Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Глазовский государственный педагогический институт имени

В.Г. Короленко»

**ОТЧЕТ**

**По учебной практике**

**ПМ 01 Разработка программных модулей программного обеспечения для компьютерных систем**

**Специальность 09.02.03 Программирование в компьютерных системах**

Выполнил:

Студент ОП СПО группа 21

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики: ст. преподаватель кафедры

Математики и информатики Касаткин К.А

г. Глазов 2019

**Оглавление**

[Введение 3](#_Toc9677821)

[Аналоги приложения 4](#_Toc9677822)

[Требования к программе 6](#_Toc9677823)

[Описание алгоритма работы программы 7](#_Toc9677824)

[Исходный код программы 9](#_Toc9677825)

[**Интерфейс и код программы** 9](#_Toc9677826)

[Использованные компоненты 17](#_Toc9677827)

[XAMPP 17](#_Toc9677828)

[MySQL 18](#_Toc9677829)

[PhpMyAdmin 19](#_Toc9677830)

[Структура базы данных 20](#_Toc9677831)

[**Тестирование программы** 21](#_Toc9677832)

[**Заключение** 22](#_Toc9677833)

[**Список литературы** 23](#_Toc9677834)

# Введение

Актуальность: в современном мире тестирование является простейшим способом проверки человека на знания. А цифровая версия тестов гораздо сильно упрощает процесс тестирования.

Задача: изучить язык программирования PHP с применением MySQL.

Цель: написать программу для тестирования пользователя на языке PHPс использованием баз данных MySQL.

Данная программа может быть использована практически во всех видах обучения или контроля знаний у человека.

# Аналоги приложения

Google Forms – онлайн приложение для создания форм с тестами. Позволяет быстро и просто создавать опросы, тесты. Имеет значительный набор функций, которые позволяют проводить опрос в практически inавтоматическом режиме.

Рисунок 1 – Google Forms

Айрен — это бесплатная программа, позволяющая создавать тесты для проверки знаний и проводить тестирование в локальной сети, через интернет или на одиночных компьютерах.

Тесты могут включать в себя задания различных типов: с выбором одного или нескольких верных ответов, с вводом ответа с клавиатуры, на установление соответствия, на упорядочение и на классификацию.

