**Приложение A**

Controllers/AdminController.php

<?php  
  
include\_once ROOT . '/models/Details.php';  
include\_once ROOT . '/models/ChpuFor.php';  
include\_once ROOT . '/models/Parameter.php';  
include\_once ROOT . '/models/Save.php';  
  
class AdminController  
{  
 public static function actionDetails($del = null)  
 {  
 $errors = array();  
 if (isset($del)) :  
 Details::delete($del);  
 endif;  
  
 if (isset($\_POST['add'])) :  
 if (isset($\_POST['name'])) :  
 if (Details::chekDetails($\_POST['name']) !== true):  
 $uploaddir = 'template/uploads/'; // Relative path under webroot  
 $filename = $\_FILES['file']['name'];  
 $ext = explode('.', $filename);  
 $name = Save::translit($\_POST['name']).'.'.array\_pop($ext);  
 $uploadfile = $uploaddir . basename($name);  
  
 if (move\_uploaded\_file($\_FILES['file']['tmp\_name'], $uploadfile)) {  
 echo "File is valid, and was successfully uploaded.\n";  
 Details::setDetails($\_POST['name'], basename($name));  
 } else {  
 echo "File uploading failed.\n";  
 }  
 else:  
 $errors[] = 'Такое наименование уже есть';  
 endif;  
 else:  
 $errors[] = 'Введите Наименование';  
 endif;  
 endif;  
  
 $details = Details::getList();  
  
 require\_once(ROOT . '/views/admin/details.php');  
 return true;  
 }  
  
 public static function actionFor($del = null)  
 {  
 $errors = array();  
 if (isset($del)) :  
 ChpuFor::delete($del);  
 endif;  
  
 if (!empty($\_POST['add'])):  
 if (!empty($\_POST['name']) and !empty($\_POST['code'])) {  
 if (ChpuFor::chekForbyCode($\_POST['code']) !== true) :  
 ChpuFor::setFor($\_POST['name'], $\_POST['code']);  
 else:  
 $errors[] = 'Цикл с таким кодом уже добавлен';  
 endif;  
 } else {  
 $errors[] = 'Одно из полей не заполнено';  
 }  
 endif;  
  
 $for = ChpuFor::getList();  
  
 require\_once(ROOT . '/views/admin/for.php');  
 return true;  
 }  
  
 public static function actionParameter($for, $del = null)  
 {  
 $errors = array();  
 if (isset($del)) :  
 Parameter::delete($del);  
 endif;  
  
 if (!empty($\_POST['add'])):  
 if (!empty($\_POST['name']) and !empty($\_POST['code']) and !empty($\_POST['ext'])) {  
 if (Parameter::checkByCode($for, $\_POST['code']) !== true and Parameter::checkByName($for, $\_POST['name']) !== true) :  
 Parameter::setParameter($\_POST['name'], $\_POST['code'], $\_POST['ext'], $for);  
 else:  
 $errors[] = 'Введные данные, совпадают с данными в базе данных';  
 endif;  
 } else {  
 $errors[] = 'Одно из полей не заполнено';  
 }  
 endif;  
  
 if (!empty($\_POST['update'])) {  
 if (!empty($\_POST['upd\_name']) and !empty($\_POST['upd\_code']) and !empty($\_POST['upd\_ext'])) {  
 Parameter::updateById($\_POST['upd\_id'], $\_POST['upd\_name'], $\_POST['upd\_code'], $\_POST['upd\_ext']);  
 } else {  
 $errors[] = 'Одно из полей не заполнено';  
 }  
 }  
  
 $parameter = Parameter::getListByForId($for);  
  
 require\_once(ROOT . '/views/admin/parameter.php');  
 return true;  
 }  
  
 public static function actionDetailsUpd($id){  
 if (!empty($\_POST['update'])) {  
 if (!empty($\_POST['upd\_name']) or empty($\_POST['upd\_file'])) {  
 $uploaddir = 'template/uploads/'; // Relative path under webroot  
 $filename = $\_FILES['upd\_file']['name'];  
 $ext = explode('.', $filename);  
 $name = Save::translit($\_POST['upd\_name']).'.'.array\_pop($ext);  
 $uploadfile = $uploaddir . basename($name);  
  
 if (move\_uploaded\_file($\_FILES['upd\_file']['tmp\_name'], $uploadfile)) {  
 echo "File is valid, and was successfully uploaded.\n";  
 } else {  
 echo "File uploading failed.\n";  
 }  
 Details::updateById($\_POST['upd\_name'], $name, $id);  
 } else {  
 $errors[] = 'Одно из полей не заполнено';  
 }  
 }  
  
