

**Некоммерческое
акционерное
общество**



**АЛМАТИНСКИЙ
УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И
СВЯЗИ**

Кафедра
«Информационные
системы»

WEB-ПРОГРАММИРОВАНИЕ

Методические указания по выполнению курсовых работ
для студентов специальности 5В060200 - Информатика

Алматы 2016

СОСТАВИТЕЛИ: К.Б. Альмуратова. Web-программирование. Методические указания по выполнению курсовых работ для студентов специальности 5В060200 - Информатика. – Алматы: АУЭС, 2015. – 23 с.

Методические указания содержат основные задачи курсовой работы, требования к ее выполнению, рекомендации и указания по выполнению курсовой работы и варианты заданий.

Методические указания предназначены для выполнения курсовой работы по дисциплине «Web-программирование» студентами специальности 5В060200–Информатика. В данных методических указаниях рассмотрена технология разработки web-систем на языке PHP.

Библиогр. – 6 назв.

Рецензент: канд.техн.наук., доц. К.С. Чежимбаева

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2015 г.

© НАО «Алматинский университет энергетики и связи», 2016 г.

СОСТАВИТЕЛИ: К.Б. Альмуратова. Web-программирование. Методические указания по выполнению курсовых работ для студентов специальности 5В060200 - Информатика. – Алматы: АУЭС, 2015. – 23 с.

Методические указания содержат основные задачи курсовой работы, требования к ее выполнению, рекомендации и указания по выполнению курсовой работы и варианты заданий.

Методические указания предназначены для выполнения курсовой работы по дисциплине «Web-программирование» студентами специальности 5В060200–Информатика. В данных методических указаниях рассмотрена технология разработки web-систем на языке PHP.

Библиогр. – 6 назв.

Рецензент: канд.техн.наук., доц. К.С. Чежимбаева

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2015 г.

© НАО «Алматинский университет энергетики и связи», 2016 г.

Введение

Web-программирование - раздел программирования, ориентированный на разработку web-приложений (программ, обеспечивающих функционирование динамических сайтов Всемирной паутины).

Языки web-программирования-это языки, которые в основном предназначены для работы с веб-технологиями. Языки web-программирования можно условно разделить на две пересекающиеся группы: клиентские и серверные.

Как следует из названия, программы на клиентских языках обрабатываются на стороне пользователя, как правило, их выполняет браузер. Это и создает главную проблему клиентских языков — результат выполнения программы (скрипта) зависит от браузера пользователя. То есть, если пользователь запретил выполнять клиентские программы, то они исполняться не будут, как бы ни желал этого программист. Кроме того, может произойти такое, что в разных браузерах или в разных версиях одного и того же браузера один и тот же скрипт будет выполняться по-разному. С другой стороны, если программист возлагает надежды на серверные программы, то он может упростить их работу и снизить нагрузку на сервер за счет программ, исполняемых на стороне клиента, поскольку они не всегда требуют перезагрузки (генерацию) страницы.

Когда пользователь дает запрос на какую-либо страницу (переходит на неё по ссылке или вводит адрес в адресной строке своего браузера), то вызванная страница сначала обрабатывается на сервере, то есть выполняются все программы, связанные со страницей, и только потом возвращается к посетителю по сети в виде файла. Этот файл может иметь расширения: HTML, PHP, ASP, ASPX, Perl, SSI, XML, DHTML, XHTML.

Работа программ уже полностью зависима от сервера, на котором расположен сайт, и от того, какая версия того или иного языка поддерживается. К серверным языкам программирования можно отнести: PHP, Perl, Python, Ruby, любой .NET язык программирования (технология ASP.NET), Java, Groovy.

Важной стороной работы серверных языков является возможность организации непосредственного взаимодействия с системой управления базами данных (или СУБД) — сервером, на котором упорядоченно хранится информация, которая может быть вызвана в любой момент.

Основной целью курсовой работы является приобретение практических навыков по разработке баз данных, программной реализации приложений БД и методов защиты БД для опеределенной предметной области на основе конкретной СУБД (My SQL, HTML, JavaScript, PHP и т.д.).

