|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **ПРИЛОЖЕНИЕ В**  (обязательное)  Разработка web-приложения  «Сервис для создания полиграфической наградной и рекламной продукции»  Листинг исходного кода | | | | | | |
|  | | | | | | |
|  |  |  |  |  | **КП.09.02.03.ПКИПТ.297С.216.19 ПЗ** | Лист |
|  |  |  |  |  | 37 |
| Изм | Лист | № документа | Подпись | Дата |

**php/auth.php**

<?php

include\_once "const.php";

include\_once "mysql.php";

include\_once "files.php";

function setAuthCookie($id, $hash) {

setcookie(AUTH\_COOKIE\_ID, $id, time() + 60 \* 60 \* 24 \* 30, "/");

setcookie(AUTH\_COOKIE\_HASH, $hash, time() + 60 \* 60 \* 24 \* 30, "/");

}

function removeAuthCookie() {

unset($\_COOKIE[AUTH\_COOKIE\_ID]);

unset($\_COOKIE[AUTH\_COOKIE\_HASH]);

setcookie(AUTH\_COOKIE\_ID, "", time() - 3600, "/");

setcookie(AUTH\_COOKIE\_HASH, "", time() - 3600, "/");

}

function signin($login=null, $password=null) {

$user;

$hash;

if(isset($login) && isset($password)) {

$mysql = new MySQLConnection();

$mysql->query("SELECT id, hash FROM @users WHERE login='$login'");

$user = $mysql->toObject();

$hash = md5(md5($password));

} else {

return AUTH\_NO\_COOKIE;

}

if($user == NULL) {

return AUTH\_NO\_USER;

}

if($user->hash == $hash) {

setAuthCookie($user->id, $user->hash);

return RFC\_SUCCESS;

}

return RFC\_FAIL;

}

function signup($login, $password, $name, $surname) {

$loginV = validate($login, USER\_LOGIN\_PATTERN, USER\_LOGIN\_MIN, USER\_LOGIN\_MAX);

$passwordV = validate($password, USER\_PASSWORD\_PATTERN, USER\_PASSWORD\_MIN, USER\_PASSWORD\_MAX);

$nameV = validate($name, USER\_NAME\_PATTERN, USER\_NAME\_MIN, USER\_NAME\_MAX);

$surnameV = validate($surname, USER\_SURNAME\_PATTERN, USER\_SURNAME\_MIN, USER\_SURNAME\_MAX);

$response = array();

if($loginV != USER\_VALID) {

$response["login"] = $loginV;

}

if($passwordV != USER\_VALID) {

$response["password"] = $passwordV;

}

if($nameV != USER\_VALID) {

$response["name"] = $nameV;

}

if($surnameV != USER\_VALID) {

$response["surname"] = $surnameV;

}

if(count($response) == 0) {

$hash = md5(md5($password));

$mysql = new MySQLConnection();

$mysql->query("INSERT INTO @users (login, hash, name, surname) VALUES ('$login', '$hash', '$name', '$surname')");

if($mysql->result) {

$mysql->query("SELECT id FROM @users WHERE login='$login'");

$user = $mysql->toObject();

setAuthCookie($user->id, $hash);

$dir = new File("../users/".($user->id)."/configs/");

$dir->mkdir();

$dir = new File("../users/".($user->id)."/uploads/");

$dir->mkdir();

return RFC\_SUCCESS;

} else {

return USER\_LOGIN\_BUSY;

}

} else {

return $response;

}

}

function currentRole() {

if(isset($\_COOKIE[AUTH\_COOKIE\_ID]) && isset($\_COOKIE[AUTH\_COOKIE\_HASH])) {

$mysql = new MySQLConnection();

$mysql->query("SELECT role, hash FROM @users WHERE id='".$\_COOKIE[AUTH\_COOKIE\_ID]."'");

$user = $mysql->toObject();

if($user->hash != $\_COOKIE[AUTH\_COOKIE\_HASH]) {

signout(true);

} else {

return $user->role == "admin" ? USER\_ROLE\_ADMIN : USER\_ROLE\_USER;

}

}

return USER\_ROLE\_GUEST;

}

function currentID() {

if(currentRole() != USER\_ROLE\_GUEST) {

return $\_COOKIE[AUTH\_COOKIE\_ID];

}

return USER\_ID\_GUEST;

}

function signout($force=FALSE) {

if(!$force && currentRole() == USER\_ROLE\_GUEST) {

return ACCESS\_DENIED;

}

removeAuthCookie();

return RFC\_SUCCESS;

}

function checkAccess($level=USER\_ROLE\_ADMIN) {

return currentRole() >= $level;

}

function validate($string, $pattern, $min, $max) {

$len = strlen($string);

if($len < $min) {

return USER\_INVALID\_SHORT;

} else if($len > $max) {

return USER\_INVALID\_LONG;

}

$valid = (boolean)preg\_match($pattern, $string);

return $valid ? USER\_VALID : USER\_INVALID;

}

?>

**php/config.php**

<?php

include\_once "const.php";

include\_once "auth.php";

include\_once "files.php";

include\_once "tools.php";

function recoverCfg($broken) {

$pattern = json\_decode(file\_get\_contents(CONFIG\_PATTERN\_PATH));

foreach($broken as $key => $value) {

$pattern->$key = $value;

}

return $pattern;

}

function resolvePath($uid=null) {

$path = "../users/".currentID()."/configs/";

if($uid != null) {

$path .= "$uid.json";

}

return $path;

}

function createCfg() {

$uid = uniqidReal();

$data = recoverCfg(array("uid" => $uid));

saveCfg($uid, $data);

return $data;

}

function loadCfg($uid) {

if(!checkAccess(USER\_ROLE\_USER)) {

return ACCESS\_DENIED;

}

$path = resolvePath($uid);

if(!file\_exists($path)) {

return RFC\_NOT\_FOUND;

}

$file = file\_get\_contents($path);

$file = json\_decode($file);

return recoverCfg($file);

}

function saveCfg($uid, $cfg) {

if(!checkAccess(USER\_ROLE\_USER)) {

return ACCESS\_DENIED;

}

file\_put\_contents(resolvePath($uid), json\_encode($cfg));

}

function removeCfg($uid) {

if(!checkAccess(USER\_ROLE\_USER)) {

return ACCESS\_DENIED;

}

$file = new File(resolvePath($uid));

if(!$file->exists()) {

return RFC\_NOT\_FOUND;

}

$file->delete();

return RFC\_SUCCESS;

}

function listCfg() {

if(!checkAccess(USER\_ROLE\_USER)) {

return ACCESS\_DENIED;

}

$dir = new File(resolvePath());

return $dir->listNames("\*.json", false);

}

?>

**php/const.php**

<?php

$\_CONST = [];

function defConst($name, $val) {

global $\_CONST;

$\_CONST[$name] = $val;

define($name, $val);

}

define("SEGMENTS\_DIR", "../segments/");

define("TYPES\_LIST\_PATH", SEGMENTS\_DIR."types.json");

defConst("AUTH\_COOKIE\_ID", "id");

defConst("AUTH\_COOKIE\_HASH", "hash");

defConst("RFC\_FAIL", 0x00);

defConst("RFC\_SUCCESS", 0x01);

defConst("RFC\_NOT\_FOUND", 0x02);

defConst("RFC\_EXISTS", 0x03);

defConst("FIELD\_NOT\_CHANGED", null);

defConst("AUTH\_NO\_COOKIE", 0x10);

defConst("AUTH\_NO\_USER", 0x11);

defConst("USER\_ID\_GUEST", 0);

defConst("USER\_VALID", 0x20);

defConst("USER\_INVALID", 0x21);

defConst("USER\_INVALID\_SHORT", 0x22);

defConst("USER\_INVALID\_LONG", 0x23);

defConst("USER\_ROLE\_GUEST", 0x24);

defConst("USER\_ROLE\_USER", 0x25);

defConst("USER\_ROLE\_ADMIN", 0x26);

defConst("USER\_LOGIN\_BUSY", 0x27);

defConst("USER\_LOGIN\_PATTERN", "/^[a-zA-Z0-9@\_]+$/");

defConst("USER\_LOGIN\_MIN", 4);

defConst("USER\_LOGIN\_MAX", 16);

defConst("USER\_PASSWORD\_PATTERN", "/^[a-zA-Z0-9\_@#$%&+-]+$/");

defConst("USER\_PASSWORD\_MIN", 8);

defConst("USER\_PASSWORD\_MAX", 32);

defConst("USER\_NAME\_PATTERN", "/^[a-zA-Zа-яА-Я]+$/u");

defConst("USER\_NAME\_MIN", 1);

defConst("USER\_NAME\_MAX", 32);

defConst("USER\_SURNAME\_PATTERN", "/^[a-zA-Zа-яА-Я]+$/u");

defConst("USER\_SURNAME\_MIN", 1);

defConst("USER\_SURNAME\_MAX", 32);

defConst("ACCESS\_DENIED", 0x30);

defConst("NO\_SUCH\_GROUP", 0x40);

defConst("NO\_SUCH\_METHOD", 0x41);

defConst("EDITABLE\_SEGMENT\_PREVIEW", "Текст");

define("CONFIG\_PATTERN\_PATH", "../empty\_configuraton.json");

?>

**php/draw.php**

<?php

function draw($uid, $w, $h) {

include\_once 'const.php';

include\_once 'config.php';

include\_once 'image.php';

include\_once 'types.php';

include\_once 'segments.php';

if(gettype($uid) == "array") {

$cfg = $uid;

$uid = $uid["uid"];

} else {

$cfg = get\_object\_vars(loadCfg($uid));

}

$types = listTypes();

$bgcolor = $cfg["bgcolor"];

$w = round($w);

$image = new Image($w, $h, $bgcolor[0], $bgcolor[1], $bgcolor[2]);

$texts = new Image($w, $h, $bgcolor[0], $bgcolor[1], $bgcolor[2]);

foreach ($types as $type) {

if(isset($cfg[$type->id])) {

$parts = $cfg[$type->id];

if(strpos($type->type, "editable") !== FALSE) {

foreach ($parts as $part) {

drawText($texts, $type->id, $part->uid, $part->x \* $w, $part->y \* $h, $part->size \* $w, $part->color, $part->value);

}

}

}

}

$texts->drawOn($image->image, 0, 0);

foreach ($types as $type) {

if(isset($cfg[$type->id])) {

$parts = $cfg[$type->id];

if($type->type == "static") {

drawStatic($image, $type->id, $parts, $w, $h);

} else if(strpos($type->type, "editable") === FALSE) {

foreach ($parts as $part) {

drawResizable($image, $type->id, $part["uid"], $part["x"] \* $w, $part["y"] \* $h, $part["width"] \* $w, $part["height"] \* $h);

}

}

}

}

$image->out();

}

function drawStatic($image, $type, $uid, $width, $height) {

$src = new Image(findPathToSegment($type, $uid));

$src->resize($width, $height);

$src->drawOn($image->image, 0, 0);

}

function drawText($image, $type, $uid, $x, $y, $size, $color, $value) {

$size = ((int)$size)/1;

$font = "..".($uid->file);

$cl = imagecolorallocate($image->image, $color[0], $color[1], $color[2]);

$lines = explode("\n", $value);

$bounds = imageftbbox($size, 0.0, $font, $lines[0]);

$x += abs($bounds[0]);

$y += abs($bounds[5]) + abs($bounds[7]);

$w = abs($bounds[2] - $bounds[0]);

$h = abs($bounds[7] - $bounds[1]);

foreach($lines as $text) {

$b = imagettfbbox($size, 0.0, $font, $text);

$lw = abs($b[2] - $b[0]);

imagettftext($image->image, $size, 0.0, $x + ($w - $lw) / 2, $y, $cl, $font, $text);

$y += $h\*1.25;

}

}

function drawResizable($image, $type, $uid, $x, $y, $width, $height) {

$part = new Image(findPathToSegment($type, $uid));

$part->resize($width, $height);

$part->drawOn($image->image, $x, $y);

$part->destroy();

}

?>

**php/files.php**

<?php

class File {

var $path;

function \_\_construct($path) {

$this->path = $path;

if($this->isDirectory() && substr($this->path, -1) != "/" && substr($this->path, -1) != "\\") {

$path .= "/";

}

}

function isFile() {

return !is\_dir($this->path);

}

function isDirectory() {

return is\_dir($this->path);

}

function listFiles($wrap=false, $pattern="\*") {

if(!$this->isDirectory()) {

return null;

}

$path = $this->path;

$result = glob($path.$pattern);

if(!$wrap) {

return $result;

}

$count = count($result);

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

$result[$i] = new File($result[$i]);

}

return $result;

}

function listNames($pattern="\*", $withext=true) {

if(!$this->isDirectory()) {

return null;

}

$path = $this->path;

$result = glob($path.$pattern);

$count = count($result);

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

$file = new File($result[$i]);

$result[$i] = $file->getName($withext);

}

return $result;

}

function lastModified() {

return filemtime($this->path);

}

function size() {

return filesize($this->path);

}

function delete() {

if($this->isFile()) {

unlink($this->path);

} else {

$content = $this->listFiles();

foreach($content as $fpath) {

$file = new File($fpath);

$file->delete();

}

rmdir($this->path);

}

}

function mkdir() {

mkdir($this->path, 0700, true);

}

function create() {

$parent = $this->getParent();

$parent->mkdir();

file\_put\_contents($this->path, "");

}

function exists() {

return file\_exists($this->path);

}

function getPath() {

return $this->path;

}

function getName($withext=true) {

$index = strrpos($this->path, "/");

if(!$index) {

$index = strrpos($this->path, "\\");

}

$name = substr($this->path, $index + 1, strlen($this->path));

if($this->isFile() && !$withext) {

$name = substr($name, 0, strrpos($name, "."));

}

return $name;

}

function getParent() {

$index = strrpos($this->path, "/");

if(!$index) {

$index = strrpos($this->path, "\\");