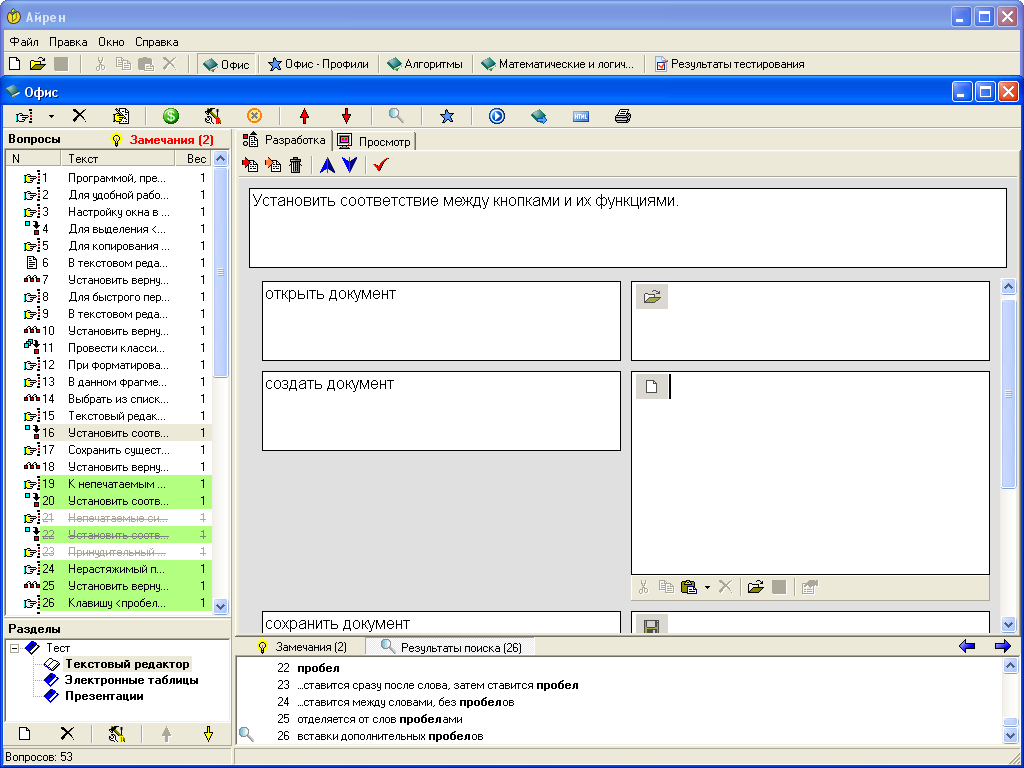


Рисунок 2 – Интерфейс приложения Айрен

Система тестирования INDIGO – это профессиональный инструмент автоматизации процесса тестирования и обработки результатов, который предназначен для решения широкого спектра задач:

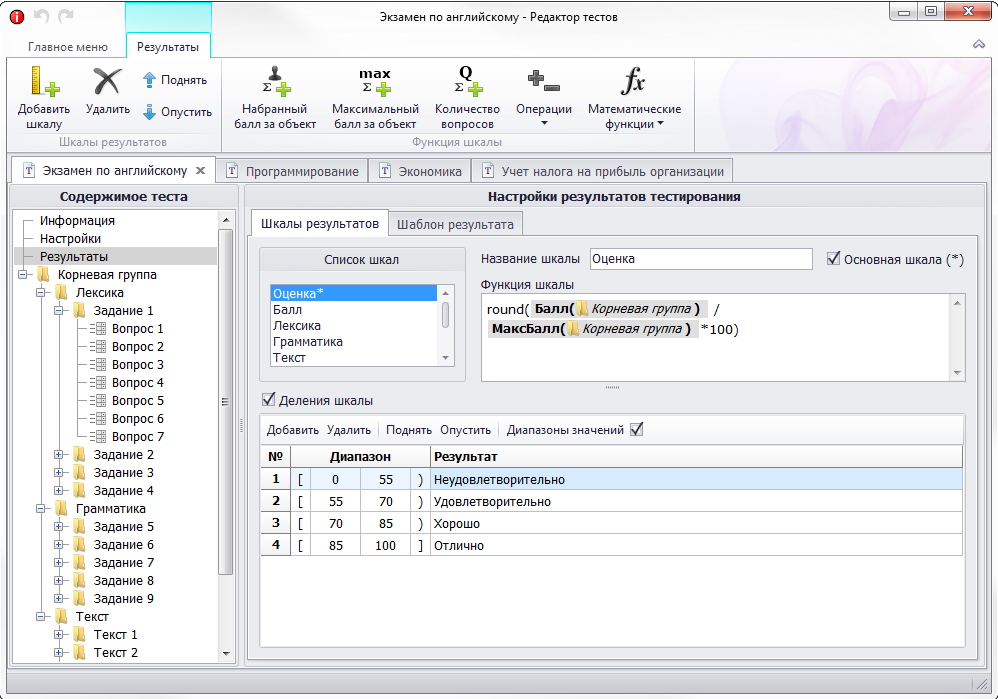


Рисунок 3 – Интерфейс программы INDIGO

# Требования к программе

Программа должна иметь панель авторизации, регистрации, тестирования, вывода сертификата о пройденном тесте. Вся информация должна храниться в базе данных MySQL.

