Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра программного обеспечения
информационных технологий

С. С. Куликов

Д. В. Деменковец

Д. В. Аврамец

ВЕБ-ТЕХНОЛОГИИ

Методическое пособие
к лабораторным работам
для студентов специальности
«Программное обеспечение информационных технологий»
всех форм обучения

Минск, БГУИР, 2020

СОДЕРЖАНИЕ

Данное методическое пособие представляет собой руководство по установке и настройке необходимого программного обеспечения и выполнению лабораторных работ.

Лабораторные работы представлены в двух видах: стандартном, предназначенном для общего использования, и расширенном, предназначенном для студентов, обладающих углубленными знаниями в предметной области дисциплины "Веб-технологии". Выбор стандартного или расширенного варианта лабораторных работ осуществляется студентом самостоятельно. Стандартный вариант лабораторных работ подразумевает выполнение одного из вариантов заданий, который назначается преподавателем.

Выполнение лабораторных работ подразумевает использование свободно распространяемого программного обеспечения, а именно:

– веб-сервер – Apache;
– СУБД – MySQL;
– среда исполнения и язык программирования – PHP;
– средство проектирования БД – phpMyAdmin;
– среда разработки программ на PHP – Notepad++.

Основная терминология, используемая в данном пособии.

Веб-клиент – приложение, выполняемое на локальной рабочей станции пользователя и устанавливающее соединения с веб-сервером по мере необходимости обмена данными. Некоторые операции могут выполняться на стороне клиента в том случае, если они не требуют информации с сервера и ориентированы, в основном, на работу с пользователем. Наиболее распространёнными веб-клиентами являются браузеры.

Веб-сервер – программное обеспечение, получающее HTTP-запросы от клиентов (см. веб-клиент) и генерирующее ответы, которые, как правило, представляют собой HTML-документы и связанные с ними данные (графические изображения, файлы CSS, XML и тому подобное) Наиболее известными веб-серверами являются Apache, Lighttpd, Nginx, Miscrosoft IIS.

Среда исполнения – программное окружение, изолирующее приложение, написанное на некотором высокоуровневом языке программирования, от операционной системы и аппаратного обеспечения. Основная задача среды исполнения – обеспечение переносимости приложений между программными и аппаратными платформами. Наиболее известными средами исполнения являются: PHP, Java Runtime Environment (JRE), Microsoft .NET Framework.

В разделе "Установка и настройка необходимого программного обеспечения" приведено подробное описание процесса подготовки рабочего места для выполнения лабораторных работ. Основной упор сделан на придание рабочему месту свойств реальных хостинговых платформ, чем обусловлено использование отдельных программных средств вместо готовых "пакетов веб-ПО".

Установка и настройка необходимого программного обеспечения

Для выполнения лабораторных работ необходимо загрузить из сети Интернет следующее программное обеспечение.

Веб-сервер Apache (версии 2.2.x или новее) по адресу:
https://www.apachelounge.com/download/
СУБД MySQL (версии 5.1.x, или 5.5.x или новее) по адресу:
https://dev.mysql.com/downloads/mysql/
Среду исполнения PHP (версии 5.3.x или новее) по адресу:
https://windows.php.net/download/
Среду проектирования БД phpMyAdmin (версии 3.3.x или новее) по адресу:
https://www.phpmyadmin.net/
Средство разработки программ на PHP Notepad++ (версии 5.8.x или новее) по адресу:
https://notepad-plus-plus.org/

Последовательность установки программного обеспечения такова.

1. Перед установкой веб-сервера Apache необходимо выполнить из командной строки команду "telnet 127.0.0.1 80" и убедиться, что соединение не может быть установлено. Это означает, что 80-й порт, по которому будет работать Apache, свободен. После этого установите веб-сервер Apache со всеми настройками по умолчанию.

2. Установите PHP, указав в опциях инсталлятора, что PHP должен работать как модуль веб-сервера Apache. Обязательно укажите, что следует использовать следующие расширения: mysql, mysqli, gd, mbstring.

3. Установите MySQL со всеми настройками по умолчанию.

4. Установите Notepad++ со всеми настройками по умолчанию.

5. Теперь необходимо правильно настроить Apache и PHP.

В файле настроек Apache httpd.conf измените значение параметра DocumentRoot на c:/www, предварительно создав такую папку. Также замените на c:/www все пути, совпадающие со старым значением DocumentRoot. Измените значение параметра DirectoryIndex: перед index.html добавьте index.php. В файле настроек PHP php.ini установите следующие значения параметров: short_open_tag = On, output_buffering = Off, max_execution_time = 30, max_input_time = 60, memory_limit = 128M, error_reporting = E_ALL, display_errors = On, post_max_size = 64M, upload_max_filesize = 64M, session.save_path="C:\WINDOWS\Temp" (удостоверьтесь, что такая папка существует!), date.timezone = 'Europe/Minsk'

6. Распакуйте содержимое архива с дистрибутивом phpMyAdmin в папку c:/www/ phpmyadmin/.

7. Перезагрузите компьютер. Всё готово. Также вы можете посмотреть видеоинструкцию по установке в материалах курса.

Лабораторная работа №1. Основы HTML и CSS

Цель работы: изучение основ языков гипертекстовой разметки HTML и управления визуальным оформлением HTML CSS.

Порядок выполнения работы

1. Изучить темы 2.1–2.4 лекционного материала.
2. Выполнить задание по лабораторной работе.
3. Представить для проверки результат выполнения работы в виде файлов HTML, CSS.

Рекомендации

Выбор типа и структуры сайта, создание статических HTML-шаблонов страниц сайта, оформление их при помощи CSS.

HTML 5

- Использовать блочные структурные элементы для организации семантики страницы:

<header> в качестве первого элемента страницы, который может включать в себя логотип, слоган, имиджевую картинку фоном; <nav> в качестве списка ссылок, которые ведут на разные страницы сайта; <h1> в качестве заголовка страницы; <section> в качестве раздела страницы; <article> в качестве основного содержимого страницы (например, основного текста) ; <sidebar> - боковая колонка, <footer> в качестве последнего элемента страницы, расположенного внизу и т.п.