}

return new File(substr($this->path, 0, $index));

}

function read() {

return file\_get\_contents($this->path);

}

function write($text) {

file\_put\_contents($this->path, $text);

}

}

?>

**php/image.php**

<?php

class Image {

var $image;

var $type;

function \_\_construct() {

$args = func\_get\_args();

$argc = func\_num\_args();

if($argc == 1) {

$image\_info = getimagesize($args[0]);

$this->type = $image\_info[2];

switch($this->type) {

case IMAGETYPE\_JPEG: $this->image = imagecreatefromjpeg($args[0]); break;

case IMAGETYPE\_GIF: $this->image = imagecreatefromgif($args[0]); break;

case IMAGETYPE\_PNG: $this->image = imagecreatefrompng($args[0]); break;

}

} else {

$this->type = IMAGETYPE\_PNG;

$this->image = imagecreatetruecolor($args[0], $args[1]);

$color = imagecolorallocate($this->image, $args[2], $args[3], $args[4]);

imagefilledrectangle($this->image, 0, 0, $args[0], $args[1], $color);

}

imageAlphaBlending($this->image, false);

imageSaveAlpha($this->image, true);

}

function save($filename, $compression=75) {

switch($this->type) {

case IMAGETYPE\_JPEG: imagejpeg($this->image, $filename, $compression); break;

case IMAGETYPE\_GIF: imagegif($this->image, $filename); break;

case IMAGETYPE\_PNG: imagepng($this->image, $filename); break;

}

}

function out() {

header("Content-Type: ".($this->mimeType()));

switch($this->type) {

case IMAGETYPE\_JPEG: imagejpeg($this->image); break;

case IMAGETYPE\_GIF: imagegif($this->image); break;

case IMAGETYPE\_PNG: imagepng($this->image); break;

}

}

function mimeType() {

$type = "";

switch($this->type) {

case IMAGETYPE\_JPEG: $type = "jpeg"; break;

case IMAGETYPE\_GIF: $type = "gif"; break;

case IMAGETYPE\_PNG: $type = "png"; break;

}

return "image/$type";

}

function getWidth() {

return imagesx($this->image);

}

function getHeight() {

return imagesy($this->image);

}

function resizeToHeight($height, $high\_quality=true) {

$ratio = $height / $this->getHeight();

$width = $this->getWidth() \* $ratio;

$this->resize($width, $height, $high\_quality);

}

function resizeToWidth($width, $high\_quality=true) {

$ratio = $width / $this->getWidth();

$height = $this->getheight() \* $ratio;

$this->resize($width, $height, $high\_quality);

}

function scale($scale, $high\_quality=true) {

$width = $this->getWidth() \* $scale/100;

$height = $this->getheight() \* $scale/100;

$this->resize($width, $height, $high\_quality);

}

function resize($width=-1, $height=-1, $high\_quality=true) {

if($width == -1) {

return $this->resizeToHeight($height, $high\_quality);

}

if($height == -1) {

return $this->resizeToWidth($width, $high\_quality);

}

$new\_image;

if($high\_quality) {

$new\_image = imagecreatetruecolor($width, $height);

} else {

$new\_image = imagecreate($width, $height);

}

imageAlphaBlending($new\_image, false);

imageSaveAlpha($new\_image, true);

imagecopyresampled($new\_image, $this->image, 0, 0, 0, 0, $width, $height, $this->getWidth(), $this->getHeight());

$this->image = $new\_image;

}

function drawOn($dst, $dst\_x, $dst\_y, $src\_x=0, $src\_y=0, $src\_w=-1, $src\_h=-1, $pct=100) {

if($src\_w < 0) { $src\_w = $this->getWidth(); }

if($src\_h < 0) { $src\_h = $this->getHeight(); }

mergeImages($dst, $this->image, $dst\_x, $dst\_y, $src\_x, $src\_y, $src\_w, $src\_h, $pct);

}

function destroy() {

imagedestroy($this->image);

}

}

function mergeImages($dst\_im, $src\_im, $dst\_x, $dst\_y, $src\_x, $src\_y, $src\_w, $src\_h, $pct){

$cut = imagecreatetruecolor($src\_w, $src\_h);

imagecopy($cut, $dst\_im, 0, 0, $dst\_x, $dst\_y, $src\_w, $src\_h);

imagecopy($cut, $src\_im, 0, 0, $src\_x, $src\_y, $src\_w, $src\_h);

imagecopymerge($dst\_im, $cut, $dst\_x, $dst\_y, 0, 0, $src\_w, $src\_h, $pct);

}

?>

**php/mysql\_config.php**

<?php

define("MYSQL\_SERVER", "localhost");

define("MYSQL\_USER", "root");

define("MYSQL\_PASSWORD", null);

define("MYSQL\_DATABASE", "mydiploma");

define("MYSQL\_USERS\_TABLE", "users");

?>

**php/mysql.php**

<?php

include\_once "mysql\_config.php";

class MySQLConnection {

public $mysql;

public $result;

function \_\_construct() {

$this->mysql = new mysqli(MYSQL\_SERVER, MYSQL\_USER, MYSQL\_PASSWORD);

$this->query("CREATE DATABASE IF NOT EXISTS ".MYSQL\_DATABASE);

$this->query("USE ".MYSQL\_DATABASE);

$this->query("CREATE TABLE IF NOT EXISTS @users(".

"id INT NOT NULL AUTO\_INCREMENT, ".

"login VARCHAR(64) NOT NULL UNIQUE, ".

"hash VARCHAR(64) NOT NULL, ".

"name VARCHAR(64) NOT NULL, ".

"surname VARCHAR(64) NOT NULL, ".

"role ENUM ('user', 'admin') NOT NULL DEFAULT 'user', ".

"PRIMARY KEY(id))");

}

function query($query) {

$query = str\_replace("@users", MYSQL\_USERS\_TABLE, $query);

$this->result = $this->mysql->query($query);

return $this->result;

}

function toObject() {

if($this->result->num\_rows == 0) return null;

return (object)$this->result->fetch\_assoc();

}

function toArray() {

return $this->result->fetch\_array();

}

function get($key) {

return $this->toObject()[$key];

}

function allRows() {

$result = array();

for($i = 0; $i < $this->result->num\_rows; $i++) {

$result[] = $this->toObject();

}

return $result;

}

function \_\_destruct() {

$this->mysql->close();

}

}

function replaceConfig($comment, $value, $text) {

return preg\_replace("/\"$comment\", ?\".\*\"/", "\"$comment\", \"$value\"", $text);

}

function editConfigFile($host=FIELD\_NOT\_CHANGED, $user=FIELD\_NOT\_CHANGED, $password=FIELD\_NOT\_CHANGED, $database=FIELD\_NOT\_CHANGED, $table=FIELD\_NOT\_CHANGED) {

$file = file\_get\_contents("mysql\_config.php");

if($host != FIELD\_NOT\_CHANGED) {

$file = replaceConfig("MYSQL\_SERVER", $host, $file);

}

if($user != FIELD\_NOT\_CHANGED) {

$file = replaceConfig("MYSQL\_USER", $user, $file);

}

if($password != FIELD\_NOT\_CHANGED) {

$file = replaceConfig("MYSQL\_PASSWORD", $password, $file);

}

if($database != FIELD\_NOT\_CHANGED) {

$file = replaceConfig("MYSQL\_DATABASE", $database, $file);

}

if($table != FIELD\_NOT\_CHANGED) {

$file = replaceConfig("MYSQL\_USERS\_TABLE", $table, $file);

}

file\_put\_contents("mysql\_config.php", $file);

}

?>

**php/rfc.php**

<?php

include\_once "const.php";

$response = array("code"=>RFC\_SUCCESS, "body"=>"");

if(isset($\_GET["auth"])) {

include\_once "auth.php";

switch($\_POST["method"]) {

case "signin":

$response["code"] = signin($\_POST["login"], $\_POST["password"]);

break;

case "signup":

$r = signup($\_POST["login"], $\_POST["password"], $\_POST["name"], $\_POST["surname"]);

if(gettype($r) == "array") {

$response["code"] = RFC\_FAIL;

$response["body"] = $r;

} else {

$response["code"] = $r;

}

break;

case "signout":

$response["code"] = signout();

break;

case "role":

$response["body"] = currentRole();

break;

case "id":

$response["body"] = currentID();

break;

default:

$response["code"] = NO\_SUCH\_METHOD;

}

} else if(isset($\_GET["types"])) {

include\_once "types.php";

switch($\_POST["method"]) {

case "list":

$response["body"] = listTypes();

break;

case "add":

$response["code"] = addType($\_POST["id"], $\_POST["name"], $\_POST["storage"], $\_POST["type"]);

break;

case "edit":

$response["code"] = editType($\_POST["id"],

isset($\_POST["name"]) ? $\_POST["name"] : FIELD\_NOT\_CHANGED,

isset($\_POST["storage"]) ? $\_POST["storage"] : FIELD\_NOT\_CHANGED,

isset($\_POST["type"]) ? $\_POST["type"] : FIELD\_NOT\_CHANGED);

break;

case "remove":

$response["code"] = removeType($\_POST["id"]);

break;

default:

$response["code"] = NO\_SUCH\_METHOD;

}

} else if(isset($\_GET["tools"])) {

include\_once "tools.php";

switch($\_POST["method"]) {

case "const": $response["body"] = $\_CONST; break;

case "uniqid": $response["body"] = uniqidReal(); break;

default: $response["code"] = NO\_SUCH\_METHOD;

}

} else if(isset($\_GET["segments"])) {

include\_once "segments.php";

switch($\_POST["method"]) {

case "list":

$response["body"] = listSegments($\_POST["type"], $\_POST["offset"], $\_POST["count"]);

break;

case "load":

loadSegment($\_POST["type"], $\_POST["uid"], $\_POST["width"], $\_POST["height"]);

exit;

break;

case "remove":

$response["code"] = removeSegment($\_POST["type"], $\_POST["uid"]);

break;

default:

if(isset($\_GET["type"]) && isset($\_GET["upload"])) {

$response["code"] = uploadSegment($\_GET["type"], $\_FILES["image"]);

break;

}

var\_dump($\_POST["method"]);

$response["code"] = NO\_SUCH\_METHOD;

}

} else if(isset($\_GET["config"])) {

include\_once "config.php";

switch($\_POST["method"]) {

case "create": $response["body"] = createCfg(); break;

case "load": $response["body"] = loadCfg($\_POST["uid"]); break;

case "save": $response["body"] = saveCfg($\_POST["uid"], $\_POST["config"]); break;

case "draw":

include\_once "draw.php";

draw($\_POST["uid"], $\_POST["width"], $\_POST["height"]);

exit;

break;

case "remove": $response["code"] = removeCfg($\_POST["uid"]); break;

case "list": $response["body"] = listCfg(); break;

default: $response["code"] = NO\_SUCH\_METHOD;

}

} else if(isset($\_GET["user"])) {

include\_once "user.php";

switch($\_POST["method"]) {

case "exists": $response["code"] = isUserExists($\_POST["login"]); break;

case "read": $response["body"] = readUser(isset($\_POST["id"]) ? $\_POST["id"] : currentID()); break;

case "save":

$r = saveUser(

isset($\_POST["login"]) ? $\_POST["login"] : FIELD\_NOT\_CHANGED,

isset($\_POST["old\_password"]) ? $\_POST["old\_password"] : FIELD\_NOT\_CHANGED,

isset($\_POST["new\_password"]) ? $\_POST["new\_password"] : FIELD\_NOT\_CHANGED,

isset($\_POST["name"]) ? $\_POST["name"] : FIELD\_NOT\_CHANGED,

isset($\_POST["surname"]) ? $\_POST["surname"] : FIELD\_NOT\_CHANGED

);

if(gettype($r) == 'array') {

$response["code"] = RFC\_FAIL;

$response["body"] = $r;

} else {

$response["code"] = $r;

} break;

case "delete": $response["body"] = deleteUser(isset($\_POST["id"]) ? $\_POST["id"] : currentID()); break;

case "make\_admin": makeAdmin($\_POST["id"], $\_POST["admin"]); break;

case "configs": $response["body"] = listConfigs(isset($\_POST["id"]) ? $\_POST["id"] : currentID()); break;

case "uploads": $response["body"] = listUploads(isset($\_POST["id"]) ? $\_POST["id"] : currentID()); break;

case "list": $response["body"] = listUsers($\_POST["params"]); break;

default: $response["code"] = NO\_SUCH\_METHOD;

}

} else { $response["code"] = NO\_SUCH\_GROUP; }

echo json\_encode($response);

?>

**php/segments.php**

<?php

include\_once "const.php";

include\_once "image.php";

include\_once "types.php";

include\_once "files.php";

include\_once "auth.php";

include\_once "tools.php";

function listSegments($type, $offset, $length) {

$types = listTypes();

$type = searchType($type, $types);

if($type == -1) {

return RFC\_NOT\_FOUND;

}

$type = $types[$type];

$list;

if($type->storage == "json") {

$list = json\_decode(file\_get\_contents(SEGMENTS\_DIR."/".($type->id).".json"));

} else {

$list = new File(SEGMENTS\_DIR.($type->id)."/");

$list = $list->listNames("\*.png", false);

}

if(isset($offset)) {

$list = array\_slice($list, $offset, $length);

}

return $list;

}

function findPathToSegment($type, $uid) {

return SEGMENTS\_DIR."$type/$uid.png";

}

function loadSegment($type, $uid, $width, $height) {

$image = new Image(findPathToSegment($type, $uid));

if(isset($width) && isset($height)) {

$image->resize($width, $height);

} else if(isset($width)) {

$image->resizeToWidth($width);

} else if(isset($height)) {

$image->resizeToHeight($height);

}

$image->out();

}

function uploadSegment($type, $file) {

if(!checkAccess()) {

return ACCESS\_DENIED;

}

$uid = uniqidReal();

