполнен *на русском языке* с соблюдением его грамматических и синтаксических правил. Оформление текстовых и графических материалов курсовой работы [3] производится в соответствии с требованиями комплекса стандартов Единой системы конструкторской документации (ЕСКД). Электронный вариант отчета о курсовой работе готовится с помощью текстового процессора MS Word для односторонней распечатки на стандартных листах бумаги формата А4. Размеры полей должны быть следующими: левое, верхнее и нижнее – по 20 мм, правое – 15 мм. Основной текст набирается шрифтом Times New Roman размером 12 pt, кроме титульного листа, содержимое которого набирается шрифтом 14 pt. Настройки абзаца: выравнивание – по ширине, первая строка – красная (отступ 1,25 см), интервал (отступ)

перед абзацем – 6 pt, междустрочный интервал – одинарный. Фрагменты исходных текстов набирать шрифтом Courier, Courier new.

Названия структурных элементов отчета, заголовки разделов и подразделов основной части работы выделяются жирным шрифтом размера 14 pt с использованием стилей заголовков различного уровня для последующей вставки содержания. Отдельные слова для усиления смысловой нагрузки можно выделять жирным или курсивом, не злоупотребляя эффектами начертания.

Каждый структурный элемент отчета и каждый новый раздел основной части начинается с новой страницы. Все страницы отчета должны иметь сквозную нумерацию. Номер страницы пишется арабскими цифрами размера 12 pt и располагается снизу по центру. Титульный лист первым включается в общую нумерацию, но при этом номер на нем не проставляется.

Контрольные задания

Вариант 1

Пользователь выбирает цвет фона для станицы, при повторном посещении настройки сохраняются. Реализовать при помощи технологии Cookie.

Вариант 2

При повторном посещении страницы сохраняются предыдущие результаты табулирования функции (все, только на концах промежутка). Реализовать при помощи технологии Cookie.

Вариант 3

При регистрации пользователя запретить ввод логинов, которые ранее были заняты другими пользователями. Реализовать при помощи технологии Cookie.

Вариант 4

Пользователь с логином admin, устанавливает количество разбиений для табулирования. Остальные пользователи могут устанавливать только значения конца и начала промежутка, количество интервалов, то которое установил admin. Реализовать при помощи технологии Cookie.

Вариант 5

Хранить пароль в зашифрованном виде (обратная последовательность символов). Реализовать при помощи технологии Cookie.

Вариант 6

На форме регистрации присутствует флаг «Запомнить меня», если он включен, то при каждом посещении страницы выводить количество посещений данным пользователем. Реализовать при помощи технологии Cookie.

Вариант 7

Пользователь выбирает язык страницы. При последующем посещении настройки сохраняются. Реализовать при помощи технологии Cookie.

Вариант 8

Пользователь выбирает цвет фона для станицы, при повторном посещении настройки сохраняются. Реализовать при помощи технологии сессий.

Вариант 9

При повторном посещении страницы сохраняются предыдущие результаты табулирования функции (все, только на концах промежутка). Реализовать при помощи технологии сессий.

Вариант 10

При регистрации пользователя запретить ввод логинов, которые ранее были заняты другими пользователями. Реализовать при помощи технологии сессий.

Вариант 11

Пользователь с логином admin, устанавливает количество разбиений для табулирования. Остальные пользователи могут устанавливать только значения конца и начала промежутка, количество интервалов, то которое установил admin. Реализовать при помощи технологии сессий.

Вариант 12

Хранить пароль в зашифрованном виде (обратная последовательность символов). Реализовать при помощи технологии сессий.

Лабораторная работа №3

ДОСТУП К БАЗЕ ДАННЫХ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ AJAX

Целью данной лабораторной работы является изучение на практике работы с СУБД MySQL и изучение принципов построения архитектуры веб-приложения с использованием технологии Ajax.

СУБД MySQL – одна из множества баз данных, поддерживаемых в PHP. Система MySQL распространяется бесплатно и обладает достаточной мощностью для решения реальных задач.

Краткое введение в MySQL

SQL – это аббревиатура от слов Structured Query Language, что означает структурированный язык запросов. Этот язык является стандартным средством для доступа к различным базам данных.

Система MySQL представляет собой сервер, к которому могут подключаться пользователи удаленных компьютеров.

