

МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ И КОММУНИКАЦИИ РЕСПУБЛИКИ  
УЗБЕКИСТАН

КАРШИНСКИЙ ФИЛИАЛ ТУИТ  
ФАКУЛЬТЕТ «КОМПЬЮТЕР ИНЖИНИРИНГ»

## САМОСТОЯТЕЛЬНАЯ РАБОТА

---

По предмету: «Web-программирование»

На тему: «Создание страницы электронного журнала »

Подготовила: студентка 4-курса группы ИКТ-42 Л. Юмакаева

Принял: А. Э. Чупонов

Карши – 2015 год

## **Содержание:**

|   |           |
|---|-----------|
| <b>ВВЕДЕНИЕ.....</b>                                      | <b>2</b>  |
| <b>I. ТЕХНОЛОГИИ ВЕБ-ПРОГРАММИРОВАНИЯ.....</b>            | <b>5</b>  |
| <b>1.1ActionScript.....</b>                               | <b>5</b>  |
| <b>1.2PHP.....</b>  | <b>5</b>  |
| <b>1.3ASP.NET.....</b>                                    | <b>7</b>  |
| <b>1.4JavaServerPages.....</b>                            | <b>7</b>  |
| <b>II. КЛАССИФИКАЦИЯ ЯЗЫКОВ WEB-ПРОГРАММИРОВАНИЯ.....</b> | <b>9</b>  |
| <b>2.1Клиентские языки.....</b>                           | <b>9</b>  |
| <b>2.2Серверные языки.....</b>                            | <b>10</b> |
| <b>III. КЛАССИФИКАЦИЯ БАЗ ДАННЫХ.....</b>                 | <b>11</b> |
| <b>IV. РЕАЛИЗАЦИЯ.....</b>                                | <b>13</b> |
| <b>4.1Проектируем структуру базы данных.....</b>          | <b>14</b> |
| <b>4.2Практическая часть.....</b>                         | <b>16</b> |
| <b>4.3Скрипт.....</b>                                     | <b>23</b> |
| <b>ЗАКЛЮЧЕНИЕ.....</b>                                    | <b>27</b> |
| <b>ГЛОССАРИЙ.....</b>                                     | <b>28</b> |
| <b>ЛИТЕРАТУРА.....</b>                                    | <b>30</b> |

## ВВЕДЕНИЕ

Специальность веб-мастера со временем теряет свою популярность, а технология построения информационно-справочного ресурса выходит из стен лабораторий на просторы промышленного освоения. Верный признак этого – это растущие инвестиции в Интернет-проекты и осознание того, что веб-ресурсы пригодны не только для распространения on-line развлечений, но и для серьезной образовательной, научной и легальной коммерческой деятельности. А раз так, то и подход к изучению Интернет-технологий должен быть серьезным. Для начала необходимо разобраться, в каком направлении движутся Интернет-технологии и разобраться в той системе, которую используют сегодня разработчики. Выделим несколько интересующих нас современных тенденций:

- в веб-технологиях: какие грядут архитектурные подходы, программное обеспечение, специальная аппаратура, серверы так далее, и как сделать из Интернет-ресурса – рекламного буклета нормальный веб-ресурс с внешней Интернет, партнерской экстрапет и внутренней интранет частью;
- экономике строительства сайта: анализ рынков (кто, где и сколько платит за веб-развитие) и связанный с анализом рынков обзор инвестиционных перспектив отрасли;
- маркетинге веб-дизайна и веб-программинга: какие аргументы и бизнес-модели используют лидеры отрасли, чтобы организовать поток заказов, контент (содержание) – происхождение и потребление;
- людских ресурсах отрасли: откуда в отрасль приходят специалисты, куда разработчики из отрасли уходят, какие еще в отрасли есть профессии, кроме веб-мастера, и кто учит профессионалов, и сколько времени нужно учиться, сколько платят в отрасли разным специалистам и так далее;

- о социальных аспектах: от проблем приватности до проблем соблюдения смежных прав (копирайта) на визуальный дизайн.

Веб-ресурсы – это любые программные комплексы, имеющие Интернет-интерфейс. В полном соответствии с названием: веб-ресурс – место в паутине. Все труднее сообразить, какие программные комплексы не имеют этот самый Интернет-интерфейс. Поэтому программы становятся веб-ресурсами, а строительство Интернет-ресурса и прикладное программирование становятся неразличимыми.

Собственно, если программы становятся Интернет-ресурсами, то все инструменты программирования становятся инструментами веб-строительства. Это вызов современному программированию, потому, что у этой отрасли по-прежнему нет ответа на вопрос о том, как многим желающим непрофессионалам построить свой сайт быстро и безошибочно писать сложные многопользовательские и многозадачные приложения с графическими интерфейсами.

Сегодня все отрасли человеческой деятельности автоматизируются и многие из них переходят на электронный вид. Не является исключением и преподавательская деятельность, где почти все приобрело электронный вид. Ярким примером можно привести применение систем дистанционного обучения, которые можно разместить как на локальной компьютерной сети учебного заведения, так и в сети Интернет. Еще одним примером является электронный журнал оценивания знаний учащихся. Его так же можно разместить в сети Интернет либо использовать локально внутри учебного заведения. Применение электронного журнала обладает рядом преимуществ: он открыт, то есть любой желающий (родители, другие преподаватели, администрация учебного заведения, сами учащиеся) имеют определенный доступ к данным журнала; может быть использована функция СМС оповещения, которая будет сообщать родителям данные об успехах и посещаемости их детей; освобождает преподавателя от ведения бумажного варианта журнала, и многие другие.

Самостоятельная работа посвящена разработке «Электронного классного журнала» (ЭКЖ) и ее целью является приобретение навыков разработки web-приложения для работы с базой данных. Задачи, которые необходимо решить при создании программного продукта:

- Анализ предметной области;
- Проектирование базы данных (БД);
- Разработка приложения для взаимодействия с БД.

## **I. ТЕХНОЛОГИИ ВЕБ-ПРОГРАММИРОВАНИЯ**

### **1.1 ActionScript**

ActionScript – это объектно-ориентированный язык программирования, один из диалектов EcmaScript, который добавляет интерактивность, обработку данных и многое другое в содержимое Flash-приложений. ActionScript исполняется виртуальной машиной (ActionScriptVirtualMachine), которая является составной частью приложения FlashPlayer. ActionScript компилируется в байт-код, который включается в SWF-файл.

SWF-файлы исполняются FlashPlayer. Сам FlashPlayer существует в виде плагина к веб-браузеру, а также как самостоятельное исполняемое приложение. Во втором случае возможно создание исполняемых exe-файлов, когда swf-файл включается во FlashPlayer.

С помощью ActionScript можно создавать интерактивные мультимедиа-приложения, игры, Интернет-ресурсы и многое другое.

### **1.2 PHP**

PHP (Hypertext PreProcessor) – это один из самых распространенных инструментов веб-программирования на стороне сервера. Работа PHP в самом простом варианте сводится к обработке HTTP запроса клиента. Обработка запроса, в свою очередь, заключается в программном формировании гипертекста в соответствии с параметрами запроса, после чего полученная разметка возвращается клиенту. Когда клиент (Интернет-браузер) запрашивает обычную статическую Интернет страницу (чаще всего с расширением HTML), сервер в качестве ответа возвращает ему содержимое этой страницы без изменений. Если запрашивается php-страница, то в процессе обработки запроса содержимое указанной

страницы сначала обрабатывается интерпретатором PHP, и только потом результат этой обработки отправляется клиенту.

