

СОЗДАНИЕ МНОГОУРОВНЕВОГО СЕТЕВОГО ВЕБ-ПРИЛОЖЕНИЯ «ТЕСТИРОВАНИЕ» С ПРИМЕНЕНИЕМ ТЕХНОЛОГИЙ PHP И СЕРВЕРА MySQL ДЛЯ ТОНКОГО КЛИЕНТА140

Дистанционное тестирование знаний; автоматизация процесса тестирования. Описан процесс разработки лёгкой для понимания конечного пользователя информационной системы, для автоматизации процесса оценки уровня знаний тестируемых, с использованием сети интернет и веб-технологий.

Алыстыктан билимини текшеруу; билиминин текшеруу автоматташтыруу жараян. Маалыматтык системаны куруунун жонокой тушундурмосу жаны баштагандар учун, интернет менен веб-технологиянын жардамы менен автоматикалык процесстин кадамдарын текшеруу.

Remote testing of knowledge; automation process of testing. Described process of create of an easy-to-understand end-user information system to automate the process of assessing the level of knowledge users with using the Internet and web technologies.

Использование тестирования в качестве средства оценки знаний позволяет проверить гораздо большее количество тестируемых за меньшее время, данный показатель можно улучшить, используя автоматизированную систему создания, прохождения, проверки и поиска тестов, о разработке которой далее и пойдёт речь.

Основная концепция использования веб-приложения следующая, после регистрации и авторизации в системе, пользователь сможет проходить существующие тесты, а также создавать свои по определённым областям знаний. Для того чтобы результаты прохождения теста были занесены в базу данных необходимо добавить группу тестируемых лиц: университет и факультет, либо школу и класс, либо предприятие и отдел. Пользователь, создавший тест, имеет возможность получить отчёт о прохождении теста тестируемыми из созданного им списка. Результаты теста будут записаны в базу данных после того, как пользователь будет инициализирован системой как участник тестирования, введя своё Ф.И.О. или любое другого значение/ключ, известный только группе тестируемых, либо пройдя регистрацию и выбрав необходимую группу пользователей веб-приложения.

При регистрации пользователь выбирает необходимую группу, у каждой группы свои возможности работы с системой, они разделены на следующие шесть типов:

1. Пользователь – тип регистрации, при котором имеется возможность создавать пользовательские тесты, писать статьи, оставлять комментарии и редактировать созданный собой контент соответственно. Проходить все тесты без занесения результата в базу данных.

2. Учитель – тип регистрации, позволяющий создавать как пользовательские тесты, так и тесты для школ и университетов, добавлять списки тестируемых учеников, получать отчёт о результатах прохождения ими тестов, а также писать статьи.

3. Студент – выбрав данный тип регистрации пользователь заполняет анкету где указывает страну, город, ВУЗ, факультет и группу где проходит обучение, получая возможность проходить тесты созданные учителями того учебного заведения, где учится сам студент, с занесением результатов в базу. Также студенты могут проходить любые тесты без занесения результатов в базу, имеют возможность создавать пользовательские тесты и писать статьи.

4. Школьник – тип регистрации, включающий в себя предварительную необходимость внесения данных о стране и городе проживания, месте учёбы, указания названия (номера) школы. Данный тип регистрации позволяет проходить тесты созданные учителями школы, в которой учится пользователь с занесением результата в базу. Так же

школьник может проходить все тесты без записи результата в базу данных, и создавать пользовательские тесты и писать статьи.

5. Начальник – выбрав этот тип регистрации, пользователь должен указать страну и город где расположена организация, вписать название организации, создать список сотрудников. Также пользователь с типом регистрации начальника может создавать как тесты для сотрудников организации – организационные тесты, так и простые пользовательские тесты, и получать отчёт о прохождении организационных тестов.

6. Сотрудник – регистрация под данным типом позволяет пользователю проходить организационные тесты с добавлением результата прохождения в базу данных. Пользователь вводит страну, город и организацию в которой трудится, для идентификации его руководителем создавшим тест. Также сотрудники имеют возможность создавать пользовательские тесты, писать статьи и проходить любые другие тесты без занесения результатов в базу.

Не зарегистрированные пользователи, могут проходить любые тесты без занесения результатов в базу соответственно, также у них есть возможность оставить комментарий на статью после ввода цифр со специального изображения. В качестве универсальной бальной системы оценки используется подсчёт правильных ответов и вывод их количества в процентах от общего числа вопросов. Процентный результат, после ответа пользователя на последний вопрос теста, легко преобразуется в привычную пяти бальную или десяти бальную систему.

Вопросов в тесте может быть до 500, а вариантов ответа на каждый вопрос до 30, тестирующий может произвольно задавать время на выполнения теста в минутах. Есть дополнительная возможность создания вопроса без вариантов ответа – участнику тестирования потребуется ввести правильный ответ в текстовое поле, впоследствии этот ответ будет сопоставлен с верным ответом из базы на стороне сервера. После ответа на все вопросы, будет отображён результат тестирования с занесением данных в базу в случае наличия пользователя в списке тестируемых. В качестве вопроса можно использовать как текстовый вид информации, так и графический, указав ссылку на изображение, которое необходимо опознать и интерпретировать участнику тестирования. Каждый раз вопросы теста выдаются в случайном порядке, что препятствует недобросовестному получению желаемого результата.