 $details = Details::findById($id);  
  
  
 require\_once(ROOT . '/views/admin/detailsUpd.php');  
 return true;  
 }  
  
}

Controllers/ChpuController.php

<?php  
include\_once ROOT . '/models/Chpu.php';  
include\_once ROOT . '/models/ChpuFor.php';  
include\_once ROOT . '/models/Save.php';  
  
class ChpuController  
{  
  
 public function actionIndex()  
 {  
 if (isset($\_POST['submit'])) {  
 $text = '';  
 $textFull = '';  
 $maxCount = 0;  
 $sum = 0;  
 $arr\_item = array();  
 $for = Chpu::getParameter($\_POST['for']);  
 foreach ($for as $item) :  
 $arr\_item[$item['code']] = explode(';', trim($\_POST[$item['code']]));  
 $maxCount < count($arr\_item[$item['code']]) ? $maxCount = count($arr\_item[$item['code']]) : $maxCount = $maxCount;  
 if (isset($\_POST[$item['code']])) :  
 $sum = $sum + str\_replace(',', '.', $\_POST[$item['code']]) > 0 ? 1 : 0;  
 endif;  
 endforeach;  
 if ($sum) :  
 for ($i = 0; $i < $maxCount; $i++) :  
 foreach ($for as $item) :  
 if (!empty($arr\_item[$item['code']][$i])) {  
 if (!empty(ChpuFor::getType($\_POST['for']))) :  
 $textFull .= $arr\_item[$item['code']][$i] . ', ';  
 $text .= '<strong>' . substr($item['code'], 0, -1) . $i . '</strong>';  
 $text .= str\_replace(',', '.', trim($arr\_item[$item['code']][$i])) . ' ';  
 else:  
 $textFull .= $arr\_item[$item['code']][$i] . ', ';  
 $text .= '<strong>' . trim($item['code']) . '</strong>' . str\_replace(',', '.', trim($arr\_item[$item['code']][$i])) . ' ';  
 endif;  
 } else {  
 $textFull .= '0, ';  
 }  
 endforeach;  
 endfor;  
 Chpu::setSave($\_POST['details'], $\_POST['for'], $text, $textFull);  
 endif;  
 }  
  
 if (isset($\_POST['clear'])) {  
 Save::clear();  
 }  
  
 $details = Chpu::getDetails();  
 $save = Save::getSave();  
  
 if (isset($\_POST['save'])) {  
 Save::saveFile();  
 Save::downloadFile();  
 }  
  
 require\_once(ROOT . '/views/chpu/index.php');  
  
 return true;  
 }  
  
 public function actionView($id, $page = 1)  
 {  
 $model = Chpu::getParameter($id, $page);  
 require\_once(ROOT . '/views/chpu/view.php');  
  
 return true;  
 }  
  
 public function actionFor($id)  
 {  
 $details = Chpu::getDetails();  
 $detailsId = $id;  
 $operations = Chpu::getOperations();  
 require\_once(ROOT . '/views/chpu/for.php');  
  
 return true;  
 }  
  
 public function actionParameter($detailsId, $forId)  
 {  
 $model = Chpu::getParameter($forId);  
 require\_once(ROOT . '/views/chpu/parameter.php');  
  
 return true;  
 }  
  
 public function actionAdd()  
 {  
 print\_r($\_POST);  
 return true;  
 }  
  
 public function actionDetailsImg($id){  
 $img = Details::getImg($id);  
 echo $img;  
 return true;  
 }  
  
  
}

Models/Chpu.php

<?php  
  
include\_once ROOT . '/components/Db.php';  
  
class Chpu  
{  
 const SHOW\_BY\_DEFAULT = 2;  
  
 public static function getNewsList()  
 {  
 $db = Db::getConnection();  
 $result = $db->query('SELECT \* FROM `chpu\_operation` ORDER BY id DESC LIMIT 10');  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
  
