МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

ОТЧЁТ

к курсовой работе

по дисциплине

Банки и структуры данных

(наименование дисциплины)

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Балашова Т.И.

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Видюльцев Д.И.,

(подпись) Курилин Е.С.,

Лашук А.Д.

(фамилия, и.,о.)

17-АС

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2020

**Цель работы**:

Разработка информационной системы управления данных для школьного дневника. Используется для хранения информации об учениках, учителях, заданиях, журналы с оценками и т.д.

## Требования к структуре базы данных

База данных должна хранить следующую информацию:

**Студенты:** ФИО, класс, телефон, родители

**Семья:** ФИО, телефон, место работы.

**Пользователи:** профиль, логин, пароль, тип

**Учителя:** ФИО, телефон, класс

**Успеваемость:** оценка, предмет, дата

**Предмет:** название, наличие экзамена

**Домашнее задание:** ID, класс, предмет, учитель, задание

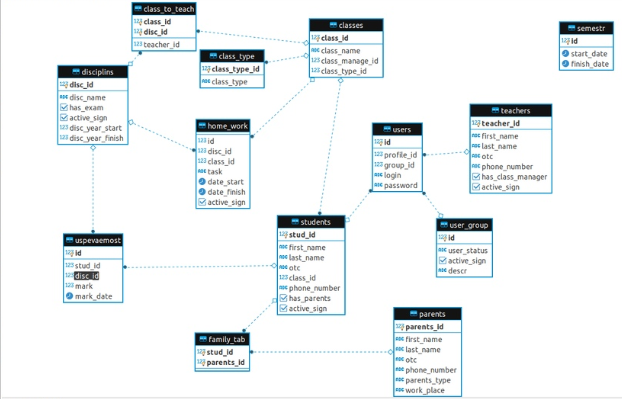
**Класс:** название, тип

Данный проект должен обеспечить удобный и гибкий способ работы с базой данных для учителей и учеников. Для этого необходимо реализовать необходимое количество форм и запросов, которые позволяя вводить, редактировать и при необходимости удалять информацию из базы данных.

# Проектирование базы данных

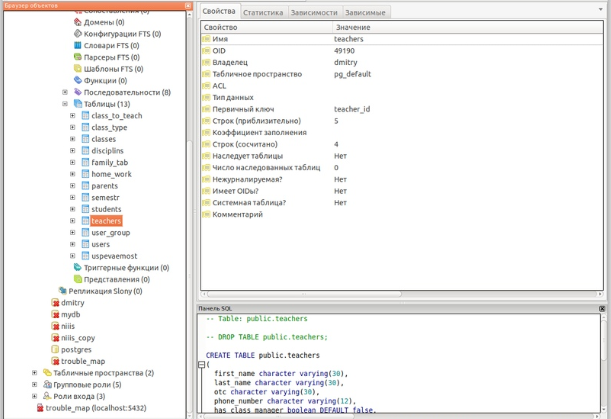
## Логическая модель

Общим способом представления логической модели БД является построение ER-диаграмм (Entity-Relationship – сущность-связь). В этой модели сущность определяется как дискретный объект, для которого сохраняются элементы данных, а связь описывает отношение между двумя объектами.



**Создание базы данных в среде PostgreSQL.**

Для работы по созданию базы данных и таблиц будем использовать PostgreSQL.



**Выбор СУБД**

Система управления базами данных (СУБД) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

## Основные функции СУБД

* управление данными во внешней памяти (на дисках);
* управление данными в оперативной памяти с использованием дискового кэша;
* журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
* поддержка языков БД (язык определения данных, язык манипулирования данными).

## Классификации СУБД

**По модели данных**

* + Иерархические
  + Сетевые
  + Реляционные
  + Объектно-ориентированные

**По степени распределённости**

* Локальные СУБД (все части локальной СУБД размещаются на одном компьютере)
* Распределённые СУБД (части СУБД могут размещаться на двух и более компьютерах).

**По способу доступа к БД**

* Файл-серверные

В файл-серверных СУБД файлы данных располагаются централизованно на файл-сервере. СУБД располагается на каждом клиентском компьютере (рабочей станции). Доступ СУБД к данным осуществляется через локальную сеть. Синхронизация чтений и обновлений осуществляется посредством файловых блокировок. Преимуществом этой архитектуры является низкая нагрузка на ЦП сервера. Недостатки: потенциально высокая загрузка локальной сети; затруднённость централизованного управления; затруднённость обеспечения таких важных характеристик как высокая надёжность, высокая доступность и высокая безопасность. Применяются чаще всего в локальных приложениях, которые используют функции управления БД.

* Клиент-серверные

Клиент-серверная СУБД располагается на сервере вместе с БД и осуществляет доступ к БД непосредственно, в монопольном режиме. Все клиентские запросы на обработку данных обрабатываются клиент-серверной СУБД централизованно. Недостаток клиент-серверных СУБД состоит в повышенных требованиях к серверу. Достоинства: потенциально более низкая загрузка локальной сети; удобство централизованного управления; удобство обеспечения таких важных характеристик как высокая надёжность, высокая доступность и высокая безопасность.