Для выполнения курсовой работы необходимо знание принципов и методов проектирования баз данных, языковых средств СУБД, принципов

реализации приложений баз данных и наличие навыков работы в среде конкретной СУБД.

1 Основные задачи курсовой работы и требования к ее выполнению

1.1 Содержание курсовой работы

В рамках курсовой работы по дисциплине «Web-программирование» студент должен разработать информационную систему для учёта информации определенной предметной области, согласно индивидуальному заданию.

Разрабатываемая система должна быть построена по 3-х звенной клиент-серверной архитектуре.

В качестве клиентского приложения должен использоваться любой современный браузер. В качестве сервера приложений должен использоваться HTTP-сервер Apache. Информация должна храниться в базе данных, на СУБД MySQL.

Разрабатываемая система должна иметь модульную структуру.

Она должна решать следующие задачи:

- добавление данных;
- редактирование данных;
- удаление данных;
- представление данных.

Разрабатываемая система должна упрощать ввод и редактирование данных.

1.2 Оформление и защита курсового проекта

Тема курсовой работы, указываемой студентом в документах, формируется по шаблону: «Разработка web-системы для учёта [предметная область]».

Например: «Разработка web-системы для учёта постановок театра».

Объём пояснительной записки к курсовой работе должен быть в пределах 20-50 страниц печатного текста, оформленный согласно правилам нормоконтроля, принятым на кафедре.

Нумерация разделов начинается с раздела «Постановки задачи». Исходные коды файлов и содержимое файла sql-скрипта базы данных выносятся в приложения. Основная и графическая часть курсовой работы должны быть оформлены в соответствии с требованиями государственных стандартов.

Для сплошного текста по всем четырём сторонам листа остаются поля (левое – 25 мм, правое – 18 мм, верхнее - 20 мм, нижнее - 25 мм), используется шрифт Times New Roman №12 с полуторным междустрочным интервалом (Times New Roman №14 с одинарным междустрочным интервалом).

Текст основной части делится на разделы, разделы делятся на подразделы. Разделы должны иметь порядковые номера в пределах всей работы, обозначаемые арабскими цифрами, после номера раздела точка не ставится. Наименование разделов записывают прописными буквами с абзацного отступа, в конце наименования раздела точка не ставится. Каждый раздел рекомендуется начинать с нового листа.

Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номера раздела и подраздела, разделенных точкой, после номера подраздела точка не ставится. Наименование подраздела оформляется как заголовок строчными буквами, первая буква - прописная. В конце наименования подраздела точка не ставится.

Содержание пояснительной записки:

- титульный лист (оформленный по стандарту с обязательным указанием темы);
- пояснительная записка;
- задание на курсовой проект;
- содержание;
- постановка задачи (условие поставленной задачи, функции системы применительно к своему варианту);
- описание структуры базы данных (описание таблиц и связей между ними);
- описание структуры проекта (описание файлов, каталогов и функций);
- инструкция пользователю (подробное описание разработанной системы, использование, установка и т. д.);
- список источников и литературы;
- приложения.

При этом все листы (страницы) должны иметь сквозную нумерацию, начиная с титульного листа и включая приложение. Номер листа пишется в правом верхнем углу без точки (на титульном листе номер страницы не проставляется). Наименования, включённые в содержание, записывают строчными буквами (кроме первой). Слова «СОДЕРЖАНИЕ», «ВВЕДЕНИЕ», «ЗАКЛЮЧЕНИЕ», «СПИСОК ЛИТЕРАТУРЫ», записываются в виде заголовка прописными буквами и не нумеруются. Каждое приложение начинается с нового листа. Если их более одного, то приложения обозначают заглавными буквами русского алфавита. После слова «Приложение» следует буква, обозначающая его последовательность. Слово «Приложение» пишут строчными буквами с первой прописной и надпись располагают наверху посередине страницы. Если приложение располагается на нескольких листах, то на каждом последующем листе необходимо писать «Продолжение приложения А».