 $i = 0;  
 while ($rows = $result->fetch()) {  
 $newsList[$i]['id'] = $rows['id'];  
 $newsList[$i]['id\_operation'] = $rows['id\_operation'];  
 $newsList[$i]['code'] = $rows['code'];  
 $newsList[$i]['name'] = $rows['name'];  
 $i++;  
 }  
  
 return $newsList;  
  
 }  
  
 public static function getNewsItemById($id)  
 {  
 $id = intval($id);  
 $db = Db::getConnection();  
 $result = $db->query('SELECT \* FROM `chpu\_operation` WHERE id = ' . $id);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
  
 $newsItem = $result->fetch();  
  
 return $newsItem;  
 }  
  
  
 public static function getDetails()  
 {  
 $db = Db::getConnection();  
  
 $result = $db->query('SELECT \* FROM `Details` ORDER BY id');  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
  
 $i = 0;  
 while ($rows = $result->fetch()) {  
 $list[$i]['id'] = $rows['id'];  
 $list[$i]['name'] = $rows['name'];  
 $i++;  
 }  
  
 return $list;  
 }  
  
 public static function getOperations()  
 {  
 $db = Db::getConnection();  
 $result = $db->query('SELECT \* FROM `chpu\_for` ORDER BY id');  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
  
 $i = 0;  
 while ($rows = $result->fetch()) {  
 $list[$i]['id'] = $rows['id'];  
 $list[$i]['code'] = $rows['code'];  
 $list[$i]['name'] = $rows['name'];  
 $i++;  
 }  
  
 return $list;  
 }  
  
 public static function getParameter($id)  
 {  
 $db = Db::getConnection();  
 $id = intval($id);  
  
 $sql = "SELECT \* FROM `parameter` WHERE id\_for = :id\_for";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':id\_for', $id);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
  
 $i = 0;  
 while ($rows = $result->fetch()) {  
 $list[$i]['id'] = $rows['id'];  
 $list[$i]['code'] = $rows['code'];  
 $list[$i]['name'] = $rows['name'];  
 $list[$i]['ext'] = $rows['ext'];  
 $i++;  
 }  
  
 return $list;  
 }  
  
 public static function setSave($detailsId, $forId, $text, $textFull)  
 {  
 $db = Db::getConnection();  
 $detailsId = intval($detailsId);  
 $forId = intval($forId);  
  
 $sql = "INSERT INTO save (id\_details, id\_for, text, type, id\_save) "  
 . "VALUES (:id\_details, :id\_for, :text, NULL, NULL)";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':id\_details', $detailsId, PDO::PARAM\_INT);  
 $result->bindParam(':id\_for', $forId, PDO::PARAM\_INT);  
 $result->bindParam(':text', $text, PDO::PARAM\_STR);  
  
 $result->execute();  
 $lastId = $db->lastInsertId();  
  
 $sql = "INSERT INTO save (id\_details, id\_for, text, type, id\_save) "  
 . "VALUES (:id\_details, :id\_for, :text, 1, :id\_save)";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':id\_details', $detailsId, PDO::PARAM\_INT);  
 $result->bindParam(':id\_for', $forId, PDO::PARAM\_INT);  
 $result->bindParam(':text', $textFull, PDO::PARAM\_STR);  
 $result->bindParam(':id\_save', $lastId, PDO::PARAM\_INT);  
  
 return $result->execute();  
 }  
  
  
 public static function getDetailsName($id)  
 {  
  
 $db = Db::getConnection();  
 $id = intval($id);  
  
 $sql = "SELECT name FROM `Details` WHERE id = :id";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':id', $id);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
 $newsItem = $result->fetch();  
  
 return $newsItem['name'];  
 }  
  
 public static function getForName($id)  
 {  
 $db = Db::getConnection();  
 $id = intval($id);  
  
 $sql = "SELECT name FROM `chpu\_for` WHERE id = :id";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':id', $id);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
 $newsItem = $result->fetch();  
  
 return $newsItem['name'];  
 }  
}

Models/ChpuFor.php

<?php  
  
include\_once ROOT . '/components/Db.php';  
  
class ChpuFor  
{  
 const STATUS\_FOR\_SERIAL = '1';  
  
 private static function findModel($id){  
 $db = Db::getConnection();  
 $id = intval($id);  
  