Можно сказать, что PHP – это генератор гипертекста, поскольку в большинстве случаев его работа – это программная генерация HTML разметки по содержимому базы данных или по любой другой структурированной информации, размещенной на сервере.

PHP – это язык программирования, который поддерживает практически все основные конструкции процедурного программирования: переменные, условные операторы, циклы, функции и так далее. PHP – это объектно-ориентированный язык программирования, он поддерживает классы и объекты, а также привычное наследование на уровне классов.

PHP – это язык веб-программирования, поскольку он в первую очередь создан для разработки динамических Интернет-ресурсов и поэтому содержит большое количество готовых решений, применяемых в этой сфере, таких как:

- обработка и извлечение параметров HTTP-запросов GET и POST;
- формирование и отправка HTTP-заголовков;
- инфраструктура для хранения данных сеанса;
- программные сервисы для работы с cookies.

Cookies – это текстовые данные, сохраняемые браузером на компьютере клиента, которые чаще всего содержат параметры доступа (логин и пароль) или персональные настройки пользователя. Cookies формируются браузером и автоматически, в ходе каждого удаленного обращения пересылаются серверу в заголовках HTTP-запроса.

- работа с файлами по FTP-протоколу;
- работа с базами данных посредством SQL-запросов;
- поддержка регулярных выражений;
- поддержка HTTP авторизации;
- обмен сообщениями по электронной почте и многое другое.

### **1.3 ASP.NET**

ASP.NET – это технология создания динамических веб-приложений. Она является частью платформы .NET Framework. Приложения ASP.NET можно писать на любом языке, совместимом с технологиями .NET. В их число входят VisualBasic, C# и J#. Страницы ASP.NET (веб-формы) проходят компиляцию, что обеспечивает лучшую производительность по сравнению с приложениями на основе сценариев. Веб-формы позволяют создавать многофункциональные веб-страницы. При разработке можно использовать серверные элементы управления ASP.NET для создания общих элементов пользовательского интерфейса и программирования общих задач для них. Процесс создания веб-форм ускоряется за счет использования стандартных встроенных компонентов, таких как представления GridView и DetailsView, и пользовательских компонентов, помогающих при написании кода страницы. ASP.NET значительно упрощает создание веб-приложений, имеющих практическое применение. Серверные элементы управления ASP.NET позволяют применять декларативную модель программирования, схожую с HTML, в которой для создания динамической страницы требуется гораздо меньше кода, нежели при использовании классической технологии ASP. Отображение данных, проверка ввода и загрузка файлов происходят гораздо проще. Главное же преимущество страниц ASP.NET состоит в том, что они работают в любом обозревателе, включая обозреватели Netscape, Opera, AOL и InternetExplorer.

### **1.4 JavaServerPages**

Технология JavaServerPages (JSP) – это технология, позволяющая веб-разработчикам динамически генерировать HTML, XML и другие веб-страницы. Технология позволяет внедрять Java-код, а также EL

(ExpressionLanguage) в статичное содержимое страницы. Также могут использоваться библиотеки JSP тегов для внедрения их в JSP-страницы. Страницы компилируются JSP-компилятором в сервлеты, представляющие собой Java-классы, которые выполняются на сервере. Сервлеты также могут быть написаны разработчиком, не используя JSP-страницы. Эти технологии могут дополнять друг друга.

JSP – это одна из высокопроизводительных технологий, так как весь код страницы транслируется в java-код сервлета с помощью компилятора JSP страниц Jasper, и затем компилируется в байт-код виртуальной машины java (JVM).

Основным отличием от других подобных технологий, например ASP и PHP, является то, что код, находящийся внутри специальных тэгов не интерпретируется при обращении к странице, а предварительно компилируется в JavaServlet. Технология JSP удачно объединяет шаблонный подход к построению сайтов и все преимущества Java-платформы.

Важным шагом к расширению шаблонного подхода стали библиотеки тэгов. Они предоставляют хорошую возможность интегрировать стандартные, сторонние, или собственные программные компоненты в страницы. Простота создания и использования привели к большой популярности библиотек тэгов.

JSP не привязана к конкретной платформе, благодаря использованию Java. Кроссплатформенность является одним из основных преимуществ этой технологии. Также к достоинствам JSP следует отнести простоту разработки, наличие большого количества библиотек и легкость их использования, мощные и разнообразные среды разработки.

## II. КЛАССИФИКАЦИЯ ЯЗЫКОВ WEB-ПРОГРАММИРОВАНИЯ

### 2.1 Клиентские языки

Клиентские языки обрабатываются на стороне клиента, то есть программы на клиентском языке обрабатывает браузер. Отсюда следует и недостаток – это то, что обработка скрипта зависит от браузера пользователя, и пользователь имеет полномочия настроить свой браузер так, чтобы он вообще игнорировал написанные вами скрипты. При этом, если браузер старый, он может не поддерживать тот или иной язык или версию языка, на которую вы опираетесь. Преимущество же клиентского языка заключается в том, что обработка скриптов на таком языке может выполняться без отправки документа на сервер.

Достоинством клиентских языков программирования является то, что нет необходимости отправлять веб-документ для обработки на сервер. Эта особенность значительно повышает скорость чтения скрипта. Уменьшается трафик, что экономит деньги и снижается нагрузка на сервер. Так же нет необходимости ставить дополнительное программное обеспечение для обработки скриптов, так как их обрабатывают браузеры. Подготовленный web-мастер способен установить на сайт с помощью скриптов разнообразные «фишки», которые позволяют существенно улучшить восприятие Интернет-ресурса и управление им, значительно повысить его юзабилити.

Недостатки клиентских языков программирования также вытекают из особенностей. Разные браузеры могут интерпретировать скрипты, исходя из своих индивидуальных предпочтений. Поэтому, результаты показа одних и тех же документов могут отличаться. Так же к недостаткам можно отнести и открытость кода таких языков для пользователей. Самыми распространенными клиентскими языками программирования являются: JavaScript, VBScript, ActionScript, Java.

## 2.2 Серверные языки

Серверные языки программирования соответственно работают на стороне сервера. Во взаимодействии с базами данных они поддерживают связь между пользователем и сервером. Получая запрос с адресом веб-документа от браузера, серверные программы связываются с базой данных. База данных отдаёт информацию о веб-странице скриптам сервера, и те обработав её, отсылают для интерпретации браузеру клиента, который и выводит результат совместной работы на монитор.

Достоинством серверных языков является безграничные возможности и то, что их работа не подвержена воздействию пользователей и скрыта от их взоров. Недостаток – зависимость от программного обеспечения хостера. Так же к недостаткам можно отнести и сложность освоения новичками серверных языков программирования. К серверным языкам программирования можно отнести HTML, PHP, ASP, PERL, SSI, XML, DHTML, XHTML.

### III. КЛАССИФИКАЦИЯ БАЗ ДАННЫХ

База данных (БД) – это представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины.