# C:\Users\WEXLE\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Untitled Diagram.pngОписание алгоритма работы программыC:\Users\WEXLE\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Untitled Diagram (2).png

Рисунок 5 – Блок схема процесса регистрации

Рисунок 4 – Блок схема работы программы

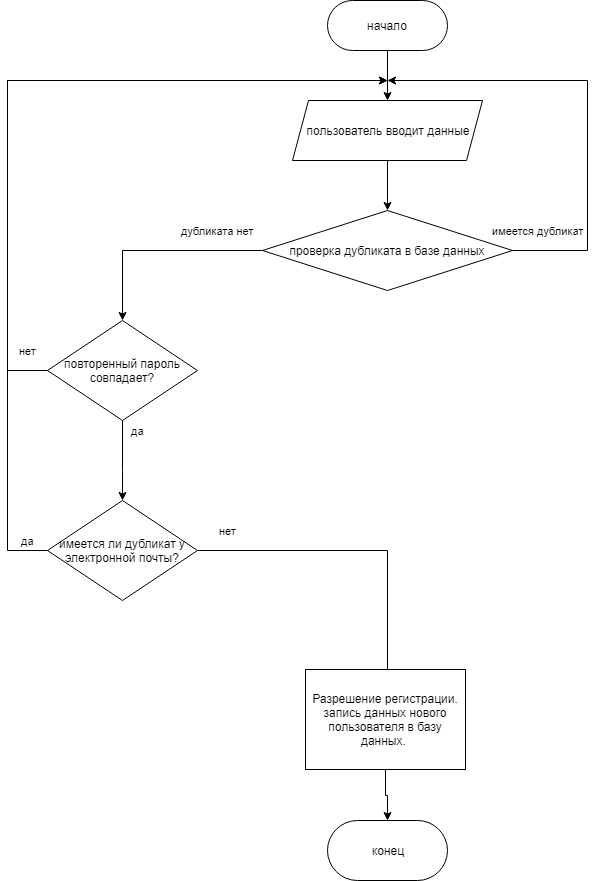


Рисунок 6 – Процесс регистрации

# Исходный код программы

Для разработки был выбран язык программирования php так как она используется вместе с http. Это позволяет иметь доступ к тестированию в любом месте, где есть доступ к сети Интернет. Это избавляет от многочисленных установок программы, если требуется протестировать большую группу людей.

**Интерфейс и код программы**

Главная страница – здесь пользователю предлагается выбор: регистрация или авторизация.

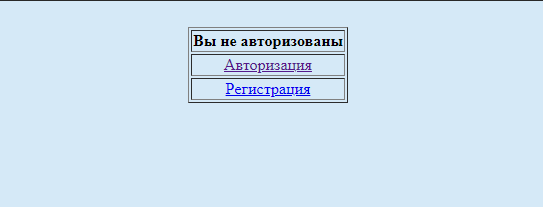


Рисунок 7 – Главное окно

<?php

require 'db.php';

?>

<body bgcolor="d5e9f7">

<?php if ( isset ($\_SESSION['logged\_user']) ) : ?>

<center>

<br/><table border="1">

<tr><td><center><strong>Авторизован!</strong></center></td></tr>

<tr><td><center>Привет, <?php echo $\_SESSION['logged\_user']->login; ?>!<br/></center></td></tr>

<tr><td><center><a href="logout.php">Выйти</a></center></td></tr>

</center>

<?php else : ?>

<center>

<table border="1">

<tr><td><center><strong>Вы не авторизованы</strong></center></td></tr>

<tr><td><center><a href="login.php">Авторизация</a></center></td></tr>

<tr><td><center><a href="signup.php">Регистрация</a></center></td></tr>

Окно регистрации –регистрация новых пользователей. Данные вносятся в базу данных.

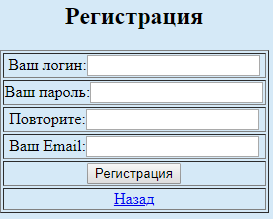


Рисунок 8 – окно регистрации окно

﻿<?php

require 'db.php';

$data = $\_POST;

//если кликнули на button

if ( isset($data['do\_signup']) )

{

// проверка формы на пустоту полей

$errors = array();

if ( trim($data['login']) == '' )

{

$errors[] = 'Введите логин';

}

if ( trim($data['email']) == '' )

{

$errors[] = 'Введите Email';

}

if ( $data['password'] == '' )

{

$errors[] = 'Введите пароль';

}

if ( $data['password\_2'] != $data['password'] )

{

$errors[] = 'Повторный пароль введен не верно!';

}

//проверка на существование одинакового логина

if( R::count('users', "login = ?", array($data['login'])) > 0)