* Встраиваемые

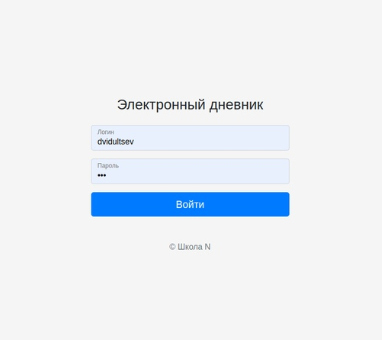
Встраиваемая СУБД — СУБД, которая может поставляться как составная часть некоторого программного продукта, не требуя процедуры самостоятельной установки. Встраиваемая СУБД предназначена для локального хранения данных своего приложения и не рассчитана на коллективное использование в сети. Физически встраиваемая СУБД чаще всего реализована в виде подключаемой библиотеки. Доступ к данным со стороны приложения может происходить через SQL либо через специальные программные интерфейсы.

Пользователь – текущий пользователь системы, от имени которого был осуществлен вход в систему.

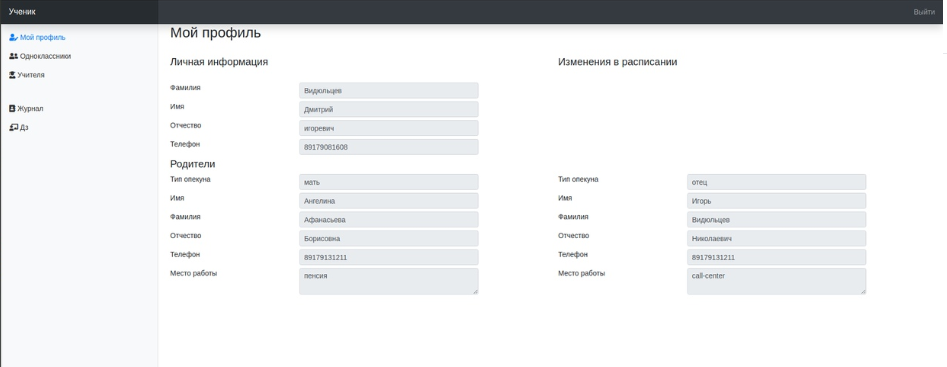
Статус администратора – наличие у текущего пользователя привилегий администратора системы.

**Руководство пользователя:**

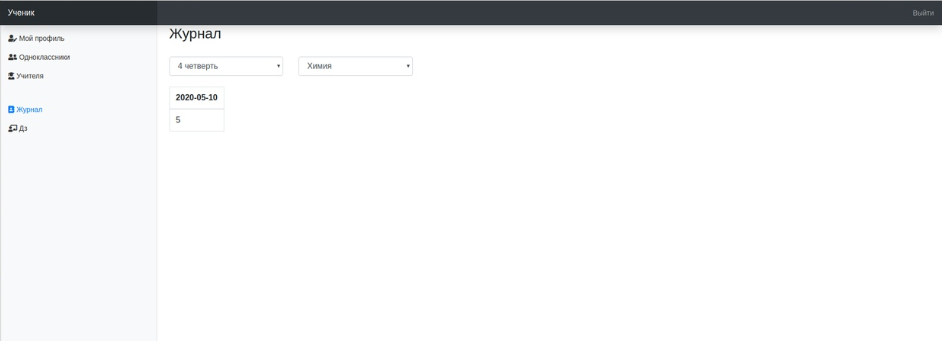
При открытии веб-сайта Ученик должен ввести пароль.



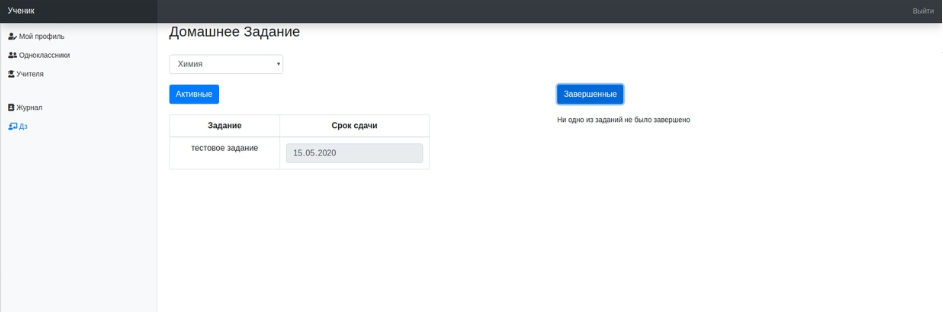
При вхождении будет показана главная страница, располагающая вкладки, при нажатии на которые пользователю будет представлена информация, соответствующая названию нажатой вкладки.