$path = findPathToSegment($type, $uid);

$moved = move\_uploaded\_file($file["tmp\_name"], $path);

if($moved) {

return RFC\_SUCCESS;

}

return RFC\_NOT\_FOUND;

}

function removeSegment($type, $uid) {

if(!checkAccess()) {

return ACCESS\_DENIED;

}

$file = new File(findPathToSegment($type, $uid));

if(!$file->exists()) {

return RFC\_NOT\_FOUND;

}

$file->delete();

return RFC\_SUCCESS;

}

?>

**php/tools.php**

<?php

function uniqidReal($length = 13) {

// uniqid gives 13 chars, but you could adjust it to your needs.

if (function\_exists("random\_bytes")) {

$bytes = random\_bytes(ceil($length / 2));

} elseif (function\_exists("openssl\_random\_pseudo\_bytes")) {

$bytes = openssl\_random\_pseudo\_bytes(ceil($length / 2));

} else {

throw new Exception("no cryptographically secure random function available");

}

return substr(bin2hex($bytes), 0, $length);

}

function listFiles($path) {

include\_once "files.php";

$f = new File($path);

$files = $f->listFiles(true);

$result = array();

foreach($files as $file) {

$result[] = array(

"path" => $file->getPath(),

"size" => $file->size(),

"lastModified" => $file->lastModified(),

"type" => $file->isFile() ? "file" : "dir"

);

}

return $result;

}

?>

**php/types.php**

<?php

include\_once "const.php";

function listTypes() {

$types = file\_get\_contents(TYPES\_LIST\_PATH);

$types = json\_decode($types);

return $types;

}

function searchType($id, $list=null) {

$types = ($list == null ? listTypes() : $list);

foreach($types as $key => $type) {

if($type->id == $id) {

return $key;

}

}

return -1;

}

function addType($id, $name, $storage, $type) {

$origin = listTypes();

if(searchType($id, $origin) != -1) {

return RFC\_EXISTS;

}

$origin[] = array(

"id" => $id,

"name" => $name,

"storage" => $storage,

"type" => $type

);

file\_put\_contents(TYPES\_LIST\_PATH, json\_encode($origin));

return RFC\_SUCCESS;

}

function editType($id, $name=FIELD\_NOT\_CHANGED, $storage=FIELD\_NOT\_CHANGED, $type=FIELD\_NOT\_CHANGED) {

$origin = listTypes();

$key = searchType($id, $origin);

if($key == -1) {

return RFC\_NOT\_FOUND;

}

if($name != FIELD\_NOT\_CHANGED) {

$origin[$key]->name = $name;

}

if($storage != FIELD\_NOT\_CHANGED) {

$origin[$key]->$storage = $storage;

}

if($type != FIELD\_NOT\_CHANGED) {

$origin[$key]->$type = $type;

}

file\_put\_contents(TYPES\_LIST\_PATH, json\_encode($origin));

return RFC\_SUCCESS;

}

function removeType($id) {

$origin = listTypes();

$key = searchType($id, $list);

if($key == -1) {

return RFC\_NOT\_FOUND;

}

unset($origin[$key]);

file\_put\_contents(TYPES\_LIST\_PATH, json\_encode($origin));

return RFC\_SUCCESS;

}

?>

**php/user.php**

<?php

include\_once "const.php";

include\_once "mysql.php";

include\_once "auth.php";

function isUserExists($login) {

$mysql = new MySQLConnection();

$mysql->query("SELECT id FROM @users WHERE login='$login'");

return $mysql->result->num\_rows === 1;

}

function makeAdmin($id, $admin) {

if(!checkAccess()) {

return ACCESS\_DENIED;

}

$role = $admin == "true" ? "admin" : "user";

$mysql = new MySQLConnection();

$mysql->query("UPDATE @users SET role='$role' WHERE id=$id");

return RFC\_SUCCESS;

}

function readUser($id) {

if(!checkAccess() && $id != currentID()) {

return ACCESS\_DENIED;

}

$mysql = new MySQLConnection();

$mysql->query("SELECT id, login, name, surname, role FROM @users WHERE id='$id'");

if($mysql->result->num\_rows == 1) {

$user = $mysql->toObject();

return $user;

}

return RFC\_NOT\_FOUND;

}

function saveUser($login=FIELD\_NOT\_CHANGED, $old\_password=FIELD\_NOT\_CHANGED, $password=FIELD\_NOT\_CHANGED, $name=FIELD\_NOT\_CHANGED, $surname=FIELD\_NOT\_CHANGED) {

if(currentRole() == USER\_ROLE\_GUEST) {

return ACCESS\_DENIED;

}

$mysql = new MySQLConnection();

$mysql->query("SELECT login, hash, name, surname, role FROM @users WHERE id=".currentID());

$user = $mysql->toObject();

$response = array();

if($login != FIELD\_NOT\_CHANGED) {

$loginV = validate($login, USER\_LOGIN\_PATTERN, USER\_LOGIN\_MIN, USER\_LOGIN\_MAX);

if($loginV != USER\_VALID) {

$response["login"] = $loginV;

}

} else {

$login = $user->login;

}

if($old\_password != FIELD\_NOT\_CHANGED) {

$passwordV = validate($password, USER\_PASSWORD\_PATTERN, USER\_PASSWORD\_MIN, USER\_PASSWORD\_MAX);

if($passwordV != USER\_VALID) {

$response["password"] = $passwordV;

}

if($passwordV == USER\_VALID) {

if($user->hash != md5(md5($old\_password))) {

$response["password"] = ACCESS\_DENIED;

} else {

$hash = md5(md5($password));

setAuthCookie(currentID(), $hash);

}

}

} else {

$hash = $user->hash;

}

if($name != FIELD\_NOT\_CHANGED) {

$nameV = validate($name, USER\_NAME\_PATTERN, USER\_NAME\_MIN, USER\_NAME\_MAX);

if($nameV != USER\_VALID) {

$response["name"] = $nameV;

}

} else {

$name = $user->name;

}

if($surname != FIELD\_NOT\_CHANGED) {

$surnameV = validate($surname, USER\_SURNAME\_PATTERN, USER\_SURNAME\_MIN, USER\_SURNAME\_MAX);

if($surnameV != USER\_VALID) {

$response["surname"] = $surnameV;

}

} else {

$surname = $user->surname;

}

if(count($response) == 0) {

$mysql->query("UPDATE @users SET login='$login', hash='$hash', name='$name', surname='$surname' WHERE id=".currentID());

return RFC\_SUCCESS;

}

return $response;

}

function deleteUser($id) {

if(!checkAccess() && $id != currentID()) {

return ACCESS\_DENIED;

}

$mysql = new MySQLConnection();

$mysql->query("DELETE FROM @users WHERE id=$id");

if($mysql->result) {

if($id == currentID())

signout();

return RFC\_SUCCESS;

}

return RFC\_FAIL;

}

function listUploads($id) {

if(!checkAccess() && $id != currentID()) {

return ACCESS\_DENIED;

}

$dir = new File("../users/$id/uploads/");

return $dir->listNames(".png", false);

}

function listConfigs($id) {

if(!checkAccess() && $id != currentID()) {

return ACCESS\_DENIED;

}

$dir = new File("../users/$id/configs/");

$d= $dir->listNames("\*.json", false);

return $d;

}

function listUsers($request) {

if(!checkAccess()) {

return ACCESS\_DENIED;

}

$query = "SELECT ";

if(isset($request["fields"])) {

$query .= $request["fields"];

} else {

$query .= "id, login, name, surname, role";

}

$query .= " FROM @users";

if(isset($request["sort"])) {

$sort = $request["sort"];

$query .= " ORDER BY ".$sort["field"];

if($sort["to"] == "down") {

$query .= " ASC";

} else {

$query .= " DESC";

}

}

if(isset($request["limit"])) {

$limit = $request["limit"];

$query .= " LIMIT ".($limit["offset"]).", ".$limit["length"];

}

$mysql = new MySQLConnection();

$mysql->query($query);

return $mysql->allRows();

}

?>

**css/admin.css**

@CHARSET "UTF-8";

#no-access {

position: fixed;

top: 0px;

left: 0px;

width: 100%;

height: 100%;

display: flex;

flex-direction: column;

justify-content: center;

text-align: center;

}

#no-access IMG {

width: 10%;

margin: auto;

}

#content {

width: 100%;

padding: 0px;

}

#users {

border-collapse: collapse;

width: 48%;

font-size: 125%;

margin: 1%;

float: left;

box-shadow: 0px 0px 10px black;

border-radius: 0px;

border-top-left-radius: 30px;

border-top-right-radius: 30px;

}

#users TD {

text-align: center;

padding: 2px;

}

#users TH {

background: #FDD212;

}

#users TR {

border-bottom: solid 3px #120A8F;

}

#users .cell {

border-right: dotted 1px #120A8F;

padding: 10px;

}

#users .tool {

height: 0px;

}

#segments {

width: 48%;

padding: 1%;

display: inline-block;

overflow-x: hidden;

}

#controls {

width: 100%;

text-align: center;

margin-bottom: 10px;

}

#segments .tool {

display: inline-block;

float: left;

margin-left: 10px;

}

#type {

background: #FDD212;

color: #120A8F;

border: solid 3px #120A8F;

border-radius: 30px;

padding: 10px;

text-align: center;

font-size: 15pt;

display: inline-block;

float: left;

}

.segment-list {

width: 100%;

}

.segment-item {

margin-right: 10px;

margin-top: 10px;

}

.refresh-window {

width: 33%;

padding: 5%;

background: #FCF0BD;

border-radius: 30px;

font-size: 22pt;

margin-left: 31%;

}

**css/common.css**

@CHARSET "UTF-8";

@font-face {

font-family: Stylo;

src: url("../fonts/Stylo.ttf");

}

@font-face {

font-family: Gagalin;

src: url("../fonts/gagalin.otf");

}

HTML {

height: 100%;

}

HTML, BODY {

padding: 0px;

margin: 0px;

}

\* {

outline: 0;

font-family: Stylo;

}

BODY {

background: #FCF0BD;

color: #120A8F;

}

#header {

width: 99%;

height: 7%;

max-height: 75px;

padding: 0.5%;

background: #FDD212;

box-shadow: 0px 0px 10px black;

position: fixed;

top: 0px;

left: 0px;

z-index: 10;

}

#logo {

float: left !important;

}

#header img {

height: 100%;

}

#sign-controls {

position: relative;

height: 100%;

float: right;

padding-left: 20px;

}

#content {

position: relative;

}

@keyframes sign-control-rotation {

to {

transform: rotate(360deg);

}

}

@keyframes scroll-arrow-up-rotation {

to {

transform: rotate(180deg);

}

}

@keyframes scroll-arrow-down-rotation {

from {

transform: rotate(180deg);

}

to {

transform: rotate(0deg);

}

}

.nice-input-container {

background: #FCF0BD;

color: #120A8F;

border: solid 3px #FDD212;

border-radius: 30px;

padding: 10px;

margin-top: 3%;

margin-left: 1%;

margin-right: 1%;

text-align: left;

}

.nice-input-container[invalid] {

color: #7F0000;

border: solid 3px #7F0000;

border-radius: 30px;

padding: 10px;

}

.nice-input-container[focused][invalid] {

color: #FF0000;

border: solid 3px #FF0000;

border-radius: 30px;

padding: 10px;

}

.nice-input-container[focused] {

color: #120A8F;

border: solid 3px #120A8F;

border-radius: 30px;

padding: 10px;

}

.nice-input-input {

background: #FCF0BD;

color: #120A8F;

border: none;

width: 100%;

font-size: 15pt;

text-align: center;

}

.nice-input-input[invalid] {

color: #7F0000;

}

.nice-input-input[invalid][focused] {

color: #FF0000;

}

.nice-input-hint {

background: #FCF0BD;

color: #120A8F;

padding: 5px;

padding-bottom: 0px;

position: relative;

font-size: 15pt;

}

.nice-input-hint[invalid] {

color: #7F0000;

}

.nice-input-hint[invalid][focused] {

color: #FF0000;

}

@keyframes loading {

from {

transform: rotate(0deg);

}

to {

transform: rotate(360deg);

}

}

#loading-tools {

position: relative;

height: 100%;

float: right;

padding-left: 20px;

cursor: pointer;

}

.modal-container {

width: 100%;

height: 100%;

top: 0px;

left: 0px;

position: fixed;

background: rgba(0, 0, 0, 0.5);

z-index: 15;

flex-direction: column;

justify-content: center;

text-align: center;

display: flex;

}

.modal-loading {

width: 75%;

max-width: 300px;

margin: auto;

animation: loading 1s infinite linear;

}

.sign-container {

width: 30%;

padding: 2%;

border-radius: 30px;

margin-left: 33%;

background: #FCF0BD;

color: #120A8F;

box-shadow: 0px 0px 10px black;

}

.sign-title {

width: 100%;

font-size: 150%;

}

.sign-button {

width: 45%;

border: solid 3px #120A8F;

box-shadow: 1px 1px 3px #120A8F;

border-radius: 30px;

background: #FDD212;

color: #120A8F;

padding: 2%;

font-size: 18pt;

margin-top: 5%;

margin-left: 1%;

}

.downloading-window {

width: 50%;

padding: 2%;

border-radius: 30px;

margin-left: 25%;

background: #FCF0BD;

color: #120A8F;

font-size: 125%;

box-shadow: 0px 0px 10px black;

}

.downloading-options {

width: 33%;

display: inline-block;

float: left;

}

.downloading-title {

width: 100%;

font-size: 150%;

margin-bottom: 5%;

}

.downloading-quality {

background: #FDD212;

color: #120A8F;

border: solid 3px #120A8F;

border-radius: 30px;

padding: 10px;

}

.downloading-preview {

width: 50%;