Для работы с базами данных удобно пользоваться средством, входящее в комплект Web-разработчика: Denwer PHPMyAdmin. Здесь можно создать новую базу данных, создать новую таблицу в выбранной базе данных, заполнить таблицу данными, а также добавлять, удалять и редактировать данные [5].

AJAX расшифровывается как Asynchronous JavaScript + XML (асинхронный JavaScript+XML).

Клиент-серверная технология в чистом виде не может удовлетворить всем требованиям, которые предъявляются к некоторым веб-приложениям. В синхронной модели браузер отправляет запрос на сервер и висит, ждет, пока тот совершит всю необходимую работу. Сервер выполняет запросы к базе данных, заворачивает ответ в необходимый формат и выводит его. Браузер, получив ответ, вызывает функцию показа. Все процессы выполняются последовательно, один за другим.

В асинхронной модели запрос отсылается, и можно заняться чем-то другим. Когда запрос выполнен – запускается заранее подготовленная программистом функция показа сообщения сервера. Здесь сервер сразу же уведомляет браузер о том, что запрос принят в обработку и освобождает его для дальнейшей работы. Когда ответ будет готов – сервер перешлет его, и на браузере будет вызвана соответствующая функция показа, но пока этот ответ формируется и пересылается – браузер свободен. Пользователь может написать комментарии, запол-

нить и отослать форму и т.п. Могут производиться новые асинхронные запросы.

Асинхронная модель характеризуется почти мгновенной реакцией на действия пользователя, так что создается впечатление удобного и быстрого приложения [7-8].

Задание № 1 для выполнения лабораторной работы

При подготовке к выполнению лабораторной работы следует изучить синтаксис языка запросов SQL, а также функции PHP для работы с базами данных.

1. При помощи приложения PHPMyAdmin создать базу данных, состоящую из одной таблицы. Таблица должна содержать минимум пять полей, должны присутствовать поля строкового типа, числового, логического и поле типа дата.

2. Создать PHP-скрипты, обеспечивающие сохранение полученной из пользовательской формы информации в базе данных.

3. Обеспечить возможность просмотра, редактирования и удаления ранее введенных данных.

Задание № 2 для выполнения лабораторной работы

При подготовке к выполнению второй части лабораторной работы следует выполнить следующее упражнение:

Добавление комментариев без перезагрузки страницы

Скопируйте в папку с приложением файлы из папки comment. Создайте форму:

```
<form action="#" method="POST" enctype="multipart/form-data" name="addcom"
id="addcom" onSubmit="return false">
Ваше имя:<br>
<input name="author" type="text" size="30" class="pole"
id="author"><br><br>
Текст комментария:<br>
<textarea name="text" rows="5" cols="50" class="text"></textarea><br>
<br>
<input name="id" type="hidden" value="<?php echo $id; ?>">
<input class="adscom" name="button" type="button" value="Добавить коммен-
тарий" onclick="doLoad(document.getElementById('addcom'))">
</form>
```

В форме содержится скрытое поле id, которое нужно для того, чтобы мы знали для какой именно статьи данный комментарий. Так же важно то чтобы у формы был атрибут enctype="multipart/form-data" без

него ничего работать не будет. Теперь нужно подключить между `<head></head>` javascript-ы для корректной работы:

```
<script type="text/javascript" src="js/comment.js"></script>
<script type="text/javascript" src="js/JsHttpRequest.js"></script>
```

Теперь мы создаем на странице над формой специальный блок, в котором и будут отображаться результаты работы, будь то успех или ошибка. И присвоим ему идентификатор `error`

```
<div id="cerror"></div>
```

После этого мы текстовым редактором откроем файл **comment.js**. В этом файле в строке **`err=document.getElementById('cerror');`** идет обращение к нашему блоку с идентификатором `cerror`. Следующая строка **`err.innerHTML = "
<p>Ваш комментарий обрабатывается...</p>";`** содержит сообщение и картинку, показываемые при обработке. Так же было бы не плохо, если бы форма очищалась при удачном добавлении комментария. Для этого пропишем следующие строки:

```
form = document.getElementById('addcom');
form.text.value = "";
form.author.value = "";
```

