# ЛАБОРАТОРНЯ РАБОТА №8 «Исследование архитектуры MVC приложения и возможностей обработки данных HTML-форм на стороне сервера с использованием языка PHP»

## Цель работы

Исследовать особенности построения приложений с использованием архитектуры MVC. Изучить основы синтаксиса PHP и принципы функционирования MVC приложения на стороне сервера. Приобрести практические навыки использования языка PHP для генерации HTML-кода и обработки HTML-форм в MVC приложении.

## Порядок выполнения работы

1. В соответствии с рекомендациями, представленными в п. 1.2.8 реструктурировать код персонального сайта на основе шаблона MVC.
2. Для страницы «Фотоальбом» реализовать хранение данных фото (подписей и имен файлов) в соответствующих константах модели Photo. Вывод таблицы, содержащей фото, реализовать в представлении с использованием операторов циклов.
3. Для страницы «Мои интересы» реализовать хранение данных (название категорий, интересов, описание интересов) в соответствующих константах модели Interest. Вывод меню ссылок и интересов реализовать в представлении с использованием операторов циклов.
4. Реализовать класс FormValidation, выполняющий валидацию данных форм, передаваемых на сторону сервера. Рекомендуемая структура класса:

* Rules – поле(массив), содержащее набор правил для проверки валидности данных;
* Errors – поле(массив), содержащее тексты ошибок возникших при проверке валидности данных;
* isNotEmpty(data) – метод проверки является ли значение data не пустым возвращает сообщение об ошибке, если таковая имеется;
* isInteger(data) – метод проверки является ли значение data строковым представлением целого числа – возвращает сообщение об ошибке, если таковая имеется;
* isLess(data, value) – метод проверки является ли значение data строковым представлением целого числа и не меньшим, чем value – возвращает сообщение об ошибке, если таковая имеется;
* isGreater(data, value) – метод проверки является ли значение data строковым представлением целого числа и не большим, чем value – возвращает сообщение об ошибке, если таковая имеется;
* isEmail(data) – метод проверки является ли значение data строковым представлением email – возвращает сообщение об ошибке, если таковая имеется;
* SetRule(field\_name, validator\_name) – метод, добавляющий в массив Rules проверку для поля field\_name типа validator\_name; Validate(post\_array) – метод выполняющий проверку элементов в массиве post\_array, в соответствии с правилами Rules и сохраняющий сообщения об ощибках в поле Errors;
* ShowErrors() – метод, выводящий все сообщения об ошибках из поля Errors в формате HTML.

1. С использованием разработанного класса реализовать валидацию форм «Контакт» и «Тест по дисциплине “…”».
2. Реализовать дочерний класс CustomFormValidation от класса FormValidation, дополнив его возможностью выполнения специализированной проверки формы «Тест по дисциплине» на стороне сервера.
3. Реализовать дочерний класс ResultsVerification от класса CustomFormValidation, дополнив его возможностью проверки правильности ответов, введенных пользователем на странице "Тест по дисциплине" (реализовать проверку правильности для вопросов с элементами ввода типа RadioButton, ComboBox или однострочный текст) и вывода результатов проверки пользователю.

## Ход работы

### Изучив язык программирования PHP персональный сайт, разработанный в прошлых лабораторных работах, был реструктурирован в соответствии с шаблоном MVC (Рисунок 1.1).