{

$errors[] = 'Пользователь с таким логином уже существует!';

}

//проверка на существование одинакового email

if( R::count('users', "email = ?", array($data['email'])) > 0)

{

$errors[] = 'Пользователь с таким Email уже существует!';

}

if( empty($errors) )

{

//ошибок нет, теперь регистрируем

$user = R::dispense('users');

$user->login = $data['login'];

$user->email = $data['email'];

$user->password = password\_hash($data['password'], PASSWORD\_DEFAULT); //пароль нельзя хранить в открытом виде, мы его шифруем при помощи функции password\_hash дляphp> 5.6

R::store($user);

echo '<div style="color:dreen;">Вы успешно зарегистрированы!</div><hr>';

}else

{

echo '<div id="errors" style="color:red;">' .array\_shift($errors). '</div><hr>';

}

}

?>

<body bgcolor="d5e9f7">

<form action="signup.php" method="POST">

<center>

<table border="1">

<h2>Регистрация</h2>

<tr><td><center>Ваш логин:<input type="text" name="login" value="<?php echo @$data['login']; ?>"><br/></center></td></tr>

<tr><td><center>Ваш пароль:<input type="password" name="password" value="<?php echo @$data['password']; ?>"><br/></center></td></tr>

<tr><td><center>Повторите:<input type="password" name="password\_2" value="<?php echo @$data['password\_2']; ?>"><br/></center></td></tr>

<tr><td><center>Ваш Email:<input type="email" name="email" value="<?php echo @$data['email']; ?>"><br/></center></td></tr>

<td><center><button type="submit" name="do\_signup">Регистрация</button></center></td></tr>

<tr><td><center><a href="index.php">Назад</a></button></center></td></tr>

</center>

</form>

Окно авторизации – пользователю необходимо ввести свои данные для входа. Данные считываются с базы данных.

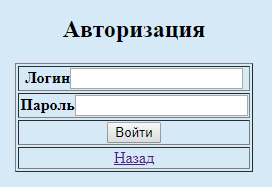


Рисунок 9 – окно авторизации

﻿<?php

require 'db.php';

$data = $\_POST;

if ( isset($data['do\_login']) )

{

$user = R::findOne('users', 'login = ?', array($data['login']));

if ( $user )

{

//логин существует

if ( password\_verify($data['password'], $user->password) )

{

//если пароль совпадает, то нужно авторизовать пользователя

$\_SESSION['logged\_user'] = $user;

echo '<divstyle="color:green;">Вы авторизованы!<br/> Можете перейти на <a href="home.php">главную</a> страницу.</div><hr>';

}else

{

$errors[] = 'Неверно введен пароль!';

}

}else

{

$errors[] = 'Пользователь с таким логином не найден!';

}

if( !empty($errors) )

{

//выводим ошибки авторизации

echo '<div id="errors" style="color:red;">' .array\_shift($errors). '</div><hr>';

}

}

?>

<form action="login.php" method="POST">

<body bgcolor="d5e9f7">

<center>

<h2>Авторизация</h2>

<table border="1">

<tr><td><center><strong>Логин</strong><input type="text" name="login" value="<?php echo @$data['login']; ?>"><br/></center></td></tr>

<tr><td><center><strong>Пароль</strong><input type="password" name="password" value="<?php echo @$data['password']; ?>"><br/></center></td></tr>

<tr><td><center><button type="submit" name="do\_login">Войти</button></center></td></tr>

<tr><td><center><a href="index.php">Назад</a></button></center></td></tr>

</form>

Окно личного кабинета – здесь пользователь может начать тестирование или вывести сертификат.

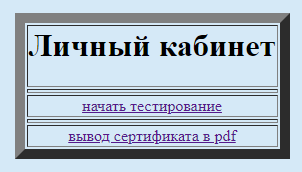


Рисунок 10 –Окно личного кабинета окно

<meta charset="utf-8">

<body bgcolor="d5e9f7">

<center>

<br/><table border="10">

<tr><td><center><strong><h1><div align=left>Личный кабинет</strong></center></td></tr>

<tr><td><center><strong><tr><td><center><a href="test.php">начать тестирование<center></td></tr></strong></center></td></tr>

<tr><td><center><strong><tr><td><center><a href="pdf.php">вывод в pdf</center></td></tr></strong></center></td></tr>

Окно теста – здесь пользователь проходит тест.