 $sql = "SELECT \* FROM `chpu\_for` WHERE id = :id";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':id', $id);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
 $model = $result->fetch();  
 return $model;  
 }  
  
 public static function getCode($id){  
 $model = self::findModel($id);  
 return $model['code'];  
 }  
 public static function getName($id){  
 $model = self::findModel($id);  
 return $model['name'];  
 }  
 public static function getType($id){  
 $model = self::findModel($id);  
 return $model['type'];  
 }  
 public static function getCount(){  
 $db = Db::getConnection();  
  
 $sql = "SELECT id FROM `chpu\_for`";  
 $result = $db->prepare($sql);  
 $result->execute();  
  
 return $result->rowCount();  
 }  
  
  
 public static function getList()  
 {  
 $db = Db::getConnection();  
  
 $result = $db->query('SELECT \* FROM `chpu\_for` ORDER BY id');  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
  
 $i = 0;  
 while ($rows = $result->fetch()) {  
 $list[$i]['id'] = $rows['id'];  
 $list[$i]['name'] = $rows['name'];  
 $list[$i]['code'] = $rows['code'];  
 $i++;  
 }  
  
 return $list;  
 }  
  
 /\*  
 \* Добавляем деталь  
 \*/  
 public static function setFor($name, $code){  
 $db = Db::getConnection();  
 $sql = "INSERT INTO `chpu\_for` (name, code, type) VALUES (:name, :code, NULL )";  
 $result = $db->prepare($sql);  
 $result->bindParam(':name', $name);  
 $result->bindParam(':code', $code);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
 }  
  
 /\*  
 \* Проверка на наличие такой детали в базе  
 \*/  
 public static function chekForbyCode($code){  
 $db = Db::getConnection();  
  
 $sql = "SELECT COUNT(\*) FROM `chpu\_for` WHERE code = :code";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':code', $code);  
 $result->execute();  
 return $result->fetchColumn(0) > 0 ? true : false;  
 }  
  
 /\*  
 \* Удаялем элемент  
 \*/  
 public static function delete($id){  
 $db = Db::getConnection();  
 $sql = "DELETE FROM `chpu\_for` WHERE id = :id";  
 $stmt = $db->prepare($sql);  
 $stmt->bindParam(':id', $id, PDO::PARAM\_INT);  
 $stmt->execute();  
 }  
  
}

Models/Details.php

<?php  
  
include\_once ROOT . '/components/Db.php';  
  
class Details  
{  
 public static function findById($id){  
 $db = Db::getConnection();  
 $id = intval($id);  
  
 $sql = "SELECT \* FROM `Details` WHERE id = :id";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':id', $id);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
 $model = $result->fetch();  
 return $model;  
 }  
  
  
 public static function getName($id){  
 $model = self::findById($id);  
 return $model['name'];  
 }  
  
 public static function getImg($id){  
 $model = self::findById($id);  
 return $model['img'];  
 }  
  
 public static function getCount(){  
 $db = Db::getConnection();  
  
 $sql = "SELECT id FROM `Details`";  
 $result = $db->prepare($sql);  
 $result->execute();  
  
 return $result->rowCount();  
 }  
  
 public static function getList()  
 {  
 $db = Db::getConnection();  
  
 $result = $db->query('SELECT \* FROM `Details` ORDER BY id');  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
  
 $i = 0;  
 while ($rows = $result->fetch()) {  
 $list[$i]['id'] = $rows['id'];  
 $list[$i]['name'] = $rows['name'];  
 $list[$i]['img'] = $rows['img'];  
 $i++;  
 }  
  
 return $list;  
 }  
  
 /\*  
 \* Добавляем деталь  
 \*/  
 public static function setDetails($name, $uploadfile){  
 $db = Db::getConnection();  
 $sql = "INSERT INTO `Details` (name, img) VALUES (:name, :img)";  
 $result = $db->prepare($sql);  
 $result->bindParam(':name', $name);  
 $result->bindParam(':img', $uploadfile);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
 }  
  
 /\*  
 \* Проверка на наличие такой детали в базе  
 \*/  
 public static function chekDetails($name){  
 $db = Db::getConnection();  
  
 $sql = "SELECT id FROM `Details` WHERE name = :name";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':name', $name);  
 $result->execute();  
 return $result->fetchColumn(0) > 0 ? true : false;  
 }  
  