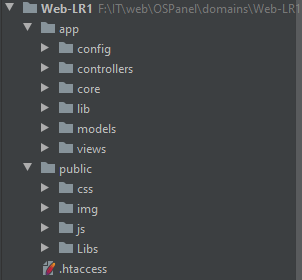


Рисунок 1.1 – Структура сайта

### Реализация модели фотоальбома представлена на рисунке 1.2. Полная реализации представлена в приложении А.

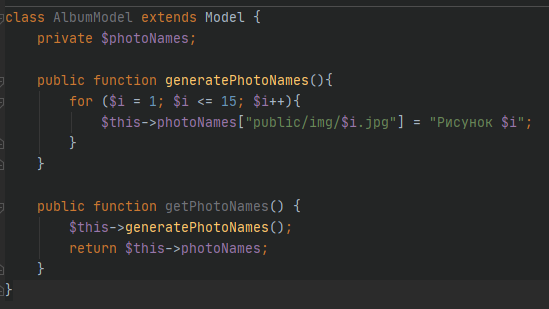


Рисунок 1.2 – Реализация модели фотоальбома

### Реализация модели «Мои интересы» представлена на рисунке 1.3. Полная реализации представлена в приложении А.

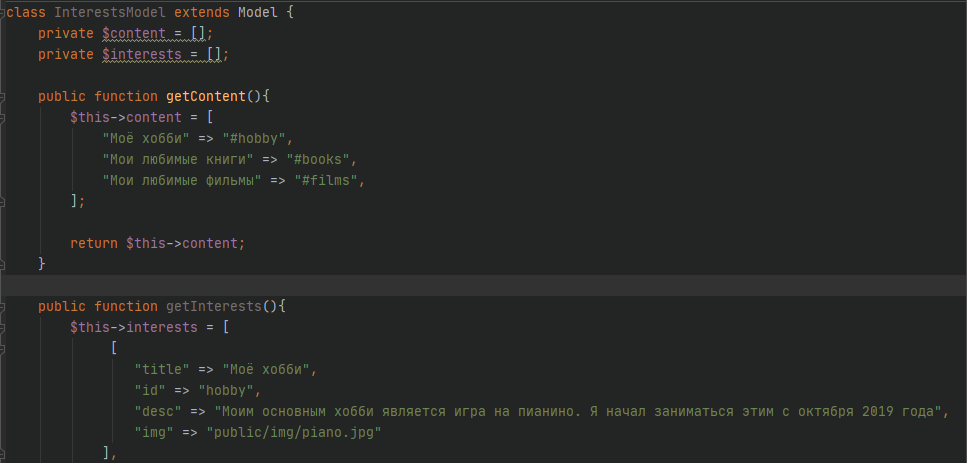


Рисунок 1.3 – Реализация модели «Мои интересы»

### Реализация класса FormValidation представлена на рисунке 1.4. Данный класс выполняет валидацию данных форм, передаваемых на сторону сервера. Полная реализации представлена в приложении А.

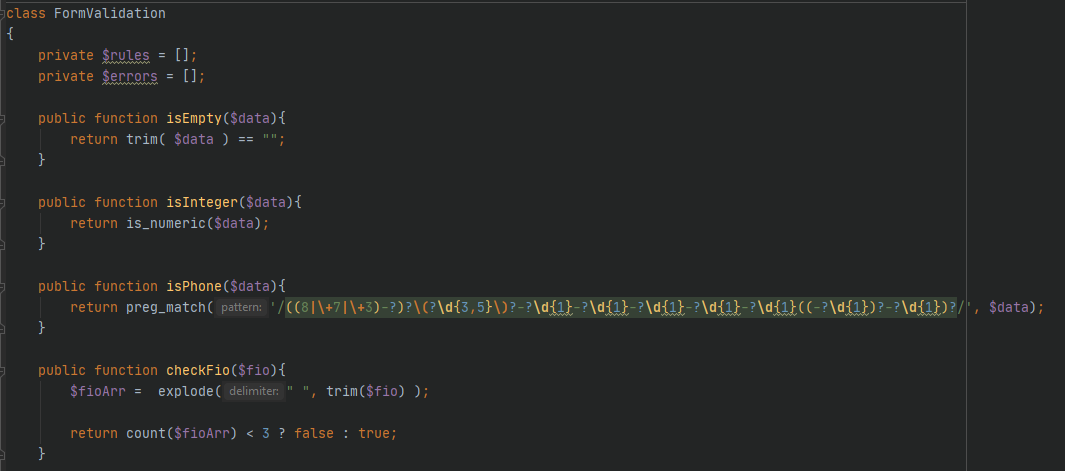


Рисунок 1.4 – Реализация класса FormValidation

### Реализация валидации страницы «Контакт» представлена на рисунке 1.5. Реализация валидации страницы «Тест по дисциплине» представлена на рисунке 1.6. Для каждого поля была добавлена своя валидация.



Рисунок 1.5 – Реализация валидации страницы «Контакт»

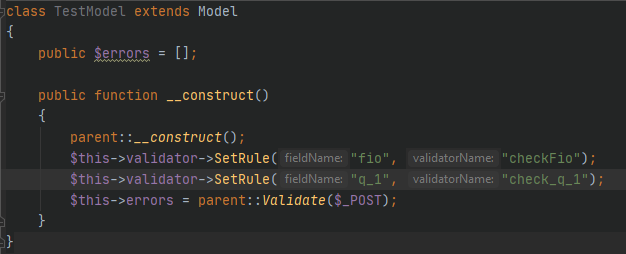


Рисунок 1.6 – Реализация валидации страницы «Тест по дисциплине»

### Реализация дочернего класса CustomFormValidation представлена на рисунке 1.7. В нём реализована функция проверки правильности ввода вопроса из теста.

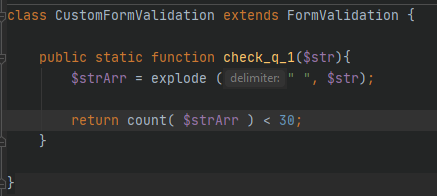


Рисунок 1.7 – Дочерний класс CustomFormValidation

# Вывод

В ходе выполнения данной лабораторной работы был изучен язык программирования PHP. Кроме этого был изучен шаблон проектирования MVC, с помощью которого был реструктурирован персональный сайт, написанный в прошлых лабораторных работах. Изучены способы построения приложений с использованием MVC, а также валидация форм.