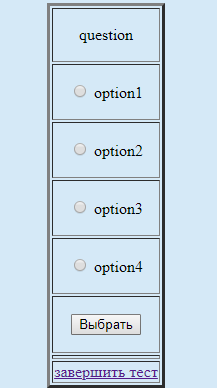


Рисунок 11 – окно тестирования  окно

<!DOCTYPE html>

<center>

<table border='3'>

<?php

$link = mysqli\_connect('localhost', 'e95082j6\_root', 'seshkek228', 'e95082j6\_root');

$quC = mysqli\_query($link, "SELECT q FROM q");

$quR = mysqli\_fetch\_array($quC);

$an = mysqli\_query($link, "SELECT \* FROM a");

$an1R = mysqli\_fetch\_array($an);

$an2R = mysqli\_fetch\_array($an);

$an3R = mysqli\_fetch\_array($an);

$an4R = mysqli\_fetch\_array($an);

$rightC = mysqli\_query($link, "SELECT r FROM a");

$anR = mysqli\_fetch\_array($rightC);

$qr = $an1R['r'];

if( isset( $\_POST['check'] ) )

{if (!$\_POST['ans']){

echo ("<table border='3'><tr><td><center><p>нет ответа</p></center></td></tr>");}

else if ($\_POST['ans']==$qr){

echo ("<tr><td><center><p>Правильный ответ</p></center></td></tr>");

}

else {

echo ("<tr><td><center><p>Неправильный ответ!</p></center></td></tr>");

}

}

?>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">

</head>

<body>

<body bgcolor="d5e9f7">

<form method="POST">

<tr><td><center><p><?php echo $quR['q'] ?></p>

<tr><td><center><p><input name="ans" type="radio" value="1"><?php echo $an1R['a1'] ?></p></center></td></tr>

<tr><td><center><p><input name="ans" type="radio" value="2"><?php echo $an1R['a2'] ?></p></center></td></tr>

<tr><td><center><p><input name="ans" type="radio" value="3"><?php echo $an1R['a3'] ?></p></center></td></tr>

<tr><td><center><p><input name="ans" type="radio" value="4"><?php echo $an1R['a4'] ?></p></center></td></tr>

<tr><td><center><p><input type="submit" name="check" value="Выбрать"></p></center></td></tr>

<tr><td><center><strong><tr><td><center><a href="home.php">завершить тест</a></center></td></tr></strong></center></td></tr>

</form>

</body>

</html>

</center>

Окно сертификата.

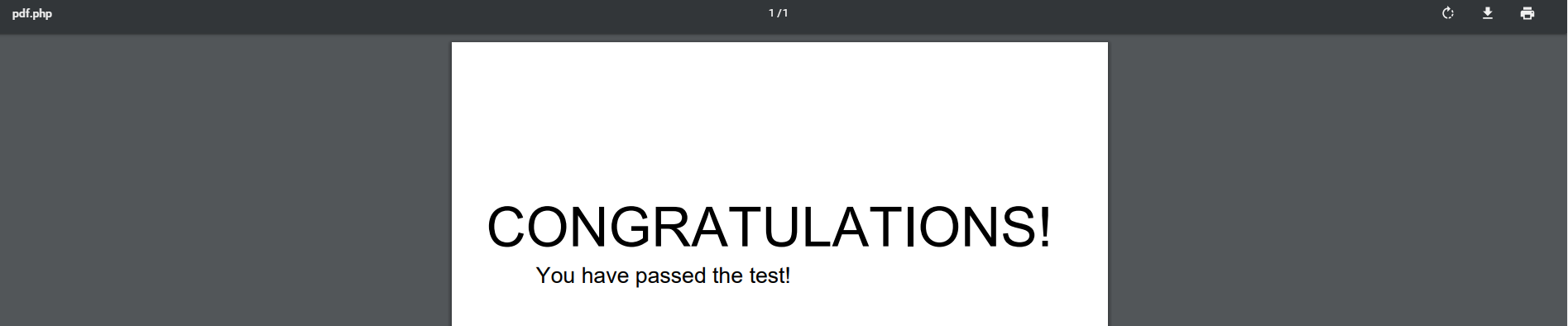


Рисунок 12 – окно сертификата окно

<?php

require( 'fpdf/fpdf.php' );