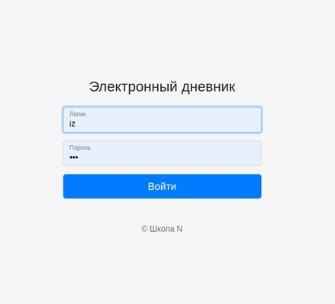
При нажатии на вкладку «Журнал» пользователю будет представлена информация об оценках и предметах, содержащаяся в базе данных.



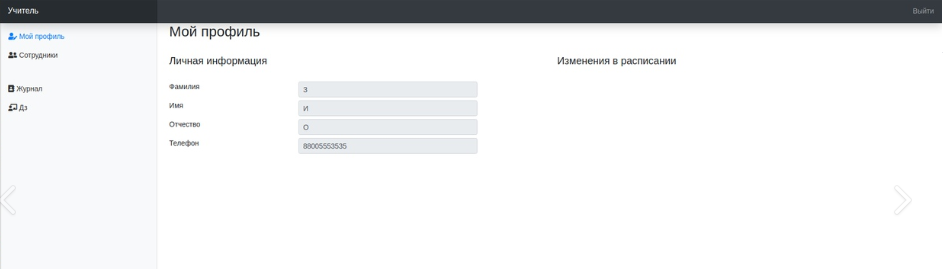
При нажатии на вкладку «Домашнее задание» пользователю будет представлена информация об домашних заданиях, содержащаяся в базе данных. Можно посмотреть, какие задания активны, а какие уже завершены.



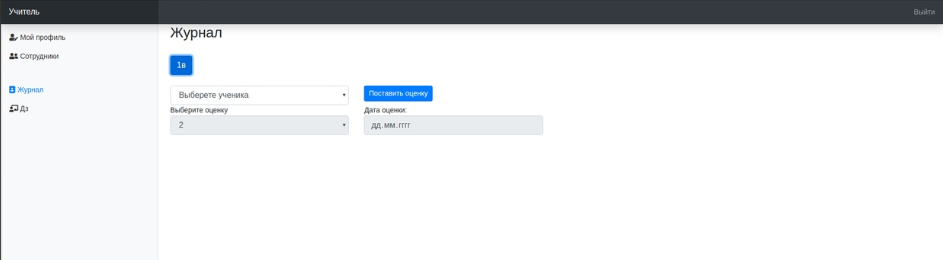
При открытии веб-сайта Ученик должен ввести пароль.



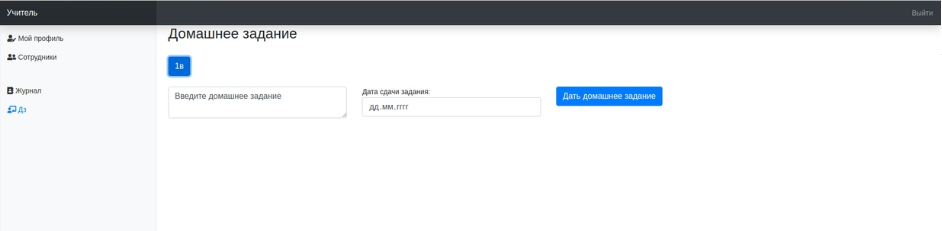
При вхождении будет показана главная страница, располагающая вкладки, при нажатии на которые пользователю будет представлена информация, соответствующая названию нажатой вкладки.



При нажатии на вкладку «Журнал» пользователю будет представлена информация об оценках и предметах, содержащаяся в базе данных, так же есть возможность добавить оценки.



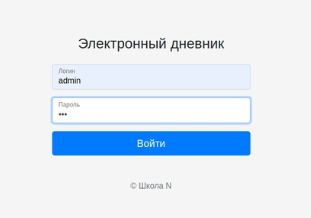
При нажатии на вкладку «Домашнее задание» пользователю будет представлена информация об домашних заданиях, содержащаяся в базе данных. Можно добавить домашнее задание для определенного класса.



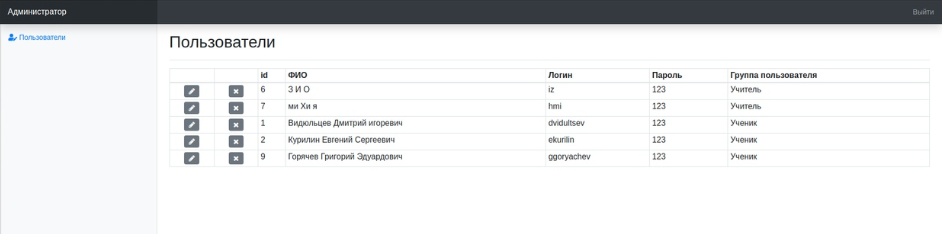
Для внесения данных нужно обратиться к администратору.

**Руководство администратора:**

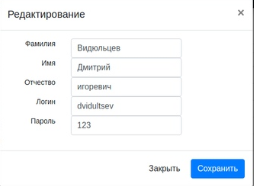
Только Администратор имеет права на изменение и удаление пользователей в таблице данных. Для этого нужно иметь специальный пароль и логин от информационной системы.