Сразу же укажем путь до нашего обработчика **comment.php** в строке **`req.open(null, 'comment.php', true);`** Сохраним изменения, если Вы их внесли и откроем файл **comment.php**. В самом начале у нас идет запрет на кэширование, чтобы нам не выводились предыдущие результаты. Затем идут переменные `$log` и `$error`, в которые будут заноситься ошибки. Далее идет подключение файла **JsHttpRequest.php**, который нужен для правильной работы и указывается кодировка. Затем идут различные проверки для введенных данных. Если они все проходят успешно, то добавляем все данные в базу и присваиваем их переменной `$ok`, заносим их в глобальный массив. То, что содержится в этой переменной и будет отображаться на экране. Если же произошли ошибки, то они заносятся в переменную `$log` и также заносятся в глобальный массив и выводятся на экран. Ну вот в принципе форма уже будет работать, осталось только разобраться с базой данных. Структура таблицы комментариев будет следующей:

	Поле	Тип	Сравнение	Дополнительно
<input type="checkbox"/>	id	int(5)		auto_increment
<input type="checkbox"/>	post	int(5)		
<input type="checkbox"/>	author	varchar(30)	utf8_general_ci	
<input type="checkbox"/>	text	text	utf8_general_ci	
<input type="checkbox"/>	date	varchar(30)	utf8_general_ci	

В поля **author**, **text** и **date** будут заноситься соответственно автор, текст и дата добавления. А в поле **post** будет записываться **id** статьи, который мы передавали в скрытом поле. Чтобы мы могли получать доступ к базе данных мы должны соединиться с ней. Все настройки будут храниться в файле **bd.php**, который нужно подключить в самом верху страницы **comment.php** и **index.php** [9].

Задание к лабораторной работе:

На основе первой части работы и выполненного упражнения «Добавление комментариев без перезагрузки страницы» реализовать возможности добавления, удаления, редактирования и просмотра записей базы данных, используя технологию Ajax.

После выполнения лабораторной работы необходимо представить веб-приложение, позволяющее выполнять описанные выше задачи, а также отчет о проделанной работе (в электронном и печатном виде).

Отчет должен содержать:

1. **Титульный лист** (приложение А).
2. В **содержании** указываются названия всех частей отчета (кроме титульного листа и самого содержания) со ссылками на соответствующие страницы. Содержание должно быть сформировано и вставлено в отчет средствами MS Word.
3. **Введение** должно содержать цель и задачи лабораторной работы, задание, анализ предметной области.
4. **Основная часть** должна содержать листинг кода страницы.
5. **Заключение** должно содержать краткие выводы по результатам проделанной работы.
6. **Список использованных источников**. Отчет должен содержать полный список использованных литературных источников с соблюдением ссылок при цитировании. При составлении списка следует соблюдать требования ГОСТов [2, 3].

Отчет должен быть выполнен *на русском языке* с соблюдением его грамматических и синтаксических правил. Оформление текстовых и графических материалов курсовой работы [3] производится в соответ-

ствии с требованиями комплекса стандартов Единой системы конструкторской документации (ЕСКД). Электронный вариант отчета о курсовой работе готовится с помощью текстового процессора MS Word для односторонней распечатки на стандартных листах бумаги формата А4. Размеры полей должны быть следующими: левое, верхнее и нижнее – по 20 мм, правое – 15 мм. Основной текст набирается шрифтом Times New Roman размером 12 pt, кроме титульного листа, содержимое которого набирается шрифтом 14 pt. Настройки абзаца: выравнивание – по ширине, первая строка – красная (отступ 1,25 см), интервал (отступ) перед абзацем – 6 pt, междустрочный интервал – одинарный. Фрагменты исходных текстов набирать шрифтом Courier, Courier new.

Названия структурных элементов отчета, заголовки разделов и подразделов основной части работы выделяются жирным шрифтом размера 14 pt с использованием стилей заголовков различного уровня для последующей вставки содержания. Отдельные слова для усиления смысловой нагрузки можно выделять жирным или курсивом, не злоупотребляя эффектами начертания.

Каждый структурный элемент отчета и каждый новый раздел основной части начинается с новой страницы. Все страницы отчета должны иметь сквозную нумерацию. Номер страницы пишется арабскими цифрами размера 12 pt и располагается снизу по центру. Титульный лист первым включается в общую нумерацию, но при этом номер на нем не проставляется.

Контрольные вопросы

1. Что представляет собой технология Ajax?
2. Какие технологии необходимы для реализации Ajax-приложения?
3. Чем приложение созданное с использованием технологии Ajax отличается от синхронного клиент-серверного приложения?
4. По средством чего происходит взаимодействие между клиентской и серверной частью Ajax-приложения?
5. Какие возможности предоставляет пользователю технология Ajax?
6. Какие функциональные возможности можно реализовать на сайте с использованием технологии Ajax?