# Приложение А

Файл config/routes.php

<?php  
  
return [  
 "" => [  
 "controller" => "main",  
 "action" => "index"  
 ],  
  
 "album" => [  
 "controller" => "album",  
 "action" => "index"  
 ],  
  
 "aboutMe" => [  
 "controller" => "aboutMe",  
 "action" => "index"  
 ],  
  
 "interests" => [  
 "controller" => "interests",  
 "action" => "index"  
 ],  
  
 "education" => [  
 "controller" => "education",  
 "action" => "index"  
 ],  
  
 "contacts" => [  
 "controller" => "contacts",  
 "action" => "index"  
 ],  
  
 "test" => [  
 "controller" => "test",  
 "action" => "index"  
 ],  
  
 "history" => [  
 "controller" => "history",  
 "action" => "index"  
 ],  
];

Файл controllers/AboutMeController.php

<?php  
  
namespace app\controllers;  
  
use app\core\Controller;  
  
class AboutMeController extends Controller {  
  
 public function indexAction(){  
 $vars = [  
 "img" => "public/img/eeEgrrRIERE.jpg",  
 "text" => "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do  
 eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut  
 enim ad minim veniam, quis nostrud exercitation ullamco laboris  
 nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in  
 reprehenderit in voluptate velit esse cillum dolore eu fugiat  
 nulla pariatur. Excepteur sint occaecat cupidatat non proident,  
 sunt in culpa qui officia deserunt mollit anim id est laborum.",  
 "menuIndex" => 2  
 ];  
  
 $this->view->render("Обо мне", $vars);  
 }  
  
}

Файл controllers/AlbumController.php

<?php  
  
namespace app\controllers;  
  
use app\core\Controller;  
  
class AlbumController extends Controller {  
 private $photoNames;  
  
 public function indexAction(){  
 $photoNames = $this->model->getPhotoNames();  
  
 $vars = [  
 "photoNames" => $photoNames,  
 "menuIndex" => 5  
 ];  
  
 $this->view->render("Фотоальбом", $vars);  
 }  
  
}

Файл controllers/ContactsController.php

<?php  
  
namespace app\controllers;  
  
use app\core\Controller;  
  
class ContactsController extends Controller {  
  
 public function indexAction(){  
 $vars = [  
 "menuIndex" => 6  
 ];  
  
 if( !empty( $\_POST ) ){  
 $vars["errors"] = $this->model->errors;  
 }  
  
 $this->view->render("Контакты", $vars);  
  
 }  
  
}

Файл controllers/EducationController.php

<?php  
  
namespace app\controllers;  
  
use app\core\Controller;  
  
class EducationController extends Controller {  
  
 public function indexAction(){  
 $vars = [  
 "menuIndex" => 4  
 ];  
  
 $this->view->render("Учёба", $vars);  
 }  
  
}

Файл controllers/HistoryController.php

<?php  
  
namespace app\controllers;  
  
use app\core\Controller;  
  
class HistoryController extends Controller {  
  
 public function indexAction(){  
 $vars = [  
 "menuIndex" => 8  
 ];  
  
 $this->view->render("История", $vars);  
 }  
  
}

Файл controllers/InterestsController.php

<?php  
  
namespace app\controllers;  
  
use app\core\Controller;  
  
class InterestsController extends Controller {  
 private $content = [];  
 private $interests = [];  
  
 public function indexAction(){  
 $this->content = $this->model->getContent();  
 $this->interests = $this->model->getInterests();  
  
 $vars = [  
 "menuIndex" => 3,  
 "content" => $this->content,  
 "interests" => $this->interests  
 ];  
  
 $this->view->render("Мои интересы", $vars);  
 }  
  
}

Файл controllers/MainController.php

<?php  
  
namespace app\controllers;  
  
use app\core\Controller;  
  
class MainController extends Controller  
{  
  
 public function indexAction(){  
 $vars = [  
 "fio" => "Шведенко Александр Сергеевич",  
 "img" => "public/img/eeEgrrRIERE.jpg",  
 "group" => "ПИ/б-18-1-о",  
 "desc" => "Лабораторная работа №1. Исследование возможностей языка разметки гипертекстов HTML и каскадных таблиц  
 стилей CSS",  
 "menuIndex" => 1  
 ];  
  
 $this->view->render("Главная", $vars);  
 }  
  
}

Файл controllers/TestController.php

<?php  
  
namespace app\controllers;  
  
use app\core\Controller;  
  
class TestController extends Controller {  
  
 public function indexAction(){  
 $vars = [  
 "menuIndex" => 7  
 ];  
  
 if( !empty( $\_POST ) ){  
 $vars["errors"] = $this->model->errors;  
 }  
  
 $this->view->render("Тест по дисциплине (БЖД)", $vars);  
 }  
  
}

Файл core/Controller.php

<?php  
  
namespace app\core;  
  
class Controller {  
  
 public $route;  
 public $view;  
 public $model;  
  
 public function \_\_construct($route) {  
 $this->route = $route;  
 $this->view = new View($route);  
 $this->model = $this->loadModel( $this->route['controller'] );  
 }  
  
 public function loadModel($name) {  
 $path = "app\models\\" . ucfirst($name) . "Model";  
  
 if( class\_exists($path) ){  
 return new $path;  
 }  
 }  
  
}

Файл core/Model.php

<?php  
  
namespace app\core;  
  
use app\models\validators\FormValidation;  
  
class Model {  
 public $validator;  
  
 public function \_\_construct()  
 {  
 $this->validator = new FormValidation();  
 }  
  
 public function Validate($postData){  
 return $this->validator->Validate($postData);  
 }  
}