$pdf = new FPDF();

$pdf->SetFont('Arial','',50);

$pdf->AddPage();

$pdf->Cell(16,100,"CONGRATULATIONS!");

$pdf->SetFont('Arial','',20);

$pdf->Cell(10,130,"You have passed the test", 10);

$pdf->Output();

?>

# Использованные компоненты

## XAMPP

XAMPP полностью бесплатный и простой в установке дистрибутив Apache, содержащий MySQL, PHP и Perl. XAMPP создан с открытым исходным кодом, чтобы быть невероятно простым в установке и в использовании. XAMPP является самой популярной средой разработки PHP XAMPP — это акроним:

X (любая из четырех операционных систем)

Apache

MySQL

PHP

Perl.

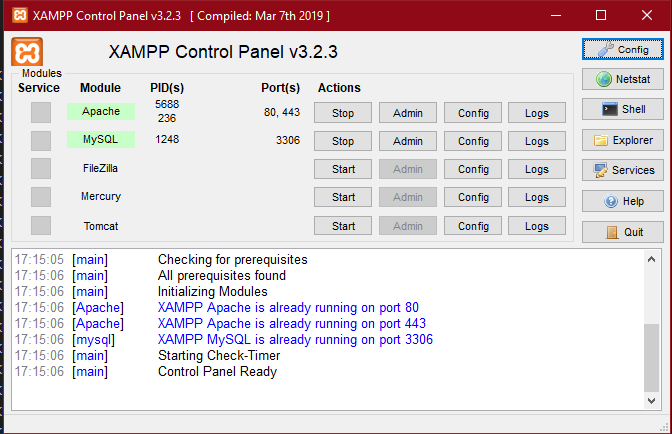


Рисунок 10 – контрольная панель XAMPP окно

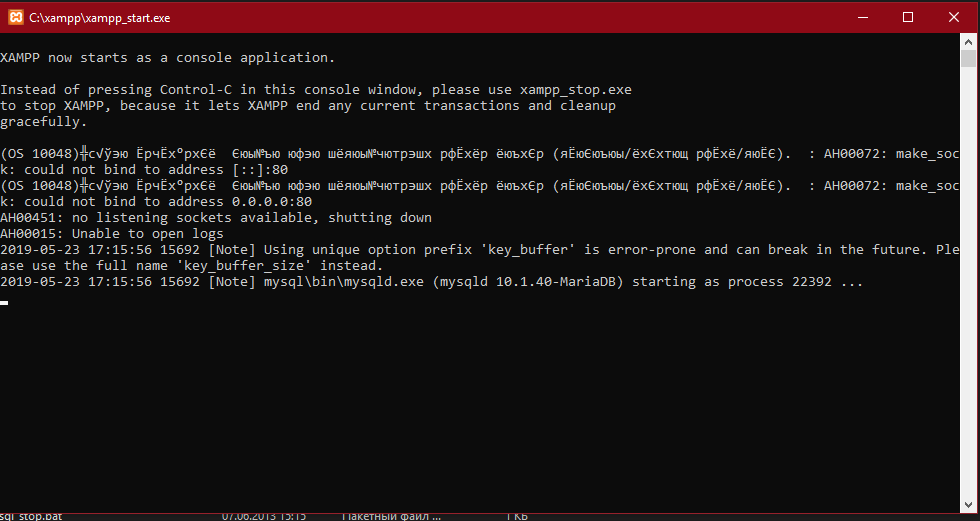
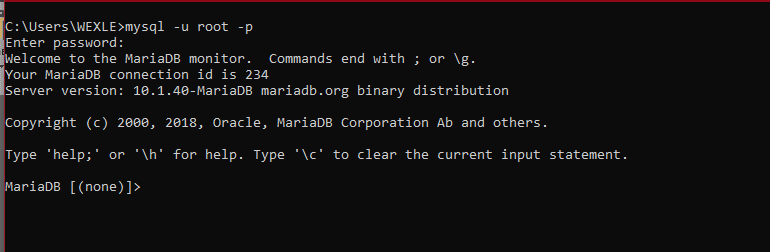