7. Контрольные задания

База данных «Компьютерная фирма»

Схема БД состоит из четырех таблиц

- Product (maker, model, type)
- PC (code, model, speed, ram, hd, cd, price)
- Laptop (code, model, speed, ram, hd, screen, price)
- Printer (code, model, color, type, price)

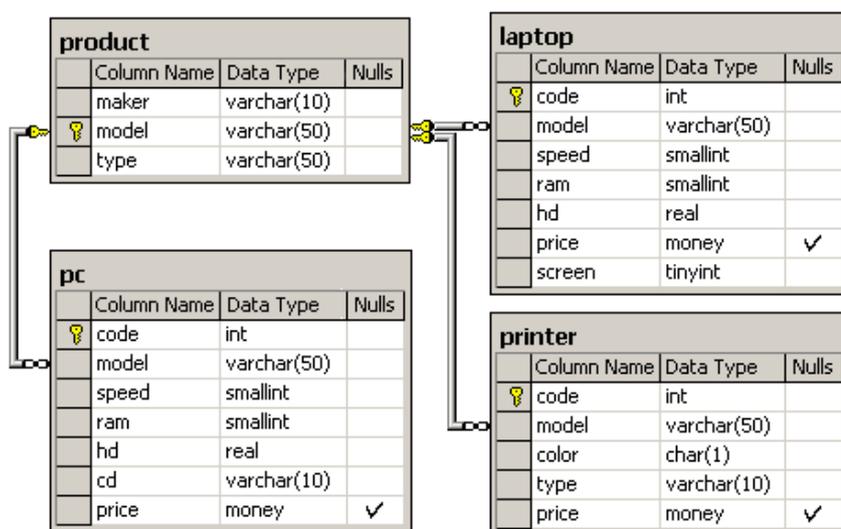


Таблица Product содержит поля:

производитель – maker;

номер модели – model;

тип – type (PC – ПК, Laptop — портативный компьютер или Printer – принтер).

Предполагается, что в этой таблице номера моделей уникальны для всех производителей и типов продуктов.

Таблица PC содержит поля:

скорость процессора – speed (МГц);

общий объем оперативной памяти – ram (Мбайт);

размер диска – hd (в Гбайт);

скорость считывающего устройства – cd (например, '4x');

цена – price.

Таблица Laptop содержит поля:

размер экрана – screen (в дюймах);

общий объем оперативной памяти – ram (в мегабайтах);

размер диска – hd (в гигабайтах);

скорость считывающего устройства – cd (например, '4x');
цена – price.

Таблице Printer содержит поля:

является ли принтер цветным – color ('y', если цветной);
тип принтера – type (лазерный – Laser, струйный – Jet или матричный – Matrix);
цена – price.

Вариант 1

Найдите номер модели, скорость и размер жесткого диска для всех ПК стоимостью менее 500 долларов. Вывести: model, speed и hd.

Вариант 2

Найдите номер модели, объем памяти и размеры экранов портативных компьютеров, цена которых превышает 1000 долларов.

Вариант 3

Найдите номер модели, скорость и размер жесткого диска ПК, имеющих 12x или 24x CD и цену менее 600 долларов.

Вариант 4

Укажите производителя и скорость портативных компьютеров с жестким диском объемом не менее 10 Гбайт.

Вариант 5

Найдите номера моделей и цены всех продуктов (любого типа) выпущенных производителем В (латинская буква).

Вариант 6

Найдите производителя, продающего ПК, но не портативные компьютеры.

Вариант 7

Найдите самые дорогие принтеры. Вывести: model, price.

Вариант 8

Найдите среднюю скорость ПК.

Вариант 9

Найдите среднюю скорость ПК, выпущенных производителем А.

Вариант 10

Найдите размеры жестких дисков, совпадающих у двух и более РС. Вывести: HD.

Вариант 11

Найдите портативные компьютеры, скорость которых меньше скорости любого ПК. Вывести: type, model, speed.

Вариант 12

Найдите производителей самых дешевых цветных принтеров. Вывести: maker, price.

Вариант 13

Найдите производителей, выпускающих по меньшей мере три различных модели ПК. Вывести: Maker, число моделей.