Существует огромное количество разновидностей баз данных, отличающихся по различным критериям:

1) по характеру хранимой информации базы данных делятся на:

- фактографические базы данных – это картотеки, где хранится краткая информация в строго определенном формате;
- документальные базы данных – это архивы, где хранятся всевозможные документы, это могут быть не только текстовые документы, но и графика, видео и звук (мультимедиа).

2) по способу хранения данных делит базы данных на:

- централизованные базы данных – где вся информация хранится на одном компьютере;
- распределенные базы данных – используются в локальных и глобальных компьютерных сетях;
- неоднородная база данных – фрагменты распределённой базы данных в разных узлах сети поддерживаются средствами более одной системы управления базами данных;
- однородная база данных – фрагменты распределённой базы данных в разных узлах сети поддерживаются средствами одной и той же системы управления базами данных;
- фрагментированная или секционированная база данных – методом распределения данных является фрагментация (партиционирование, секционирование), вертикальное или горизонтальное;

— тиражированная база данных — методом распределения данных является тиражирование (репликация).

3) по структуре организации данных:

- реляционная база данных;
- иерархическая база данных;
- сетевая база данных;
- объектная и объектно-ориентированная базы данных;
- объектно-реляционная база данных;
- функциональная база данных.

4) по среде постоянного хранения:

- во вторичной памяти, или традиционная — в ней средой постоянного хранения является периферийная энергонезависимая память (вторичная память), как правило, жёсткий диск;
- в оперативную память система управления базами данных (СУБД) помещает лишь кэш и данные для текущей обработки;
- в оперативной памяти — все данные на стадии исполнения находятся в оперативной памяти;
- в третичной памяти — в ней средой постоянного хранения является отсоединяемое от сервера устройство массового хранения (третичная память), как правило, на основе магнитных лент или оптических дисков;
- во вторичной памяти сервера хранится лишь каталог данных третичной памяти, файловый кэш и данные для текущей обработки; загрузка же самих данных требует специальной процедуры.

5) по содержимому:

- географическая база данных;
- историческая база данных;
- научная база данных;
- мультимедийная база данных и так далее.

## IV. РЕАЛИЗАЦИЯ

Конечно, было бы заманчиво реализовать базу данных для ведения настоящего электронного журнала целиком.

В рамках данной самостоятельной работы мы ограничимся одной страницей журнала. Более точно, речь пойдет об одном разделе журнала, посвященном одному предмету, причем (это дополнительное упрощающее ограничение) класс не разбивается на группы. Если часов по предмету много, то соответствующий раздел в журнале скорее всего физически будет занимать несколько страниц, мы этот факт не будем считать существенным (он ведь не имеет никакого отношения к структуре самих данных — можно было бы взять просто журнал очень большого формата).

Практически во всех учебных заведениях используется одна и та же утвержденная типовая форма журнала, поэтому мы с вами, взяв с полки в учительской любой журнал и открыв его на типичной странице, увидим примерно одно и то же. Так что же мы увидим?

| Predmet web-programmirovaniye |                         |       |       |       |       |       |       |       |       |       |       |       |       |
|-------------------------------|-------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| #                             | Familija, imya          | 03/02 | 04/02 | 05/02 | 06/02 | 10/02 | 11/02 | 12/02 | 13/02 | 17/02 | 18/02 | 19/02 | 20/02 |
| 1                             | Ahmedjanov Temur        |       | N     |       | 2     |       |       |       | 3     |       |       | 3     |       |
| 5                             | Hudoykulova Nilufar     | 3     |       |       |       |       | 3     |       |       |       | 3     |       | 3     |
| 2                             | Jomurotova Sitora       |       |       |       |       | 3     |       |       | 2     |       |       |       | 2     |
| 3                             | Nomozova Shahnoza       |       |       |       |       | 3     |       |       |       | 3     |       |       | 3     |
| 4                             | Nurmamatova Anbarabegim |       |       | N     |       | 2     | 2     |       |       | 2     |       |       | 2     |
| 6                             | Yumakayeva Luiza        |       |       |       |       |       |       | 3     |       |       | 2     |       |       |
| 7                             | Yusupov Bobur           |       |       |       | 3     |       |       | 2     |       |       |       | 3     |       |

| Data  | Tema uroka                                 | Domashnaye zadaniye   |
|-------|--|---|
| 03/02 | Uvedeniye v web-programmirovaniye.         | Uznat' kakiye provayderi internet deystvuyut v Uzbekistane. |
| 04/02 | Uvedeniya v HTML.                          | Nayti informatsiyu o tarifah interneta.                     |
| 05/02 | Formi i freymi v HTML.                     |   |
| 06/02 | Programmirovaniye so storoni klijenta      | Nayti informatsiyu o serverah                               |
| 10/02 | Perevemenniye v JavaScript                 | -   |
| 11/02 | Elementi upravleniya logikoy programmi     | -   |
| 12/02 | Tipi ob'ektor na JavaScript                | -   |
| 13/02 | Sozdaniye ob'ektorov                       | -   |
| 17/02 | Standartniye ob'ekti JavaScript            | -   |
| 18/02 | Svoystva okna brauzera                     | Sozdat' sayt na HTML  |
| 19/02 | Osnovniye standartniye funktsii JavaScript | -   |
| 20/02 | Bezopasnost' v JavaScript                  | -   |

## 4.1 Проектируем структуру базы данных

В подобной ситуации обычно оказывается программист-проектировщик, приступая к знакомству с предметной областью, когда ему необходимо положить сложившийся и всем привычный документооборот на музыку реляционной базы данных. Вообще говоря, конечно, проектирование — вопрос серьезный. С указанной точки зрения работа проектировщика обычно состоит из трех этапов.

**Этап 1.** Выделение сущностей (объектов) в предметной области. Применительно к нашей задаче “обычный человек” видит на странице журнала даты, темы, домашние задания, фамилии учеников и т.д. Программист же видит здесь всего две сущности — ученик и урок. Каждая сущность станет затем отдельной таблицей.

**Этап 2.** Определение свойств сущностей. Под этим мы понимаем как выделение собственно набора характеристик сущностей, так и типов этих характеристик — числа, даты, строки и т.п. Каждое свойство станет полем соответствующей таблицы.

**Этап 3.** Определение связей между сущностями и типов этих связей (“один к одному”, “один ко многим”, “многие ко многим”). Для реализации связей вида “многие ко многим” в базе данных потребуется завести таблицы.

Применим описанный алгоритм к нашей задаче, объединив первые два этапа (поскольку база данных совсем крошечная, это вполне допустимо).