- использовать блочную верстку для блочных элементов; для строчных элементов.
- Корректно создать блок <head> с определением Doctype, языка, кодировки, meta-тегов.
- На одной из страниц создать простую форму <form> 2 поля ввода, текстовое поле, кнопка отправки.

CSS 3

- задать свойства заголовков, основного текста, элементов форм, цветового оформления, отступов в блочных элементах
- использовать свойство display (block, inline-block, flex и т.д.).
- в верстке использовать технологии блочной верстки float или flex.
- желательно использовать адаптивную верстку (@media) для адаптации под мобильные устройства.

Фреймворки (типа Bootstrap) не следует использовать. Приветствуется системная, смысловая организация классов CSS кода (подобно БЭМ-методологии и т.п.)

Задание
1. Выбрать тематику сайта.
2. Сделать набросок структуры страниц (расположение заголовка, меню, основной области, боковых блоков, подвала и т.д. см. пример рис.1)
3. Разработать набор веб-страниц (HTML, CSS) выбранной тематики. Наполнить страницу содержимым соответствующим предметной области выбранного сайта. На одной из страниц обязательно должна быть форма. Использовать блочную вёрстку.

4. Каждый набор должен включать не менее пяти страниц разного типа (титульная, новости, поиск, карта сайта, каталог товаров и тому подобное).

Пример структуры сайта а показан на рисунке 1.

Рисунок 1 – Пример страницы

В процессе выполнения данной лабораторной работы особое внимание следует уделить корректности HTML и CSS кода и вопросам совместимости с различными браузерами. Изменение контента (добавление и удаление текста, изображений и тому подобное) не должно приводить к нарушению структуры HTML-страниц. По возможности следует уделить внимание вопросам дизайна, цветового оформления и использования графических элементов, а также вопросам удобства использования (юзабилити).

Особое задание:
Организовать WI-Fi сеть (локальную) и предусмотреть запуск разработанного сайта/страницы на другом устройстве или компьютере.

Полезные ссылки:
http://htmlbook.ru/
https://webref.ru/
https://webref.ru/html
https://webref.ru/css
https://ru.bem.info/methodology/
https://html5book.ru/css3-flexbox/

Лабораторная работа №2. Основы PHP, операции с массивами, строками, формы, GET- и POST-запросы.

Цель работы: изучение основ языка программирования PHP, работы с переменными, операторами, условными конструкциями, циклами. Изучить принципы работы с формами, GET/POST-запросами.

Порядок выполнения работы

1. Изучить темы 3.1–3.7 лекционного материала.
2. Выполнить задание по лабораторной работе.
3. Представить для проверки результат выполнения работы в виде файлов, распечатки и демонстрации выполнения на компьютере

Рекомендации

В данной лабораторной работе особое внимание следует уделить оформлению кода (отступы, комментарии, наименования переменных и функций), а также вопросам универсальности алгоритма.

Разработанный алгоритм должен быть независимым от данных, т. е. продолжать корректно функционировать, если входные данные будут заменены на иной произвольный набор значений – как корректный, так и некорректный.

Таким образом, в данной лабораторной работе мы постепенно приближаемся к вопросам обеспечения качества приложений в контексте устойчивости к входным данным.

При именовании переменных и функций, а также при оформлении кода рекомендуется придерживаться следующих правил:

- все имена пишутся в одном стиле;
- имена переменных пишутся в нижнем регистре, состоят из не более чем 2–3 слов, разделённых знаком подчёркивания и представляющих собой существительные или прилагательные;
- имена функций пишутся в нижнем регистре, состоят из не более чем 2–3 слов, разделённых знаком подчёркивания и представляющих собой глаголы или существительные;
- имена переменных и функций являются мнемоничными (отражают смысл хранимых данных или выполняемых действий);
- рекомендуемое количество комментариев – одна строка на 3–5 строк кода программы;
- отступы оформляются знаком табуляции или пятью пробелами;
- варианты поведения программы в условных конструкциях заключаются в операторные скобки даже тогда, когда состоят из одного оператора;
- в формах делать проверку на корректность введенных данных с соответствующей реакцией программы;
- создать файл .htaccess с кодировкой UTF-8 и выводом ошибок на экран.

Общее задание:
На экран вывести ссылки меню с названиями (например "О компании", "Услуги", "Прайс", "Контакты"). При клике по ссылке меняется и остается измененным цвет фона вокруг активной ссылки. Весь код на одной странице. Не использовать javascript. Использовать GET-запросы.

Задания по вариантам:

Вариант 1: Пользователь вводит в поле формы названия городов через запятую. После отправки отсортируйте города по алфавиту, первые буквы должны быть в верхнем регистре, остальные в нижнем. Если в запросе будут одинаковые города, они не должны повторяться.

Вариант 2: сгенерировать HTML-таблицу с количеством строк, которое вводится в поле формы (вводить количество не менее 10). Каждая 5-я строка должна иметь зеленый фон, а фон остальных строк меняется в оттенках серого от #000 до #FFF (шестнадцатеричная система) с одновременным увеличением выбранного компонента цвета в каждой следующей строке.

Вариант 3: в произвольном тексте, введенном через форму, каждое третье слово перевести в верхний регистр, каждую третью букву всех слов сделать фиолетовой, подсчитать общее количество встречающихся в тексте букв "о" и "О".

Вариант 4: Создайте 2 массива с целыми числами через 2 поля формы, объедините два массива в один (не используя специальные функции PHP типа array_merge(arr1,arr2)!), Выведите все чётные числа из получившегося массива.

Вариант 5: Введите через поле ввода строку состоящую из слов разделенных запятой (например, 'Один, два, три, четыре, пять.'). Первое слово начинается с большой буквы, в конце точка. Расставьте слова в обратном порядке. Выведите строку. Только первое слово новой строки должно начинаться с большой буквы, в конце предложения точка.

Вариант 6: Объявить пятимерный массив с целыми числами (не менее 20-ти элементов), вывести массив на экран таким образом, чтобы элементы первого уровня отображались красным цветов, второго – синим, третьего – зелёным, четвёртого – фиолетовым, пятого – жёлтым. После этого произвести сортировку элементов, где четные числа будут отображаться красным, нечетные фиолетовым, нулевые значения отображаться не будут. Данные в исходный массив вводить через поля ввода веб-формы в виде строк, где данные разделены запятыми.