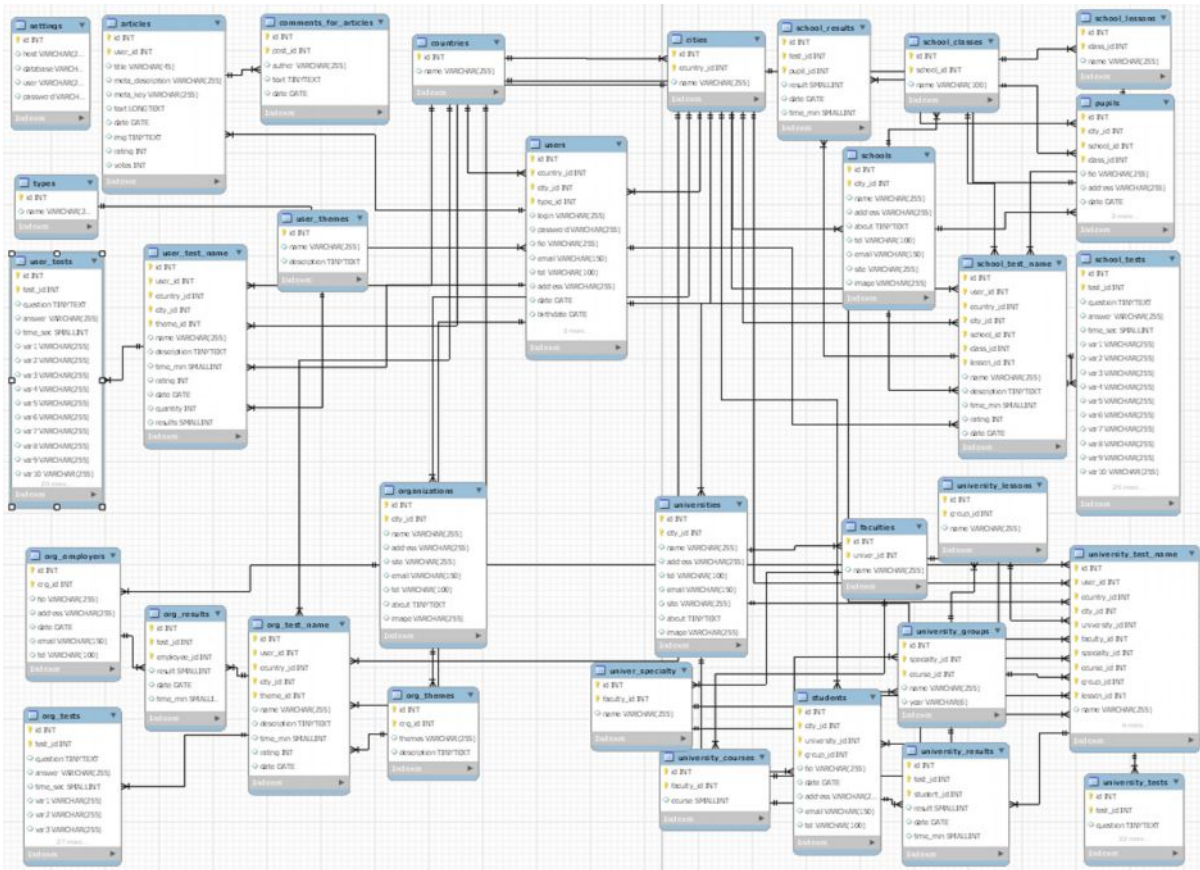
Причина выбора веб-технологий для реализации данного проекта – это повсеместное распространение интернета, глобальной сети, доступ к которой клиент получает через специализированное приложение веб-браузер. Веб-браузер является тонким клиентом посылающим запросы к веб-серверам и интерпретирующий полученный ответ в удобный для восприятия пользователя вид. Потенциально у любого человека имеющего доступ в сеть интернет, есть возможность в полной мере воспользоваться данным веб-приложением для дистанционного тестирования знаний.

Информационная система состоит из нескольких уровней-звеньев, на каждом уровне используется своя технология:

- на первом уровне представления данных пользователю, используется язык разметки гипертекста HTML с каскадной таблицей стилей CSS, и интерпретируемый язык с динамической типизацией JavaScript с библиотекой JQuery для более удобной кроссбраузерной реализации визуальных эффектов и AJAX для асинхронных запросов к серверу.

- на следующем уровне серверной логики приложения, используется язык PHP версии 5.4, в которой улучшена скорость интерпретации кода и внесены некоторые новые возможности для объектно-ориентированного программирования, например, такие как пространства имён.

- заключительное звено информационной системы это сервер баз данных MySQL, в котором хранится спроектированная и приведённая к третьей нормальной форме база данных со всей информацией о тестах и результатах их прохождения.