Вариант 14

Найдите производителей, которые производили бы как ПК со скоростью не менее 750 МГц, так и портативные компьютеры со скоростью не менее 750 МГц. Вывести: Maker.

Вариант 15

Найдите номер модели продукта (ПК, ПК-блокнота или принтера), имеющего самую высокую цену. Вывести: model.

Вариант 16

Найдите номер модели продукта (ПК, ПК-блокнота или принтера), имеющего самую высокую цену. Вывести: model [6].

Лабораторная работа №4

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ НА PHP

Целью данной лабораторной работы является изучение методики объектно-ориентированного программирования на языке PHP.

Стратегию ООП лучше всего описать как смещение приоритетов в процессе программирования от функциональности приложения к структурам данных. Это позволяет программисту моделировать в создаваемых приложениях реальные объекты и ситуации. Технология ООП обладает тремя главными преимуществами:

- она проста для понимания: ООП позволяет мыслить категориями повседневных объектов;
- повышено надежна и проста для сопровождения – правильное проектирование обеспечивает простоту расширения и модификации объектно-ориентированных программ. Модульная структура позволяет вносить независимые изменения в разные части программы, сводя к минимуму риск ошибок программирования;
- ускоряет цикл разработки – модульность и здесь играет важную роль, поскольку различные компоненты объектно-ориентированных программ можно легко использовать в других программах, что уменьшает избыточность кода и снижает риск внесения ошибок при копировании.

Специфика ООП заметно повышает эффективность труда программистов и позволяет им создавать более мощные, масштабируемые и эффективные приложения.

Объектно-ориентированное программирование основано на:

- инкапсуляции;
- полиморфизме;
- наследовании

PHP до недавнего времени обеспечивал лишь некоторую поддержку ООП. Однако, после выхода PHP5 поддержка ООП в PHP стала практически полной [1].

Задание для выполнения лабораторной работы

При подготовке к выполнению лабораторной работы следует изучить основы объектно-ориентированного программирования на PHP.

1. Используя объектно-ориентированный подход создать РНР-скрипты, обеспечивающие сохранение полученной из пользовательской формы информации в базе данных.

2. Обеспечить возможность просмотра, редактирования и удаления ранее введенных данных.

После выполнения лабораторной работы необходимо представить веб-приложение, позволяющее выполнять описанные выше задачи, а также отчет о проделанной работе (в электронном и печатном виде).

Отчет должен содержать:

1. **Титульный лист** (приложение А).

2. **В содержании** указываются названия всех частей отчета (кроме титульного листа и самого содержания) со ссылками на соответствующие страницы. Содержание должно быть сформировано и вставлено в отчет средствами MS Word.

3. **Введение** должно содержать цель и задачи лабораторной работы, задание, анализ предметной области.

4. **Основная часть** должна содержать листинг кода страницы.

5. **Заключение** должно содержать краткие выводы по результатам проделанной работы.

6. **Список использованных источников.** Отчет должен содержать полный список использованных литературных источников с соблюдением ссылок при цитировании. При составлении списка следует соблюдать требования ГОСТов [2, 3].

Отчет должен быть выполнен *на русском языке* с соблюдением его грамматических и синтаксических правил. Оформление текстовых и графических материалов курсовой работы [3] производится в соответствии с требованиями комплекса стандартов Единой системы конструкторской документации (ЕСКД). Электронный вариант отчета о курсовой работе готовится с помощью текстового процессора MS Word для односторонней распечатки на стандартных листах бумаги формата А4. Размеры полей должны быть следующими: левое, верхнее и нижнее – по 20 мм, правое – 15 мм. Основной текст набирается шрифтом Times New Roman размером 12 pt, кроме титульного листа, содержимое которого набирается шрифтом 14 pt. Настройки абзаца: выравнивание – по ширине, первая строка – красная (отступ 1,25 см), интервал (отступ) перед абзацем – 6 pt, междустрочный интервал – одинарный. Фрагменты исходных текстов набирать шрифтом Courier, Courier new.

Названия структурных элементов отчета, заголовки разделов и подразделов основной части работы выделяются жирным шрифтом размера 14 pt с использованием стилей заголовков различного уровня для последующей вставки содержания. Отдельные слова для усиления смысловой нагрузки можно выделять жирным или курсивом, не злоупотребляя эффектами начертания.