Итак, в нашей предметной области имеются две сущности: ученик и урок. Опишем структуру соответствующих таблиц.

| Ученик        |             |
|---------------|-------------|
| Имя поля      | Тип поля    |
| Номер ученика | Целое число |
| Имя           | Строка      |
| Фамилия       | Строка      |

| Урок             |             |
|------------------|-------------|
| Имя поля         | Тип поля    |
| Номер урока      | Целое число |
| Дата             | Дата        |
| Тема             | Строка      |
| Домашнее задание | Строка      |

Теперь давайте разберемся с пропусками. Правильнее отмечать пропуски уроков, так как в один и тот же день ученик вполне может пропустить один урок и быть на другом. С точки зрения структуры базы данных пропуски представляют собой связь вида “многие ко многим” между сущностями “ученик” и “урок”. Ведь один ученик может пропустить *много* уроков и один урок могут пропустить *много* учеников. Для реализации подобной связи требуется завести таблицу. Например, такую:

| Пропуск       |             |
|---------------|-------------|
| Имя поля      | Тип поля    |
| Номер ученика | Целое число |
| Номер урока   | Целое число |

Обратите внимание на то, что эта таблица имеет составной ключ. Ни номер ученика, ни номер урока по отдельности не являются ключом, но уникальной является именно их комбинация!

Теперь давайте разберемся с оценками. Что такое оценка? Это тоже (как и пропуск) связь между учеником и уроком. Если бы за каждый урок ученик мог получить не более одной оценки, то для реализации этой связи можно было бы завести таблицу, похожую по структуре на таблицу “Пропуск”:

| Оценка        |             |
|---------------|-------------|
| Имя поля      | Тип поля    |
| Номер ученика | Целое число |
| Номер урока   | Целое число |
| Оценка        | Целое число |

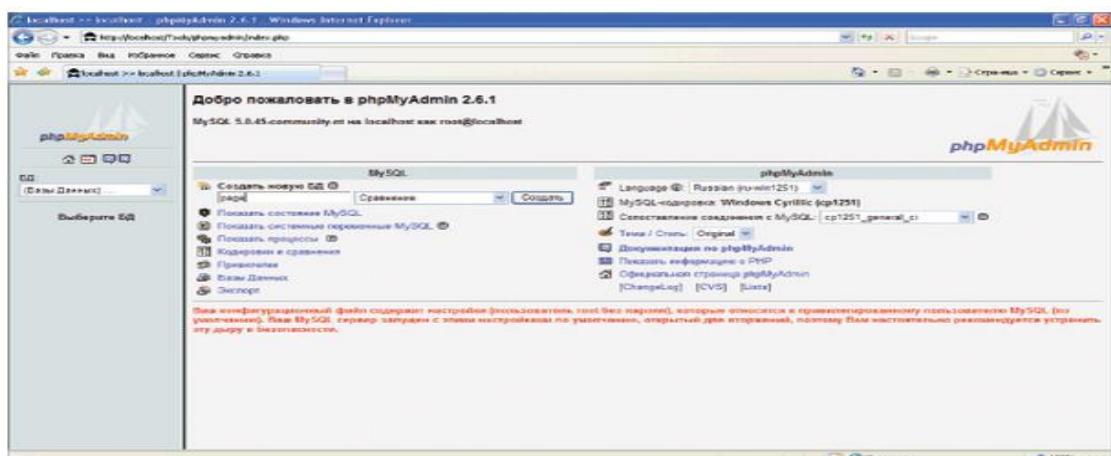
Но... К сожалению, не все так просто. Ведь в журналах очень часто встречаются как минимум двойные оценки вида 2/5. Поэтому придется для оценок завести чуть более сложную таблицу, в которой каждая оценка тоже будет иметь свой уникальный номер.

| Оценка          |             |
|-----------------|-------------|
| Имя поля        | Тип поля    |
| Номер оценки    | Целое число |
| Номер ученика   | Целое число |
| Номер урока     | Целое число |
| Оценка          | Целое число |
| Описание оценки | Строка      |

Поле “Описание оценки” - для обеспечения дополнительной функциональности. Оно позволяет, в частности, в оценках вида 2/5 пояснить, за что именно получен каждый балл: 2 за грамотность, 5 за содержание и т.п.

## 4.2 Практическая часть

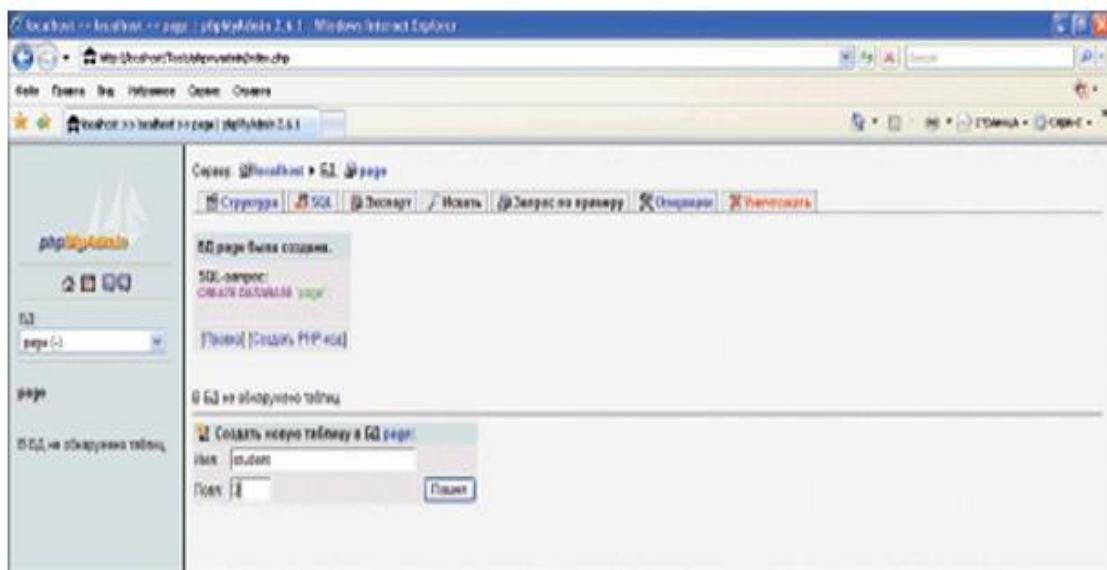
Теперь мы на бумаге имеем структуру нашей базы данных. Пора заняться ее “оцифровкой”. Для этого, запустим Denwer и перейдите на страницу phpMyAdmin. В поле “Создать базу данных” введем имя создаваемой базы — например, journal2 (см. *рис. 1*).



*Рис. 1*

Обратите внимание на то, что phpMyAdmin лишь предоставляет удобный интерфейс для выполнения некоторых операций. Создать базу данных можно было бы и “руками”, посредством команды SQL CREATE DATABASE page (см. *рис. 2*).

На том же *рис. 2* показано следующее действие: необходимо ввести имя создаваемой таблицы (student) и количество полей в ней (3). Разумеется, количество полей в дальнейшем можно менять, здесь оно введено только для удобства.



*Рис. 2*

На следующей вкладке нам потребуется задать имена и типы полей таблицы *student*. Обратите внимание на то, что на *рис. 3* поле *student\_id* помечено как ключевое (радиокнопка под пиктограммой с ключом в правой части рисунка).