При вхождении будет показана главная страница, где можно будет посмотреть, какие пользователи существуют, а так же провести изменения и удаление аккаунтов.



Так выглядит изменение администратором записи пользователя.



# Заключение

Данная работа посвящена созданию информационной системы, предназначенной, например, для школы с дистанционным обучением, что бы помочь учителям и ученикам общаться в не школы. Все задачи, поставленные в начале работы, достигнуты. Разработанная информационная система удовлетворяет всем требованиям.

# Приложение

**System\_configuration.php:**

<?php

class conf{

static $ROOT;

static $PROJECT\_NAME;

static $SITE\_URL;

static $SITE\_URL\_SHORT;

static $SITE\_PATH;

static $SITE\_NAME;

static $SESSION\_TIME;

static $DB\_SERVER\_NAME;

static $DB\_PORT;

static $DB\_NAME;

static $DB\_USER\_NAME;

static $DB\_PASSWORD;

static $SITE\_LOCAL;

static $SITE\_SUBDIR\_INDEX; // вложенность сайта - количество дополнительных поддиректорий

static $YEAR\_SMALL\_ABIT; // Год для импорта лицевого счета

static $VERSION; // Версия - используется для принудительного предотвращения кэширования js,css

}

conf::$VERSION = 2; // Устанавливается в system.php вызовом sys::version()

conf::$SITE\_NAME = 'Цифровая фабрика ИТ-процессов';

conf::$PROJECT\_NAME = 'diary';

conf::$SITE\_URL\_SHORT = $\_SERVER['SERVER\_NAME'];

conf::$SITE\_PATH = dirname(\_\_FILE\_\_).DIRECTORY\_SEPARATOR.'../';

conf::$SESSION\_TIME = 2592000;

conf::$ROOT = rtrim($\_SERVER['DOCUMENT\_ROOT'],'/').'/';

conf::$SITE\_URL = 'http://'.$\_SERVER['SERVER\_NAME'].'/';

conf::$SITE\_LOCAL=true;

conf::$SITE\_SUBDIR\_INDEX=0; // например http://site.ru/level1/level2/ - 2

conf::$DB\_SERVER\_NAME='localhost';

conf::$DB\_PORT='5432';

conf::$DB\_NAME='LAB\_DB';

conf::$DB\_USER\_NAME='dmitry';

conf::$DB\_PASSWORD='ubvyfpbz50';

error\_reporting(0);

ini\_set( 'display\_errors', 1 );

ini\_set("memory\_limit", "200M"); // для экспорта надо много

**system\_function.php:**

<?php

class sys {

static $mysqli, $mssql\_link,$PDO;

static function db\_connect($db\_host, $db\_port, $db\_user, $db\_password, $db) {

sys::$mssql\_link = $dbconn3 = pg\_connect("host=$db\_host port=$db\_port dbname=$db user=$db\_user password=$db\_password");

if (!sys::$mssql\_link) {

throw new Exception("Failed to connect to database");

}

$PDO = new PDO("pgsql:host='$db\_host';dbname=$db", "$db\_user", "$db\_password");

sys::$PDO=$PDO;

}

static function db\_disconnect() {

mssql\_close();

}

//Проверка авторизации пользователя

static function is\_autorised() {

if (isset($\_SESSION['diary']['user\_id'])){

return true;

}else{

return false;

}

}

static function secure($text) {

// return str\_replace("'", "''",$text);

return str\_replace("'", "''", str\_replace('%', '[%]', str\_replace('\_', '[\_]', $text)));

// return mysql\_real\_escape\_string($text);

}

//Авторизация пользователя

static function autorization($name, $password) {

//Проверяем наличие такого пользователя с email

$SQL = "SELECT

u.profile\_id,g.user\_status

FROM users u left join

user\_group g on g.id = u.group\_id

where u.LOGIN=:login

and u.PASSWORD=:password";

$R = array('result' => '0');

$autorization = '';

$ip = $\_SERVER['REMOTE\_ADDR'];

$q = sys::$PDO->prepare($SQL);

$q->execute(array('login'=>sys::secure($name),'password'=>sys::secure($password)));

$result = $q->fetchAll();

if ($result) {

// нашли юзера с заданным логином, паролем

$row = $result[0];

$\_SESSION['diary']['user\_id'] = (int) $row['profile\_id'];

$\_SESSION['diary']['user\_status'] = $row['user\_status'];

// если admin, можем менять пользователей не делая relogin

// В остальных случаях не использовать $\_SESSION['hostel']['is\_super\_admin'] !