 /\*  
 \* Удаялем элемент  
 \*/  
 public static function delete($id){  
 $db = Db::getConnection();  
 $sql = "DELETE FROM `Details` WHERE id = :id";  
 $stmt = $db->prepare($sql);  
 $stmt->bindParam(':id', $id, PDO::PARAM\_INT);  
 $stmt->execute();  
 }  
  
 public static function updateById($name, $file, $id)  
 {  
 $db = Db::getConnection();  
 $stmt = $db->prepare("UPDATE Details set name = :name, img = :img where id=:id");  
 $stmt->bindParam(':name', $name, PDO::PARAM\_STR);  
 $stmt->bindParam(':img', $file, PDO::PARAM\_STR);  
 $stmt->bindParam(':id', $id, PDO::PARAM\_INT);  
 $stmt->execute();  
 }  
  
  
}

Models/Parameter.php

<?php  
  
include\_once ROOT . '/components/Db.php';  
  
class Parameter  
{  
 private static function findById($id)  
 {  
 $db = Db::getConnection();  
 $id = intval($id);  
  
 $sql = "SELECT \* FROM `parameter` WHERE id = :id";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':id', $id);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
 $model = $result->fetch();  
 return $model;  
 }  
  
 public static function getName($id)  
 {  
 $model = self::findById($id);  
 return $model['name'];  
 }  
  
 public static function getListByForId($id)  
 {  
 $db = Db::getConnection();  
 $id = intval($id);  
 $sql = "SELECT \* FROM `parameter` WHERE id\_for = :id";  
 $result = $db->prepare($sql);  
 $result->bindParam(':id', $id);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
  
 $i = 0;  
 while ($rows = $result->fetch()) {  
 $list[$i]['id'] = $rows['id'];  
 $list[$i]['code'] = $rows['code'];  
 $list[$i]['name'] = $rows['name'];  
 $list[$i]['ext'] = $rows['ext'];  
 $i++;  
 }  
  
 return $list;  
 }  
  
 /\*  
 \* Добавляем деталь  
 \*/  
 public static function setParameter($name, $code, $ext, $id\_for)  
 {  
 $db = Db::getConnection();  
 $sql = "INSERT INTO `parameter` (name, code, ext, id\_for) VALUES (:name, :code, :ext, :id\_for)";  
 $result = $db->prepare($sql);  
 $result->bindParam(':name', $name);  
 $result->bindParam(':code', $code);  
 $result->bindParam(':ext', $ext);  
 $result->bindParam(':id\_for', $id\_for);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
 }  
  
 /\*  
 \* Проверка на наличие такой детали в базе  
 \*/  
 public static function checkByCode($for, $code)  
 {  
 $db = Db::getConnection();  
  
 $sql = "SELECT id FROM `parameter` WHERE code = :code and id\_for = :id\_for";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':code', $code);  
 $result->bindParam(':id\_for', $for);  
 $result->execute();  
 return $result->fetchColumn(0) > 0 ? true : false;  
 }  
  
 /\*  
 \* Проверка на наличие такой детали в базе  
 \*/  
 public static function checkByName($for, $name)  
 {  
 $db = Db::getConnection();  
  
 $sql = "SELECT id FROM `parameter` WHERE name = :name and id\_for = :id\_for";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':name', $name);  
 $result->bindParam(':id\_for', $for);  
 $result->execute();  
 return $result->fetchColumn(0) > 0 ? true : false;  
 }  
  
 /\*  
 \* Удаялем элемент  
 \*/  
 public static function delete($id)  
 {  
 $db = Db::getConnection();  
 $sql = "DELETE FROM `parameter` WHERE id = :id";  
 $stmt = $db->prepare($sql);  
 $stmt->bindParam(':id', $id, PDO::PARAM\_INT);  
 $stmt->execute();  
 }  
  
 /\*  
 \* Обновляем занчние КОДА по АйДи  
 \*/  
 public static function updateById($id, $name, $code, $ext)  
 {  
 $db = Db::getConnection();  
 $stmt = $db->prepare("UPDATE parameter set code = :code, name = :name, ext = :ext where id=:id");  
 $stmt->bindParam(':id', $id, PDO::PARAM\_INT);  
 $stmt->bindParam(':code', $code, PDO::PARAM\_STR);  
 $stmt->bindParam(':name', $name, PDO::PARAM\_STR);  
 $stmt->bindParam(':ext', $ext, PDO::PARAM\_STR);  
 $stmt->execute();  
 }  
  