Файл core/Router.php

<?php  
  
namespace app\core;  
  
class Router {  
 private $routes = [];  
 private $params = [];  
  
 public function \_\_construct() {  
 $arr = require "app/config/routes.php";  
 foreach ($arr as $key => $val){  
 $this->add($key, $val);  
 }  
 }  
  
 public function add($route, $params){  
 $route = "#^$route$#";  
 $this->routes[$route] = $params;  
 }  
  
 public function match(){  
 $url = trim($\_SERVER['REQUEST\_URI'], '/');  
  
 foreach ($this->routes as $route => $params) {  
 if( preg\_match($route, $url, $matches) ){  
 $this->params = $params;  
 return true;  
 }  
 }  
  
 return false;  
 }  
  
 public function run(){  
 if( $this->match() ){  
 $path = "app\controllers\\" . ucfirst($this->params['controller']) . "Controller";  
 if( class\_exists($path) ){  
 $action = $this->params['action'] . "Action";  
 if( method\_exists($path, $action) ){  
 $controller = new $path($this->params);  
 $controller->$action();  
 } else {  
 View::errorCode(404);  
 }  
 } else {  
 View::errorCode(404);  
 }  
  
 } else View::errorCode(404);  
 }  
}

Файл core/View.php

<?php  
  
namespace app\core;  
  
class View {  
  
 public $route;  
 public $path;  
 public $layout = "default";  
  
 public function \_\_construct($route) {  
 $this->route = $route;  
 $this->path = $route['controller'] . "/" . $route['action'];  
 }  
  
 public function render($title, $vars = []){  
 extract($vars);  
 $viewPage = "app/views/$this->path.php";  
 $menu = [  
 "/" => "Главная",  
 "/aboutMe" => "Обо мне",  
 "/interests" => [  
 "pageName" => "Мои интересы",  
 "/interests#hobby" => "Моё хобби",  
 "/interests#books" => "Мои любимые книги",  
 "/interests#films" => "Мои любимые фильмы",  
 ],  
 "/education" => "Учёба",  
 "/album" => "Фотоальбом",  
 "/contacts" => "Контакты",  
 "/test" => "Тест по дисциплине",  
 "/history" => "История",  
 ];  
  
 if( file\_exists( $viewPage ) ){  
 ob\_start();  
 require $viewPage;  
 $content = ob\_get\_clean();  
 require "app/views/layouts/$this->layout.php";  
 } else {  
 echo "Не найден вид $this->path";  
 }  
 }  
  
 public function redirect($url){  
 header("location: $url");  
 exit;  
 }  
  
 public static function errorCode($code){  
 http\_response\_code($code);  
 $errorPage = "app/views/errors/$code.php";  
 if( file\_exists( $errorPage ) ){  
 require $errorPage;  
 }  
 exit;  
 }  
}

Файл models/AlbumModel.php

<?php  
  
namespace app\models;  
  
use app\core\Model;  
  
class AlbumModel extends Model {  
 private $photoNames;  
  
 public function generatePhotoNames(){  
 for ($i = 1; $i <= 15; $i++){  
 $this->photoNames["public/img/$i.jpg"] = "Рисунок $i";  
 }  
 }  
  
 public function getPhotoNames() {  
 $this->generatePhotoNames();  
 return $this->photoNames;  
 }  
}

Файл models/ContactsModel.php

<?php  
  
  
namespace app\models;  
  
use app\core\Model;  
  
class ContactsModel extends Model  
{  
 public $errors = [];  
  
 public function \_\_construct()  
 {  
 parent::\_\_construct();  
 $this->validator->SetRule("fio", "checkFio");  
 $this->validator->SetRule("email", "isEmail");  
 $this->validator->SetRule("birthday", "isEmpty", "дата рождения");  
 $this->validator->SetRule("message", "isEmpty", "текст сообщения");  
 $this->validator->SetRule("phone", "isPhone");  
 $this->errors = parent::Validate($\_POST);  
 }  
}

Файл models/InterestsModel.php

<?php  
  
namespace app\models;  
  
use app\core\Model;  
  
class InterestsModel extends Model {  
 private $content = [];  
 private $interests = [];  
  
 public function getContent(){  
 $this->content = [  
 "Моё хобби" => "#hobby",  
 "Мои любимые книги" => "#books",  
 "Мои любимые фильмы" => "#films",  
 ];  
  
 return $this->content;  
 }  
  
 public function getInterests(){  
 $this->interests = [  
 [  
 "title" => "Моё хобби",  
 "id" => "hobby",  
 "desc" => "Моим основным хобби является игра на пианино. Я начал заниматься этим с октября 2019 года",  
 "img" => "public/img/piano.jpg"  
 ],  
 [  
 "title" => "Мои любимые книги",  
 "id" => "books",  
 "desc" => "Моими любимыми книгами являются:<ol>".  
 "<li>Мастер и Маргарита</li>".  
 "<li>Маленький принц</li>".  
 "</ol>",  
 "img" => "public/img/books.jpg"  
 ],  
 [  
 "title" => "Мои любимые фильмы",  
 "id" => "films",  
 "desc" => "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.\",",  
 "img" => "public/img/films.jpg"  
 ],  
 ];  
  
 return $this->interests;  
 }  
}

Файл models/TestModel.php

<?php  
  
  
namespace app\models;  
  
use app\core\Model;  
  
class TestModel extends Model  
{  
 public $errors = [];  
  
 public function \_\_construct()  
 {  
 parent::*\_\_construct*();  
 $this->validator->SetRule("fio", "checkFio");  
 $this->validator->SetRule("q\_1", "check\_q\_1");  
 $this->errors = parent::*Validate*($\_POST);  
 }  
}