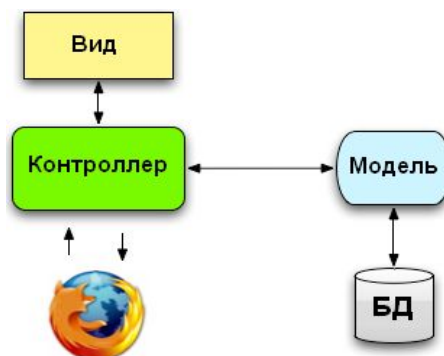


Модель сущность-связь, общая схема ER-модели базы данных (Рис. 1)

Первый этап в разработки информационной системы это проектирование базы данных, которая будет в полной мере отражать предметную область. Для этого использовался специальный программный комплекс MySQL Workbench версии 5.2 от компании Oracle, интегрирующий в себе инструменты визуального проектирования баз данных, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL. Спроектированная и приведённая к третьей нормальной форме, база данных содержит в себе 33 таблицы (см. Рис. 1) связанные между собой внешними ключами, что обеспечивает целостность данных и их атомарность.

Результат проектирования возвращается в виде программного кода на языке SQL, который будет исполнен на сервере баз данных MySQL.

Следующий этап разработки информационной системы создание архитектуры веб-приложения, в качестве её основы избрана широко используемая концепция MVC (от англ. – Model-модель, View-вид, Controller-контроллер), она представляет собой схему использования нескольких шаблонов проектирования, с помощью которых модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента (Рис. 2):



Общая схема использования MVC концепции (Рис. 2)

1. **Модель** - модель предоставляет данные из БД и методы работы с этими данными. Не содержит информации о визуализации.

2. **Представление, вид** - отвечает за отображение информации (визуализацию). В качестве представления выступает HTML шаблон, содержащий минимальное количество кода на PHP в котором содержится результат обработки пришедших от контроллера данных.

3. **Контроллер** - обеспечивает связь между пользователем и системой, контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

Модификация одного из компонентов оказывает минимальное воздействие на остальные, либо вообще не затрагивает другие пакеты. Для корректного функционирования системы необходимо задать следующие параметры в корневом файле .htaccess :

```
RewriteEngine on RewriteCond %{REQUEST_FILENAME} !-f RewriteCond
%{REQUEST_FILENAME} !-d RewriteRule !.(js|gif|png|jpg|jpeg|css|bmp)$ index.php
```

В данном случае все запросы пользователя с адресной строки браузера будут направлены в один файл index.php, который по типу запроса подключает соответствующий контроллер, данный файл содержит кода на языке PHP:

```
session_start();
try{    set_include_path(get_include_path()
.PATH_SEPARATOR.'server/controllers'
.PATH_SEPARATOR.'server/models'
.PATH_SEPARATOR.'server/views');
function __autoload($name)
{require_once $name.'.php';}
$front=FrontController::get();
$front->route();
echo $front->getBody(); }
catch(Exception $e) { echo 'Ошибка в файле: '$e->getFile().' На строке '$e-
->getLine().'<br>'.$e->getMessage(); }
```

Созданный каркас приложения можно модифицировать, создавая новые классы контроллеров и моделей, наследовать существующие классы, изменяя их внутреннюю реализацию, создавая новые шаблоны для представления данных пользователю либо использовать существующие. Итог работы – многофункциональное, расширяемое веб-приложение «Тестирование» позволяющее проходить и создавать тесты из окна веб-браузера, находясь в любой точке нашей планеты, где есть доступ к сети интернет.

Литература:

1. Мэтт Зандстра, «PHP: объекты, шаблоны и методики программирования», И.Д.Вильямс, 2011 ., 560с.
2. Е. С. Бенкен, Г.А. Самков, «AJAX: программирование для Интернета», СПб.: БХВ-Петербург, 2009. — 464 с.90.
3. Chris Pitt, «Pro PHP MVC», Apress, 2012., – 500 с.
4. И. М. Ибрагимов, «Информационные технологии и средства дистанционного обучения», Академия 2007., - 336 с.
5. Максим Кузнецов, Игорь Симдянов, «MySQL 5. В подлиннике», БХВ-Петербург, 2010., - 1024 с.

6. К. Дари, Б. Бринзаре, Ф. Черчез-Тоза, М. Бусика, «AJAX и PHP. Разработка динамических веб-приложений», Символ-Плюс 2009., - 336 с.
7. Дэвид Флэнаган, «JavaScript. Подробное руководство», Символ-Плюс 2008., - 992 с.
8. Максим Кузнецов, Игорь Симдянов, «Объектно-ориентированное программирование на PHP», БХВ-Петербург, 2007., - 608 с.
9. Фримен Эрик, Фримен Элизабет, «HTML, XHTML и CSS», Питер, 2012., - 656 с.
10. Джошуа Кериевски, «Рефакторинг с использованием шаблонов», И.Д.Вильямс, 2008., - 400 с.
11. Роберт К. Мартин, «Чистый код. Создание, анализ и рефакторинг», Питер, 2013., - 464 с.