Рисунок 11 – окно консоли XAMPPс сообщением об успешном запуске виртуального сервера

## MySQL

Рисунок 12 – консоль программы MySQL



MySQL— свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой SunMicrosystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU GeneralPublicLicense, 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.

## PhpMyAdmin

PhpMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PHPMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд, предоставляя дружественный интерфейс.

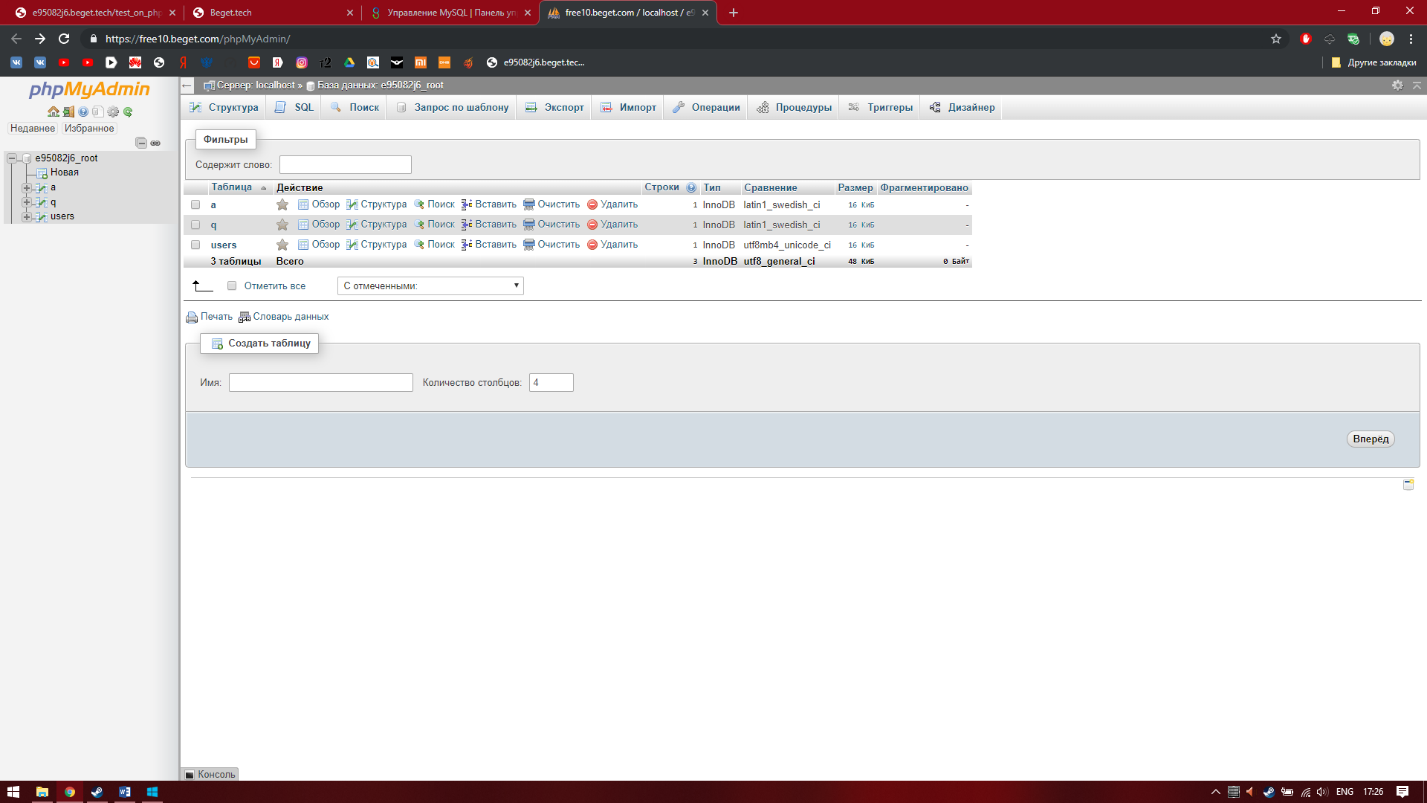
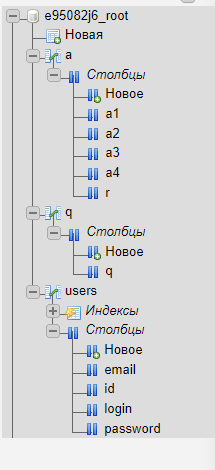
На сегодняшний день PHPMyAdmin широко применяется на практике. Последнее связано с тем, что разработчики интенсивно развивают свой продукт, учитывая все нововведения СУБД MySQL. Подавляющее большинство российских провайдеров используют это приложение в качестве панели управления для того, чтобы предоставить своим клиентам возможность администрирования выделенных им баз данных.