Файл models/validators/FormValidation.php

<?php  
  
namespace app\models\validators;  
  
class FormValidation  
{  
 private $rules = [];  
 private $errors = [];  
  
 public function isEmpty($data){  
 return trim( $data ) == "";  
 }  
  
 public function isInteger($data){  
 return is\_numeric($data);  
 }  
  
 public function isPhone($data){  
 return preg\_match('/((8|\+7|\+3)-?)?\(?\d{3,5}\)?-?\d{1}-?\d{1}-?\d{1}-?\d{1}-?\d{1}((-?\d{1})?-?\d{1})?/', $data);  
 }  
  
 public function checkFio($fio){  
 $fioArr = explode(" ", trim($fio) );  
  
 return count($fioArr) < 3 ? false : true;  
 }  
  
 public function isEmail($data){  
 return filter\_var($data, *FILTER\_VALIDATE\_EMAIL*);  
 }  
  
 public function SetRule($fieldName, $validatorName, $errorText = ""){  
 $this->rules[$fieldName] = [$validatorName, $errorText];  
 }  
  
 public function Validate($postData){  
  
 if( !empty($postData) ){  
 foreach ($this->rules as $fieldName => [$validatorName, $errorText]){  
 switch ($validatorName){  
 case "isEmpty":  
 if( $this->isEmpty( $postData[$fieldName] ) ) $this->errors[] = "Поле <span>$errorText</span> должно быть заполнено!";  
 break;  
 case "isInteger":  
 if( !$this->isInteger( $postData[$fieldName] ) ) $this->errors[] = "Поле должно быть числом!";  
 break;  
 case "isEmail":  
 if( !$this->isEmail( $postData[$fieldName] ) ) $this->errors[] = "Неверно введён email!";  
 break;  
 case "check\_q\_1":  
 if( CustomFormValidation::*check\_q\_1*( $postData[$fieldName] ) ) $this->errors[] = "В первом вопросе должно быть минимум 30 слов!";  
 break;  
 case "isPhone":  
 if( !$this->isPhone( $postData[$fieldName] ) ) $this->errors[] = "Неверно введён телефон!";  
 break;  
 case "checkFio":  
 if( !$this->checkFio( $postData[$fieldName] ) ) $this->errors[] = "Введите фамилию имя отчество!";  
 break;  
 }  
 }  
 }  
  
 return $this->errors;  
 }  
  
}

Файл models/validators/FormValidation.php

<?php  
  
  
namespace app\models\validators;  
  
  
class CustomFormValidation extends FormValidation {  
  
 public static function check\_q\_1($str){  
 $strArr = explode (" ", $str);  
  
 return count( $strArr ) < 30;  
 }  
  
}

Файл views/aboutMe/index.php

<main>  
 <section class="page-name">  
 <span><?= $title ?></span>  
 </section>  
 <div class="content-wrapper">  
 <div class="person-info-wrapper grey-block">  
 <img src="<?= $img ?>" alt=""/>  
 <div class="person-info">  
 <p>  
 <?= $text ?>  
 </p>  
 </div>  
 </div>  
 </div>  
</main>

Файл views/album/index.php

<main>  
 <section class="page-name">  
 <span><?= $title ?></span>  
 </section>  
 <div class="wrapper\_album\_img">  
 <div id="black\_bg"></div>  
 <div id="album\_img">  
 <img src="" alt=""/>  
 <span></span>  
 <div class="next-img"><span>ᐳ</span></div>  
 <div class="prev-img"><span>ᐸ</span></div>  
 <div class="count-img">  
 Фото <span class="num-this-img">1</span> из  
 <span class="total-count"><?php echo count($photoNames); ?></span>  
 </div>  
 <div class="close">X</div>  
 </div>  
 </div>  
  
 <div class="content-wrapper grey-block album">  
 <?php  
 $i = 0;  
 foreach ($photoNames as $photoSrc => $name):  
 $i++;  
 ?>  
 <figure>  
 <img class='foto\_img' data-img-id=<?= $i - 1 ?> title='<?= $name ?>' src='<?= $photoSrc ?>'>  
 <figcaption><?= $name ?></figcaption>  
 </figure>  
 <?php endforeach; ?>  
 </div>  
</main>

Файл views/contacts/index.php

<main>  
 <section class="page-name">  
 <span><?= $title ?></span>  
 </section>  
  
 <div class="content-wrapper grey-block">  
 <form action="/contacts" class="contact-form" method="post">  
 <div class="modal-wrapper">  
 <div class="modal">  
 <p></p>  
 <div class="modal-btns">  
 <button id="yes\_modal" class="main-btn" type="button">  
 Да  
 </button>  
 <button id="no\_modal" class="main-btn" type="button">  
 Нет  
 </button>  
 </div>  
 </div>  
 <div class="overlay"></div>  
 </div>  
  
 <div class="tooltip" data-tooltip-text="Ваши Фамилия Имя Отчество">  
 <p style="margin-top: 0">ФИО</p>  
 <input  
 type="text"  
 name="fio"  
 id="fio"  
 data-error-text="Заполните ФИО!"  
 value="<?php if( !empty($\_POST["fio"]) ) echo $\_POST["fio"]; ?>"  
 />  
 <p class="error">Заполните ФИО!</p>  
 <div class="tooltip-block"></div>  
 </div>  
  