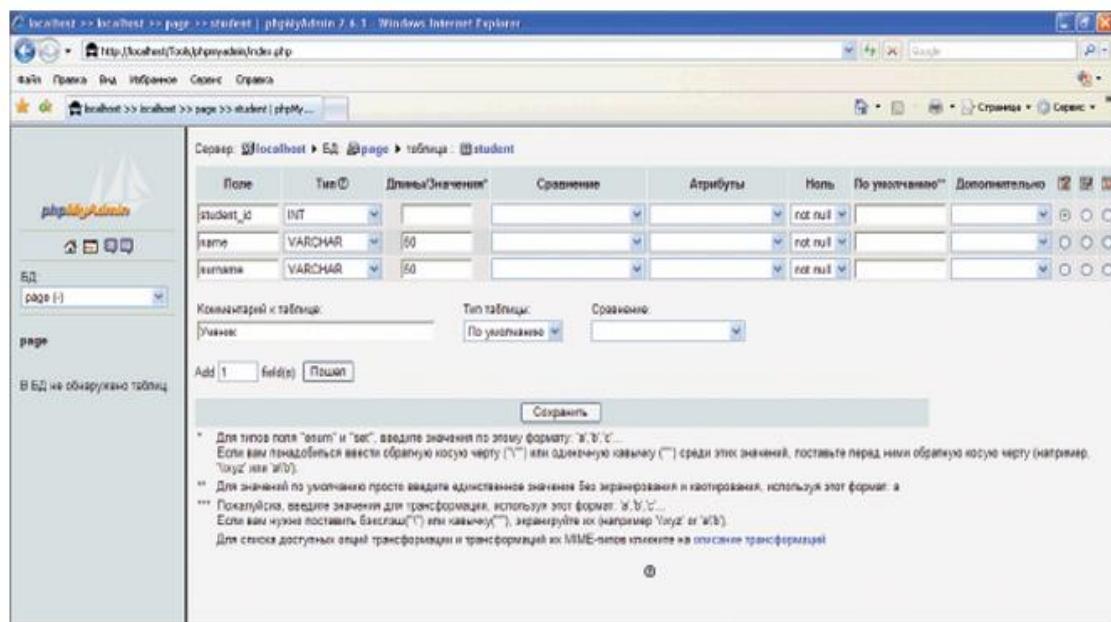


Рис. 3

Результат операции создания таблицы показан на рис. 4.

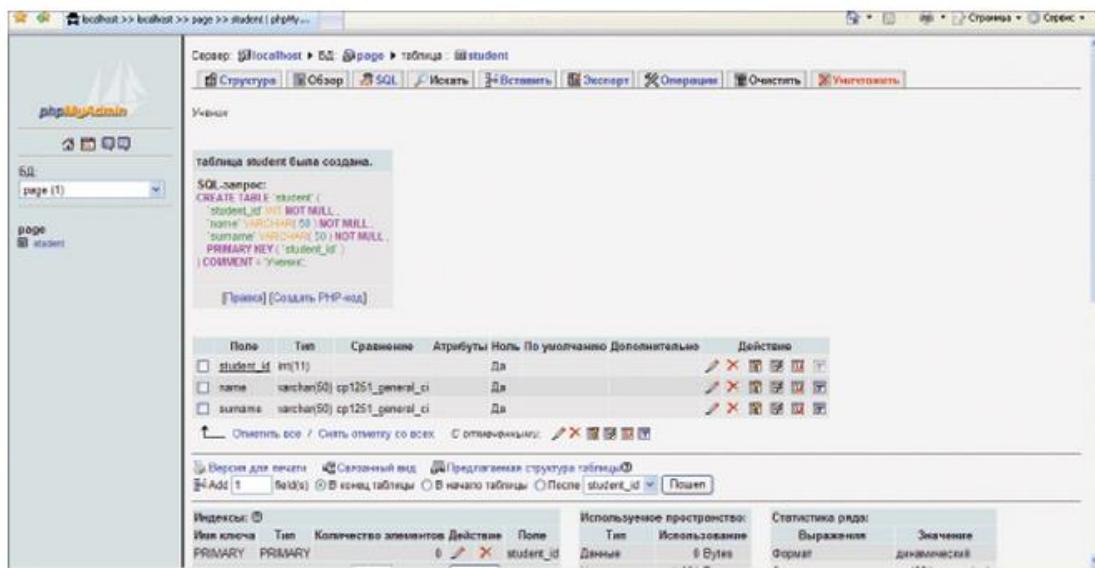


Рис. 4

Чтобы перейти к созданию следующей таблицы, надо кликнуть на имя базы данных (page). Мы попадем на страницу, показанную на рис. 5. Здесь можно ввести имя очередной таблицы.

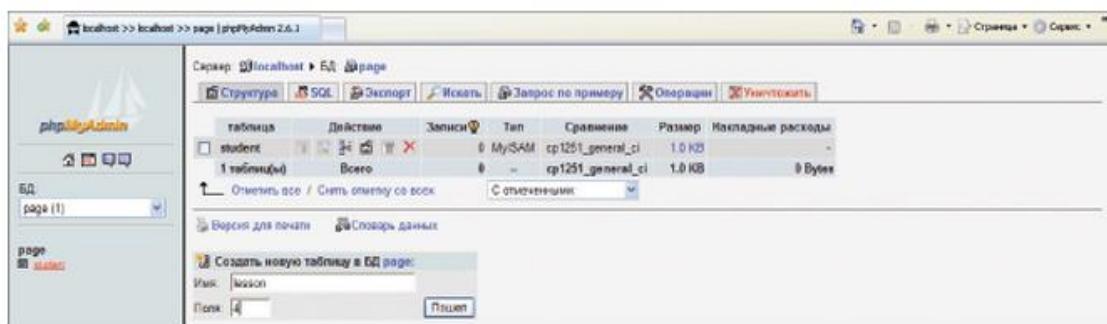


Рис. 5

На следующих рисунках показаны структуры всех оставшихся таблиц нашей базы данных.