Вариант 7: Объявить пятимерный массив с произвольными данными (не менее 20-ти элементов), в этом массиве (программно!) удалить все целые числа, дроби округлить до сотых, все текстовые элементы перевести в верхний регистр. Отсортировать все данные по возрастанию в строковом режиме. Отобразить на экран в табличном виде (или графически) исходный и полученный массивы. Данные в исходный массив вводить через поля ввода веб-формы в виде строк, где данные разделены запятыми.

Вариант 8: У нас есть массив часть элементов которого повторяются, необходимо вывести этот массив без дублей при помощи лишь одного цикла foreach без использования функций группировки элементов массива и не нарушая данный массив. Массив получать через веб-форму.

Вариант 9: Написать скрипт, который считает сумму цифр числа, введенного пользователем. Также сделать проверку на корректность введения данных пользователем. Число получать через веб-форму.

Полезные ссылки:
http://php.net/manual/ru/langref.php
https://metanit.com/web/php/2.1.php

Лабораторная работа №3. Специальные функции PHP, работа с файлами.

Цель работы: изучение функций, определяемых пользователем, функций по работе с датой и временем, функций по работе с файловой системой языка программирования PHP.

Порядок выполнения работы

1. Изучить темы 3.8–3.10 лекционного материала.
2. Выполнить задание по лабораторной работе.
3. Представить для проверки результат выполнения работы в виде файлов, распечатки и демонстрации выполнения на компьютере

Рекомендации

Взаимодействие с файловой системой и работа с датами – операции, при выполнении которых пользователи часто совершают ошибки. Ваша задача – реализовать полученное задание настолько устойчивым к нестандартным ситуациям, насколько это возможно.

Так, например, должны быть проверки на корректность и соответствие здравому смыслу введённых дат, на существование и тип объектов файловой системы, с которыми пользователь собирается выполнять операции и тому подобное.

В случае возникновения нештатной ситуации ваша программа должна в максимально удобной для пользователя форме реагировать на происходящее, предлагая варианты решения и предотвращая необратимые действия пользователя, которые могут привести к повреждению или потере данных.

Для всех программ, в которых данные запрашиваются через веб-форму, в случае некорректного ввода данных веб-форма должна отображаться вновь. При этом все поля должны сохранить введённые пользователем значения, а неверно заполненные поля должны быть отмечены красным цветом и дополнены подсказкой, поясняющей, в чём суть допущенной пользователем ошибки ввода.

Касательно оформления кода выполнять все требования, описанные в Лабораторной работе №2.

Задания по вариантам

Вариант 1: Работа с файлом с информацией о товарах.
Создать файл list.csv (разделитель полей запятая). На странице создать форму для добавления в базу информации о товарах: id, name (название), price (цена), description (описание). На экран вывести список имен товаров (в виде ссылок). При клике по ссылке - справа от списка показать информацию о товаре (id, name, price, description). Также дополнительно показать "price" со скидкой 15%. Округлить до 2 цифр после запятой.

Вариант 2: Работа с файлом с информацией о компаниях.
Создать файл companies.csv (разделитель полей запятая). На странице создать форму для добавления в базу информации о компаниях: name, address, phone, email. В случае отсутствия (name) или существовании такой компании в базе выдать предупреждение. Вывести информацию из файла в отдельном блоке. Сделать поиск по названию.

Вариант 3: написать функцию, определяющую точный возраст человека (с точностью до одного дня) по его дате рождения. Дату рождения получать через веб-форму. Определить дату, когда человеку исполнится, например, 10000 дней (получать через веб-форму). Определить год человека по восточному календарю.

Вариант 4: написать функцию, формирующую полный список файлов в указанном каталоге (включая подкаталоги) и считающую общий объём файлов. Имя каталога, в котором следует выполнять поиск, получать через веб-форму. Отобразить в табличном виде.
Вариант 5: написать функцию, формирующую календарь на год. Календарь представить в виде HTML-таблицы, в которой основные праздники отображаются жирным красным шрифтом, и отобразить подсказку (hint). Год, за который следует формировать календарь, получать через веб-форму. Сохранить результат в текстовый html-файл для последующего просмотра в браузере.
Вариант 6: написать функцию, формирующую список файлов в указанном каталоге (включая подкаталоги), время создания которых лежит в указанном диапазоне, а имя содержит указанное сочетание символов. Данные для поиска получать через веб-форму.
Вариант 7: написать функцию, формирующую полный список файлов и подкаталогов в указанном каталоге. Для всех элементов списка выводить размер в килобайтах (для подкаталогов считать размер их содержимого), дату и время создания, модификации и последнего обращения, а также иконку. Для всех текстовых файлов отобразить первые 100 символов. А также написать функцию, определяющую процентное отношение объёма графических файлов в произвольном каталоге (включая подкаталоги) к общему объёму данных в этом каталоге. Имя анализируемого каталога получать через веб-форму.
Вариант 8: написать функцию, формирующую календарь учебного года с указанием номера учебной недели для определенного курса. Первой неделей учебного года считается неделя, на которую приходится 1-е сентября. Номера учебных недель – от 1-го до 4-х. Год, для которого следует формировать календарь учебных недель, а также номер курса, получать через веб-форму. Время каникул и сессии отобразить жирным шрифтом и другим цветом. Сохранить результат в текстовый html-файл для последующего просмотра в браузере.
Вариант 9: “Onliner”. Необходимо вывести дату ближайшей доставки в формате: "12 апреля 2020". Желаемую дату доставки вводить через форму в формате "11.04.2020" (DD.MM.YYYY). Алгоритм следующий: если сегодня времени меньше, чем 12-00, то доставка завтра, если более 12-00, то послезавтра! Если день доставки попадает на праздничный или выходной день, то доставка переносится на следующий день после праздника. Праздники и выходные хранятся в текстовом файле "holidays.txt" в формате: "месяц-день": '08-03-2020' - 8 марта 2020.