Каждый структурный элемент отчета и каждый новый раздел основной части начинается с новой страницы. Все страницы отчета должны иметь сквозную нумерацию. Номер страницы пишется арабскими цифрами размера 12 pt и располагается снизу по центру. Титульный лист первым включается в общую нумерацию, но при этом номер на нем не проставляется.

Контрольные вопросы

1. Каковы основные принципы объектно-ориентированного программирования?
2. Дайте определение понятию *класс*.
3. Дайте определение понятию *объект*.
4. Что такое *полиморфизм*?
5. Что такое *инкапсуляция*?
6. Что такое *наследование*?

ЗАДАНИЕ ДЛЯ ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

CMS – это система управления контентом и содержимым сайта. Под контентом и содержимым сайта понимают: текст, картинки, видеоданные и файлы. CMS позволяет в удобном виде создавать и управлять (редактировать, удалять) текстовыми материалами и мультимедиа документами на сайте [10-15].

При подготовке к выполнению самостоятельной работы следует ознакомиться с методикой установки CMS на локальный компьютер или хостинг; изучить принципы работы и настройки CMS.

Задание к самостоятельной работе:

1. Провести обзор и сравнительный анализ популярных CMS.
2. Установить и настроить CMS WordPress, Drupal, Joomla!.
3. Разработка сайта с применением CMS WordPress в соответствии с вариантом.

После выполнения самостоятельной работы необходимо представить 3 работающих сайта, созданных при помощи CMS; сайт WordPress должен быть модифицирован согласно варианту. Также необходимо представить отчет о проделанной работе (в электронном и печатном виде).

Варианты заданий [13-14]

№	Тема	Плагин
1	2 колонки. Правое меню	WP-Cumulus – «живое» облако тегов Плагин создает облако тегов в виде подвижной объемной сферы (flash)
2	2 колонки. Левое меню	Tagnetic Poetry интересное flash облако тегов flash-плагин для вывода облака тегов
3	3 колонки. Меню 2 стороны	WP-Polls – опросы Плагин позволяет проводить опросы на сайте, можно вставить опрос как в запись, так и в сайдбар с помощью виджета
4	4 колонки. Портфолио	NextGEN Gallery галерея NextGen Gallery – самая популярная галерея для WordPress, с помощью этого плагина легко создавать галереи, группировать их в альбомы, множество настроек и стилей для отображения фотографий
5	3 колонки. Верхнее меню	Search Unleashed – умный поиск Плагин, который позволяет искать текст в постах,

№	Тема	Плагин
		страницах, комментариях, заголовках и ссылках, подсвечивая искомое слово или фразу
6	2 колонки. Верхнее меню	Search Everything Позволяет искать не только в записях, а также в страницах, цитатах, черновиках, вложениях, тегах, комментариях и произвольных полях, есть возможность исключить определенные категории, посты и страницы из поиска
7	3 колонки. Левое меню	Яндекс.Карты для Wordpress Плагин позволяет встраивать Яндекс. Карты в ваши записи и страницы в WordPress. В отличии от Google Maps, этот плагин позволяет вставлять карты больших городов Украины и России
8	3 колонки. Меню внизу	Wordpress Thread Comment – древовидные комментарии Плагин «Wordpress Thread Comment» позволяет отображать комментарии к записям в древовидном виде, т.е. есть возможность отвечать на конкретный комментарий
9	2 колонки. Меню внизу	English spam – пропускать комментарии только на русском Анти-спам плагин English spam – это плагин, который пропускает комментарии только на русском языке
10	Резиновый шаблон	Wp-postratings – рейтинг записей Плагин позволяет посетителям оценивать записи (посты и страницы), возможность отображения самых популярных записей

Отчет должен содержать:

1. **Титульный лист** (приложение А).
2. В **содержании** указываются названия всех частей отчета (кроме титульного листа и самого содержания) со ссылками на соответствующие страницы. Содержание должно быть сформировано и вставлено в отчет средствами MS Word.
3. **Введение** должно содержать цель и задачи самостоятельной работы, задание, анализ предметной области.

4. **Основная часть** должна содержать сравнительный анализ систем WordPress, Drupal, Joomla!, а также описание процесса настройки CMS WordPress.

5. **Заключение** должно содержать краткие выводы по результатам проделанной работы.

6. **Список использованных источников.** Отчет должен содержать полный список использованных литературных источников с соблюдением ссылок при цитировании. При составлении списка следует соблюдать требования ГОСТов [7, 8].