Курсовая работа может включать таблицы и рисунки (графики, схемы, диаграммы и т. п.), которые нумеруются (если их в тексте больше одного) в пределах документа.

Слово «Таблица» записывается в левом углу с абзацного отступа с прописной буквы, после номера таблицы ставится дефис, после которого с прописной буквы идет название таблицы. Слово «Рисунок» записывается под рисунком с абзацного отступа с прописной буквы, после номера рисунка ставится дефис, после которого с прописной буквы идет название рисунка. После названия таблицы (рисунка) точка не ставится.

Допускается нумерация таблиц (рисунков) в пределах раздела. В этом случае номер таблицы (рисунка) состоит из номера раздела и порядкового номера таблицы (рисунка), разделенных точкой. На все таблицы (рисунки) должны быть приведены ссылки в тексте документа.

К защите допускаются работы, выполненные в соответствии с приведёнными выше требованиями, после проверки преподавателем. На защите студент должен показать знание основных определений и понятий, используемых в работе, уметь обосновать выбор принятого решения, а также знать в подробностях все разработанные вопросы.

Защита курсовой работы проводится с демонстрацией работы разработанного приложения на компьютере.

2 Рекомендации к выполнению курсовой работы

2.1 Разработка информационной модели и проектирование базы данных

Выполняется анализ предметной области, выбранной по индивидуальному заданию в соответствии с номером по списку в группе или в соответствии с профилем работы. Определяются объекты, которые должны интересовать конечного пользователя.

Индивидуальное задание назначается преподавателем при собеседовании со студентом. Логическая модель базы данных формируется преподавателем.

Примерный перечень предметных областей, на основании которых может быть сформирован индивидуальный вариант задания:

- учёт абонентов телефонной сети (Ф. И. О., адрес, номер телефона, тариф, стоимость);
- учёт участников студенческой конференции (Ф. И. О., факультет, группа, курс, название научной секции, руководитель, название доклада);
- «Календарь погоды» (дата, температура воздуха, направление ветра, осадки);
- учёт участников предвыборной компании депутатов горсовета (Ф. И. О., возраст, политическая партия, должность, семейное положение);

- учёт книг в домашней библиотеке (название, автор, литературный жанр, издательство, цена);
- учёт автомобилей на платной стоянке (номер автомобиля, фамилия владельца, время пользования стоянкой, стоимость);
- учёт результатов баскетбольного турнира, данные о командах-участницах турнира (название, город, фамилия тренера, количество проведённых игр, количество побед);
- учёт компьютерного оборудования в учебном классе, данные о компьютере (учётный номер, код модели, тип процессора, размер диска, объём памяти);
- учёт результатов сдачи экзаменов студентами одной группы за один семестр (Ф. И. О. студента, номер зачётки, оценка по информатике, оценка по ин. яз., оценка по математике, оценка по физике);
- формирование графиков отпусков сотрудников кафедры, данные о сотрудниках (Ф. И. О., должность, табельный номер, продолжительность отпуска);
- «Электронный журнал преподавателя», учёт посещаемости и успеваемости студентов (Ф. И. О. студента, дисциплина, дата, посещение, оценка);
- «Расписание движения поездов по станции Ухта», информация о поездах, данные о поездах (номер, пункт отправления, станция назначения, время в пути, время прибытия);
- учёт клиентов библиотеки, данные о клиентах (Ф.И.О., индивидуальный номер, название книги, автор, инвентарный номер книги, срок сдачи);
- формирование и учёт больничных листов, данные о больничных листах (Ф. И. О. больного, диагноз, номер больничного листа, фамилия врача, дата выдачи, дата закрытия, количество дней);
- учёт работы копировального аппарата (дата, фамилия (кто копировал), отдел (из какого отдела), количество страниц, количество копий, цена одной копии, процент надбавки на каждые сто листов, итого);
- учёт ассортимента обуви на складе (артикул (тип обуви – мужская, женская, детская), наименование, стоимость 1 пары, покупатель, отпущено (в парах), процент скидки за объём партии, стоимость партии);
- учёт оплаты за жилую площадь (жилец, адрес, площадь жилого помещения, оплата (за один кв. метр), доплата за лишнюю жилплощадь (рассчитывается в зависимости от превышения жилой площади на одного проживающего), количество жильцов, итого);
- учёт клубных взносов в спортивном клубе (Ф. И. О. спортсмена, вид спорта, соревнование, место, клубный взнос, скидка на клубный взнос (рассчитывается в зависимости от занятого места), сумма взноса);
- учёт клиентов парикмахерского салона (клиент, мастер, квалификация мастера, услуга, дата, скидка постоянному клиенту, надбавка за квалификацию мастера, стоимость услуги);