}

.downloading-button {

width: 96%;

border: solid 3px #120A8F;

box-shadow: 1px 1px 3px #120A8F;

border-radius: 30px;

background: #FDD212;

color: #120A8F;

padding: 2%;

font-size: 18pt;

margin-top: 5%;

display: block;

}

**css/index.css**

@CHARSET "UTF-8";

BODY {

height: 100%;

}

#controls {

width: 10%;

max-width: 80px;

padding: 10px;

position: fixed;

right: 10px;

bottom: 10px;

z-index: 2;

}

#controls img {

width: 100%;

border-radius: 50%;

margin-top: 10px;

background: #FDD212;

box-shadow: 0px 0px 20px black;

}

#contacts {

width: 10%;

max-width: 80px;

padding: 10px;

position: fixed;

left: 10px;

bottom: 10px;

z-index: 2;

}

#contacts img {

width: 100%;

border: solid 3px #FDD212;

border-radius: 50%;

margin-top: 10px;

background: #FDD212;

box-shadow: 0px 0px 20px black;

}

#landing {

width: 100%;

height: 100%;

display: flex;

flex-direction: column;

justify-content: center;

text-align: center;

border-bottom: solid 5px black;

}

#logo {

width: 34%;

margin-left: 33%;

margin-right: 33%;

z-index: 1;

font-size: 65pt;

text-shadow: 0px 0px 35px black;

color: #FDD212;

}

#welcome {

width: 50%;

z-index: 1;

margin-left: auto;

margin-right: auto;

color: #FDD212;

font-size: 175%;

}

.splash {

width: 100%;

height: 100%;

z-index: 0;

position: absolute;

top: 0px;

left: 0px;

}

#splash-shadow {

background: black;

opacity: 0.6;

z-index: 1;

}

#content {

width: 50%;

max-width: 1000px;

font-size: 15pt;

margin: auto;

text-align: justify;

}

#author {

margin-bottom: 10%;

}

#sign-controls {

height: auto !important;

padding-left: 0px !important;

}

**css/user.css**

@CHARSET "UTF-8";

\* {

font-size: 125%;

}

#no-access {

position: fixed;

top: 0px;

left: 0px;

width: 100%;

height: 100%;

display: flex;

flex-direction: column;

justify-content: center;

text-align: center;

}

#no-access IMG {

width: 10%;

margin: auto;

}

#content {

width: 60%;

padding-top: 10px;

margin: auto;

text-align: center;

}

#user-info {

width: 60%;

margin: auto;

}

#id {

font-size: 15pt;

text-align: center;

}

#user-info BUTTON {

width: 100%;

border: solid 3px #120A8F;

box-shadow: 1px 1px 3px #120A8F;

border-radius: 30px;

background: #FDD212;

color: #120A8F;

padding: 10px;

margin-top: 3%;

margin-left: 1%;

margin-right: 1%;

font-size: 15pt;

}

#user-info > .container {

width: 50%;

}

#works {

width: 100%;

overflow-x: scroll;

}

#works-images {

width: 0px;

font-size: 90%;

overflow-y: hidden;

}

#works-images > \* {

margin-right: 10px;

#uploads {

width: 100%;

overflow-x: scroll;

}

#uploads-images {

width: 0px;

font-size: 90%;

overflow-y: hidden;

#uploads-images > \* {

margin-right: 10px;

}

#error {

font-size: 50%;

}

.lock-window {

width: 100%;

height: 100%;

display: flex;

flex-direction: column;

justify-content: center;

text-align: center;

background: rgba(0, 0, 0, 0.35);

position: fixed;

z-index: 20;

position: relative;

}

.lock-loading {

width: 75%;

margin: auto;

animation: loading 1s infinite linear;

}

**css/workbench.css**

@CHARSET "UTF-8";

\* {

font-size: 100%;

}

#content {

font-size: 125%;

}

#content > DIV {

padding: 1%;

}

#options {

width: 18%;

max-width: 300px;

background: #F9E8A2;

box-shadow: 0px 0px 5px black;

float: left;

}

#bgcolor-container {

border-radius: 0px;

border-top-right-radius: 20px;

border-bottom-right-radius: 20px;

box-shadow: 0px 0px 5px black;

margin-bottom: 5%;

padding: 0px;

padding-bottom: 2%;

background: #F7D962;

}

#bgcolor-title {

background: #F7D962;

border-radius: 0px;

border-top-right-radius: 20px;

border-bottom-right-radius: 20px;

width: 94%;

padding: 3%;

padding-bottom: 1%;

display: block;

text-align: center;

}

#bgcolor-content {

height: 0px;

background: #F9E389;

width: 100%;

border-radius: 0px;

border-bottom-right-radius: 20px;

overflow: hidden;

}

#bgcolor-picker {

padding: 5%;

width: 90%;

height: 90%;

overflow: hidden;

}

#bgcolor-picker> CANVAS {

border-radius: 50%;

border: solid 2px #120A8F;

overflow: hidden;

}

#texteditor-container[disabled] {

background: #C0C0C0;

color: black;

box-shadow: 0px 0px 3px black;

opacity: 0.5;

}

#texteditor-container[disabled] #texteditor-title {

background: #C0C0C0;

}

#texteditor-container {

border-radius: 0px;

border-top-right-radius: 20px;

border-bottom-right-radius: 20px;

box-shadow: 0px 0px 5px black;

margin-bottom: 5%;

padding: 0px;

padding-bottom: 2%;

background: #F7D962;

}

#texteditor-title {

background: #F7D962;

border-radius: 0px;

border-top-right-radius: 20px;

border-bottom-right-radius: 20px;

width: 94%;

padding: 3%;

padding-bottom: 1%;

display: block;

text-align: center;

}

#texteditor-content {

background: #F9E389;

width: 100%;

border-radius: 0px;

border-bottom-right-radius: 20px;

overflow: hidden;

}

#texteditor-content > DIV {

padding: 5%;

}

#texteditor-value {

width: 90%;

padding: 5%;

font-size: 120%;

border: solid 2px #120A8F;

border-radius: 25px;

background: #FCF0BD;

color: #120A8F;

resize: none;

}

#texteditor-picker {

width: 100%;

margin-top: 5%;

margin-bottom: 5%;

}

#texteditor-picker > CANVAS {

border-radius: 50%;

border: solid 2px #120A8F;

}

#texteditor-font {

padding: 5px;

border-radius: 0px;

border-top-left-radius: 30px;

border-bottom-left-radius: 30px;

margin: 0px;

border: solid 2px #120A8F;

background: #FCF0BD;

color: #120A8F;

}

#texteditor-size {

padding: 5px;

border-radius: 0px;

border-top-right-radius: 30px;

border-bottom-right-radius: 30px;

margin: 0px;

border: solid 2px #120A8F;

background: #FCF0BD;

color: #120A8F;

}

#texteditor-controls > BUTTON {

width: 100%;

padding: 3%;

margin-top: 5%;

border: solid 2px #120A8F;

border-radius: 25px;

background: #F7D962;

box-shadow: 1px 1px 3px #120A8F;

color: #120A8F;

}

#bench {

position: absolute;

}

#segments {

width: 28%;

max-width: 550px;

background: #F9E8A2;

box-shadow: 0px 0px 5px black;

float: right;

overflow: hidden;

}

#blank {

background: white;

margin: auto;

box-shadow: 0px 0px 10px black;

z-index: 3;

}

.segment-container {

border-radius: 0px;

border-top-left-radius: 20px;

border-bottom-left-radius: 20px;

box-shadow: 0px 0px 5px black;

margin-bottom: 3%;

padding: 0px;

padding-bottom: 1%;

background: #F7D962;

}

.segment-list {

overflow-y: scroll;

height: 0px;

background: #F9E389;

width: 100%;

border-radius: 0px;

border-bottom-left-radius: 20px;

}

.segment-title {

background: #F7D962;

border-radius: 0px;

border-top-left-radius: 20px;

border-bottom-left-radius: 20px;

width: 96%;

padding: 2%;

padding-bottom: 1%;

display: block;

text-align: center;

}

IMG.segment {

margin-left: 10px;

margin-top: 10px;

}

SPAN.segment {

display: inline-block;

margin-left: 10px;

margin-top: 10px;

text-align: center;

}

.dragged-segment {

position: absolute;

z-index: 5;

}

.resizable-tools-container {

position: absolute;

}

.resizable-tool {

position: absolute;

top: 0px;

left: 0px;

display: none;

}

.editable-item {

display: block;

position: absolute;

}

**js/workbench/color\_picker.js**

ColorPicker = function(parent) {

var $r = this.$root = parent;

var width = $r.width();

var height = $r.width();

var $c = this.$circle = $("<canvas></canvas>");

$c.width(width).height("100%");

$r.append($c);

$c.attr("width", width).attr("height", height);

var ctx = $c[0].getContext('2d');

var $img = new Image();

$img.onload = function() {

ctx.drawImage($img, 0, 0, $img.width, $img.height, 0, 0, width, height);

};

$img.src = "/img/color\_ring.png";

var MOUSE\_DOWN = false;

var thiz = this;

var on\_pick\_color = function(x, y) {

var color = ctx.getImageData(x, y, 1, 1).data;

var circleOffset = $c.offset();

if(color[3] == 255) {

//$s.offset({ top: circleOffset.top + y - 2, left: circleOffset.left + x - 2});

thiz.onColorPicked(color);

return true;

}

return false;

}

$c.on("mousedown", function(event) {

if(on\_pick\_color(event.offsetX, event.offsetY)) {

MOUSE\_DOWN = true;

}

});

$(document).on("mousemove", function(event){

if(MOUSE\_DOWN) {

on\_pick\_color(event.offsetX, event.offsetY);

}

});

$(document).on("mouseup", function(event) {

MOUSE\_DOWN = false;

});

}

ColorPicker.prototype.onColorPicked = function(color) {};

**js/workbench/config.js**

Config = {

\_\_current: null,

load: function() {

var data = {};

if(currentRole() == USER\_ROLE\_GUEST) {

var data = localStorage.getItem("config");

if(data && data != "null") {

Config.\_\_current = JSON.parse(data);

this.\_\_import();

return;

} else {

localStorage.removeItem("config-uid");

}

}

var uid = document.location.search.replace("?", "");

if(uid == "") {

uid = localStorage.getItem("config-uid");

if(!uid || uid == null) {

Config.create();

return;

}

} else {

history.pushState(null, "", "http://mydiploma.ru/workbench.php");

}

callRemoteFunction("config", "load", { uid: uid }, function(data) {

if(data.body == RFC\_NOT\_FOUND) {

localStorage.removeItem("config-uid");

Config.load();

} else {

Config.\_\_current = data.body;

Config.\_\_import();

}

});

},

create: function() {

callRemoteFunction("config", "create", { async: false }, function(data) {

Config.\_\_current = data.body;

});

},

\_\_import: function() {

Config.setBackground(Config.\_\_current["bgcolor"]);

for(var i in SEGMENT\_TYPES) {

var type = SEGMENT\_TYPES[i];

var width = $("#blank").width();

var height = $("#blank").height();

if(Config.\_\_current[type.id]) {

if(~type.type.indexOf("static")) {

Config.setStatic(type.id, Config.\_\_current[type.id]);

} else if(~type.type.indexOf("resizable")) {

var list = Config.\_\_current[type.id];

for(var i in list) {

Config.\_\_show\_resizable(i+1, type.id, list[i].uid, list[i].x \* width, list[i].y \* height, list[i].width \* width, list[i].height \* height, "");

}

} else if(~type.type.indexOf("editable")) {

var list = Config.\_\_current[type.id];

for(var i in list) {

var color = [];

for(var j in list[i].color) {

color.push(list[i].color[j]);

}

Config.\_\_show\_editable(parseInt(i)+1, type.id, list[i].uid, list[i].x \* width, list[i].y \* height, color, list[i].size \* $("#blank").width(), list[i].value);

}

}

}

}

},

\_\_export: function() {

var config = {};

for(var i in this.\_\_current) {

if(typeof this.\_\_current[i] == 'object' && this.\_\_current[i].length > 0) {

config[i] = [];

for(var j in this.\_\_current[i]) {

if(this.\_\_current[i][j] && this.\_\_current[i][j] != null) {

config[i].push(this.\_\_current[i][j]);

}

}

} else {

config[i] = this.\_\_current[i];

}

}

return config;

},

save: function() {

if(currentRole() == USER\_ROLE\_GUEST) {

localStorage.setItem("config", JSON.stringify(this.\_\_export()));

} else {

localStorage.setItem("config-uid", this.\_\_current.uid);

var config = this.\_\_export();

callRemoteFunction("config", "save", { uid: this.\_\_current.uid, config: config }, function(data) {});

}

},

setStatic: function(type, uid) {

var $blank = $("#blank");

this.\_\_current[type] = uid;

var xhr = callRemoteFunction("segments", "load", {

type: type,

uid: uid,

width: $blank.width(),

height: $blank.height()

}, function(data) {

$("#" + type).attr("src", URL.createObjectURL(xhr.response));

}, "blob")

Config.save();

},

\_\_show\_resizable: function(index, type, uid, x, y, width, height, defSrc) {

var resizable = new Resizable(type, uid, x, y, width, height, {

type: type,

index: index-1

}, defSrc);

},

addResizable: function(type, uid, x, y, width, height, defSrc) {

type = type.id;

if(!this.\_\_current[type]) {

this.\_\_current[type] = [];

}

var index = this.\_\_current[type].push({ uid: uid });

this.changeResizable({

type: type,

index: index-1

}, x, y, width, height);

this.\_\_show\_resizable(index, type, uid, x, y, width, height, defSrc);

},

changeResizable(anchor, x, y, width, height) {

var type = this.\_\_current[anchor.type];

if(x != FIELD\_NOT\_CHANGED)

type[anchor.index].x = x / $("#blank").width();

if(y != FIELD\_NOT\_CHANGED)

type[anchor.index].y = y / $("#blank").height();

if(width != FIELD\_NOT\_CHANGED)

type[anchor.index].width = width / $("#blank").width();

if(height != FIELD\_NOT\_CHANGED)

type[anchor.index].height = height / $("#blank").height();