}

Models/Save.php

<?php  
  
include\_once ROOT . '/components/Db.php';  
include\_once ROOT . '/models/Details.php';  
  
class Save  
{  
 const IMAGE\_PATH = 'template/uploads/';  
 /\*  
 \* Сохраняем параметры цикла в таблицу  
 \*/  
 public static function getSave()  
 {  
 if (self::getCountSave()) {  
 $db = Db::getConnection();  
  
 $result = $db->query('SELECT \* FROM `save` WHERE type IS NULL ORDER BY id\_details');  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
  
 $i = 0;  
 while ($rows = $result->fetch()) {  
 $list[$i]['id'] = $rows['id'];  
 $list[$i]['id\_details'] = $rows['id\_details'];  
 $list[$i]['id\_for'] = $rows['id\_for'];  
 $list[$i]['text'] = $rows['text'];  
 $i++;  
 }  
 } else {  
 $list = null;  
 }  
  
 return $list;  
 }  
  
 public static function getSaveByType()  
 {  
 if (self::getCountSave()) {  
 $db = Db::getConnection();  
  
 $result = $db->query('SELECT \* FROM `save` WHERE type = 1 ORDER BY id\_details');  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
  
 $i = 0;  
 while ($rows = $result->fetch()) {  
 $list[$i]['id'] = $rows['id'];  
 $list[$i]['id\_details'] = $rows['id\_details'];  
 $list[$i]['id\_for'] = $rows['id\_for'];  
 $list[$i]['text'] = $rows['text'];  
 $i++;  
 }  
 } else {  
 $list = null;  
 }  
  
 return $list;  
 }  
  
 /\*  
 \* Получаем количество записей в таблице  
 \*/  
 public static function getCountSave()  
 {  
 $db = Db::getConnection();  
  
 $result = $db->query('SELECT count(id) AS count FROM `save`');  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $row = $result->fetch();  
  
 return $row['count'];  
 }  
  
 /\*  
 \* Запись файла  
 \*/  
 public static function saveFile(){  
 $fileName = 'save.txt';  
 $nameDEtails='';  
 unlink($fileName);  
 $create\_file = fopen($fileName, "w");  
 foreach (self::getSaveByType() as $item) {  
 fwrite(  
 $create\_file,  
 self::translit(ChpuFor::getName($item['id\_for']))  
 . "\r\n"  
 . trim(ChpuFor::getCode($item['id\_for']))  
 . '('  
 . str\_replace(array('<strong>', '</strong>'), '', trim(trim($item['text']), ','))  
 . ')'  
 . "\r\n" . "\r\n");  
 }  
 fclose($create\_file);  
 }  
  
 /\*  
 \* Скачиваем файл  
 \*/  
 public static function downloadFile($file = "save.txt")  
 {  
 header('Content-Description: File Transfer');  
 header('Content-Type: application/octet-stream');  
 header('Content-Disposition: attachment; filename=' . basename($file));  
 header('Content-Transfer-Encoding: binary');  
 header('Expires: 0');  
 header('Cache-Control: must-revalidate');  
 header('Pragma: public');  
 header('Content-Length: ' . filesize($file));  
 readfile($file);  
 exit;  
 }  
  
 /\*  
 \* Очистка таблицы  
 \*/  
 public static function clear(){  
 $db = Db::getConnection();  
 $result = $db->query('TRUNCATE TABLE `save`');  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $row = $result->fetch();  
 }  
  
 /\*  
 \* Удаляем строку  
 \*/  
 public static function delete($id){  
 $db = Db::getConnection();  
 $id = intval($id);  
  
 $sql = "DELETE FROM save WHERE id = :id";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':id', $id);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
  
 $sql = "DELETE FROM save WHERE id\_save = :id\_save";  
  
 $result = $db->prepare($sql);  
 $result->bindParam(':id\_save', $id);  
 $result->setFetchMode(PDO::FETCH\_ASSOC);  
 $result->execute();  
 }  
  