- учёт продажи билетов в театре (название, жанр, режиссёр, стоимость билета, продано билетов, скидка на групповые заявки, дата, остаток билетов, выручено за спектакль);
- учёт отработанных часов и начисленной заработной платы сотрудников в отделе кадров (сотрудник, отдел, количество отработанных лет, отработанные часы, тариф, надбавка за выслугу лет, начислено);
- учёт доставки продуктов клиентам (клиент, продукт, количество, дальность доставки, надбавка за дальность доставки, скидка постоянному клиенту, сумма);
- учёт проката видео/аудио дисков (заказчик, название, тип носителя, формат записи, дата выдачи, дата возврата, количество дней, тариф в сутки проката, надбавка за задержку, итого);
- учёт работ строительно-отделочной организации (заказчик, вид услуги, стоимость, срок выполнения фирмой, срок выполнения заказчика, надбавка за срочность, стоимость материала заказчика, итоговая стоимость);
- учёт междугородных телефонных переговоров (абонент, город соединения, дата, коэффициент зоны, количество минут, льгота в ночное время, стоимость);
- учёт работы Интернет-клуба (пользователь, № компьютера, начало сеанса, время пользования, тариф (от быстродействия компьютера), стоимость услуги консультанта, скидка за ночное время, стоимость);
- учёт клиентов Интернет-магазина (фамилия, телефон, код товара, дата покупки, стоимость единицы товара, количество, итоговая сумма);
- учёт клиентов салона по продаже автомобилей (фамилия, № паспорта, марка машины, год выпуска, сумма, скидка, итоговая сумма);
- учёт оптовой продажи стройматериалов (№ заказа, дата продажи, вид товара, стоимость единицы, кол-во товара, сумма, скидка, итоговая сумма);
- абитуриентов вуза, данные об абитуриентах (Ф. И. О., адрес, № паспорта, специальность, средний балл, проходной балл, зачисление).

2.2 Разработка инструкций для работы с базой данных

Разрабатываются инструкции пользователя, сопровождающего программиста и администратора БД.

Инструкции пользователя должны содержать инструкции по работе с приложением для конечного пользователя с иллюстрациями в виде форм интерфейса.

Инструкции для сопровождающего программиста должны содержать описание необходимых технических и программных средств, описание файлов системы, структуры их хранения.

Инструкции для администратора БД должны содержать указания о способе хранения и изменения пароля для входа в систему, методах и средствах администрирования БД (при использовании сервера БД).

2.3 Порядок сдачи и защиты работы

Работа защищается в два этапа.

На первом этапе студент должен предоставить исходные файлы проекта и файла sql-скрипта базы данных в таком виде, чтобы преподаватель смог проверить их работоспособность.

После защиты проекта студент предоставляет документацию (пояснительную записку). В процессе защиты на втором этапе студент должен ответить на вопросы, касающиеся технологий, используемых при выполнении курсовой работы.

Заключение

Выполнение курсовой работы по дисциплинам, связанных с изучением технологии баз данных и web-программирования позволит студентам приобрести практические навыки по разработке БД и построению приложений БД с использованием методов защиты и безопасности для определенной предметной области на основе конкретной СУБД.