Отчет должен быть выполнен *на русском языке* с соблюдением его грамматических и синтаксических правил. Оформление текстовых и графических материалов курсовой работы [7] производится в соответствии с требованиями комплекса стандартов Единой системы конструкторской документации (ЕСКД). Электронный вариант отчета о курсовой работе готовится с помощью текстового процессора MS Word для односторонней распечатки на стандартных листах бумаги формата А4. Размеры полей должны быть следующими: левое, верхнее и нижнее – по 20 мм, правое – 15 мм. Основной текст набирается шрифтом Times New Roman размером 12 pt, кроме титульного листа, содержимое которого набирается шрифтом 14 pt. Настройки абзаца: выравнивание – по ширине, первая строка – красная (отступ 1,25 см), интервал (отступ) перед абзацем – 6 pt, междустрочный интервал – одинарный. Фрагменты исходных текстов набирать шрифтом Courier, Courier new.

Названия структурных элементов отчета, заголовки разделов и подразделов основной части работы выделяются жирным шрифтом размера 14 pt с использованием стилей заголовков различного уровня для последующей вставки содержания. Отдельные слова для усиления смысловой нагрузки можно выделять жирным или курсивом, не злоупотребляя эффектами начертания.

Каждый структурный элемент отчета и каждый новый раздел основной части начинается с новой страницы. Все страницы отчета должны иметь сквозную нумерацию. Номер страницы пишется арабскими цифрами размера 12 pt и располагается снизу по центру. Титульный лист первым включается в общую нумерацию, но при этом номер на нем не проставляется.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Материалы сайта «Php.su. Все о PHP» [Электронный ресурс]. – Режим доступа: <http://www.php.su/php/?php>
2. ГОСТ 7.82-2001. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления. Введен впервые 01.07.2002. – Минск: Издательство стандартов, 2001. – 23 с.
3. ГОСТ 7.1-2003. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. Взамен ГОСТ 7.1-84; введен 01.07.2004. – Москва: ИПК Издательство стандартов, 2004. – 166 с.
4. Материалы сайта «Учебник по PHP 4 » [Электронный ресурс] – Режим доступа: http://www.softtime.ru/bookphp/gl8_2.php
5. Материалы сайта «Web-технологии» [Электронный ресурс]. – Режим доступа: <http://htmlweb.ru/php/mysql.php>
6. Материалы сайта «SQL и решение. Учебник. Сергей Моисеенко» [Электронный ресурс]. – Режим доступа: <http://www.sql-tutorial.ru/>
7. Материалы сайта «AJAX. Так что же это? » [Электронный ресурс]. – Режим доступа: <http://www.ixbt.com/soft/ajax.shtml>
8. Материалы сайта «Центральный Javascript-ресурс » [Электронный ресурс]. – Режим доступа: <http://javascript.ru/ajax/intro>
9. Материалы сайта «ajaxs.ru» [Электронный ресурс]. – Режим доступа: http://ajaxs.ru/lesson/ajax/110-dobavlenie_kommentariev_bez_perezagruzki_stranicy.html
10. Материалы сайта «Портал CMSList.ru» [Электронный ресурс]. – Режим доступа: <http://cmslist.ru>
11. Материалы сайта «Wordpress.org Русский» [Электронный ресурс]. – Режим доступа: <http://ru.wordpress.org>
12. Материалы сайта «WordPress Codex, справочник WordPress» [Электронный ресурс]. – Режим доступа: <http://codex.wordpress.org>
13. Материалы сайта «Мастер шаблонов» [Электронный ресурс]. – Режим доступа: <http://template-master.ru/wordpress/wordpress-template-files-for-content.html>
14. Материалы сайта «Русские шаблоны WordPress» [Электронный ресурс]. – Режим доступа: <http://wp-templates.ru/>
15. Материалы сайта «WordPress Русифицированные плагины» [Электронный ресурс]. – Режим доступа: <http://mywordpress.ru/plugins/>

ПРИЛОЖЕНИЕ А

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУЗБАССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ Т. Ф. ГОРБАЧЕВА»

Кафедра прикладных информационных технологий

Отчет по лабораторной работе по дисциплине
«Информационные системы и технологии»
на тему
«**НАЗВАНИЕ РАБОТЫ**»

Выполнил:
студент группы ПИБ-121
Фамилия И.О.
Проверила: Тайлакова А. А.

Кемерово – 2016