Расширенные задания

1. Написать часть системы управления сайтом, отвечающую за добавление, удаление и перемещение файлов. Фактически требуется реализовать примитивный веб-ориентированный файловый менеджер для управления файловой системой на стороне сервера. Использовать добавленный файл, если графический отобразить, текстовый – отобразить первые 30 символов.
Дополнительными возможностями такого файлового менеджера может быть управление каталогами, редактирование текстовых файлов, конвертация графических файлов, шифрование и дешифрование файлов, создание и распаковка архивов и тому подобное.
2. Написать элементарный шаблонизатор, выполняющий поиск и подстановку в шаблонах элементов без параметров (например, {TIME}, {DATE} и тому подобное) В качестве шаблонов использовать результат задания первой ла-бораторной работы.
Результатом данной работы долен стать скрипт, генерирующий динамически часть содержимого страниц сайта.
Логика работы скрипта такова: определить запрашиваемую страницу, прочитать с диска её исходный код, провести подстановку элементов, отобразить результат.

Полезные ссылки:
http://php.net/manual/ru/funcref.php
http://php.net/manual/ru/book.filesystem.php

Лабораторная работа №4. Регулярные выражения в PHP

Цель работы: изучение основ регулярных выражений и их использования в языке программирования PHP.

Порядок выполнения работы

1. Изучить тему 3.11 лекционного материала.
2. Выполнить задание по лабораторной работе (во всех вариантах использование регулярных выражений является ОБЯЗАТЕЛЬНЫМ!).
3. Представить для проверки результат выполнения работы в виде файлов и демонстрации выполнения на компьютере

Рекомендации по выполнению заданий

- Основное требование при выполнении данной работы – универсальность и ориентированность на производительность.
- Не допускается выполнение данной лабораторной работы в виде, когда программа реагирует лишь на некоторые частные случаи вхождения искомых элементов в текст. Следует помнить, что отдельные части искомых элементов могут быть разделены переносами строк, пробелами и иными символами, которые автор документа мог использовать для форматирования текста.
- Производительность скриптов следует анализировать, разрабатывая несколько альтернативных вариантов и выбирая наиболее быстродействующий.
- В некоторых случаях алгоритмически более простым является выполнение задачи в несколько этапов с предварительным приведением данных к формату, позволяющему использовать более простые регулярные выражения, нежели в случае анализа исходного текста без предобработки.
- Ещё одним важным показателем качества программы является использование оперативной памяти. Поскольку вложенные регулярные выражения приводят к рекурсивным вызовам и геометрической прогрессии занимаемого анализируемыми данными объёма оперативной памяти, следует предусмотреть особое поведение программы для случаев, когда объём входных данных приближается к объёму доступной программе оперативной памяти.
- В предложенных выше вариантах заданий нет строгого требования к анализу текста в различных кодировках, однако при выполнении задания рекомендуется предусмотреть представление входных данных в кодировке UTF8.
- Во всех вариантах заданий результатом работы программы должна являться корректная HTML-страница, содержащая как исходный, так и преобразованный согласно заданию текст.

Задания по вариантам

Вариант 1: Напишите регулярное выражение которое будет определять корректно адреса электронной почты в формате, например, name@server.com, name.surname@subserver.server.com и т.д.. Поместите его в функцию, принимающую аргументом строку с электронной почтой. Воз-

вратите соответствующие значения результата проверки. Данные принимать через форму (Имя, E-mail) . Корректные адреса вместе с именем записывать в текстовый файл.

Вариант 2: Написать программу оставления отзывов. Форма: Имя, Сообщение. Отправленные сообщения хранить в файле и выводить в блоке выше формы. Все ссылки в отправляемом сообщении типа http://mysite.by, https://www.mysite.by/price и т.д. должны заменяться на текст #Внешние ссылки запрещены#. Ссылки на сайт БГУИР типа http://bsuir.by, https://www.bsuir.by/ru/kafedry-bguir и т.д. остаются в тексте. Вся проверка в одном регулярном выражении.

Вариант 3: Написать программу оставления отзывов. Форма: Имя, Сообщение. Отправленные сообщения хранить в файле и выводить в блоке выше формы. Все упоминания e-mail в отправляемом сообщении типа mymail@mail.com, name.surname@subserver.server.com и т.д. должны заменяться на текст #Стоп e-mail#. Упоминание почты БГУИР mymail@bsuir.by, mymail.poit@bsuir.by и т.д. остаются в тексте. Вся проверка в одном регулярном выражении.

Вариант 4: в произвольном тексте все целые числа вывести синим цветом, все дроби вывести красным цветом и округлить до десятых, все слова, начинающиеся с большой буквы, вывести зелёным цветом. Текст вводить через форму.

Вариант 5: в произвольном тексте все URL'ы вывести красным цветом и привести к виду URL. Если до преобразования присутствовала человекочитаемая часть URL'a, выводить URL в виде URL; человекочитаемая_часть. Текст загружать из файла.

Вариант 6: в произвольном тексте все e-mail адреса вывести красным цветом и привести к виду EMAIL. Дополнительно, в отдельном блоке вывести отдельно все e-mail адреса.Текст загружать из файла.

Вариант 7: в произвольном тексте все номера телефонов (предусмотреть не менее пяти вариантов записи номера) вывести зелёным цветом. При этом номера сотовых телефонов (начинаются с "+КОД-") подчеркнуть. Текст вводить через форму.

Вариант 8: в произвольном тексте все даты (в формате DD.MM.YYYY и MM/DD/YYYY, причём день и месяц могут быть однозначными, а год – двузначным) вывести красным цветом, при этом увеличить год на единицу. Формат MM/DD/YYYY привести к DD.MM.YYYY также с использованием регулярного выражение. Текст получать из файла.

Вариант 9: в произвольном тексте все слова, состоящие из английских букв, вывести синим цветом, все слова, состоящие из русских букв, вывести красным цветом, все числа вывести зелёным цветом. Текст вводить через форму.

Вариант 10: в произвольном тексте последовательности из двух и более пробельных символов заменить на один пробел, каждое предложение оформить в виде отдельного абзаца, все аббревиатуры (например ОАО, АСУ и т.п.) подчеркнуть, все числа вывести синим цветом. Текст получать из файла.