// для проверки принадлежности к админу, а использовать sys::user\_group()

if($row['user\_status']==='administrator'){

$\_SESSION['diary']['login'] = $name;

sys::session\_set('is\_super\_admin', 1, false);

}else{

sys::session\_set('is\_super\_admin', 0, false);

}

// --если admin, можем менять пользователей не делая relogin

$R = array('result' => '1');

} else {

$R = array('result' => '0');

}

return $R;

}

static function check\_ip($ip, $mask) {

$re = ' {^ ( (\\*|\d+).(\\*|\d+).(\\*|\d+).(\\*|\d+) )$ }xs';

preg\_match($re, $ip, $IP\_us);

preg\_match($re, $mask, $IP\_db);

for ($k = 2; $k < 6; $k++)

if ($IP\_db[$k] == '\*')

$IP\_db[$k] = $IP\_us[$k];

if ($IP\_us[2] == $IP\_db[2] && $IP\_us[3] == $IP\_db[3] && $IP\_us[4] == $IP\_db[4] && $IP\_us[5] == $IP\_db[5]) // Проверка на совпадение IP

return true;

else

return false;

}

static function generate($type = 'salt') {

$uniq = false;

while (!$uniq) {

$arr = array('q', 'w', 'e', 'r', 't', 'y', 'u', 'i', 'o', 'p',

'a', 's', 'd', 'f', 'g', 'h', 'j', 'k', 'l',

'z', 'x', 'c', 'v', 'b', 'n', 'm',

'1', '2', '3', '4', '5', '6', '7', '8', '9', '0');

$salt = '';

for ($i = 0; $i < 20; $i++) {

$index = rand(0, count($arr) - 1);

$salt .= $arr[$index];

}

switch ($type) {

case 'salt':

$SQL = "SELECT count(\*) FROM users WHERE password\_s='" . $salt . "'";

break;

case 'email':

$SQL = "SELECT count(\*) FROM users WHERE email\_activation\_string='" . $salt . "'";

break;

case 'session':

$SQL = "SELECT count(\*) FROM sessions WHERE hash='" . $salt . "'";

break;

}

sys::$mysqli->real\_query($SQL);

$result = sys::$mysqli->store\_result();

$result = $result->fetch\_row();

if ($result[0] === '0')

$uniq = true;

}

return $salt;

}

/\*\*

\* Возвращает true, если файл с данным id загружен на сервер

\* @param type $id номер файла

\*/

static function check\_user\_file($user, $id) {

$ank\_id = (int) $user;

$file\_id = (int) $id;

$target\_dir = "registration/uploads/";

$target\_file = $target\_dir . $ank\_id . '\_' . $file\_id . '.jpg';

return file\_exists($target\_file);

}

/\*\*

\* Возвращает login текущего юзера

\* @return string

\*/

static function user\_login() {

$R = '';

if (isset($\_SESSION['diary']['user\_id'])) {

$id = (int) $\_SESSION['diary']['user\_id'];

$sql="select LOGIN from USERS where ID=:id";

$q = sys::$PDO->prepare($sql);

$q->execute(array('id' => $id));

$result=$q->fetchAll();

if($result){

$row=$result[0];

$R=$row['login'];

}

}

return $R;

}

/\*\*

\* Установка переменных сессии

\* Если есть n\_token сессии и $cookie\_set=true, сохраняем сессию в базу и обновляем n\_token

\* @param type $var

\* @param type $val

\* @param type $cookie\_set

\*/

static function session\_set($var,$val,$cookie\_set=true){

$\_SESSION['diary'][$var]=$val;

if($cookie\_set && isset($\_COOKIE['n\_token'])){

// Установим куки

sys::cookie\_set();

}

}

static function cookie\_set(){

$user\_id=(int)sys::session('user\_id');

// $is\_super\_user=empty(sys::session('is\_super\_admin'))?0:1;

if(!empty($user\_id)){

$expire=time()+conf::$SESSION\_TIME;

$token=hash('sha256', $user\_id.microtime());

$ip=$\_SERVER['REMOTE\_ADDR'];

$agent=mb\_substr($\_SERVER['HTTP\_USER\_AGENT'],0,1000);

$session='';

$i=0;

foreach($\_SESSION['diary'] as $k=>$v){

if($i++ !== 0){$session.=",";}

$session.="$k=$v";

}

$sql="update user\_session set active=0 where user\_id=:user\_id and site=:site and ip=:ip and agent=:agent";

$q=sys::$PDO->prepare($sql);

$q->execute(array('user\_id'=>$user\_id,'site'=>conf::$SITE\_URL\_SHORT,'ip'=>$ip,'agent'=>$agent));

$sql="insert into user\_session (user\_id,token,ip,session,agent,date\_create,active,site) values (:user\_id,:token,:ip,:session,:agent,getdate(),1,:site)";

$q=sys::$PDO->prepare($sql);

$q->execute(array('user\_id'=>$user\_id,'token'=>$token,'ip'=>$ip,'session'=>$session,'agent'=>$agent,'site'=>conf::$SITE\_URL\_SHORT));

setcookie('n\_token', $token, $expire, $path='/', $domain='', $secure=false, $httponly=true);

}

}

static function cookie\_reset(){

setcookie('n\_token', '', strtotime('01.01.2000'), $path='/', $domain='');

}

static function session($var){

if(isset($\_SESSION['diary'][$var])){

return $\_SESSION['diary'][$var];