В процессе выполнения курсовой работы студентами приобретаются необходимые практические знания принципов и методов проектирования баз данных, языковых средств СУБД, принципов проектирования приложений баз данных и web-приложений, навыки работы в среде конкретной СУБД.

Знания и навыки, полученные в процессе выполнения курсовой работы, в дальнейшем могут быть использованы при изучении дисциплин, связанных с разработкой различных систем на основе технологии баз данных и web-программирования, а также при выполнении дипломного проектирования.

Приложение А

Пример реализации проекта

Информационная система «Информация о студентах» предназначена для автоматизации учёта основной информации о студентах.

Система построена по модульной архитектуре.

Для удобства ввода и редактирования информация о группе студента вводится с помощью поля с выпадающим списком.

Система состоит из следующих файлов:

- Index.php – основной файл проекта;
- Function.php – файл, содержащий набор необходимых функций;
- Grupa.php – вывод информации о группах;
- Grupaform.php – содержит описание форм для работы с таблицей «Группа»;
- Gruparost.php – содержит скрипты для добавления редактирования и удаления информации в таблице «Группа»;
- Student.php – выводит информацию о студентах;
- Stidentform.php – содержит описание форм для работы с таблицей «Студент»;
- Studentpost.php – содержит скрипты для добавления, редактирования и удаления информации в таблице «Студент».

Содержание файла «Index.php»

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Информация о студентах</title>
</head>
<body>
<h1>Информация о студентах</h1>
<p><a href='index.php?p=1'>Студенты</a> &nbsp;&nbsp;&nbsp;<a href='index.php?p=2'>Группы</a><p>

<?php
$p=$_REQUEST['p'];

if (($p==0)||($p==1))
    include_once 'student.php';

if ($p==11)
    include_once 'studentform.php';
```

```

if ($p==12)
    include_once 'studentpost.php';

if ($p==2)
    include_once 'grupa.php';

if ($p==21)
    include_once 'grupaform.php';

if ($p==22)
    include_once 'grupapost.php';

// echo "<p>$fiostudent<p>";
?>

<p>&copy; Толегенов Сакен Маратович<p>
</body></html>

```

Содержание файла «Function.php»

```

<?php
error_reporting( E_ERROR );

function dbconnect()
{
    $db = new mysqli('localhost', 'mysql', 'mysql', 'student');

    if (mysqli_connect_errno())
    {
        $error=mysqli_connect_error();
        $errorno=mysqli_connect_errno();
        exit ("<p>Подключение невозможно: $errorno - $error<p>");
    }
    else
    {
        return $db;
    }
}

function student()
{
    $query =
        "SELECT  grupa.grupa,  student.idstudent,  student.idgrupa,  student.
fiostudent, student.nshifr

```

```
FROM student INNER JOIN grupa ON student.idgrupa = grupa.idgrupa
ORDER BY grupa.grupa, student.fiostudent";
```

```
$handle=dbconnect();
$result = $handle->query($query);
return $result;
}
```

```
function studentid($id)
{
    $query =
    "SELECT idstudent, fiostudent, idgrupa, nshifr
    FROM student
    WHERE idstudent=$id";
```

```
$handle=dbconnect();
$result = $handle->query($query);
return $result;
}
```

```
function studentins($fiostudent, $idgrupa, $nshifr)
{
    $query =
    "INSERT INTO student (fiostudent, idgrupa, nshifr)
    VALUES ('$fiostudent', $idgrupa, '$nshifr')";

    $handle=dbconnect();
    $result = $handle->query($query);
    return $result;
}
```

```
function studentupd($idstudent, $fiostudent, $idgrupa, $nshifr)
{
    $query =
    "UPDATE student SET fiostudent='$fiostudent', idgrupa=$idgrupa,
nshifr='$nshifr'
    WHERE idstudent=$idstudent";

    $handle=dbconnect();
    $result = $handle->query($query);
    return $result;
}
```

```
function studentdel($idstudent)
```

```

{
    $query =
    "DELETE FROM student WHERE idstudent=$idstudent";