 /\*  
 \* Транслит  
 \*/  
 static function translit($s) {  
 $s = (string) $s; // преобразуем в строковое значение  
 $s = strip\_tags($s); // убираем HTML-теги  
 $s = str\_replace(array("\n", "\r"), " ", $s); // убираем перевод каретки  
 $s = preg\_replace("/\s+/", ' ', $s); // удаляем повторяющие пробелы  
 $s = trim($s); // убираем пробелы в начале и конце строки  
 $s = function\_exists('mb\_strtolower') ? mb\_strtolower($s) : strtolower($s); // переводим строку в нижний регистр (иногда надо задать локаль)  
 $s = strtr($s, array('а'=>'a','б'=>'b','в'=>'v','г'=>'g','д'=>'d','е'=>'e','ё'=>'e','ж'=>'j','з'=>'z','и'=>'i','й'=>'y','к'=>'k','л'=>'l','м'=>'m','н'=>'n','о'=>'o','п'=>'p','р'=>'r','с'=>'s','т'=>'t','у'=>'u','ф'=>'f','х'=>'h','ц'=>'c','ч'=>'ch','ш'=>'sh','щ'=>'shch','ы'=>'y','э'=>'e','ю'=>'yu','я'=>'ya','ъ'=>'','ь'=>''));  
 $s = preg\_replace("/[^0-9a-z-\_ ]/i", "", $s); // очищаем строку от недопустимых символов  
 $s = str\_replace(" ", "-", $s); // заменяем пробелы знаком минус  
 return $s; // возвращаем результат  
 }  
}

Views/chpu/index.php

<?php  
include ROOT . '/views/laouyts/header.php';  
include\_once ROOT.'/models/Chpu.php';  
include\_once ROOT.'/models/ChpuFor.php';  
?>  
 <div class="row col-lg-12">  
 <div class=""> </div>  
 <div class="col-lg-3">  
 <?php foreach ($details as $item) : ?>  
 <button type="button"  
 data-id="<?= $item['id']; ?>"  
 id="details<?= $item['id']; ?>"  
 class="btn btn-primary btn-block for\_list"><?= $item['name'] ?>  
 </button>  
 <?php endforeach; ?>  
 </div>  
 <div class="col-lg-3 for-content"></div>  
 <div class="col-lg-6 parameter-content"></div>  
 </div>  
 <div class="col-lg-12 row text-center" style="margin: 15px" >  
 <img src="" alt="" class="img-responsive text-center" id="details-img">  
 </div>  
  
<?php if ($save) : ?>  
 <div class="row col-lg-10 col-lg-offset-1" style="margin-top: 30px">  
 <div href="#" class="list-group-item active" style="color: white">  
 Сохранненая конфигурация  
 <div style="float: right; margin-top: -8px; margin-right: -12px">  
 <form action="" method="post">  
 <input type="submit" name="save" class="btn btn-default" value="Сохранить"/>  
 <input type="submit" name="clear" class="btn btn-warning" value="Очистить"/>  
 </form>  
 </div>  
 </div>  
 <table class="col-md-12 table table-bordered table-responsive" style="z-index: 10000; background: white" >  
 <?php foreach ($save as $item) : ?>  
 <tr id="for\_param\_<?=$item['id']?>">  
 <td class="col-md-2">  
 <?= Chpu::getDetailsName($item['id\_details']) ?>  
 </td>  
 <td class="col-md-4">  
 <?= ChpuFor::getCode($item['id\_for']).'('.ChpuFor::getName($item['id\_for']).')' ?>  
 </td>  
 <td class="col-md-6">  
 <?= $item['text'] ?>  
 </td>  
 <td>  
 <button type="button" class="btn btn-link btn-param-delete" data-id = "<?=$item['id']?>" title="Удалить запись"><span class="glyphicon glyphicon-trash"></span> </button>  
 </td>  
 </tr>  
 <?php endforeach; ?>  
 </table>  
 </div>  
<?php endif ?>  
  
<?php  
include ROOT . '/views/laouyts/footer.php'  
?>

Views/chpu/for.php

<?php  
include\_once ROOT.'/models/Details.php'  
?>  
  
 <a href="#" class="list-group-item active">  
 Циклы(<?= Details::getName($detailsId)?>)  
 </a>  
 <?php foreach ($operations as $item) : ?>  
 <a href="#" data-id="<?= $detailsId.';'.$item['id']; ?>" class="list-group-item parameter\_list"><?= $item['name'] ?></a>  
 <?php endforeach; ?>  
  