}else{

return FALSE;

}

}

/\*\*

\* Находится ли текущий пользователь в одной из групп, переданных в $groups

\* @param string $groups Группы через запятую

\* @return boolean

\*/

static function user\_in\_group($groups){

$R=false;

$Groups=explode(',',$groups);

$UG=sys::user\_group(); // Массив с правами пользователя

foreach($Groups as $group){

if(isset($UG[$group])){

$R=true;

break;

}

}

return $R;

}

//Проверка на супер админа

static function is\_super\_admin(){

if (!empty(sys::session('is\_super\_admin'))){

return true;

}

else{

return false;

}

}

/\*\*

\* Возвращает массив статусов пользователя

\* Сейчас у каждого пользователя возможен только один статус,

\* однако массив сделан на всякий случай на будущее, для более гибкой настройки

\* Возможные статусы: admin,user,reader

\* Таблица статусов: user\_group

\* @return array Например: array['user']=true

\*/

static function user\_group(){

$R=array();

$R[sys::session('user\_status')]=true;

return $R;

}

/\*\*

\* Версия по времени изменения файла для предотвращения кэширования

\* @return string 4 последних символа (секунды)

\*/

static function version(){

$version='';

$filename=conf::$SITE\_PATH.'.deployment';

if(file\_exists($filename)){

$filetime=filemtime ( $filename );

$version=substr($filetime, -4);

}

return $version;

}

static function strtodatetime($str){

// return date('d.m.Y H:m', strtotime($str));

$date = explode("-", $str);

$time = explode(' ',$date[2])[1];

$date[2] = explode(' ',$date[2])[0];

$time = explode(':',$time);

$datetime = $date[2].'.'.$date[1].'.'.$date[0].' '.$time[0].':'.$time[1];

return $datetime;

}

/\*\*

\* Функция возвращает строку для инклуда файла предотвращая кэширование

\* @param string $type тип файла, css или javascript на выбор

\* @param string $path путь файла от корневой директории

\*/

static function inc\_no\_cache($type, $path){

if($type == 'css'){

echo '<link href="'.conf::$SITE\_URL.$path.'?'.microtime(true).'" rel="stylesheet" type="text/css">';

} else if($type == 'javascript'){

echo '<script src="'.conf::$SITE\_URL.$path.'?'.microtime(true).'" type="text/javascript"></script>';

}

}

}

**Login\_view.php:**

<?php

$data['message'] = '';

if (isset($\_POST['login']) && isset($\_POST['password'])) {

$aut = sys::autorization($\_POST['login'], $\_POST['password']);

if ($aut['result'] == '1') {

header('location:' . $\_SERVER["REQUEST\_URI"]);

} else {

$data['message'] = 'Ошибка! Неверный логин или пароль!';

}

}

?><!DOCTYPE html>

<html lang="ru">

<head>

<title><?= $data['title']; ?></title>

<meta charset="UTF-8" />

<?php

sys::inc\_no\_cache('css', 'css/bootstrap.css');

sys::inc\_no\_cache('js', 'js/bootstrap.js');

sys::inc\_no\_cache('css', 'css/login.css');

sys::inc\_no\_cache('js', 'js/login.js');

?>

</head>

<body>

<form class="form-signin login\_form" method="post">

<div class="text-center mb-4">

<img class="mb-4" src="<?php echo conf::$SITE\_URL ?>images/diary\_sm.png" alt="" width="72" height="72">

<h1 class="h3 mb-3 font-weight-normal">Электронный дневник</h1>

</div>

<?php if (!empty($data['message'])): ?>

<div class="alert alert-danger" role="alert">

<?= $data['message'] ?>

<button type="button" class="close" data-dismiss="alert" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<?php endif; ?>

<div class="form-label-group">

<input type="text" id="inputLogin" class="form-control" name="login" placeholder="Логин" required autofocus>

<label for="inputLogin">Логин</label>

</div>

<div class="form-label-group">

<input type="password" id="inputPassword" class="form-control" name="password" placeholder="Пароль" required>

<label for="inputPassword">Пароль</label>

</div>

<button class="btn btn-lg btn-primary btn-block" type="submit">Войти</button>

<p class="mt-5 mb-3 text-muted text-center">&copy; Школа N<br></p>

</form>

</body>

</html>

**admin\_cab\_view.php:**

<?php

//require\_once conf::$ROOT . 'system/etc/functions.php';

// получаем адрес страницы без учета параметров

if ($\_SERVER['REQUEST\_URI'] == '/') {

$CURRENT\_PAGE = 'admin';

$CURRENT\_PAGE\_HARD = conf::$SITE\_URL;

} else {

$CURRENT\_PAGE = basename($\_SERVER['REQUEST\_URI']); // получаем адрес

// способ 2

// парсим адрес

$arr = parse\_url($CURRENT\_PAGE);

$CURRENT\_PAGE\_HARD = $CURRENT\_PAGE; // сохраняем для жестких ссылок прописанных с учетом параметров в запросе