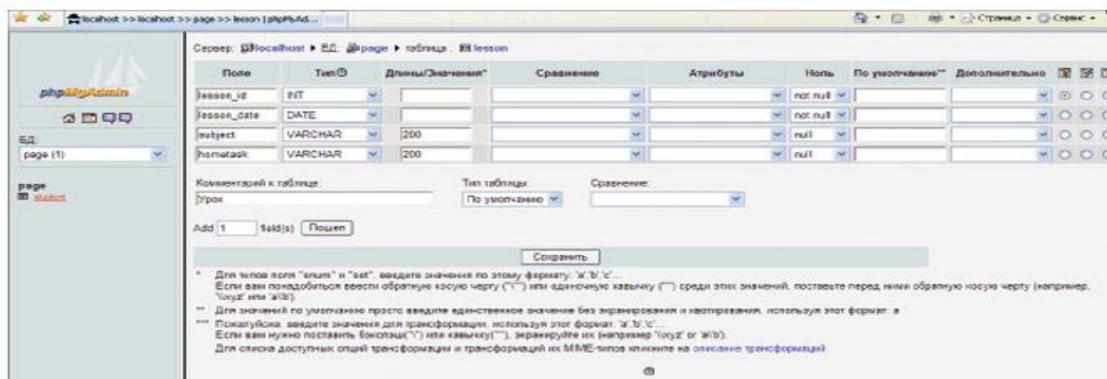


Рис. 6

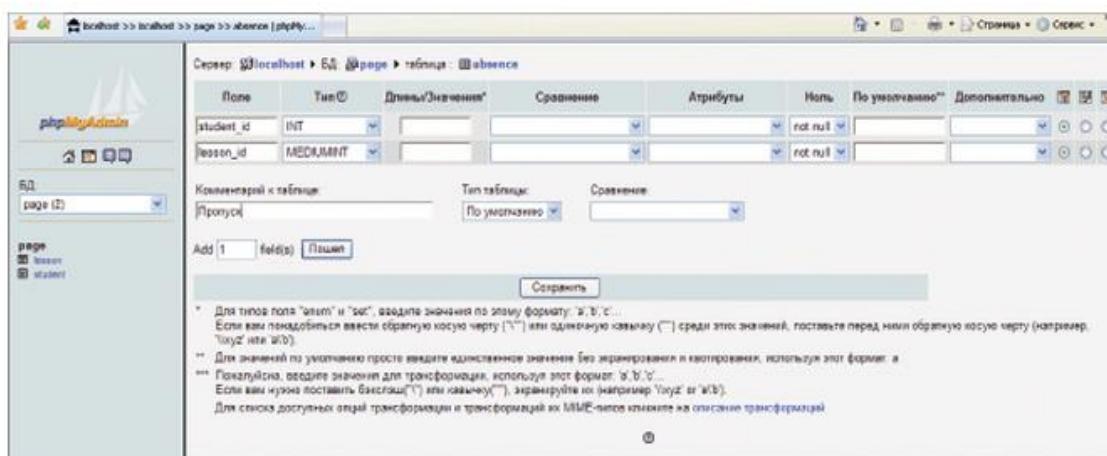
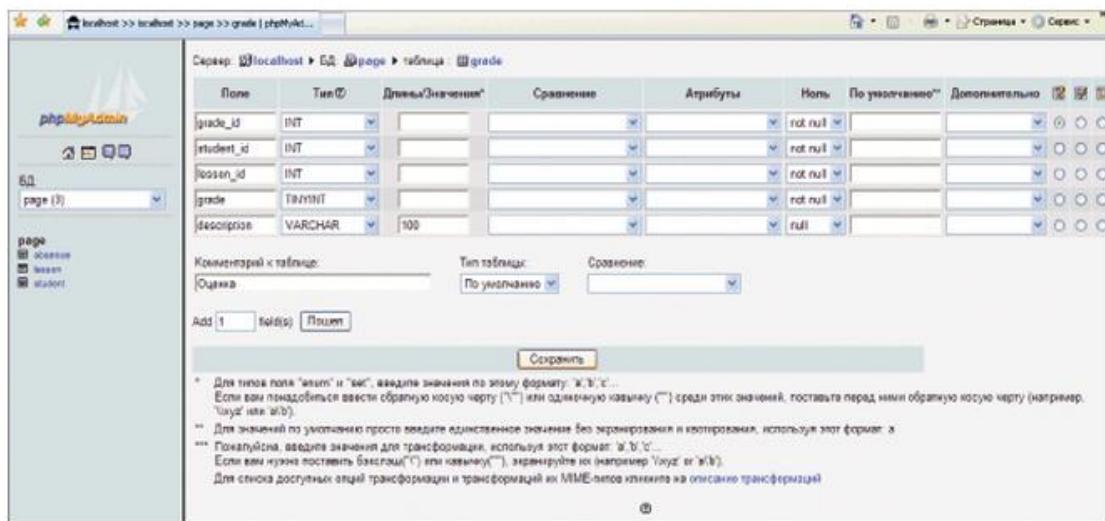
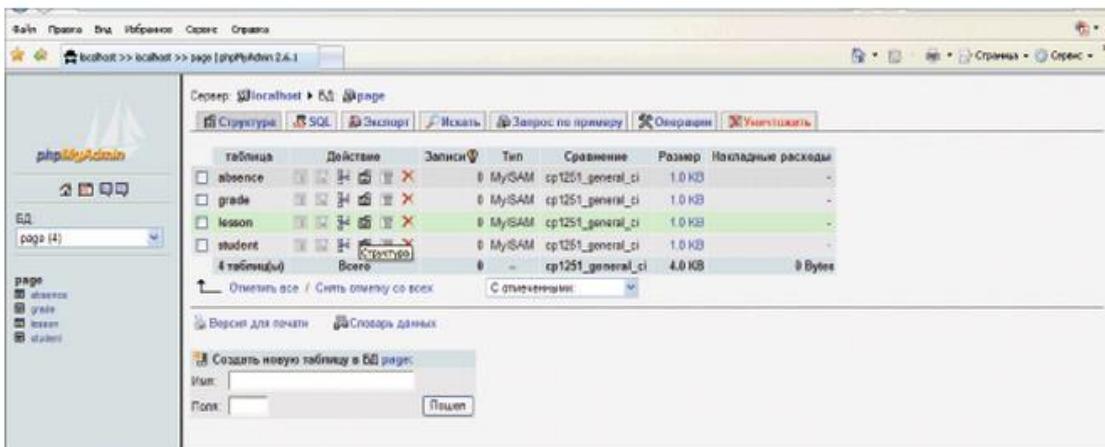


Рис. 7



Puc. 8

В нашей базе данных теперь содержится четыре пустые таблицы (см. *рис9*).



Puc. 9

Теперь нужно их чем-нибудь заполнить. Кликнем на таблицу `student` и в верхнем меню выберите пункт “Вставить”. Откроется страница, показанная на *рис. 10*.

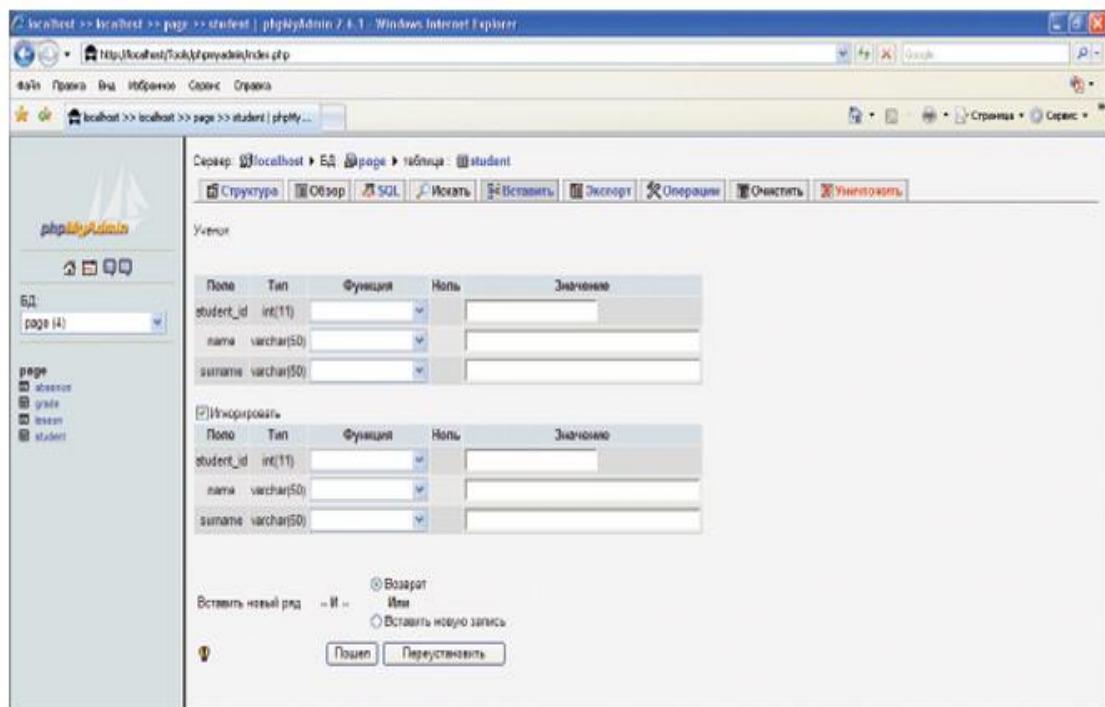


Рис. 10

Нам предстоит вставить в таблицу *student* несколько записей. За один раз (на одном экране) можно вставить две. Введем данные в поля (см. *рис. 11*).