 <div class="tooltip" data-tooltip-text="Ваш пол">  
 <p>Пол</p>  
 <p class="radio">  
 <input  
 class="radio-input"  
 type="radio"  
 name="sex"  
 value="m"  
 checked  
 />М  
 </p>  
 <p class="radio">  
 <input class="radio-input" type="radio" name="sex" value="z"/>Ж  
 </p>  
 <div class="tooltip-block"></div>  
 </div>  
  
 <div class="tooltip" data-tooltip-text="Ваш возраст">  
 <p>Возраст</p>  
 <select class="" name="age">  
 <option value="">16</option>  
 <option value="">17</option>  
 <option value="">18</option>  
 <option value="">19</option>  
 <option value="">20</option>  
 </select>  
 <div class="tooltip-block"></div>  
 </div>  
  
 <div class="tooltip" data-tooltip-text="Ваша дата рождения">  
 <div class="calendar-wrapper">  
 <p>Дата рождения</p>  
 <input  
 id="birthday"  
 type="text"  
 name="birthday"  
 data-error-text="Заполните дату рождения"  
 value="<?php if( !empty($\_POST["birthday"]) ) echo $\_POST["birthday"]; ?>"  
 />  
 <p class="error"></p>  
 <div id="calendar">  
 <div class="calendar-header">  
 <div class="left-calendar-header">  
 <span id="today\_day\_of\_week">Понедельник 1</span>  
 </div>  
 <div class="right-calendar-header">  
 <select class="" name="" id="selectMonth">  
 <option value="">January</option>  
 <option value="">February</option>  
 <option value="">March</option>  
 <option value="">April</option>  
 <option value="">May</option>  
 <option value="">June</option>  
 <option value="">Jule</option>  
 <option value="">August</option>  
 <option value="">September</option>  
 <option value="">October</option>  
 <option value="">November</option>  
 <option value="">December</option>  
 </select>  
 <input id="selectYear" type="number" value="2020"/>  
 </div>  
 </div>  
 <hr/>  
 <div class="content">  
 <div id="days">  
 <table>  
 <tr class="not-hover">  
 <td>Mon</td>  
 <td>Tue</td>  
 <td>Wed</td>  
 <td>Thu</td>  
 <td>Fri</td>  
 <td>Sat</td>  
 <td>Sun</td>  
 </tr>  
 <tr></tr>  
 <tr></tr>  
 <tr></tr>  
 <tr></tr>  
 <tr></tr>  
 <tr></tr>  
 </table>  
 </div>  
 </div>  
 </div>  
 </div>  
 <div class="tooltip-block"></div>  
 </div>  
  
 <div class="tooltip" data-tooltip-text="Ваш Email">  
 <p>Email</p>  
 <input  
 id="email"  
 type="email"  
 name="email"  
 data-error-text="Заполните Email"  
 value="<?php if( !empty($\_POST["email"]) ) echo $\_POST["email"]; ?>"  
 />  
 <p class="error">Заполните Email!</p>  
 <div class="tooltip-block"></div>  
 </div>  
  
 <div class="tooltip" data-tooltip-text="Ваш текст сообщения">  
 <p>Текст сообщения</p>  
 <textarea  
 id="message"  
 name="message"  
 data-error-text="Введите сообщение"  
 ><?php if( !empty($\_POST["message"]) ) echo trim($\_POST["message"]); ?></textarea>  
 <p class="error">Введите сообщение!</p>  
 <div class="tooltip-block"></div>  
 </div>  
  
 <div class="tooltip" data-tooltip-text="Ваш телефон">  
 <p>Телефон</p>  
 <input  
 type="text"  
 name="phone"  
 id="phone"  
 data-error-text="Заполните Телефон"  
 value="<?php if( !empty($\_POST["phone"]) ) echo $\_POST["phone"]; else echo "+7"?>"  
 />  
 <p class="error">Заполните Телефон!</p>  
 <br/>  
 <div class="tooltip-block"></div>  
 </div>  
  
 <button  
 id="sendBtn"  
 class="main-btn btn-show-modal"  
 data-modal-text="Отправить данные?"  
 data-btn-callback="sendForm"  
 type="submit"  
 name="button"  
 >  
 Отправить  
 </button>  
 <button  
 id="resetBtn"  
 class="main-btn btn-show-modal"  
 data-modal-text="Стереть данные?"  
 data-btn-callback="resetForm"  
 type="reset"  
 name="button"  
 >  
 Очистить форму  
 </button>  
 </form>  
 <div class="notification">  
 <?php  
 if( isset($errors) ):  
 foreach ($errors as $error):  
 ?>  
 <div class="notification\_\_item notification\_\_item\_red">  
 <?= $error; ?>  
 </div>  
 <?php  
 endforeach;  
 if( count($errors) == 0 ):  
 ?>  
 <div class="notification\_\_item notification\_\_item\_green">  
 Данные успешно отправлены  
 </div>  
 <?php endif;endif; ?>  
 </div>  
 </div>  
</main>