Вариант 11: в произвольном HTML-документе все заголовки вывести синим, все наклонные фрагменты текста вывести зелёным, все жирные фрагменты текста вывести красным.
Текст получать из файла.

Вариант 12: В произвольном тесте (длинный текст) выполнить. При вводе слова в форму поиска необходимо найти все упоминания этого слова в тексте и выделить (подсветить) цветом, жирным или другим настраивающим способом. В случае, если указываются 2 слова, то каждое должно искаться индивидуально, если словосочетание указывается в кавычках, то ищется как единое словосочетание. Помимо грубого поиска так же должен выполнятся поиск слова с разными окончаниями: Родина, Родины, Родиной.. Искомое слово получить через веб-форму.
Текст получать из файла.

Вариант 13: Дан длинный текст, в нём встречаются слова длиннее 7 символов! Если слово длиннее 7 символов, то необходимо: оставить первые 6 символа и добавить звёздочку. Остальные символы вырезаются. Шаблон: "я стану крутым программистом после БГУИРа" . Результат: " я стану крутым програм* после БГУИРа ". Текст вводить через форму.

Вариант 14: в произвольном тексте все e-mail адреса вывести красным цветом и привести к виду EMAIL. Также список e-mail адресов сохранить в файл. Текст вводить через форму.

Вариант 15: в произвольном тексте все слова, начинающиеся с большой буквы, но не стоящие в начале предложения, вывести красным цветом, а все такие слова, стоящие в начале предложения подчеркнуть. Набор цифр более 3 подряд вывести зеленым цветом. Текст вводить через форму.

Расширенное задание

Написать шаблонизатор (программу, управляющую сборкой готовых HTML-страниц из отдельных шаблонов). Шаблонизатор должен уметь обрабатывать следующие инструкции:

{FILE="path_to_file"} – чтение и подстановка указанного файла;

{CONFIG="value"} – чтение и подстановка значения из конфигурационного файла;

{VAR="variable_name"} – подстановка значения из массива $VARS, формируемого в процессе работы приложения;

{DB="value"} – подстановка значения из предопределённой таблицы в БД, хранящей текстовые надписи, настройки приложения и тому подобную информацию;

{IF "var_1"</>/==/!=/<=/>="var2"} PART1 {ELSE} PART2 {ENDIF} – анализ условия и удаление из шаблона той части, которая не соответствует условию; условия могут быть вложенными; часть {ELSE} может отсутствовать.

Полезные ссылки:
https://www.php.net/manual/ru/book.pcre.php
https://www.php.net/manual/ru/reference.pcre.pattern.syntax.php

Лабораторная работа №5.
Взаимодействие с реляционными СУБД (MySQL)

Цель работы: изучение возможностей языка программирования PHP по взаимодействию с реляционной СУБД MySQL.

Порядок выполнения работы

1. Изучить темы 4.1–4.4 лекционного материала.
2. Выполнить задание по лабораторной работе.
3. Представить для проверки результат выполнения работы в виде файлов и демонстрации выполнения на компьютере. БД в отдельном SQL-файле.

Рекомендации по выполнению заданий

Одной из наиболее часто встречающихся проблем при работе с базами данных является синхронизация кодировок. Чтобы избежать данной проблемы, необходимо следовать правилу: кодировки должны быть одинаковыми в:

– среде разработки приложения;
– файлах шаблонов HTML-страниц;
– конфигурационных файлах;
– таблицах БД;
– настройках взаимодействия MySQL и PHP.

Поскольку UTF8 становится стандартом де-факто, предлагается использовать именно эту кодировку, а в целях устранения проблем после установления соединения с СУБД и выбора БД следует выполнить два запроса:
SET CHARACTER SET 'UTF8' и SET NAMES 'UTF8'
Данная лабораторная работа не подразумевает разработку сложных SQL-запросов, однако следует уделить внимание корректности работы с СУБД на стороне PHP.

Для взаимодействия с MySQL в PHP предусмотрено три стандартных решения: использование расширения mysql – классический вариант, признанный устаревшим, использование расширения mysqli (MySQL Improved) – наиболее активно развивающийся способ, рекомендованный ныне к использованию большинством специалистов; использование расширения PDO (PHP Data Objects) – одно из наиболее перспективных направлений развития взаимодействия PHP с реляционными СУБД, предоставляющее дополнительный уровень абстракции и повышающий таким образом совместимость написанных на PHP программ с различными СУБД.

В данной лабораторной работе не предъявляется жёстких требований к использованию того или иного варианта взаимодействия PHP и MySQL, однако очень рекомендуется (допустим в расширенном задании, например, решив его иным методом, чем основное или различные методы в общем и основном) попробовать в решении оба современных варианта: mysqli (MySQL Improved) и PDO (PHP Data Objects), причём основной акцент сделать на применение расширения mysqli.

Общее задание

Это задание предполагает освоение общей структуры таблиц MySQL, типов полей, атрибутов, индексов (ключей), знакомство с основными SQL-запросами – поэтому рекомендуется применять приложение PhpMyAdmin.

- Создайте БД с кодировкой UTF8 (utf8_general_ci). Создайте 2 таблицы (тематика таблиц на усмотрение студента) с полями включающие поле с индексом (ключом) PRIMARY с автоинкрементом, 1-2 поля с уникальным индексом (ключом), 1-2 поля в одной из таблиц которое может быть связано с другой таблицей. Допустим, одна таблица

данных о регистрации авторов и содержит, например, поля id, name, password, ip_registration, data_registration, а другая, к примеру, содержит тематические статьи с полями id, author_id (имеется ввиду отсылка на id зарегистрированного автора из 1-й таблицы), title, text, image, data_publications, hide, opinion и т.п. Подобрать корректные и оптимальные типы данных и длину для соответствующих полей, такие как varchar, text, int, double, bigint, enum, boolean, data, timestamp и т.п.