Config.save();

},

\_\_show\_editable: function(index, type, uid, x, y, color, size, value) {

var editable = new Editable({

type: type,

index: index-1

}, uid, x, y, color, size, value);

},

addEditable: function(type, uid, x, y, color, size, value) {

type = type.id;

if(!this.\_\_current[type]) {

this.\_\_current[type] = [];

}

var index = this.\_\_current[type].push({ uid: uid });

this.changeEditable({

type: type,

index: index-1

}, x, y, uid, color, size, value);

this.\_\_show\_editable(index, type, uid, x, y, color, size, value);

},

changeEditable(anchor, x, y, uid, color, size, value) {

var type = this.\_\_current[anchor.type];

if(x != FIELD\_NOT\_CHANGED)

type[anchor.index].x = x / $("#blank").width();

if(y != FIELD\_NOT\_CHANGED)

type[anchor.index].y = y / $("#blank").height();

if(uid != FIELD\_NOT\_CHANGED)

type[anchor.index].uid = uid;

if(color != FIELD\_NOT\_CHANGED)

type[anchor.index].color = color;

if(size != FIELD\_NOT\_CHANGED)

type[anchor.index].size = size / $("#blank").width();

if(value != FIELD\_NOT\_CHANGED)

type[anchor.index].value = value;

Config.save();

},

remove: function(anchor) {

var type = this.\_\_current[anchor.type];

delete this.\_\_current[anchor.type][anchor.index];

Config.save();

},

setBackground: function(color) {

this.\_\_current.bgcolor = [color[0], color[1], color[2], 255];

$("#blank").css("background", "rgba(" + color.join(",") + ")");

Config.save();

}

}

**js/workbench/type\_list.js**

function initExpandable(view, list, height, object) {

var ex = false;

view.on("click", function() {

var h = 0;

if(ex) {

ex = false;

object.currentExpanded = null;

} else {

h = height();

ex = true;

var exp = object.currentExpanded;

if(exp && exp[0] != view[0]) {

exp.click();

}

object.currentExpanded = view;

}

list.animate({ height: h });

});

}

TypeList = function(type) {

this.type = type;

var $r = this.$root = $("<div class='segment-container'></div>");

var $s = this.$span = $("<span class='segment-title'>" + type.name + "</span>");

var $l = this.$list = $("<div class='segment-list'></div>");

$r.append($s).append($l);

initExpandable($s, $l, function(){ return TypeList.MAX\_HEIGHT; }, TypeList);

var thiz = this;

$l.on("scroll", function() {

if($l[0].scrollTop >= $l[0].scrollTopMax \* 3 / 4) {

thiz.load();

}

});

}

TypeList.prototype.load = function() {

if(this.isLoading) {

return;

}

this.isLoading = true;

var offset = this.$list.children().length;

var thiz = this;

var width = thiz.$list.width() / 3 - 20;

console.log(TypeList.MAX\_HEIGHT, Math.ceil(TypeList.MAX\_HEIGHT / width) \* 2 );

callRemoteFunction("segments", "list", { type: this.type.id, offset: offset, count: Math.ceil(TypeList.MAX\_HEIGHT / width) \* 2 }, function(response) {

var uids = response.body;

window["LOADING\_VIEWS"] += uids.length;

for(var i in uids) {

var item;

if(~thiz.type.type.indexOf("editable")) {

item = new TextListItem(thiz.type, uids[i], width);

} else {

item = new ListItem(thiz.type, uids[i], width);

}

item.appendTo(thiz.$list);

}

thiz.isLoading = false;

});

};

TypeList.prototype.appendTo = function(root) {

root.append(this.$root);

};

**js/workbench/views.js**

LOADING\_VIEWS = -1;

ListItem = **function**(type, uid, width) {

**var** $i = **this**.$item = $("<img class='segment'/>");

**var** $d = **this**.$draggable = $("<img class='dragged-segment'/>");

$i.outerWidth(width);

$d.outerWidth(width);

$i.on("load", **function**() {

window["LOADING\_VIEWS"]--;

});

**var** xhr = callRemoteFunction("segments", "load", { type: type.id, uid: uid, width: width }, **function**(data) {

**var** blob = URL.createObjectURL(xhr.response);

$i.attr("src", blob);

$d.attr("src", blob);

}, "blob");

**var** update\_draggable = **function**(x, y) {

$d.css("top", y).css("left", x);

};

**var** out\_blank = **function**(d\_off) {

**var** $b = $("#blank");

**var** b\_off = $b.offset();

**if**(d\_off.left < b\_off.left || d\_off.top < b\_off.top) {

**return** **true**;

}

**if**(d\_off.left + $d.width() > b\_off.left + $b.width() ||

d\_off.top + $d.height() > b\_off.top + $b.height()) {

**return** **true**;

}

**return** **false**;

};

**var** down= **null**;

$i.on("mousedown", **function**(event) {

down = { x: event.offsetX, y: event.offsetY };

update\_draggable(event.pageX - down.x, event.pageY - down.y);

$(document.body).append($d);

});

$(window).on("mousemove", **function**(event) {

**if**(down != **null**) {

update\_draggable(event.pageX - down.x, event.pageY - down.y);

}

})

$(window).on("mouseup", **function**(event) {

**var** offset = $d.offset();

**var** width = $d.width(), height = $d.height();

$d.fadeOut(200, **function**() {

$d.offset({ top: 0, left: 0 });

$d.remove();

$d.fadeIn();

});

**if**(down) {

**if**(out\_blank(offset)) **return**;

offset.left -= $("#blank").offset().left;

offset.top -= $("#blank").offset().top;

**if**(~type.type.indexOf("static")) {

Config.setStatic(type.id, uid);

} **else** **if**(~type.type.indexOf("resizable")) {

Config.addResizable(type, uid, offset.left, offset.top, width, height, $d.attr("src"));

}

}

down = **null**;

});

};

ListItem.prototype.appendTo = **function**(root) {

root.append(**this**.$item);

};

TextListItem = **function**(type, uid, width) {

**var** $i = **this**.$item = $("<span class='segment'>" + EDITABLE\_SEGMENT\_PREVIEW + "</span>");

**var** $d = **this**.$draggable = $("<span class='dragged-segment'>" + EDITABLE\_SEGMENT\_PREVIEW + "</span>");

$i.outerWidth(width);

$d.outerWidth(width);

$i.css("font-family", uid.font).css("font-size", "24pt");

$d.css("font-family", uid.font).css("font-size", "24pt");

**var** update\_draggable = **function**(x, y) {

$d.css("top", y).css("left", x);

};

**var** out\_blank = **function**(d\_off) {

**var** $b = $("#blank");

**var** b\_off = $b.offset();

**if**(d\_off.left < b\_off.left || d\_off.top < b\_off.top) {

**return** **true**;

}

**if**(d\_off.left + $d.width() > b\_off.left + $b.width() ||

d\_off.top + $d.height() > b\_off.top + $b.height()) {

**return** **true**;

}

**return** **false**;

};

**var** down = **null**;

$i.on("mousedown", **function**(event) {

event.preventDefault();

down = { x: event.offsetX, y: event.offsetY };

update\_draggable(event.pageX - down.x, event.pageY - down.y);

$(document.body).append($d);

});

$(window).on("mousemove", **function**(event) {

**if**(down != **null**) {

update\_draggable(event.pageX - down.x, event.pageY - down.y);

}

})

$(window).on("mouseup", **function**(event) {

**var** offset = $d.offset();

**var** width = $d.width(), height = $d.height();

$d.fadeOut(200, **function**() {

$d.offset({ top: 0, left: 0 });

$d.remove();

$d.fadeIn();

});

**if**(down) {

**if**(out\_blank(offset)) **return**;

offset.left -= $("#blank").offset().left;

offset.top -= $("#blank").offset().top;

**if**(~type.type.indexOf("editable")) {

Config.addEditable(type, uid, offset.left, offset.top, [0, 0, 0], 24, "Текст");

}

}

down = **null**;

})

window["LOADING\_VIEWS"]--;

};

TextListItem.prototype.appendTo = **function**(root) {

root.append(**this**.$item);

};

Resizable = **function**(type, uid, x, y, width, height, anchor, defSrc) {

LOADING\_VIEWS++;

x += $("#blank").offset().left;

y += $("#blank").offset().top;

**var** $r = **this**.$root = $("<div class='resizable-tools-container'></div>");

**var** $i = **this**.$image = $("<img src='" + defSrc + "'/>");

**var** $s = **this**.$resize = $("<img class='resizable-tool' src='/img/tools/resize.png' />");

**var** $d = **this**.$delete = $("<img class='resizable-tool' src='/img/tools/delete.png' />");

$r.offset({ top: y, left: x });

$d.width($("#blank").width() / 10);

$s.width($("#blank").width() / 10);

**var** xhr = callRemoteFunction("segments", "load", { type: type, uid: uid }, **function**() {

**var** blob = URL.createObjectURL(xhr.response);

$i.attr("src", blob);

LOADING\_VIEWS--;

}, "blob");

**var** update\_draggable = **function**(x, y) {

$r.offset({ top: y, left: x });

};

**var** update\_tools = **function**(width, height) {

$i.width(width).height(height);

$d.css("margin-left", width - ($d.width() / 2)).css("margin-top", -($d.width() / 2));

$s.css("margin-left", width - ($d.width() / 2)).css("margin-top", height - ($d.width() / 2));

};

update\_tools(width, height);

**var** out\_blank = **function**(d\_off) {

**var** $b = $("#blank");

**var** b\_off = $b.offset();

**if**(d\_off.left < b\_off.left || d\_off.top < b\_off.top) {

**return** **true**;

}

**if**(d\_off.left + $d.width() > b\_off.left + $b.width() ||

d\_off.top + $d.height() > b\_off.top + $b.height()) {

**return** **true**;

}

**return** **false**;

}

**var** moveDown = **null**;

$i.on("mousedown", **function**(event) {

moveDown = { x: event.offsetX, y: event.offsetY };

})

**var** resizeDown = **false**;

$s.on("mousedown", **function**(event) {

resizeDown = **true**;

});

$(window).on("mousemove", **function**(event) {

**if**(moveDown != **null**) {

update\_draggable(event.pageX - moveDown.x, event.pageY - moveDown.y);

}

**if**(resizeDown) {

**var** width = Math.abs(event.pageX - $r.offset().left);

**var** height = Math.abs(event.pageY - $r.offset().top);

update\_tools(width, height);

}

})

$(window).on("mouseup", **function**(event) {

**if**(moveDown) {

**if**(out\_blank($r.offset())) { $d.click(); **return**; };

**var** offset = $r.offset();

Config.changeResizable(anchor, offset.left - $("#blank").offset().left, offset.top - $("#blank").offset().top, $i.width(), $i.height());

}

moveDown = **null**;

resizeDown = **false**;

})

$d.on("click", **function**(event) {

Config.remove(anchor);

$r.fadeOut(300, **function**() {

$r.remove();

});

})

$r.on("mouseenter", **function**() {

$d.fadeIn(200);

$s.fadeIn(200);

});

$r.on("mouseleave", **function**() {

$d.fadeOut(200);

$s.fadeOut(200);

});

$r.append($i).append($s).append($d);

$(document.body).append($r);

};

Editable = **function**(anchor, uid, x, y, color, size, value) {

x += $("#blank").offset().left;

y += $("#blank").offset().top;

**var** $s = **this**.$span = $("<span class='editable-item'>" + value.replaceAll("\n", "<br/>") + "</span>");

$s.offset({ top: y, left: x });

$(document.body).append($s);

$s.css("font-family", uid.font).css("font-size", size + "pt");

$s.css("color", "rgba(" + color.join(",") + ")");

$s.css("text-align", "center");

**var** CURRENT\_COLOR = color;

$s.on("click", **function**() {

Options.textColorPicker.onColorPicked = **function**(color) {

Config.changeEditable(anchor, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, color, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED);

$s.css("color", "rgba(" + color.join(",") + ")");

CURRENT\_COLOR = color;

};

Options.textFont.val(JSON.stringify(uid));

Options.textFont[0].onchange = **function**() {

**var** font = JSON.parse(Options.textFont.val());

Config.changeEditable(anchor, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, font, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED);

$s.css("font-family", font.font);

};

Options.textValue.val($s.html().replaceAll("<br>", "\n"));

Options.textValue[0].onkeyup = Options.textValue[0].onchange = **function**(event) {

Config.changeEditable(anchor, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, Options.textValue.val());

$s.html(Options.textValue.val().replaceAll("\n", "<br/>"));

};

Options.textSize.val(size);

Options.textSize[0].onkeyup = Options.textSize[0].onchange = **function**(event) {

Config.changeEditable(anchor, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, Options.textSize.val(), FIELD\_NOT\_CHANGED);

$s.css("font-size", Options.textSize.val() + "pt");

} ;

Options.textDelete[0].onclick = **function**() {

Config.remove(anchor);

$s.fadeOut(400, **function**() {

$s.remove();

});

};

$("#text-editor-shadow").fadeIn();

$("#texteditor-title").click();

});

**var** update\_draggable = **function**(x, y) {

$s.offset({ top: y, left: x });

};

**var** out\_blank = **function**(d\_off) {

**var** $b = $("#blank");

**var** b\_off = $b.offset();

**if**(d\_off.left < b\_off.left || d\_off.top < b\_off.top) {

**return** **true**;

}

**if**(d\_off.left + $s.width() > b\_off.left + $b.width() ||

d\_off.top + $s.height() > b\_off.top + $b.height()) {

**return** **true**;

}

**return** **false**;

};

**var** moveDown = **null**;

$s.on("mousedown", **function**(event) {

event.preventDefault();

moveDown = { x: event.offsetX, y: event.offsetY };