    $handle=dbconnect();
    $result = $handle->query($query);
    return $result;
}

function grupa()
{
    $query =
    "SELECT idgrupa, grupa
    FROM grupa
    ORDER BY grupa ASC";

    $handle=dbconnect();
    $result = $handle->query($query);
    return $result;
}

function grupaid($id)
{
    $query =
    "SELECT idgrupa, grupa
    FROM grupa
    WHERE idgrupa=$id";

    $handle=dbconnect();
    $result = $handle->query($query);
    return $result;
}

function grupains($grupa)
{
    $query =
    "INSERT INTO grupa (grupa) VALUES ('$grupa')";
    $handle=dbconnect();
    $result = $handle->query($query);
}

function grupapud($idgrupa, $grupa)
{
    $query =

```

```

"UPDATE grupa SET grupa='$grupa' WHERE idgrupa=$idgrupa";

$handle=dbconnect();
$result = $handle->query($query);
}

function grupadel($idgrupa)
{
    $query =
    "DELETE FROM grupa WHERE idgrupa=$idgrupa";

    $handle=dbconnect();
    $result = $handle->query($query);
}

?>

```

Содержание файла «Grupa.php»

```

<?php
include_once 'function.php';

$result=grupa();
$numresult= $result->num_rows;

echo
"<table border=1>
<th>Группы</th>
<td colspan=2><a href='index.php?p=21&m=1'>Добавить</a></td>";

for ($i=0;$i<$numresult;$i++)
{
    $row=$result->fetch_assoc();
    $idgrupa=$row['idgrupa'];
    $grupa=$row['grupa'];
    $i1=$i+1;

echo
"<tr><td>$grupa</td>
<td><a
href='index.php?p=21&m=2&id=$idgrupa'>Изменить</a></td>
<td><a
href='index.php?p=21&m=3&id=$idgrupa'>Удалить</a></td>";

```

```

}
echo
"</table>
<p>Количество записей - $numresult<p>";
?>

```

Содержание файла «Grupaform.php»

```

<?php
include_once 'function.php';

$m=$_REQUEST['m'];

if ($m==1)
{
echo "<h2>Добавить группу</h2>

<form action='index.php' method=post>
<input type='hidden' name='m' value=1>
<input type='hidden' name='p' value=22>
<table border=0>
<tr>
<th>Группа</th>
<td><input type='text' name='grupa'></td>
</tr>
<tr>
<td colspan=2 align=center><input type='submit' value='
Сохранить'></td>
</tr>
</table>
</form>";
}
if ($m==2)
{

$idgrupa=$_REQUEST['id'];

$result=grupaid($idgrupa);

$row=$result->fetch_assoc();

$grupa=$row['grupa'];

echo "<h2>Изменить группу</h2>

```

```

<form action='index.php' method=post>
<input type='hidden' name='m' value=2>
<input type='hidden' name='id' value=$idgrupa>
<input type='hidden' name='p' value=22>
<table border=0>
<tr>
  <th>Группа</th>
  <td><input type='text' name='grupa' value='$grupa'></td>
</tr>
<tr>
  <td colspan=2 align=center><input type='submit' value='
Сохранить'></td>
</tr>
</table>
</form>";
}
if ($m==3)
{
  $idgrupa=$_REQUEST['id'];

  echo "<h2>Удалить группу</h2>
<form action='index.php' method=post>
<input type='hidden' name='m' value=3>
<input type='hidden' name='id' value=$idgrupa>
<input type='hidden' name='p' value=22>
<table border=0>
<tr>
  <th>Удалить данные о группе?</th>
</tr>
<tr>
  <td align=center><input type='submit' value='Да'></td>
</tr>
</table>
</form>";
}
?>

```

Содержание файла «Grupapost.php»

```

<?php
include_once 'function.php';

$m=$_REQUEST['m'];

```

```

if ($m==1)