$CURRENT\_PAGE = $arr['path']; // получаем конечный адрес

}

?>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Кабинет администратора</title>

<?php

sys::inc\_no\_cache('css', 'css/tablesorter.css');

sys::inc\_no\_cache('css', 'css/bootstrap.css');

sys::inc\_no\_cache('javascript', 'js/libraries/jquery-3.4.1.js');

sys::inc\_no\_cache('javascript', 'js/libraries/popper.min.js');

sys::inc\_no\_cache('javascript', 'js/libraries/bootstrap.min.js');

sys::inc\_no\_cache('javascript', 'js/libraries/jquery-ui.js');

sys::inc\_no\_cache('javascript', 'js/admin/admin.js');

sys::inc\_no\_cache('css', 'css/admin\_cab.css');

sys::inc\_no\_cache('css', 'css/icons-fontawesome.css');

?>

</head>

<body>

<!--верхнее меню-->

<nav class="navbar navbar-dark fixed-top bg-dark flex-md-nowrap p-0 shadow">

<a class="navbar-brand col-sm-3 col-md-2 mr-0" >Администратор</a>

<ul class="navbar-nav px-3">

<li class="nav-item text-nowrap">

<a class="nav-link" href="<?php echo conf::$SITE\_URL . 'logout' ?>">Выйти</a>

</li>

</ul>

</nav>

<!--конец верхнего меню-->

<!--начало основного блока контента-->

<div class="container-fluid">

<div class="row">

<!-------------левое меню--------------->

<nav class="col-md-2 d-none d-md-block bg-light sidebar">

<div class="sidebar-sticky">

<ul class="nav flex-column mb-2">

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='change\_users'?'active':'')?>" href="<?=conf::$SITE\_URL?>admin\_cab/change\_users">

<i class="fas fa-user-edit"></i> Пользователи

</a>

</li>

</ul>

</div>

</nav>

<!-------------конец левого меню--------------->

<!-------------Основной контент--------------->

<main role="main" class="col-md-9 ml-sm-auto col-lg-10 px-4">

<!--вставить содержимое сюда-->

<?php

if ($content\_view <> '') {

include 'system/views/admin/' . $content\_view;

} else {

}

?>

<!--конец блока содержимого-->

</main>

</div>

</div>

</body>

</html>

**student\_cab\_view.php:**

<?php

//require\_once conf::$ROOT . 'system/etc/functions.php';

// получаем адрес страницы без учета параметров

if ($\_SERVER['REQUEST\_URI'] == '/') {

$CURRENT\_PAGE = 'student\_cab';

$CURRENT\_PAGE\_HARD = conf::$SITE\_URL;

} else {

$CURRENT\_PAGE = basename($\_SERVER['REQUEST\_URI']); // получаем адрес

// способ 2

// парсим адрес

$arr = parse\_url($CURRENT\_PAGE);

$CURRENT\_PAGE\_HARD = $CURRENT\_PAGE; // сохраняем для жестких ссылок прописанных с учетом параметров в запросе

$CURRENT\_PAGE = $arr['path']; // получаем конечный адрес

}

?>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Кабинет ученика</title>

<?php

sys::inc\_no\_cache('css', 'css/tablesorter.css');

sys::inc\_no\_cache('css', 'css/bootstrap.css');

sys::inc\_no\_cache('javascript', 'js/libraries/jquery-3.4.1.js');

sys::inc\_no\_cache('javascript', 'js/libraries/popper.min.js');

sys::inc\_no\_cache('javascript', 'js/libraries/bootstrap.min.js');

sys::inc\_no\_cache('javascript', 'js/libraries/jquery-ui.js');

sys::inc\_no\_cache('javascript', 'js/admin/admin.js');

sys::inc\_no\_cache('css', 'css/admin\_cab.css');

sys::inc\_no\_cache('css', 'css/icons-fontawesome.css');

?>

</head>

<body>

<!--верхнее меню-->

<nav class="navbar navbar-dark fixed-top bg-dark flex-md-nowrap p-0 shadow">

<a class="navbar-brand col-sm-3 col-md-2 mr-0" >Ученик</a>

<ul class="navbar-nav px-3">

<li class="nav-item text-nowrap">

<a class="nav-link" href="<?php echo conf::$SITE\_URL . 'logout' ?>">Выйти</a>

</li>

</ul>

</nav>

<!--конец верхнего меню-->

<!--начало основного блока контента-->

<div class="container-fluid">

<div class="row">

<!-------------левое меню--------------->

<nav class="col-md-2 d-none d-md-block bg-light sidebar">

<div class="sidebar-sticky">

<ul class="nav flex-column mb-2">

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='student\_cab'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/">

<i class="fas fa-user-edit"></i> Мой профиль

</a>

</li>

<!-- <li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='news'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/news">

<i class="fas fa-list"></i> Новости

</a>

</li>-->

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='classmates'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/classmates">