Файл views/Education/index.php

<main>  
 <section class="page-name">  
 <span><?= $title ?></span>  
 </section>  
 <div class="content-wrapper">  
 <div class="grey-block education-page">  
 <p><span class="text-bold">Университет:</span> Севастопольский Государственный университет</p>  
 <p><span class="text-bold">Кафедра:</span> Информационные системы</p>  
 <table border="1">  
 <tr>  
 <th rowspan="3">№</th>  
 <th rowspan="3">Дисциплина</th>  
 <th colspan="12">Часов в неделю (Лекций, Лаб.раб, Практ. раб)</th>  
 </tr>  
 <tr>  
 <th colspan="6">1 курс</th>  
 <th colspan="6">2 курс</th>  
 </tr>  
 <tr>  
 <th colspan="3">1 сем</th>  
 <th colspan="3">2 сем</th>  
 <th colspan="3">3 сем</th>  
 <th colspan="3">4 сем</th>  
 </tr>  
 <tr>  
 <td>1</td>  
 <td>Экология</td>  
  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td>1</td>  
 <td>0</td>  
 <td>1</td>  
 <td></td>  
 <td></td>  
 <td></td>  
 </tr>  
 <tr>  
 <td>2</td>  
 <td>Высшая математика</td>  
  
 <td>3</td>  
 <td>0</td>  
 <td>3</td>  
 <td>3</td>  
 <td>0</td>  
 <td>3</td>  
 <td>2</td>  
 <td>0</td>  
 <td>2</td>  
 <td></td>  
 <td></td>  
 <td></td>  
 </tr>  
 <tr>  
 <td>3</td>  
 <td>Русский язык и культура речи</td>  
  
 <td>1</td>  
 <td>0</td>  
 <td>2</td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <tr>  
 <td>4</td>  
 <td>Основы дискретной математики</td>  
  
 <td>2</td>  
 <td>0</td>  
 <td>1</td>  
 <td>3</td>  
 <td>0</td>  
 <td>2</td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 </tr>  
 </tr>  
 <tr>  
 <td>5</td>  
 <td>Основы программирования и алгоритмические языки</td>  
  
 <td>3</td>  
 <td>2</td>  
 <td>0</td>  
 <td>3</td>  
 <td>3</td>  
 <td>0</td>  
 <td>0</td>  
 <td>0</td>  
 <td>1</td>  
 <td></td>  
 <td></td>  
 <td></td>  
 </tr>  
 <tr>  
 <td>6</td>  
 <td>Основы экологии</td>  
  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td>1</td>  
 <td>0</td>  
 <td>0</td>  
 <td></td>  
 <td></td>  
 <td></td>  
 </tr>  
 <tr>  
 <td>7</td>  
 <td>Теория вероятностей и математическая статистика</td>  
  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td>3</td>  
 <td>1</td>  
 <td>0</td>  
 <td></td>  
 <td></td>  
 <td></td>  
 </tr>  
 <tr>  
 <td>8</td>  
 <td>Физика</td>  
  
 <td>2</td>  
 <td>2</td>  
 <td>0</td>  
 <td>2</td>  
 <td>2</td>  
 <td>0</td>  
 <td>2</td>  
 <td>1</td>  
 <td>0</td>  
 <td></td>  
 <td></td>  
 <td></td>  
 </tr>  
 <tr>  
 <td>9</td>  
 <td>Основы электротехники и электроники</td>  
  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td>2</td>  
 <td>1</td>  
 <td>1</td>  
 <td></td>  
 <td></td>  
 <td></td>  
 </tr>  
 <tr>  
 <td>10</td>  
 <td>Численные методы в информатике</td>  
  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td>2</td>  
 <td>2</td>  
 <td>0</td>  
 <td>0</td>  
 <td>0</td>  
 <td>1</td>  
 </tr>  
 <tr>  
 <td>11</td>  
 <td>Методы исследования операций</td>  
  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td></td>  
 <td>1</td>  
 <td>1</td>  
 <td>0</td>  
 <td>2</td>  
 <td>1</td>  
 <td>1</td>  
 </tr>  
 </table>  
 </div>  
 </div>  
</main>

Файл views/errors/403.php

<p>  
 Страница 403

<a href="/">Вернуться на главную</a>  
</p>

Файл views/errors/404.php

<p>  
 Страница 404<br>  
 <a href="/">Вернуться на главную</a>  
</p>

Файл views/history/index.php

<main>  
 <section class="page-name">  
 <span><?= $title ?></span>  
 </section>  
 <div class="content-wrapper">  
 <div class="grey-block education-page">  
 <table id="history\_table">  
 <tr>  
 <td></td>  
 <td>Итория текущего сеанса</td>  
 <td>История за всё время</td>  
 </tr>  
 </table>  
 </div>  
 </div>  
</main>

Файл views/interests/index.php

<main>  
 <section class="page-name">  
 <span>Мои интересы</span>  
 </section>  
 <div class="content-wrapper">  
 <div class="grey-block content">  
 <span>Содержание</span>  
 <ul>  
 <?php foreach ( $content as $title => $link ): ?>  
 <a href="<?= $link ?>">  
 <li><?= $title ?></li>  
 </a>  
 <?php endforeach; ?>  
 </ul>  
 </div>  
  