{
    $grupa=$_REQUEST['grupa'];

if (empty($grupa))
{
    echo "<h3>Введены НЕ все данные</h3>" ;
}
else
{
    grupains($grupa);
echo "<script>location.href='index.php?p=2'</script>";
}
}

if ($m==2)
{
    $idgrupa=$_REQUEST['id'];
    $grupa=$_REQUEST['grupa'];

if (empty($grupa))
{
    echo "<h3>Введены НЕ все данные</h3>" ;
}
else
{
    grupaupd($idgrupa, $grupa);
echo "<script>location.href='index.php?p=2'</script>";
}
}

if ($m==3)
{
    $idgrupa=$_REQUEST['id'];
    grupadel($idgrupa);
echo "<script>location.href='index.php?p=2'</script>";
}
?>

```

Содержание файла «Student.php»

```
<?php
```

```

include_once 'function.php';

$result=student();
$numresult= $result->num_rows;

echo
"<table border=1>

<th>Группа</th>
<th>ФИО студента</th>
<th>№ зачетки</th>
<td colspan=2><a href='index.php?p=11&m=1'>Добавить</a></td>";

for ($i=0;$i<$numresult;$i++)
{
    $row=$result->fetch_assoc();
    $idstudent=$row['idstudent'];
    $fiostudent=$row['fiostudent'];
    $idgrupa=$row['idgrupa'];
    $grupa=$row['grupa'];
    $nshifr=$row['nshifr'];
    $i1=$i+1;
echo
"<tr><td>$grupa</td>
    <td>$fiostudent</td>
    <td>$nshifr</td>
<td><a
    href='index.php?p=11 &id=$idstudent&m=2'>Изменить</a></td>
<td><a
    href='index.php?p=11 &id=$idstudent&m=3'>Удалить</a></td>";
}

echo "</table>
<p>Количество записей - $numresult";
?>

```

Содержание файла «Stidentform.php»

```

<?php
include_once 'function.php';

$m=$_REQUEST['m'];

if ($m==1)

```

```

{
    echo "<h2>Добавить студента</h2>

<form action='index.php' method=post>
<input type='hidden' name='m' value=1>
<input type='hidden' name='p' value=12>
<table border=0>
<tr>
    <th>ФИО студента</th>
    <td><input type='text' name='fiostudent'></td>
</tr>
<tr>
    <th>№ зачетки</th>
    <td><input type='text' name='nshifr'></td>
</tr>
<tr>
    <th>Группа</th><td>

<select name='idgrupa'>
<option selected value='0'>Не выбрано</option>";

$result=grupa();
$numresult= grupa()->num_rows;

for ($i=0;$i<$numresult;$i++)
{
    $row=$result->fetch_assoc();
    $idgrupa=$row['idgrupa'];
    $grupa=$row['grupa'];

    echo
    "<option value='$idgrupa'>$grupa</option>";
}

    echo
    "</select></td></tr>
<tr>
    <td colspan=2 align=center><input
    type='submit'
value='Сохранить'></td>
</tr>
</table>
</form>";
}

if ($m==2)

```

```

{
    $idstudent=$_REQUEST['id'];
    $result=studentid($idstudent);
    $row=$result->fetch_assoc();
    $fiostudent=$row['fiostudent'];
    $nshifr=$row['nshifr'];
    $idgrupasel=$row['idgrupa'];

    echo "<h2>Изменить студента</h2>
    <form action='index.php' method=post>
    <input type='hidden' name='m' value=2>
    <input type='hidden' name='id' value=$idstudent>
    <input type='hidden' name='p' value=12>

    <table border=0>
    <tr>
        <th>ФИО студента</th>
        <td><input type='text' name='fiostudent' value='$fiostudent'></td>
    </tr>
    <tr>
        <th>№ зачетки</th>
        <td><input type='text' name='nshifr' value='$nshifr'></td>
    </tr>
    <tr>
        <th>Группа</th><td>
        <select name='idgrupa'>
        <option value='0'>Не выбрано</option>";