Рисунок 12 – интерфейс phpMyAdmin

## Структура базы данных



В базе данных имеются три таблицы: “a, q, users”.

В таблице “a” (сокращение answers) содержатся варианты ответов (а1 – а4) и правильный вариант ответа (r).



В таблице “q” (сокращение от questions) содержатся вопросы.



В таблицеusersсодержится информация о пользователях. Туда входят электронная почта, уникальный номер пользователя, логин и пароль в зашифрованном виде.

**Тестирование программы**

Программа прошла тестирование путем проверки введенной информации.

**Руководство по установке**

1. Скачать приложения XAMPP для создания локального сервера.
2. Установить XAMPP.
3. Перейти в директорию XAMPP/htdocs/.
4. Скопировать содержимое архива test\_on\_php в эту директорию.
5. Запустить командную строку.
6. В MySQL загрузить бэкап (sump.sql) базы данных командой mysql –u root –p < путь/до/dump.sql,
7. Запустить контрольную панель XAMPP.
8. Нажать START.
9. В браузере ввести localhost/test\_on\_php

**Заключение**

За двухнедельный период учебной практики достигли значительных результатов в PHP программировании и построении баз данных MySQL. Первостепенно ознакомились с прикладным программным обеспечением XAMPP. При помощи данной кроссплатформенной сборки оперативно развернули на компьютере полноценный веб-сервер. Для обеспечения корректной работы с базами данных в XAMPP производилась настройка сервера MySQL через командную строку, а также установка пути сервера MySQL. Следующим шагом стало изучение операторов MySQL и построение баз данных на её основе. Одновременно с тем осваивали язык программирования PHP, рассматривали примеры проектирования структуры базы данных. Кроме того, создали сайт на языке PHP.

**Список литературы**

1. Дэвис М., Филлипс Дж. Изучаем PHP и MySQL – СПб.: Символ+Плюс СПб, 2011. – 448 с.
2. Уроки SQL и баз данных [Электронный ресурс] – Режим доступа: https://www.site-do.ru/db/db.php (дата обращения 14.05.19)
3. Поляков Е. PHP программирование – СПб.: Наука и Техника СПб, 2017. – 256 с.
4. PHP.SU – Уроки PHP (по материалам форума PHP.SU) [Электронный ресурс] – Режим доступа: http://www.php.su/lessons/ (дата обращения 14.05.19)
5. Колисниченко Д. PHP и MySQL. Разработка веб-приложений – СПб.: БХВ-Петербург, 2017. – 640 с.
6. Приступая к работе [Электронный ресурс] – Режим доступа: https://www.php.net/manual/ru/getting-started.php (дата обращения 13.05.19)
7. Уэнц К. PHP и MySQL. Карманный справочник – Киев.: Вильямс Издательский дом, 2015. – 256 с.
8. Хопкинс К. PHP. Быстрый старт – М.: Эксмо, 2014. – 160 с.