- Сделать небольшое первичное заполнение/редактирование/выделения через PhpMyAdmin. Изучить соответствующие SQL-команды.
- Изучить импорт/экспорт файла БД в формат SQL.
- Создать скрипт PHP для доступа к созданной БД. Определить константы для доступа к БД (localhost, имя, пароль,имя БД). Подключиться к БД, обработать ошибку подключения с соответствующим сообщением, установить кодировку UTF8, вывести всю БД и (или) часть полей с применением сортировки на экран в табличном виде. Рекомендуется также вывести объединенные данные из двух таблиц (например, имена авторов и соответствующие заголовки размещенных ими статей).

Задания по вариантам

Вариант 1: написать скрипт, отображающий содержимое некоторой заданной таблицы из БД (например, список компаний и их реквизитов, список товаров и цен, часовых поясов и смещения времени по Гринвичу относительно Минска и т.п.) в графическом виде, добавление, редактирование и удаление, просмотр записей. Удаление записи предусмотреть по различным полям.
Вариант 2: написать скрипт, отображающий структуру и данные всех таблиц указанной БД. (таблиц не менее 4-х, отобразить также первичные и вторичные ключи, и типы полей).
Вариант 3: написать скрипт, формирующий в виде HTML-страницы календарь, в котором ссылками являются даты, за которые в некоторой таблице, содержащей новости, присутствуют новости. Данные о датах и содержимых новостей и картинки находятся в БД.
Вариант 4: написать скрипт, выводящий в случайном порядке заданное количество неповторяющихся записей из произвольной таблицы БД.
Вариант 5: написать скрипт, позволяющий добавить в произвольную таблицу БД произвольное количество записей со случайными данными. Скрипт должен получать в качестве входных данных имена и типы ("число" или "текст") полей таблицы, а также количество добавляемых записей.
Вариант 6: написать скрипт, получающий через форму e-mail пользователя, проверяющий его корректность и добавляющий его в таблицу БД в случае, если такого e-mail там ещё нет.
Вариант 7: написать скрипт, получающий через форму имя пользователя, пароль и подтверждение пароля. Если такого пользователя ещё нет в БД, а пароль и его подтверждение совпадают, необходимо добавить в БД имя пользователя и пароль в виде хэша sha1. А также добавить функцию изменения пароля.
Вариант 8: сформировать БД, содержащую информацию о студентах и их оценках по различным предметам (не менее пяти). Написать скрипт, формирующий список студентов с их средними баллами, а также минимальным и максимальным баллом с указанием списка предметов, за которые был получен такой балл. Предусмотреть корректировку по баллам для студента. Отобразить полный список и получившийся.
Вариант 9: написать скрипт, позволяющий выполнить произвольный запрос к СУБД или серию произвольных запросов. после чего отображающий результат выполнения запроса и статистику: время выполнения, использованная оперативная память. Запрос или серию запросов получать через веб-формы.

Вариант 10: Создать базу городов. Далее участвуют человек и компьютер. Необходимо назвать город, дальше получаем ответ от компьютера с вероятностью в 97.4% название города, чьё название начинается на последнюю букву названного игроком города. Далее ситуация повторяется, игрок должен назвать город у которого название начинается с последней буквы названным оппонентом города. Имена не могут повторяться.
Вариант 11: Написать скрипт, который бы по заданному году выводит ближайшие пять лет названия по китайскому календарю. Год получать через вебформу.

Расширенное задание

На основе результатов выполнения расширенных заданий в лабораторных работах 1–4 доработать программу формирования пользовательской и администраторской частей сайта в контексте взаимодействия с СУБД, а именно:

– реализовать хранение структуры сайта в БД;
– реализовать построение карты сайта и поиска по сайту;
– реализовать протоколирование действий администратора;
– реализовать такие модули сайта как (на выбор) голосование, показ случайного баннера, подписка на рассылку.

Полезные ссылки:

http://jtest.ru/bazyi-dannyix/
https://www.php.net/manual/ru/mysqli.quickstart.php
https://www.php.net/manual/ru/class.mysqli.php
https://www.php.net/manual/ru/book.pdo.php
https://www.php.net/manual/ru/class.pdo.php

Лабораторная работа №6.
SESSION (Сессии) и COOKIE (Куки). Хранение пользовательских данных. Идентификация, регистрация пользователя.

Цель работы: изучение механизмов управления сессиями и куки в языке программирования PHP. Освоение базовых основ процедуры регистрации пользователя.

Порядок выполнения работы

1. Изучить тему 6.6 лекционного материала.
2. Выполнить задание по лабораторной работе..

Рекомендации по выполнению заданий

Механизм сессий и куки – один из немногих способов предоставить веб-серверу возможность "опознавать" запросы, как приходящие от одного и того же пользователя.

Суть использования куки заключается в сохранении браузером данных на стороне клиента и отправке их на сторону сервера при каждом запросе.

Суть использования сессий заключается в передаче на сторону клиента специального идентификатора (который может храниться в куки или передаваться через ссылки и формы), с которым проассоциирован набор данных, сохранённый на стороне сервера.

Использование куки позволяет посетителям ресурсов применять некоторые настройки без регистрации, а серверу позволяет опознавать таких пользователей и применять их настройки, а также отслеживать статистику использования этими пользователями ресурса.

С использованием сессий и куки связано некоторое количество потенциальных проблем в области безопасности:

– куки могут быть украдены злоумышленником, а потому не следует хранить в них конфиденциальных данных;

– куки могут быть модифицированы пользователем, а потому данные, полученные из них, следует подвергать тщательной фильтрации наряду с данными, полученными через формы или ссылки;

– сессии являются более безопасным способом хранения информации, т.к. она находится только на стороне сервера, однако атака типа "кража сессии" (кража и подмена идентификатора сессии) позволяет злоумышленнику достичь своей цели и в такой ситуации, что требует принятия дополнительных мер безопасности.

Задания по вариантам

Вариант 1: Выведите форму со списками выбора цвета фона страницы, размера и цвета основного шрифта и заголовка. При отправке соответствующих значений они должны сохраниться в COOKIE пользователя. Сразу после отправки формы, а также и без отправки (например, при перезагрузке страницы или повторном вызове скрипта через некоторое время, не превышающее время жизни куки), если уже есть соответствующая COOKIE, должны быть установлены заданные параметры стилей страницы.