<i class="fas fa-user-friends"></i> Одноклассники

</a>

</li>

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='teachers'?'active':'')?>" href="<?php echo conf::$SITE\_URL ?>student\_cab/teachers">

<i class="fas fa-user-graduate"></i> Учителя

</a>

</li>

</ul><br/>

<ul class="nav flex-column mb-2">

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='journal'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/journal">

<i class="fas fa-address-book"></i> Журнал

</a>

</li>

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='homework'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/homework">

<i class="fas fa-chalkboard-teacher"></i> Дз

</a>

</li>

</ul><br/>

<!-- <ul class="nav flex-column mb-2">

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='education'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/education">

<i class="fas fa-book-reader"></i> Образование

</a>

</li>

</ul>-->

</div>

</nav>

<!-------------конец левого меню--------------->

<!-------------Основной контент--------------->

<main role="main" class="col-md-9 ml-sm-auto col-lg-10 px-4">

<!--вставить содержимое сюда-->

<?php

if ($content\_view <> '') {

include 'system/views/student/' . $content\_view;

}

?>

<!--конец блока содержимого-->

</main>

</div>

</div>

</body>

</html>

**teacher\_cab\_view.php:**

<?php

//require\_once conf::$ROOT . 'system/etc/functions.php';

// получаем адрес страницы без учета параметров

if ($\_SERVER['REQUEST\_URI'] == '/') {

$CURRENT\_PAGE = 'student\_cab';

$CURRENT\_PAGE\_HARD = conf::$SITE\_URL;

} else {

$CURRENT\_PAGE = basename($\_SERVER['REQUEST\_URI']); // получаем адрес

// способ 2

// парсим адрес

$arr = parse\_url($CURRENT\_PAGE);

$CURRENT\_PAGE\_HARD = $CURRENT\_PAGE; // сохраняем для жестких ссылок прописанных с учетом параметров в запросе

$CURRENT\_PAGE = $arr['path']; // получаем конечный адрес

}

?>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Кабинет ученика</title>

<?php

sys::inc\_no\_cache('css', 'css/tablesorter.css');

sys::inc\_no\_cache('css', 'css/bootstrap.css');

sys::inc\_no\_cache('javascript', 'js/libraries/jquery-3.4.1.js');

sys::inc\_no\_cache('javascript', 'js/libraries/popper.min.js');

sys::inc\_no\_cache('javascript', 'js/libraries/bootstrap.min.js');

sys::inc\_no\_cache('javascript', 'js/libraries/jquery-ui.js');

sys::inc\_no\_cache('javascript', 'js/admin/admin.js');

sys::inc\_no\_cache('css', 'css/admin\_cab.css');

sys::inc\_no\_cache('css', 'css/icons-fontawesome.css');

?>

</head>

<body>

<!--верхнее меню-->

<nav class="navbar navbar-dark fixed-top bg-dark flex-md-nowrap p-0 shadow">

<a class="navbar-brand col-sm-3 col-md-2 mr-0" >Ученик</a>

<ul class="navbar-nav px-3">

<li class="nav-item text-nowrap">

<a class="nav-link" href="<?php echo conf::$SITE\_URL . 'logout' ?>">Выйти</a>

</li>

</ul>

</nav>

<!--конец верхнего меню-->

<!--начало основного блока контента-->

<div class="container-fluid">

<div class="row">

<!-------------левое меню--------------->

<nav class="col-md-2 d-none d-md-block bg-light sidebar">

<div class="sidebar-sticky">

<ul class="nav flex-column mb-2">

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='student\_cab'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/">

<i class="fas fa-user-edit"></i> Мой профиль

</a>

</li>

<!-- <li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='news'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/news">

<i class="fas fa-list"></i> Новости

</a>

</li>-->

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='classmates'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/classmates">

<i class="fas fa-user-friends"></i> Одноклассники

</a>

</li>

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='teachers'?'active':'')?>" href="<?php echo conf::$SITE\_URL ?>student\_cab/teachers">

<i class="fas fa-user-graduate"></i> Учителя

</a>

</li>

</ul><br/>

<ul class="nav flex-column mb-2">

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='journal'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/journal">

<i class="fas fa-address-book"></i> Журнал

</a>

</li>

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='homework'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/homework">

<i class="fas fa-chalkboard-teacher"></i> Дз

</a>

</li>

</ul><br/>

<!-- <ul class="nav flex-column mb-2">

<li class="nav-item">

<a class="nav-link <?=($CURRENT\_PAGE=='education'?'active':'')?>" href="<?=conf::$SITE\_URL?>student\_cab/education">

<i class="fas fa-book-reader"></i> Образование

</a>

</li>

</ul>-->

</div>

</nav>

<!-------------конец левого меню--------------->

<!-------------Основной контент--------------->

<main role="main" class="col-md-9 ml-sm-auto col-lg-10 px-4">

<!--вставить содержимое сюда-->

<?php

if ($content\_view <> '') {

include 'system/views/student/' . $content\_view;

}

?>

<!--конец блока содержимого-->

</main>

</div>

</div>

</body>

</html>