    $result=grupa();
    $numresult= grupa()->num_rows;

    for ($i=0;$i<$numresult;$i++)
    {
        $row=$result->fetch_assoc();
        $idgrupa=$row['idgrupa'];
        $grupa=$row['grupa'];

        if ($idgrupasel==$idgrupa)

        echo "<option value='$idgrupa' selected>$grupa</option>";
        else

        echo "<option value='$idgrupa'>$grupa</option>";
    }
}

```

```

echo
" </select> </td> </tr>

<tr>
  <td colspan=2 align=center> <input type='submit'
value='Сохранить'> </td>
</tr>
</table>
</form>";
}

```

```

if ($m==3)
{
  $idstudent=$_REQUEST['id'];

```

```

echo "<h2>Удалить студента</h2>

```

```

<form action='index.php' method=post>
<input type='hidden' name='m' value=3>
<input type='hidden' name='id' value=$idstudent>
<input type='hidden' name='p' value=12>

```

```

<table border=0>
<tr>
  <th>Удалить данные о студенте?</th>
</tr>
<tr>
  <td align=center> <input type='submit' value='Да'> </td>
</tr>
</table>
</form>";
}
?>

```

Содержание файла «Studentpost.php»

```

<?php
include_once 'function.php';

$m=$_REQUEST['m'];
if ($m==1)
{
  $fiostudent=$_REQUEST['fiostudent'];

```

```

$nshifr=$_REQUEST['nshifr'];
$idgrupa=$_REQUEST['idgrupa'];

if (empty($fiostudent)//empty($nshifr)//($idgrupa==0))
{
    echo "<h3>Введены НЕ все данные</h3>";
}

else
{
    studentins($fiostudent, $idgrupa, $nshifr);

    echo "<script>location.href='index.php?p=1'</script>";
}

if ($m==2)
{
    $idstudent=$_REQUEST['id'];
    $fiostudent=$_REQUEST['fiostudent'];
    $nshifr=$_REQUEST['nshifr'];
    $idgrupa=$_REQUEST['idgrupa'];

    if (empty($fiostudent)//empty($nshifr)//($idgrupa==0))
    {
        echo "<h3>Введены НЕ все данные</h3><p>";
    }
    else
    {
        studentupd($idstudent,$fiostudent, $idgrupa, $nshifr);

        echo "<script>location.href='index.php?p=1'</script>";
    }

    if ($m==3)
    {
        $idstudent=$_REQUEST['id'];
        studentdel($idstudent);
        echo "<script>location.href='index.php?p=1'</script>";
    }
    ?>

```

Список литературы

- 1 Веллинг, Люк. Разработка Web-приложений с помощью PHP и MySQL / Люк Веллинг, Лаура Томсон ; пер. с англ. – 3-е изд. – М. : Издательский дом «Вильямс», 2008. – 880 с.; ил.
- 2 Веллинг, Люк. Разработка Web-приложений с помощью PHP и MySQL /Люк Веллинг, Лаура Томсон; пер. с англ. – 4-е изд. – М. : Издательский дом «Вильямс», 2008. – 848 с.; ил.
- 3 Леонард Л. И др. Разработка приложений на основе Microsoft SQL Server 2008.-М., 2010.
- 4 Тернстрем Т. Microsoft SQL Server 2008 . Разработка баз данных.-М., 2010.
- 5 Макленнен Д. SQL Server 2008 Data mining интеллектуальный анализ данных.-СПб.: «БХВ-Петербург», 2009.
- 6 Вишнеvский А. SQL Server. Эффективная работа.- СПб.: «Питер», 2009.

Содержание

Введение.....	3
1 Основные задачи курсовой работы и требования к ее выполнению	4
1.1 Содержание курсовой работы.....	4
1.2 Оформление и защита курсового проекта.....	4
2 Рекомендации к выполнению курсовой работы.....	6
2.1 Разработка информационной модели и проектирование базы данных.....	6
2.2 Разработка инструкций для работы с базой данных.....	8
2.3 Порядок сдачи и защиты работы.....	9
Заключение.....	9
Приложение А.....	10
Список литературы.....	23