Вариант 2: написать два скрипта, один из которых формирует произвольный набор данных (числа, строки, массивы) и передаёт их другому скрипту в сериализованной форме. Второй скрипт десериализует данные.

Вариант 3: Сделайте авторизацию: login, password (исп. $_SESSION для хранения при перезагрузке страницы, md5() - для хеширования password). Проверьте наличие минимального количества символов в логине (например, 2), пароле (например, 5), корректность символов. При успешной регистрации должно появляться приветствие "Здравствуйте, login". Реализуйте функцию выхода с возвратом к форме регистрации.

Вариант 4: написать скрипт, позволяющий установить пользователю произвольную куки (с произвольным именем и значением, на произвольный срок). Вывести в табличном виде список всех установленных куки. Добавить возможность удалить вновь созданную куки.
Вариант 5: Выведите форму со списками выбора цвета фона страницы, размера и цвета основного шрифта и заголовка. При отправке соответствующих значений они должны сохраниться в SESSION пользователя. Сразу после отправки формы, а также и без отправки (например, при перезагрузке страницы или повторном вызове скрипта через некоторое время), если уже есть соответствующая сессия, должны быть установлены ранее заданные параметры стилей страницы.
Вариант 6: написать скрипт, выполняющий авторизацию пользователя на сайте с возможностью долговременной авторизации через куки (функция "запомнить меня"). Используйте md5() - для хеширования password и желательно еще дополнительную защиту пароля, учитывая нюансы хранения паролей в куки. Проверьте наличие минимального количества символов в логине (например, 2), пароле (например, 5), корректность символов. При успешной регистрации должно появляться приветствие "Здравствуйте, login". Реализуйте функцию выхода с возвратом к форме регистрации.
Вариант 7: написать скрипт, позволяющий определять, какие страницы сайта посетил пользователь (при этом пользователь не регистрируется и не авторизуется на сайте).
Вариант 8: написать скрипт, позволяющий определять, когда данный пользователь посещал сайт (полный список и количество посещений). При этом пользователь не регистрируется и не авторизуется на сайте.
Вариант 9: написать скрипт, позволяющий определять "путь пользователя по сайту" (какие страницы он посетил, в какой последовательности, в какие моменты времени). При этом пользователь не регистрируется и не авторизуется на сайте.
Вариант 10: написать скрипт, реализующий "электронную корзину" в интернет-магазине. Корзина должна хранить перечень и количество выбранных пользователем товаров. При этом пользователь не регистрируется и не авторизуется на сайте, а после подтверждения пользователем заказа корзина автоматически очищается.

Расширенное задание

В программных наработках, полученных в результате выполнения лабораторных работ 1–5, реализовать механизм регистрации и авторизации пользователей, механизм авторизации администратора, механизм управления списком и набором прав пользователей.

В механизме авторизации пользователей предусмотреть возможность долговременной авторизации (функция "запомнить меня"), а также кратковременной авторизации с максимальной защитой личных данных (функция "чужой компьютер").

В случае, если разрабатываемое вами программное средство допускает использование электронной корзины, реализовать её.

В случае, если разрабатываемое вами программное средство допускает использование поиска по сайту, доработать его таким образом, чтобы для каждого посетителя (в т.ч. такого, который не зарегистрирован и не авторизован) хранилась история его поисковых запросов.

Полезные ссылки:
https://www.php.net/manual/ru/function.setcookie.php
https://www.php.net/manual/ru/reserved.variables.cookies.php
https://www.php.net/manual/ru/function.session-start.php
https://www.php.net/manual/ru/reserved.variables.session.php

Лабораторная работа №7.
Работа с почтой, отправка, рассылки.

Цель работы: изучение технологий работы с почтовыми сообщениями с помощью языка программирования PHP.

Порядок выполнения работы

1. Изучить темы 8.1–8.2 лекционного материала.
2. Выполнить задание по лабораторной работе.
3. Представить для проверки результат выполнения работы в виде одного или нескольких файлов с исходным кодом на языке программирования PHP.

Рекомендации по выполнению заданий

Особое внимание уделить заголовку сообщения, т.к. сообщения с некорректным или недостаточно заполненным блоком заголовка с большой вероятностью будут отсеиваться спам-фильтрами почтовых систем получателя.
На виртуальном сервере возможно придется настроить пересылку во временные текстовые файлы.
Отправка почты большому количеству адресатов сопряжена с риском исчерпать лимит времени, отводимый PHP на выполнение скрипта. В связи с этим рекомендуется дополнительно изучить **способы разбиения объёмных задач на подзадачи**, которые будут выполняться при последующих запусках скрипта, а также способы создания "бессмертных" скриптов, которые при достижении лимита времени запускают свою новую копию, выполняя HTTP-запрос к серверу.

Задания по вариантам

Вариант 1: Выведите форму обратной связи на сайте со следующими полями: «Имя», «Телефон», «Email», «Тема», «Текст сообщения» и кнопкой «Отправить». Получите все данные из формы, проверьте их правильность, при ошибке выведите соответствующее сообщение, оставив введенные в полях формы, при успешном результате проверки - отправьте письмо. Вышлите ответ на почту отправителя "с благодарностью за отправленное сообщение и скором ответе".

Вариант 2: Выведите форму на сайте со следующими полями: «Получатели», «Тема», «Текст сообщения» и кнопкой «Отправить». В поле «Получатели» введите через пробел или другой установленный раздельный символ(, или;) адреса электронной почты получателей. Получите все данные из формы, проверьте их правильность, при ошибке выведите соответствующее сообщение, оставив введенные в полях формы, при успешном результате проверки - разошлите письмо. Сохраните в текстовом файле список получателей.

Вариант 3: Написать скрипт, отправляющий полученное через форму письмо списку адресатов, хранящемуся в БД.

Вариант 4: Написать скрипт, отправляющий полученное через форму письмо указанному адресату. Для подтверждения отправки создать текстовую и (или) графическую капчу.

Вариант 5: написать скрипт, отправляющий полученное через форму письмо указанному адресату. Письмо может содержать произвольное количество вложений (attachments). Для подтверждения отправки создать текстовую и (или) графическую капчу.