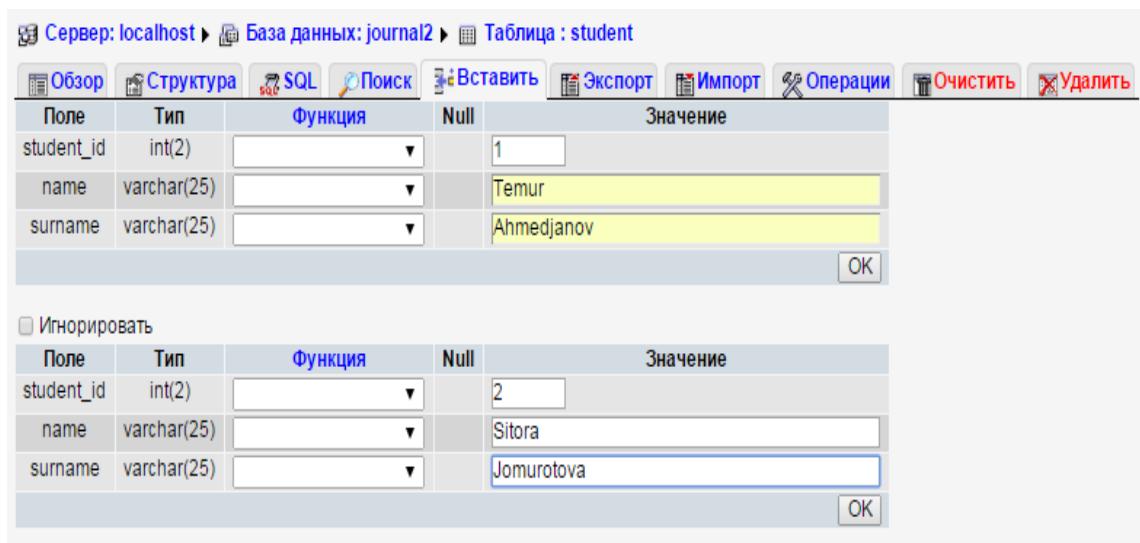


Рис. 11

Введите в эту таблицу остальных учеников и заполните другие таблицы базы данных в соответствии с примером. На *рис. 12* показан пример

заполнения таблицы `absence`. Остальные скриншоты я приводить не буду — они аналогичны.

Рис. 12

На рис. 13 показана визуализация нашей “Страницы классного журнала” на web-странице. Соответствующий скрипт (подробно прокомментированный) приведен ниже.

**Familija, imya**

| # | Familija, imya          | 03/02 | 04/02 | 05/02 | 06/02 | 10/02 | 11/02 | 12/02 | 13/02 | 17/02 | 18/02 | 19/02 | 20/02 |
|---|-------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | Ahmedjanov Temur        |       | N     |       | 2     |       |       |       | 3     |       |       | 3     |       |
| 5 | Hudoykulova Nilufar     | 3     |       |       |       |       | 3     |       |       | 3     |       | 3     |       |
| 2 | Jomirotova Sitora       |       |       |       |       | 3     |       | 2     |       |       |       | 2     |       |
| 3 | Nomozova Shahnaz        |       |       |       |       | 3     |       |       | 3     |       |       | 3     |       |
| 4 | Nurmamatova Anbarabegim |       |       |       |       |       | 2     | 2     |       | 2     |       | 2     |       |
| 6 | Yumakayeva Luisa        |       |       |       |       |       |       | 3     |       |       | 2     |       |       |
| 7 | Yusupov Bobur           |       |       |       |       | 3     |       | 2     |       |       | 3     |       |       |

**Domashnuye zadaniye**

| Data  | Tema uroka                                  | Domashnuye zadaniye   |
|-------|---|---|
| 03/02 | Ivdeniye v web-programmirovaniye.           | Uznat' kakiyu provayderi internet deystvuyut v Uzbekistane. |
| 04/02 | Ivdeniye v HTML.                            | Nayti informatsiyu o tarifah interneta.                     |
| 05/02 | Formi i frejmi v HTML.                      | -   |
| 06/02 | Programmirovaniye so storoni klyienta       | Nayti informatsiyu o serverah                               |
| 10/02 | Perevemniye v JavaScript                    | -   |
| 11/02 | Elementi upravleniya logikoy programmi      | -   |
| 12/02 | Tipi ob'ektor na JavaScript                 | -   |
| 13/02 | Sozdaniye ob'ektor                          | -   |
| 17/02 | Standartnkiye ob'ekti JavaScript            | -   |
| 18/02 | Svoystva okna brauzera                      | Sozdat' sayt na HTML  |
| 19/02 | Osnovniye standartnkiye funktsii JavaScript | -   |
| 20/02 | Bezopasnost' v JavaScript                   | -   |

Рис. 13

### 4.3 Скрипт

Для удобства *рис. 13* получен следующим образом: левый и правый “развороты” страницы выводятся различными таблицами. Одна из них выравнивается влево, вторая — вправо.

```
<html>
<head>
<title>Predmet web-programmirovaniye</title>
<style type="text/css">
<!--
.style1 {
    color: #000000;
    font-weight: bold;
}
body {
    background-color: #99CCFF;
    background-image: url();
}
-->
</style>
</head>
<body>
<h1 align="center" class="style1" style="text-align: center; margin: 0; padding: 0; font-size: 1.5em; font-weight: bold; font-family: sans-serif; border: 1px solid black; border-radius: 10px; width: fit-content; margin-left: auto; margin-right: auto; padding: 10px; text-align: center; background-color: #99CCFF; background-image: url();">Predmet web-
programmirovaniye</h1>
<?php
function DayMonth($dt) {
    //Преобразует строку yyyy-mm-dd -> dd/mm
    //Используется функция substr, которая получает подстроку из строки
    //Формат substr: (строка, начиная с, количество символов)
    //Символы в строке индексируются с нуля
    return substr($dt,8,2)."/".substr($dt,5,2);
}
```

```

}

mysql_connect("localhost", "journal2", "123456");
mysql_select_db("journal2");
$result=mysql_query("select student_id,name,surname from student order by
surname, name");
//Запомним в массиве $students данные всех учеников
while ($arr=mysql_fetch_assoc($result)) $students[]=$arr;
$result=mysql_query("select lesson_id,lesson_date,subject,hometask
from lesson order by lesson_date");
//Запомним в массиве $lessons данные всех уроков
while ($arr=mysql_fetch_assoc($result)) $lessons[]=$arr;
//Выведем левую таблицу
echo "<table border='1' align='left'><tr>";
echo "<th>#</th><th>Familiya, imya</th>";
//Выведем "шапку" таблицы -- даты уроков
foreach ($lessons as $lesson)
echo "<th>".DayMonth($lesson['lesson_date'])."</th>";
echo "</tr>";
//echo
//Выведем столбцы с номерами и фамилиями учеников
foreach ($students as $student) {
echo "<tr><td>".$student['student_id']."'</td>";
echo "<td><i>".$student['surname']."' ".$student['name']."'</i></td>";
foreach ($lessons as $lesson) {
$st_id=$student['student_id'];
$l_id=$lesson['lesson_id'];
$td=""; //Предположим, что очередная ячейка пустая...
$result=mysql_query("select * from absence
where student_id=$st_id and lesson_id=$l_id");
//Если в таблице absence есть запись о пропуске -- поменяем значение $td