});

$(window).on("mousemove", **function**(event) {

**if**(moveDown != **null**) {

update\_draggable(event.pageX - moveDown.x, event.pageY - moveDown.y);

}

});

$(window).on("mouseup", **function**(event) {

**if**(moveDown) {

**if**(out\_blank($s.offset())) {

$s.fadeOut(400, **function**() {

$s.remove();

});

Config.remove(anchor);

**return**;

};

**var** offset = $s.offset();

Config.changeEditable(anchor, offset.left - $("#blank").offset().left, offset.top - $("#blank").offset().top, uid, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED, FIELD\_NOT\_CHANGED);

}

moveDown = **null**;

});

};

**js/admin.js**

var TYPE\_DIVS = {};

var CURRENT\_TYPE\_DIV;

\_\_LOADING = new ModalWindow();

\_\_LOADING.show();

var L = 0;

$(window).on("load", function() {

callRemoteFunction("types", "list", {}, function(response) {

for(var i in response.body) {

var type = response.body[i];

TYPE\_DIVS[type.id] = $("<div id='" + type.id + "' class='type-block'></div>");

}

});

loadUsers();

loadSegments();

var check = function() {

if(L > 0) {

setTimeout(check, 500);

} else {

\_\_LOADING.hide();

}

};

check();

});

function deleteSegment() {

var type = CURRENT\_TYPE\_DIV.attr("id");

var uid = $("#delete").attr("target");

callRemoteFunction("segments", "remove", { type: type, uid: uid }, function() {

$("#" + uid).remove();

});

};

function uploadSegment() {

var input = $("<input type='file' accept='image/\*'/>");

input.click();

input.on("change", function() {

var data = new FormData();

data.append("image", input[0].files[0]);

$.ajax({

type: "POST",

url: "/api/rfc.php?segments&upload&type=" + CURRENT\_TYPE\_DIV.attr("id"),

processData: false,

contentType: false,

data: data

}).done(function(d) {

var mw = new ModalWindow($("<div class='refresh-window'>Файл успешно загружен. Через 5 секунд страница будет перезагружена</div>"));

mw.show();

setTimeout(function() {

document.location.href += " ";

}, 5000);

});

});

}

function showType(id) {

CURRENT\_TYPE\_DIV = TYPE\_DIVS[id];

$("#types").append(CURRENT\_TYPE\_DIV);

CURRENT\_TYPE\_DIV.fadeIn();

}

function loadSegments() {

$("#segments .tool").outerHeight($("#type").outerHeight());

$("#type").outerWidth($("#controls").width() - $("#segments .tool").outerHeight() \* 2 - 20);

$("#controls").outerHeight($("#type").outerHeight());

$("#delete").fadeOut();

$("#upload").fadeOut();

callRemoteFunction("types", "list", {}, function(response) {

var types = response.body;

for(var i in types) {

$("#type").append($("<option value='" + types[i].id + "'>" + types[i].name + "</option>"));

$("#type").on("change", function() {

var id = $("#type").val();

$("#upload").fadeIn();

if(CURRENT\_TYPE\_DIV) {

CURRENT\_TYPE\_DIV.fadeOut(200, function() {

CURRENT\_TYPE\_DIV.remove();

showType(id);

});

} else {

showType(id);

}

});

var div = $("<div class='segment-list' id='" + types[i].id + "'></div>");

$("#types").append(div);

var width = $("#types").width();

var count = Math.ceil(Math.max(Math.min(width / 105, 7), 3));

console.log(count);

width = width / count;

if(types[i].type.indexOf("editable") == -1) {

(function(type, div) {

callRemoteFunction("segments", "list", { type: type.id, offset: 0 }, function(response) {

var segments = response.body;

for(var i in segments) {

(function(i){

var xhr = callRemoteFunction("segments", "load", { type: type.id, uid: segments[i], width: width }, function(d) {

console.log(xhr.response);

var blob = URL.createObjectURL(xhr.response);

var img = $("<img id='" + segments[i] + "' class='segment-item' src='" + blob + "'/>");

img[0].onclick = function() {

img.animate({ opacity: 0.5 }, 200);

$("#delete").fadeIn().attr("target", segments[i]);

};

div.append(img);

img.outerWidth(width - 20);

}, "blob");

})(i);

}

});

})(types[i], div);

}

TYPE\_DIVS[types[i].id] = div;

div.hide().remove();

}

showType($("#type").val());

});

}

function loadUsers() {

callRemoteFunction("user", "list", {}, function(response) {

var users = response.body;

for(var i in users) {

var row = $("<tr></tr>");

row.append($("<td class='cell'>" + users[i].id + "</td>"));

row.append($("<td class='cell'>" + users[i].login + "</td>"));

row.append($("<td class='cell'>" + users[i].surname + " " + users[i].name + "</td>"));

var cfgs = $("<td class='cell'></td>");

row.append(cfgs);

var upls = $("<td class='cell'></td>");

row.append(upls);

$("#users").append(row);

L+=2;

(function(row, id) {

callRemoteFunction("user", "configs", { id:id }, function(response) {

row.html(response.body.length);

L--;

});

})(cfgs, users[i].id);

(function(row, id) {

callRemoteFunction("user", "uploads", { id:id }, function(response) {

row.html(response.body.length + "</td>");

L--;

});

})(upls, users[i].id);

var ctrl = $("<td></td>");

row.append(ctrl);

if(users[i].id != currentID()) {

var del = $("<img src='/img/tools/delete.png' class='tool'/>");

ctrl.append(del);

(function(row, id) {

del.on("click", function() {

callRemoteFunction("user", "delete", { id:id }, function() {

row.remove();

});

});

})(row, users[i].id);

ctrl.append(del);

var admin = $("<img class='tool'/>");

if(users[i].role == "admin") {

admin.attr("src", '/img/no\_admin.png');

} else {

admin.attr("src", '/img/admin.png');

}

(function(ctr, user) {

ctr.on("click", function() {

callRemoteFunction("user", "make\_admin", { id: user.id, admin: (user.role == "user") }, function(d) {

if(user.role == "user") {

user.role = "admin";

ctr.attr("src", '/img/no\_admin.png');

} else {

user.role = "user";

ctr.attr("src", '/img/admin.png');

}

}, "text");

});

})(admin, users[i]);

ctrl.append(admin);

} else {

ctrl.html("Вы");

}

}

$("#users .tool").outerHeight($("#users td").outerHeight());

});

}

**js/auth.js**

var \_\_AUTH\_MODAL\_WINDOW;

function signin(login, password) {

\_\_AUTH\_MODAL\_WINDOW.show();

var R;

callRemoteFunction("auth", "signin", {

login: login,

password: password,

async: false

}, function(response) {

\_\_AUTH\_MODAL\_WINDOW.hide();

R = response;

\_\_forceUpdateControls();

});

return R;

}

function signup(login, password, name, surname) {

\_\_AUTH\_MODAL\_WINDOW.show();

var R;

callRemoteFunction("auth", "signup", {

login: login,

password: password,

name: name,

surname: surname,

async: false

}, function(response) {

\_\_AUTH\_MODAL\_WINDOW.hide();

R = response;

\_\_forceUpdateControls();

});

return R;

}

function signout() {

callRemoteFunction("auth", "signout", {}, function(response) {

if(~document.location.href.indexOf("user") || ~document.location.href.indexOf("admin")) {

document.location = "/";

}

\_\_forceUpdateControls();

});

}

function currentRole() {

var role;

callRemoteFunction("auth", "role", { async: false }, function(response) {

role = response.body;

});

return role;

}

function currentID() {

var role;

callRemoteFunction("auth", "id", { async: false }, function(response) {

role = response.body;

});

return role;

}

function \_\_forceUpdateControls() {

if(window["CONTROLS\_UPDATER"]) {

clearTimeout(CONTROLS\_UPDATER);

}

\_\_updateControls();

}

var \_\_lastUserRole;

function \_\_updateControls() {

var currentUserRole = currentRole();

if(!\_\_lastUserRole && currentUserRole == USER\_ROLE\_GUEST) {

return;

}

if(\_\_lastUserRole != currentUserRole) {

\_\_lastUserRole = currentUserRole;

var href1, href2, src1, src2;

if(currentUserRole == USER\_ROLE\_GUEST) {

href1 = "#signup";

href2 = "#signin";

src1 = "/img/user/sign\_up.png";

src2 = "/img/user/sign\_in.png";

} else {

href1 = "/user.php";

href2 = "#signout";

src1 = "/img/user/user.png";

src2 = "/img/user/sign\_out.png";

}

$("#sign-controls img").css("animation", "sign-control-rotation 0.3s linear");

setTimeout(function() {

$("#first-ctrl-link").attr("href", href1);

$("#second-ctrl-link").attr("href", href2);

$("#first-ctrl-img").attr("src", src1);

$("#second-ctrl-img").attr("src", src2);

setTimeout(function() {

$("#first-ctrl-img").css("animation", "");

$("#second-ctrl-img").css("animation", "");

}, 150);

}, 150);

}

CONTROLS\_UPDATER = setTimeout(\_\_updateControls, 2500);

};

$(window).on("load", function() {

\_\_AUTH\_MODAL\_WINDOW = new ModalWindow();

\_\_AUTH\_MODAL\_WINDOW.$container.css("z-index", 999);

\_\_updateControls();

})

**js/common.js**

String.prototype.replaceAll = function(a, b) {

var s = this.toString();

while(s.indexOf(a) != -1) {

s = s.replace(a, b);

}

return s;

};

ModalWindow = function(isLoading) {

var $c = this.$container = $("<div class='modal-container'></div>");

$(document.body).append($c);

var $w;

if(!isLoading) {

$w = $("<img src='/img/loading.png' class='modal-loading'/>");

} else {

$w = isLoading;

$c.on("click", function(event) {

if(event.target == $c[0]) {

$c.fadeOut();

}

})

}

this.$window = $w;

$c.append($w);

$c.hide();

};

ModalWindow.prototype.show = function() {

this.$container.fadeIn();

};

ModalWindow.prototype.hide = function() {

this.$container.fadeOut();

};

$(window).on("load", function() {

$("#content").css("margin-top", $("#header").outerHeight(true))

})

function callRemoteFunction(group, name, data, listener, type) {

data["method"] = name;

var async = true;

if("async" in data){

async = data["async"];

delete data["async"];

}

var XHR;

var request = {

type: "post",

url: "/api/rfc.php?" + group,

data: data,

async: async

}

if(data instanceof FormData) {

request.processData = false;

}

if(type == "blob") {

request.xhr = function() {

XHR = new XMLHttpRequest();

XHR.responseType = "blob";

return XHR;

}

} else {

request.dataType = type || "json"

}

$.ajax(request).always(listener);

return XHR;

}

if(document.location.hash) {

var method = document.location.hash.split("#")[1];

if(method == "signin" || method == "signup" || method == "signout") {

history.pushState(null, "", "http://mydiploma.ru/" + method);

switch(method) {

case "signin": showSignIn(); break;

case "signup": showSignUp(); break;

case "signout": signout(); break;

}

}

}

callRemoteFunction("tools", "const", { async: false }, function(response) {

for(var name in response.body) {

window[name] = response.body[name];

}

});

$("a[href^=#]").on("click", function(event) {

if(this.href.endsWith("down")) {

event.preventDefault();

var scroll = document.scrollingElement;

$(document.scrollingElement).animate({

scrollTop: scroll.scrollTop >= scroll.scrollTopMax / 2 ? 0 : scroll.scrollTopMax

});

}

var anchor = this.href.split("#")[1];

$(window).trigger("onanchorclick", [event, anchor]);

if(anchor.startsWith("sign")) {

event.preventDefault();

switch(anchor) {

case "signin": showSignIn(); break;

case "signup": showSignUp(); break;

case "signout": signout(); break;

}

}

});

var SIGN\_IN\_MODAL\_WINDOW;

function showSignIn() {

if(SIGN\_IN\_MODAL\_WINDOW) {

SIGN\_IN\_MODAL\_WINDOW.show();

return;

}

var $c = $("<div class='sign-container'></div>");

var window = SIGN\_IN\_MODAL\_WINDOW = new ModalWindow($c);

window.show();

var $t = $("<div class='sign-title'>Вход</div>");

$c.append($t);

var $m = $("<div class='sign-message'></div>");

$c.append($m);

var login = new NiceInput($c, "Логин", "text");

var passw = new NiceInput($c, "Пароль", "password");

var $s = $('<button class="sign-button">Ок</button>');

$s.on("click", function() {

var response = signin(login.$input.val(), passw.$input.val());

switch(response.code) {

case AUTH\_NO\_USER:

login.invalidate(true);

$m.html("Пользователь с таким логином не найден");

break;

case RFC\_FAIL:

passw.invalidate(true);

$m.html("Пароль не верен");

break;

case RFC\_SUCCESS:

$m.html("Вход выполнен");

window.hide();

break;

}

});

$c.append($s);

var $l = $("<button class='sign-button'>Отмена</button>");

$l.on("click", function() {

window.hide();

});

$c.append($l);

}

var SIGN\_UP\_MODAL\_WINDOW;

function showSignUp() {

if(SIGN\_UP\_MODAL\_WINDOW) {

SIGN\_UP\_MODAL\_WINDOW.show();

return;

}

var $c = $('<div class="sign-container"></div>');

var window = SIGN\_UP\_MODAL\_WINDOW = new ModalWindow($c);

window.show();

var $t = $('<div class="sign-title">Регистрация</div>');

$c.append($t);

var $m = $('<div class="sign-message"></div>');

$c.append($m);

var login = new NiceInput($c, "Логин", "text");

var passw = new NiceInput($c, "Пароль", "password");

var name = new NiceInput($c, "Имя", "text");

var surname = new NiceInput($c, "Фамилия", "text");

var $s = $('<button class="sign-button">Ок</button>');