Расширенное задание

Создать программу типа калькулятора заказа. Выбор опций товара (например, пицца): вид (несколько на выбор), размер (несколько на выбор), доп. ингредиенты (несколько на выбор), доставка или самовывоз, имя, email, адрес покупателя (с проверками на корректность ввода), комментарии. Кнопки «Посчитать цену» (приветствуется написание блока расчета цен на языке JavaScript, чтобы пересчитывать стоимость "на лету" при выборе опций) и «Заказать». Отправить сформированный заказ (сообщение в html-формате, с соответствующим тематике оформлением, табличным видом и т.д.) на почты "менеджера" и "заказчика". Добавить заказ в БД. Создать приложение для "менеджера", где он сможет просматривать содержимое БД по заказам, устанавливать статус выполнения заказа, отсылать "заказчику" письмо об успешном выполнении или отказе.

Полезные ссылки:

https://www.php.net/manual/ru/function.mail.php (функция mail)
https://www.php.net/manual/ru/book.image.php (для капчи)
https://learn.javascript.ru/forms-controls (js для форм)

Лабораторная работа №8.
Генерация и анализ статистики. Динамический сайт.

Цель работы: изучение технологий генерации и анализа статистики использования интернет-ресурсов, доработка сайта с динамическим содержимым с помощью языка программирования PHP.

Порядок выполнения работы

1. Изучить темы 7.1–7.2 лекционного материала.
2. Выполнить задание по лабораторной работе.
3. Представить для проверки результат выполнения работы в виде одного или нескольких файлов с исходным кодом на языке программирования PHP.

Рекомендации по выполнению заданий

При сборе и анализе статистики в общем случае учитывается такая информация о посетителе ресурса: ip-адрес, браузер и операционная система, страна, откуда зашёл посетитель (можно определить по ip), домен и конкретный адрес страницы, с которой посетитель пришёл на сайт, как долго посетитель находился на той или иной странице (т. е. время между запросами страниц), путь пользователя по сайту (удобно для оптимизации навигации), клики пользователя по банерам (можно включать в общую статистику).

Также учитывается количество уникальных посетителей в час, день, неделю, месяц и т.д., полученный и переданный объём данных, какие поисковые системы и как часто индексируют сайт, по каким ключевым словам посетители находят сайт в поисковых системах и тому подобное.

Важным вопросом в контексте сбора и анализа статистики является производительность: процесс агрегации данных может занимать значительное время, а потому стоит реализовать отдельные функции для обработки оперативных данных и агрегации данных.

Результаты статистики следует хранить в БД или файлах, отображать по запросу в табличном виде.

Представление итоговых результатов также удобно реализовывать с помощью графических изображений (например, в формате PNG). Информацию о работе с изображениями вы можете найти в теме 7.2 лекционного материала.

Рекомендации по динамическому сайту в общем расширенном задании.

Задания по вариантам

Вариант 1: создайте 2 языковых версии сайта, состоящего из 2-3страниц (ini-файлы для разных языков, например, русского и английского). Сделайте парсинг нужного ini-файла в зависимости от того какой язык предпочтительнее у пользователя. Отобразите страницы сайта. Также должны быть ссылки на принудительный выбор языка страниц. При перезагрузках страницы и на других страницах сайта выбранный язык должен сохраняться.

Вариант 2: написать скрипт, собирающий статистику по используемым посетителями ресурса браузерам. Выводить результаты в виде HTML-таблицы со списком браузеров, отсортированным по убыванию количества пользующихся ими посетителей сайта.

Вариант 3: написать скрипт, собирающий статистику по используемым посетителями ресурса операционным системам. Выводить результаты в виде HTML-таблицы со списком операционных систем, отсортированным по убыванию количества пользующихся ими посетителей сайта.

Вариант 4: написать скрипт, собирающий статистику по времени посещения сайта. Выводить результаты в виде графиков активности посетителей за день, неделю, месяц, год. График представить в виде HTML.
Вариант 5: написать скрипт, собирающий статистику по ip-адресам, с которых посетители заходили на сайт. Выводить результаты в виде HTML-таблицы со списком ip-аресов, отсортированным по убыванию количества посещений с каждого адреса.
Вариант 6: написать скрипт, отправляющий администратору статистику посещения ресурса за день (название страницы, количество просмотров).
Вариант 7: выставить ссылку на другой сайт. Считать, сколько раз была нажата ссылка.
Вариант 8: сколько раз была показана картинка или размещается пиксель, который фиксирует показ конкретной картинки с определённым идентификатором.

Общее расширенное задание
Доработать до динамического статический сайт, разработанный на Лабораторной работе №1.
В случае добросовестного выполнения расширенных заданий к предыдущим лабораторным работам к данному моменту у вас должно получиться вполне работоспособное программное средство управления структурой и информационным наполнением интернет-ресурса (CMS – content management system).
В любом случае необходимо модернизировать этот сайт с учетом пройденного материала:
1). Повторяющиеся элементы страниц, такие как Header, Footer, меню должны подгружаться из шаблонов.
2). Сайт должен взаимодействовать с базой данных MySQL, генерировать меню, страницы и информацию из БД.
3). Сайт должен иметь активную форму регистрации пользователя
4). Должны присутствовать форма заказа, модуль подписки пользователей на рассылку уведомлений и модуль рассылки таких уведомлений или хотя бы форма обратной связи с функцией отправки по e-mail
5). Модуль сбора и анализа статистики посещения сайта

Полезные ссылки:

https://www.php.net/manual/ru/reserved.variables.server.php

ЛИТЕРАТУРА

1. Котеров Д.В. PHP 7 (в подлиннике) / Д.В. Котеров, И.В. Симдянов. – Спб: БХВ-Петербург, 2016
2. Костарев, А. PHP 5 в подлиннике / А. Костарев, Д. Котеров. – М.: BHV, 2009.
3. Холзнер, С. PHP в примерах / С. Холзнер. – М.: Бином, 2009.
4. Фридл, Дж. Регулярные выражения / Дж. Фридл – М.: Симовл, 2008.
5. Грабер, М. SQL / М. Грабер. – М.: Лори, 2009.
6. Кинкоф, Ш. HTML / Ш. Кинкоф. – М.: НТ Пресс, 2008.