```

```

if(mysql_fetch_assoc($result)) $td="N";
else {
//Посмотрим, есть ли у данного ученика оценки за данный урок...
$result=mysql_query("select grade from grade
where student_id=$st_id and lesson_id=$l_id");
$delimit="";
while ($grade=mysql_fetch_assoc($result)) {
//Если оценки есть, накопим их в переменной $td
$td = $td.$delimit.$grade['grade'];
$delimit="/";
}
//И, наконец, выведем, что у нас накопилось в $td
echo "<td align='center'>$td</td>";
}
echo "</tr>";
}
echo "</table>";
//Выведем правую таблицу
echo "<br>";echo "<br>";echo "<br>";echo "<br>";echo "<br>";echo
"<br>";echo "<br>";echo "<br>";echo "<br>";echo "<br>";echo "<br>";echo
"<br>";echo "<br>";echo "<br>";
echo "<table border='1' bgcolor='#EEEEEE' >";
echo "<tr><th>Data</th><th>Tema uroka</th><th>Domashneye
zadaniye</th></tr>";
foreach ($lessons as $lesson) {
echo "<tr><td>".DayMonth($lesson['lesson_date'])."</td>";
echo "<td><i>".$lesson['subject']."'</i></td>";
echo "<td><i>".$lesson['hometask']."'</i></td>";
}

```

```
}

echo "</table>";

?>

</body>

</html>
```

## ЗАКЛЮЧЕНИЕ

Проделав данную самостоятельную работу, были применены теоретические и практические навыки по дисциплине «Web-программирование». Были проанализированы и выбраны средства, а также требования для разработки данного web-приложения. Для разработки были использованы следующие языки программирования и разметки, такие как HTML, PHP и MySql.

Самостоятельная работа посвящена разработке web-приложения учета успеваемости студентов. В ходе работы были решены следующие задачи:

- Проектирование БД с помощью PHP/Admin;
- Разработка клиентского приложения с помощью языка программирования.

В соответствии с предложенным техническим заданием была создана страница «Электронный классный журнал».

Программа учета успеваемости облегчает работу классных руководителем и студентов в различных учебных заведениях. Конечно же, недостатков у разработанного продукта не мало, по причине нехватки глубоких знаний языка и функций PHP, а также недостатка опыта в сфере web-программирования. Но в будущем планируется исправить некоторые ошибки, сделать доработки и усовершенствования. Например сделать введение, фамилии, чисел и оценок непосредственно через само веб-приложение, а не через базу данных, как в нынешнем продукте. Также, добавить пункты оценок за промежуточные, итоговые контрольные работы и итоговую оценку за семестр или четверть. Планируется создать приложение, соединяющее несколько подобных листов, то есть полноценный электронный журнал группы, а также соединить несколько журналов в одну единую систему электронных журналов учебного заведения. С советами и вопросами прошу обратиться по следующему адресу электронной почты [luiza\\_yumakayeva@umail.uz](mailto:luiza_yumakayeva@umail.uz).

## ГЛОССАРИЙ

**Веб-программирование** — раздел программирования, ориентированный на разработку веб-приложений (программ, обеспечивающих функционирование динамических сайтов Всемирной паутины).

**Сайт** (от англ. *website*: *web* — «паутина, сеть» и *site* — «место», буквально «место, сегмент, часть в сети») — система электронных документов (файлов данных и кода) частного лица или организации в компьютерной сети под общим адресом (доменным именем или IP-адресом).

**База данных** - представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

**MySQL** — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

**Денвер** (от сокр. *D.n.w.p* или *DНВР* — джентльменский набор Web разработчика) — набор дистрибутивов и программная оболочка, предназначенные для создания и отладки сайтов (веб-приложений, прочего динамического содержимого интернет-страниц) на локальном ПК (без

необходимости подключения к сети Интернет) под управлением ОС Windows.

**PHP** (англ. *PHP: Hypertext Preprocessor* — «PHP: препроцессор гипертекста»; первоначально *Personal Home Page Tools* — «Инструменты для создания персональных веб-страниц»; произносится *пи-эйч-пи*) — скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

**HTML** (от англ. *HyperText Markup Language* — «язык гипертекстовой разметки») — стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

**Тег (языки разметки)** — элемент языка разметки гипертекста (например, XML, HTML).

**ActionScript** — объектно-ориентированный язык программирования, один из диалектов *ECMAScript*, который добавляет интерактивность, обработку данных и многое другое в содержимое Flash -приложений.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Конституция Республики Узбекистан. Ташкент: Узбекистан, 2009. [ISBN 978-9943-01-458-9](#)
2. 1997.08.29 №464-И [Закон Республики Узбекистан Об образовании](#) // с сайта <http://kkedu.uz/>
3. Верещагин, С.А. Perl& XML. Библиотека программиста. / С.А. Верещаг. - СПб.: Питер, 2012 г. – 208 с.
4. Основы web-программирования для школьного "сайтостроительства" - М.: Диалектика, 2009 г. – 280 с.
5. Высоковский, К.В. Perl. Библиотека программиста. / К.В. Высоковский. -СПб.: Питер, 2010 г. – 249 с.
6. Даль, В.Е. PHP. 75 готовых решений для вашего сайта. / В.Е. Даль. - СПб.: Наука и техника, 2009 г. – 256 с.
7. Каймин, В.А. Информатика. Учебник для студентов. / В.А. Каймин. - М.: ИНФРА-М, 2008 г. – 423 с.
8. Вишневский, А.В., Информатика. Учебное пособие. / А.В. Вишневский. - М.: ХГУ, 2008 г. – 278 с.
9. Пушников А.Ю. Введение в системы управления базами данных. Часть 1, часть 2; -Пб.: Символ-Плюс, 2008 г. – 270 с.
10. Справочник по базе данных MySQL. Best MySQL small cookbook. СПб.: Питер, 2010 г. – 249 с.

Поисковая система Yandex.ru, Google.com;

[www.postroika.ru](http://www.postroika.ru)

[www.lex.uz](http://www.lex.uz)

[www.internet-technologies.ru](http://www.internet-technologies.ru)

[www.htmlweb.ru](http://www.htmlweb.ru)

[www.php5.ru](http://www.php5.ru)

[ru.wikipedia.org](http://ru.wikipedia.org)