  
 <script>  
 $(document).ready(function () {  
 $(".parameter\_list").click(function () {  
 var id = $(this).attr("data-id");  
 arr = id.split(";");  
 $.post("/for/"+arr[0]+"/parameter/"+arr[1], {}, function (data) {  
 $(".parameter-content").html(data);  
 });  
  
 $(".parameter\_list").removeClass('alert-info');  
 $(this).addClass('alert-info')  
 });  
 });  
 </script>

Views/chpu/parameter.php

<form action="#" method="post">  
 <div href="#" class="list-group-item active" style="color: white">  
 Параметры  
 </div>  
 <input type="hidden" name="details" value="<?=$detailsId?>"/>  
 <input type="hidden" name="for" value="<?=$forId?>"/>  
 <?php foreach ($model as $item) : ?>  
 <div class="list-group-item parameter\_list" style="padding: 3px 5px">  
 <table>  
 <tr>  
 <td class="col-lg-10 col-sm-10 col-md-10">  
 <?= $item['name'] . '(' . $item['ext'] . ')' ?>  
 </td>  
 <td>  
 <input type="text" data-id="<?=$item['ext']?>" class="form-control parameter\_input" name="<?= $item['code'] ?>" value="0">  
 </td>  
 </tr>  
 </table>  
 </div>  
 <?php endforeach; ?>  
 <div class="list-group-item parameter\_list">  
 <div class="text-center">  
 <input type="submit" name="submit" class="btn btn-default" value="Добавить"/>  
 </div>  
 </div>  
</form>  
  
<div class="result">  
  
</div>

Views/chpu/view

<?php  
include ROOT.'/views/laouyts/main.php'  
?>  
  
<body>  
<div class="col-lg-8 col-lg-offset-2">  
  
 <nav class="navbar navbar-inverse" role="navigation">  
 <div class="container-fluid">  
 <div class="navbar-header ">  
 <a class="navbar-brand" href="/">Автоматизация загрузки параметров деталей на станках с ЧПУ "STR-30"</a>  
 </div>  
 </div>  
 </nav>  
 <div class="list-group">  
 <a href="#" class="list-group-item active">  
 Cras justo odio  
 </a>  
 <?php foreach ($model as $item) : ?>  
 <a href="#" class="list-group-item"><?= $item['id'] ?> - <?= $item['name'] ?> - <?= $item['code'] ?></a>  
 <?php endforeach; ?>  
 </div>  
</div>  
</body>  
  
<script async="" src="/template/js/bootstrap.min.js" type="text/javascript"></script>  
</html>

Views/laouyts/Footer.php

</div>  
  
</body>  
  
<script src="/template/js/jquery-3.2.1.min.js" type="text/javascript"></script>  
  
<script src="/template/js/bootstrap.min.js" type="text/javascript"></script>  
  
<script src="/template/js/main.js" type="text/javascript"></script>  
</html>

Views/laouyts/Header.php

<!doctype html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Document</title>  
 <link href="/template/css/bootstrap/css/bootstrap.min.css" rel="stylesheet">  
 <link href="/template/css/bootstrap/css/bootstrap-theme.css" rel="stylesheet">  
 <link href="/template/css/font-awesome.min.css" rel="stylesheet">  
 <link href="/template/css/site.css" rel="stylesheet">  
</head>  
<body>  
<div class=" col-lg-8 col-lg-offset-2 col-md-8 col-md-offset-2 col-sm-8 col-sm-offset-2" style="background: #F5FCFC; min-height: 800px; border: 1px dashed silver ">  
  
 <nav class="navbar navbar-inverse" role="navigation">  
  
 <div class="container-fluid">  
 <!-- Brand and toggle get grouped for better mobile display -->  
 <div class="navbar-header navbar-inverse">  
 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#bs-example-navbar-collapse-9">  
 <span class="sr-only">Toggle navigation</span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 </button>  
 <a class="navbar-brand" href="/">Автоматизация загрузки параметров деталей на станках с ЧПУ "STR-30"</a>  
 </div>  
  
 <!-- Collect the nav links, forms, and other content for toggling -->  
 <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-9" style="float: right;">  
 <ul class="nav navbar-nav">  
 <li><a href="/admin/details">Админ панель</a></li>  
 </ul>  
 </div><!-- /.navbar-collapse -->  
</div>  
 </nav>