$s.on("click", function() {

var response = signup(login.$input.val(), passw.$input.val(), name.$input.val(), surname.$input.val());

if(response.code == USER\_LOGIN\_BUSY) {

login.invalidate(true);

$m.html("Пользователь с таким логином существует");

} else if(response.code == RFC\_SUCCESS) {

$m.html("Регистрация выполнена");

window.hide();

} else {

switch(response.body.login) {

case USER\_INVALID:

$m.html("Логин может содержать символы латинского алавита, цифры, символ @ и символ подчеркивания<br/>");

login.invalidate(true);

break;

case USER\_INVALID\_SHORT:

$m.html("Логин должен содержать 4 и более символа<br/>");

login.invalidate(true);

break;

case USER\_INVALID\_LONG:

$m.html("Логин должен содержать менее 16 символов<br/>");

login.invalidate(true);

break;

}

switch(response.body.password) {

case USER\_INVALID:

$m.html($m.html() + "Пароль может содержать символы латинского алавита, цифры, символы @, #, $, %, &, +, - и символ подчеркивания<br/>");

passw.invalidate(true);

break;

case USER\_INVALID\_SHORT:

$m.html($m.html() + "Пароль должен содержать 8 и более символов<br/>");

passw.invalidate(true);

break;

case USER\_INVALID\_LONG:

$m.html($m.html() + "Пароль должен содержать менее 32 символов<br/>");

passw.invalidate(true);

break;

}

switch(response.body.name) {

case USER\_INVALID:

$m.html($m.html() + "Имя может содержать только символы русского и латинского алфавитов<br/>");

name.invalidate(true);

break;

case USER\_INVALID\_SHORT:

$m.html($m.html() + "Имя должно содержать более 1 символа<br/>");

name.invalidate(true);

break;

case USER\_INVALID\_LONG:

$m.html($m.html() + "Имя должно содержать менее 32 символов<br/>");

name.invalidate(true);

break;

}

switch(response.body.surname) {

case USER\_INVALID:

$m.html($m.html() + "Фамилия может содержать только символы русского и латинского алфавитов<br/>");

surname.invalidate(true);

break;

case USER\_INVALID\_SHORT:

$m.html($m.html() + "Фамилия должна содержать более 1 символа<br/>");

surname.invalidate(true);

break;

case USER\_INVALID\_LONG:

$m.html($m.html() + "Фамилия должна содержать менее 32 символов<br/>");

surname.invalidate(true);

break;

}

}

});

$c.append($s);

var $l = $("<button class='sign-button'>Отмена</button>");

$l.on("click", function(){

window.hide()

});

$c.append($l);

}

**js/index.js**

const SPLASH\_ANIMATION\_TIME = 1500;

const SPLASH\_TIMEOUT = 8500;

const welcome = [

"Только Вы и Ваш компьютер - никакой сложной техники и посредников. Все просто - нужно лишь начать",

"Вам больше не нужны рекламщики и дизайнеры, попробуйте сами - все очень просто и бесплатно",

"Один сервис - несколько компьютеров. Создайте аккаунт и сохраняйте свои работы на сервере автоматически. Это позволит Вам начать работу на одном комьютере и родолжить на другом",

"Мы готовы воплотить в жизнь все Ваши идеи - наши ресурсы, Ваши идеи, Ваши изделия",

"Проект родился в маленьком городе Пенза, но призван служить верным помощником большому миру"

];

var currentSplash = 0;

function showNextSplash() {

if(++currentSplash > 5) {

currentSplash = 1;

};

$("#splash-1").attr("src", "/img/splash/" + currentSplash + ".png");

$("#welcome").fadeOut(SPLASH\_ANIMATION\_TIME / 2, function(){

$("#welcome").html(welcome[currentSplash-1]);

$("#welcome").fadeIn(SPLASH\_ANIMATION\_TIME / 2);

});

$("#splash-2").fadeOut(SPLASH\_ANIMATION\_TIME, function() {

$("#splash-2").attr("src", "/img/splash/" + currentSplash + ".png").css("display", "flex");

setTimeout(showNextSplash, SPLASH\_TIMEOUT);

});

}

$(document).ready(function() {

showNextSplash();

});

var currentArrowDirection = "down";

$(window).on("scroll", function() {

var scroll = document.scrollingElement;

var animation = null, deg;

if(scroll.scrollTop > scroll.scrollTopMax / 2) {

if(currentArrowDirection == "down") {

animation = "scroll-arrow-up-rotation";

deg = 180;

currentArrowDirection = "up";

}

} else {

if(currentArrowDirection == "up") {

animation = "scroll-arrow-down-rotation";

deg = 0;

currentArrowDirection = "down";

}

}

if(animation != null) {

$("#down").css("animation", animation + " 0.300s linear");

setTimeout(function() {

$("#down").css("transform", "rotate(" + deg + "deg)");

}, 300);

}

});

**js/nice\_input.js**

NiceInput = function(parent, hint, type) {

var $c = this.$container = $("<div class='nice-input-container'></div>");

parent.append($c);

var $i = this.$input = $("<input class='nice-input-input' type='" + type + "'/>");

$c.append($i);

$c.height($i.outerHeight(true));

var $h = this.$hint = $("<span class='nice-input-hint'>" + hint + "</span>");

$h.css("top", -$i.outerHeight(true));

$c.append($h);

$h.on("click", function() {

$i.focus();

});

$i.on("focus", function(event) {

if($i.attr("readonly")) {

event.preventDefault();

} else {

$c.attr("focused", true);

$i.attr("focused", true);

$h.attr("focused", true);

}

$h.animate({

top: -$i.outerHeight(true) - $h.outerHeight(true) / 1.5,

left: $c.css("padding-left")

}, 200);

});

$i.on("blur", function() {

$c.removeAttr("focused");

$i.removeAttr("focused");

$h.removeAttr("focused");

if($i.val().length == 0) {

$h.animate({

top: -$i.outerHeight(true),

left: 0

}, 200);

}

});

var thiz = this;

$i.on("keyup", function() {

thiz.invalidate(false);

})

};

NiceInput.prototype.invalidate = function(inv) {

if(inv) {

this.$container.attr("invalid", true);

this.$input.attr("invalid", true);

this.$hint.attr("invalid", true);

} else {

this.$container.removeAttr("invalid");

this.$input.removeAttr("invalid");

this.$hint.removeAttr("invalid");

}

};

**js/user.js**

callRemoteFunction("user", "read", {}, function(data) {

if(data.code != RFC\_SUCCESS) return;

var user = data.body;

var login = new NiceInput($("#login"), "Логин", "text");

login.$container.css("display", "inline-block");

login.$input.val(user.login).attr("spellcheck", false);

login.$input.focus();

var id = new NiceInput($("#id"), "ID", "text");

id.$container.css("display", "inline-block");

id.$input.val(user.id).attr("spellcheck", false).attr("readonly", true);

id.$input.focus();

var name = new NiceInput($("#name"), "Имя", "text");

name.$container.css("display", "inline-block");

name.$input.val(user.name).attr("spellcheck", false);

name.$input.focus();

var surname = new NiceInput($("#surname"), "Фамилия", "text");

surname.$container.css("display", "inline-block");

surname.$input.val(user.surname).attr("spellcheck", false);

surname.$input.focus();

surname.$input.blur();

$("#save").on("click", function() {

save(login, name, surname);

});

});

var LOADING\_IMAGES = 0;

var LOCK = new ModalWindow();

LOCK.show();

function loadUserData() {

callRemoteFunction("config", "list", {}, function(data) {

if(data.code != RFC\_SUCCESS) return;

var cfgs = data.body;

if(cfgs.length == 0) {

$("#works-images").append("У Вас еще нет ниодной работы").width("100%");

} else {

LOADING\_IMAGES += cfgs.length;

}

for(var i in cfgs) {

viewWork(cfgs[i]);

}

});

callRemoteFunction("user", "uploads", {}, function(data) {

if(data.code != RFC\_SUCCESS) return;

var upls = data.body;

if(upls.length == 0) {

$("#uploads-images").append("Вы не загружали файлы").width("100%");

} else {

LOADING\_IMAGES += upls.length;

}

for(var i in upls) {

viewWork(upls[i]);

}

});

}

var THUMB\_WIDTH, THUMB\_HEIGHT;

$(window).on("load", function() {

var width = $("#works").outerWidth();

count = Math.min(Math.max(3, Math.floor(width / 210)), 7);

THUMB\_WIDTH = width / count;

THUMB\_HEIGHT = THUMB\_WIDTH \* 297 / 210;

$("#works-images").height(THUMB\_HEIGHT);

$("#uploads-images").height(THUMB\_HEIGHT);

loadUserData();

var check = function() {

if(LOADING\_IMAGES == 0) {

LOCK.hide();

} else {

setTimeout(check, 500);

}

};

check();

});

function viewWork(cfg) {

var xhr = callRemoteFunction("config", "draw", { uid: cfg, width: THUMB\_WIDTH, height: THUMB\_HEIGHT }, function(data) {

var blob = URL.createObjectURL(xhr.response);

var img = $("<img src='" + blob + "'/>");

img.on("load", function() {

LOADING\_IMAGES--;

});

$("#works-images").append(img);

(function(img, uid){

img.on("click", function() {

document.location.href = "/workbench.php?" + uid;

});

})(img, cfg);

$("#works-images").outerWidth($("#works-images").width() + THUMB\_WIDTH + 10);

}, "blob");

}

function viewUpload(uid) {

var xhr = callRemoteFunction("user", "upload", { uid: uid, width: THUMB\_WIDTH, height: THUMB\_HEIGHT }, function(data) {

var blob = URL.createObjectURL(xhr.response);

var img = $("<img src='" + blob + "'/>");

img.on("load", function() {

LOADING\_IMAGES--;

});

$("#uploads-images").append(img);

$("#uploads-images").outerWidth($("#uploads-images").width() + THUMB\_WIDTH + 10);

}, "blob");

}

function save(login, name, surname) {

var error = $("#error");

callRemoteFunction("user", "save", { name: name.$input.val(), surname: surname.$input.val(), login: login.$input.val() }, function(response) {

if(response.code == RFC\_SUCCESS) {

error.html("Изменения сохранены");

setTimeout(function() {

error.fadeOut(400, function() {

error.html("");

error.fadeIn();

});

}, 600);

return;

}

switch(response.body.login) {

case USER\_INVALID:

error.html("Логин может содержать символы латинского алавита, цифры, символ @ и символ подчеркивания<br/>");

login.invalidate(true);

break;

case USER\_INVALID\_SHORT:

error.html("Логин должен содержать 4 и более символа<br/>");

login.invalidate(true);

break;

case USER\_INVALID\_LONG:

error.html("Логин должен содержать менее 16 символов<br/>");

login.invalidate(true);

break;

}

switch(response.body.name) {

case USER\_INVALID:

error.html(error.html() + "Имя может содержать только символы русского и латинского алфавитов<br/>");

name.invalidate(true);

break;

case USER\_INVALID\_SHORT:

error.html(error.html() + "Имя должно содержать более 1 символа<br/>");

name.invalidate(true);

break;

case USER\_INVALID\_LONG:

error.html(error.html() + "Имя должно содержать менее 32 символов<br/>");

name.invalidate(true);

break;

}

switch(response.body.surname) {

case USER\_INVALID:

error.html(error.html() + "Фамилия может содержать только символы русского и латинского алфавитов<br/>");

surname.invalidate(true);

break;

case USER\_INVALID\_SHORT:

error.html(error.html() + "Фамилия должна содержать более 1 символа<br/>");

surname.invalidate(true);

break;

case USER\_INVALID\_LONG:

error.html(error.html() + "Фамилия должна содержать менее 32 символов<br/>");

surname.invalidate(true);

break;

}

});

}

function deleteUser() {

var container = $('<div class="sign-container"></div>');

var window = new ModalWindow(container);

window.show();

var title = $('<div class="sign-title">Удаление</div>');

container.append(title);

var text = $("<div style='font-size:85%'>Вы уверены, что хотите удалить аккаунт? Отменить это действие будет не возможно</div>");

container.append(text);

var ok = $("<button class='sign-button'>Да</button>");

ok.on("click", function() {

\_\_AUTH\_MODAL\_WINDOW.show();

callRemoteFunction("user", "delete", {}, function(response) {

document.location.href = "/";

});

});

container.append(ok);

var cancel = $("<button class='sign-button'>Отмена</button>");

cancel.on("click", function() {

window.hide();

});

container.append(cancel);

}

function changePassword() {

var $c = $('<div class="sign-container"></div>');

var window = new ModalWindow($c);

window.show();

var title = $('<div class="sign-title">Изменение пароля</div>');

$c.append(title);

var $m = $('<div class="sign-message"></div>');

$c.append($m);

var oldPass = new NiceInput($c, "Старый пароль", "password");

var newPass = new NiceInput($c, "Новый пароль", "password");

var $s = $("<button class='sign-button'>OK</button>");

$c.append($s);

$s.on("click", function() {

$m.html("");

callRemoteFunction("user", "save", { old\_password: oldPass.$input.val(), new\_password: newPass.$input.val() }, function(response) {

if(response.code == RFC\_SUCCESS) {

var error = $("#error");

error.html("Изменения сохранены");

setTimeout(function() {

window.hide();

error.fadeOut(400, function() {

error.html("");

error.fadeIn();

});

}, 600);

return;

}

switch(response.body.password) {

case USER\_INVALID:

$m.html($m.html() + "Пароль может содержать символы латинского алавита, цифры, символы @, #, $, %, &, +, - и символ подчеркивания<br/>");

newPass.invalidate(true);

break;

case USER\_INVALID\_SHORT:

$m.html($m.html() + "Пароль должен содержать 8 и более символов<br/>");

newPass.invalidate(true);

break;

case USER\_INVALID\_LONG:

$m.html($m.html() + "Пароль должен содержать менее 32 символов<br/>");

newPass.invalidate(true);

break;

case ACCESS\_DENIED:

$m.html($m.html() + "Старый пароль введен неверно");

oldPass.invalidate(true);

break;

}

});

});

var $l = $("<button class='sign-button'>Отмена</button>");

$c.append($l);

$l.on("click", function(){

window.hide();

});

}

**js/workbench.js**

SEGMENT\_TYPES = [];

TYPE\_LISTS = [];

var LOADING\_VIEWS = 0;

var LOCK = new ModalWindow();

LOCK.show();

function loadData() {

callRemoteFunction("types", "list", {}, function(types) {

SEGMENT\_TYPES = types.body;

Config.load();

for(var i in SEGMENT\_TYPES) {

var type = SEGMENT\_TYPES[i];

var list = new TypeList(type);

list.appendTo($("#segments"));

TYPE\_LISTS.push(list);

}

console.log($("#segments").height());

TypeList.MAX\_HEIGHT = $("#segments").height() - $(".segment-container").outerHeight(true) \* (SEGMENT\_TYPES.length+1);

for(var i in TYPE\_LISTS) {

TYPE\_LISTS[i].load();

}

check = function() {

if(LOADING\_VIEWS == 0) {

LOCK.hide();

} else {

setTimeout(check, 500);

}

};

setTimeout(check, 500);

});

}

window.ondragstart = function() { return false; };

window.onbeforeunload = function(event) {

Config.save();

};

$(window).on("load", function() {

var $body = $(document.body);

var $header = $("#header");

var optionsWidth = $("#options").outerWidth(true);

var segmentsWidth = $("#segments").outerWidth(true);

var $content = $("#content");

var $bench = $("#bench");

var $blank = $("#blank");

$content

.css("margin-top", $header.outerHeight(true))

.children()

.outerHeight($("HTML").height() - $header.outerHeight(true));

$bench

.css("left", optionsWidth).width($body.outerWidth() - optionsWidth - segmentsWidth);

var scale = Math.min(Math.floor($bench.width() / 210), Math.floor($bench.height() / 297));

$blank

.width(scale \* 210).height(scale \* 297)

.css("margin-top", ($bench.height() - scale \* 297) / 2);

var picker = $("#bgcolor-picker");

var bgColorPicker = new ColorPicker(picker);

bgColorPicker.onColorPicked = function(color) {

Config.setBackground(color);

};

callRemoteFunction("segments", "list", { type: "text" }, function(data) {

data = data.body;

for(var i in data) {

$("#fonts").text($("#fonts").text() + " @font-face{ font-family: \"" + data[i].font + "\"; src: url(\"" + data[i].file + "\"); } ");

var item = $("<option value='" + JSON.stringify(data[i]) + "'>" + data[i].font + "</option>");

item.css("font-family", data[i].font);

$("#texteditor-font").append(item);

}

});

Options = {};

initExpandable($("#bgcolor-title"), $("#bgcolor-content"), function(){ return $("#bgcolor-container").width() }, Options);

Options.textColorPicker = new ColorPicker($("#texteditor-picker"));

Options.textValue = $("#texteditor-value");

Options.textSize = $("#texteditor-size");

Options.textFont = $("#texteditor-font");

Options.textDelete = $("#texteditor-delete");

$("#texteditor-submit").on("click", function() {

$("#texteditor-title").click();

Options.textColorPicker.onColorPicked = function(color) {};

Options.textFont[0].onchange = function() {};

Options.textValue[0].onkeyup = Options.textValue[0].onchange = function(event) {};

Options.textSize[0].onkeyup = Options.textSize[0].onchange = function(event) {};

Options.textDelete[0].onclick = function() {};

});

var ctrl = $("#texteditor-controls").width();

$("#texteditor-font").outerWidth(ctrl \* 0.6-14).outerHeight(35);

$("#texteditor-size").outerWidth(ctrl \* 0.4).outerHeight(35);

var H = $("#texteditor-content").height();

initExpandable($("#texteditor-title"), $("#texteditor-content"), function(){ return H; }, Options);

$("#texteditor-content").outerHeight(0);

loadData();

});

$(window).bind("onanchorclick", function(event, click, anchor) {

if(anchor == "new") {

click.preventDefault();

Config.save();

Config.create();

Config.save();

document.location.href = document.location.href + " ";

}

});

function download() {

var $c = $("<div class='downloading-window'></div>");

var window = new ModalWindow($c);

var $t = $("<div class='downloading-title'>Загрузка файла</div>");

$c.append($t);

var $o = $("<div class='downloading-options'></div>");

$c.append($o);

var $l = $("<span>Выберите качество: </span>");

$o.append($l);

var $s = $("<select class='downloading-quality'></select>");

$o.append($s);

$s.append("<option value='1920'>Full HD</option>");

$s.append("<option value='1440'>HD 1080</option>");

$s.append("<option value='720'>HD 720</option>");

$s.append("<option value='576'>SD 576</option>");

$s.append("<option value='480'>SD 480</option>");

var $m = $("<div>и дождитесь загрузки предпросмотра</div>");

$o.append($m);

var $d;

var $p = $("<img class='downloading-preview'/>");

$c.append($p);

$s[0].onchange = function() {

$m.fadeIn(200);

$d.fadeOut(200);

var height = $s.val();

var width = height \* 210 / 297;

var uid = Config.\_\_current.uid;

if(currentRole() == USER\_ROLE\_GUEST) {

uid = Config.\_\_export();

}

var xhr = callRemoteFunction("config", "draw", { uid: uid, width: width, height: height }, function(data) {

var blob = URL.createObjectURL(xhr.response);

$p.attr("src", blob);

//console.log(data);

}, "blob");

};

$p.on("load", function() {

$m.fadeOut(200);

$d.fadeIn(200);

$d.attr('href', $p.attr('src'));

});

var $d = $("<a download='download.png'><button class='downloading-button'>Скачать</button></a>");

$o.append($d);

window.show();

};

**admin.php**

<!doctype>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

<meta name="description" content="Сервис для создания дипломов, грамот, открыток, благодарственных писем">

<meta name="keywords" content="Дипломы, грамоты, письма, открытки, maker, diploma, charter, letter">

<link rel="stylesheet" href="css/common.css"/>

<link rel="stylesheet" href="css/admin.css"/>

<link rel="shortcut icon" href="img/favicon.ico"/>

<style id="fonts"></style>

</head>

<body>

<?php

include\_once "api/const.php";

include\_once "api/auth.php";

if(!checkAccess()):

?>

<div id="no-access">

<div>

<img src="/img/error.png"/>

<h1>У Вас нет доступа к этой информации</h1>

</div>

</div>

<?php else: ?>

<div id="header">

<a href="/"><img id="logo" src="/img/logo.png"/></a>

<div id="sign-controls">

<a id="first-ctrl-link" href="#signup"><img id="first-ctrl-img" alt="Sign up" src="/img/user/sign\_up.png"/></a>

<a id="second-ctrl-link" href="#signin"><img id="second-ctrl-img" alt="Sign in" src="/img/user/sign\_in.png"/></a>

</div>

</div>

<div id="content">

<table id="users">

<tr>

<th class="cell" style='border-radius: 0px; border-top-left-radius: 30px'>ID</th>

<th class="cell">Логин</th>

<th class="cell">Фамилия, имя</th>

<th class="cell">Кол-во работ</th>

<th class="cell">Кол-во файлов</th>

<th style='border-radius: 0px; border-top-right-radius: 30px'>Управление</th>

</tr>

</table>

<div id="segments">

<div id="controls">

<select id="type"></select>

<img class='tool' id='upload' src='/img/tools/upload.png' onclick='uploadSegment()'/>

<img class='tool' id='delete' src='/img/tools/delete.png' onclick='deleteSegment()'/>

</div>

<div id="types"></div>

</div>

</div>

<script src="js/jquery.js"></script>

<script src="js/nice\_input.js"></script>

<script src="js/auth.js"></script>

<script src="js/common.js"></script>

<script src="js/admin.js"></script>

<?php endif; ?>

</body>

</html>

**Index.html**

<!DOCtype html>

<html lang="ru">

<head>

<meta charset="utf-8">

<title>MyDiploma</title>

<meta name="description" content="Сервис для создания дипломов, грамот, открыток, благодарственных писем">

<meta name="keywords" content="Дипломы, грамоты, письма, открытки, maker, diploma, charter, letter">

<link rel="stylesheet" href="css/common.css"/>

<link rel="stylesheet" href="css/index.css"/>

<link rel="shortcut icon" href="img/favicon.ico"/>

</head>

<body>

<div id="controls">

<a href="/workbench.php"><img id="add" alt="Add" src="/img/tools/add.png"/></a>

<div id="sign-controls">

<a id="first-ctrl-link" href="#signup"><img id="first-ctrl-img" alt="Sign up" src="/img/user/sign\_up.png"/></a>

<a id="second-ctrl-link" href="#signin"><img id="second-ctrl-img" alt="Sign in" src="/img/user/sign\_in.png"/></a>

</div>

<a href="#down"><img id="down" alt="Move to page bottom" src="/img/arrow.png"/></a>

</div>

<div id="contacts">

<a href="https://vk.com/kirill.bogatikov"><img src="/img/contact/vk.png"/></a>

<a href="http://kirillbogatikov.github.io"><img src="/img/contact/web.png"/></a>

<a href="https://facebook.com/kirill.bogatikov"><img src="/img/contact/facebook.png"/></a>

<a href="https://t.me/kirillbogatikov"><img src="/img/contact/telegram.png"/></a>

<a href="mailto:kivi15penza@gmail.com"><img src="/img/contact/email.png"/></a>

</div>

<div id="landing">

<img id="splash-1" class="splash" src="/img/splash/1.png"/>

<img id="splash-2" class="splash" src="/img/splash/1.png"/>

<div id="splash-shadow" class="splash"></div>

<div id="logo">

<span style="font-size: 200%;font-family:Gagalin">MY</span>

<span style="margin-left: -25pt;font-family:Stylo">diploma</span>

</div>

<div id="welcome">Только Вы и Ваш компьютер - никакой сложной техники и посредников. Все просто - нужно лишь начать</div>

</div>

<div id="content">

<div class="section" id="service">

<h1>О сервисе</h1>

Данный сервис разработан с одной простой целью - представить миру мощный, простой и абсолютно бесплатный инструмент для создания полиграфических изделий прямо из дома.

С помощью этого сайта Вы сможете быстро и просто создать любую грамоту, диплом, визитку, письмо и многое другое...

</div>

<div class="section" id="service">

<h1>Инструкция</h1>

Использовать наш сервис очень просто. На данной странице Вы можете зарегистрироваться, войти в уже имеющийся аккаунт или сразу приступить к разметке документа. Плюсы наличия аккаунта очевидны -

Вы сможете сохранять все работы на нашем сервере, а затем изменять и сохранять в любой удобный момент на любом удобном устройсте. Все для Вас!<br />

Для разметки документа Вы можете использовать сразу несколько параметров: цвет бумаги, рамка, положение и размер различных изображений, Вы можете добавлять текст, менять его цвет, размер и шрифт.

</div>

<div class="section" id="author">

<h1>Об авторе</h1>

Данный проект разработан в 2019 году студентом Пензенского колледжа информационных и промышленных технологий в рамках курсового проекта

</div></div>

<script src="js/jquery.js"></script>

<script src="js/auth.js"></script>

<script src="js/nice\_input.js"></script>

<script src="js/common.js"></script>

<script src="js/index.js"></script>

</body>

</html>

**user.php**

<!doctype>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

<meta name="description" content="Сервис для создания дипломов, грамот, открыток, благодарственных писем">

<meta name="keywords" content="Дипломы, грамоты, письма, открытки, maker, diploma, charter, letter">

<link rel="stylesheet" href="css/common.css"/>

<link rel="stylesheet" href="css/user.css"/>

<link rel="shortcut icon" href="img/favicon.ico"/>

<style id="fonts"></style>

</head>

<body>

<?php

include\_once "api/const.php";

include\_once "api/auth.php";

if(!checkAccess(USER\_ROLE\_USER)):

?>

<div id="no-access">

<div>

<img src="/img/error.png"/>

<h1>У Вас нет доступа к этой информации</h1>

</div>

</div>

<?php else: ?>

<div id="header">

<a href="/"><img id="logo" src="/img/logo.png"/></a>

<div id="sign-controls">

<a id="first-ctrl-link" href="#signup"><img id="first-ctrl-img" alt="Sign up" src="/img/user/sign\_up.png"/></a>

<a id="second-ctrl-link" href="#signin"><img id="second-ctrl-img" alt="Sign in" src="/img/user/sign\_in.png"/></a>

</div>

<div id="loading-tools">

<a href="/workbench.php"><img id="add" alt="Add" src="/img/tools/add.png"/></a>

<img src="/img/tools/upload.png"/>

</div>

</div>

<div id="content">

<h1>Личная информация</h1>

<table id="user-info">

<tr>

<td id="login" width="50%"></td>

<td id="id" width="50%"></td>

</tr>

<tr>

<td id="name"></td>

<td id="surname"></td>

</tr>

<tr>

<td colspan="2" id="error"></td>

</tr>

<tr>

<td id="password"><button onclick="changePassword()">Сменить пароль</button></td>

<td><button id="save">Сохранить</button></td>

</tr>

<tr>

<td colspan='2'><button onclick="deleteUser()" id="delete">Удалить</button></td>

</tr>

</table>

<h1>Ваши работы</h1>

<div id="works-block">

<div id="works">

<div id="works-images"></div>

</div>

</div>

<h1>Ваши файлы</h1>

<div id="uploads-block">

<div id="uploads">

<div id="uploads-images"></div>

</div>

</div>

</div>

<script src="js/jquery.js"></script>

<script src="js/nice\_input.js"></script>

<script src="js/auth.js"></script>

<script src="js/common.js"></script>

<script src="js/user.js"></script>

<?php endif; ?>

</body>

</html>