 <?php foreach ( $interests as $interest ): ?>  
 <div id="<?= $interest['id'] ?>" class="parallax-wrapper">  
 <div class="parallax-img" style="background: url(<?= $interest['img'] ?>) no-repeat fixed;"></div>  
 <div class="parallax">  
 <div class="parallax-title"><?= $interest['title'] ?></div>  
 <hr class="parallax-hr">  
 <div class="parallax-content"><?= $interest['desc'] ?></div>  
 </div>  
 </div>  
 <?php endforeach; ?>  
 </div>  
</main>

Файл views/layouts/default.php

<!DOCTYPE html>  
<html lang="en" dir="ltr">  
<head>  
 <meta charset="utf-8">  
 <title>Персональный сайт Шведенко Александра. <?= $title ?></title>  
 <link rel="stylesheet" href="public/css/reset.css">  
 <link rel="stylesheet" href="public/css/main.css">  
 <script src="public/Libs/jquery.js"></script>  
 <script src="public/js/index.js"></script>  
</head>  
<body>  
<header>  
 <nav>  
 <ul id="menu">  
 <?php  
 $i = 0;  
 foreach ($menu as $link => $name):  
 $i++;  
 if( !is\_array($name) ){  
 ?>  
 <li>  
 <a href="<?= $link ?>" class="<?php if( !empty($menuIndex) && $menuIndex == $i ) echo "active"; ?>"><?= $name ?></a>  
 </li>  
 <?php  
 } else {  
 ?>  
 <li class="menu-item">  
 <a href="<?= $link ?>" class="<?php if( !empty($menuIndex) && $menuIndex == $i ) echo "active"; ?>"><?= $name['pageName'] ?></a>  
 <ul class="sub-menu">  
 <?php foreach ($name as $submenuLink => $submenuName): ?>  
 <?php if( $submenuLink == "pageName" ) continue; ?>  
 <li> <a href="<?= $submenuLink ?>"><?= $submenuName ?></a></li>  
 <?php endforeach; ?>  
 </ul>  
 </li>  
 <?php  
 }  
 ?>  
  
 <?php endforeach; ?>  
 </ul>  
 <div id="time">00:00:00 1 январь 1970</div>  
 </nav>  
</header>  
  
<?= $content ?>  
  
<footer>  
  
</footer>  
<?php if( $menuIndex == 6 ): ?>  
<script src="public/js/calendar.js"></script>  
<?php endif; ?>  
</body>  
</html>

Файл views/main/index.php

<main>  
 <section class="page-name">  
 <span><?= $title ?></span>  
 </section>  
 <div class="content-wrapper">  
 <div class="person-info-wrapper grey-block">  
 <img src="<?= $img ?>" alt="">  
 <div class="person-info">  
 <p><?= $fio ?></p>  
 <p><?= $group ?></p>  
 <p><?= $desc ?></p>  
 </div>  
 </div>  
 </div>  
</main>

Файл views/test/index.php

<main>  
 <section class="page-name">  
 <span><?= $title ?></span>  
 </section>  
 <div class="content-wrapper grey-block">  
 <form class="contact-form" action="/test" method="post">  
 <div class="modal-wrapper">  
 <div class="modal">  
 <p></p>  
 <div class="modal-btns">  
 <button id="yes\_modal" class="main-btn" type="button">  
 Да  
 </button>  
 <button id="no\_modal" class="main-btn" type="button">  
 Нет  
 </button>  
 </div>  
 </div>  
 <div class="overlay"></div>  
 </div>  
  
 <div id="tooltip"></div>  
  
 <div class="tooltip" data-tooltip-text="Ваши Фамилия Имя Отчество">  
 <p style="margin-top: 0">ФИО</p>  
 <input type="text"  
 id="fio"  
 name="fio"  
 value="<?php if( !empty($\_POST["fio"]) ) echo $\_POST["fio"]; ?>"  
 />  
 <p class="error">Заполните ФИО!</p>  
 </div>  
  
 <div class="tooltip" data-tooltip-text="Ваша группа">  
 <p>Группа</p>  
 <select class="" name="age">  
 <optgroup label="1 курс">  
 <option value="">ПИ/б-20-1-о</option>  
 </optgroup>  
 <optgroup label="2 курс">  
 <option value="">ПИ/б-19-1-о</option>  
 </optgroup>  
 <optgroup label="3 курс">  
 <option value="">ПИ/б-18-1-о</option>  
 </optgroup>  
 <optgroup label="4 курс">  
 <option value="">ПИ/б-17-1-о</option>  
 </optgroup>  
 </select>  
 </div>  
  
 <div class="tooltip" data-tooltip-text="Первый вопрос">  
 <p>1. Что делать при землетрясении?</p>  
 <textarea type="text" id="q\_1" name="q\_1"><?php if( !empty($\_POST["q\_1"]) ) echo $\_POST["q\_1"]; ?></textarea>  
 <p class="error"></p>  
 </div>  
  
 <div class="tooltip" data-tooltip-text="Второй вопрос">  
 <p>2. В каком году произошёл взрыв ЧАЭС?</p>  
 <p>  
 <input  
 class="radio-input"  
 type="radio"  
 name="q\_2"  
 value=""  
 checked  
 />1975  
 </p>  
 <p>  
 <input class="radio-input" type="radio" name="q\_2" value=""/>1986  
 </p>  
 <p>  
 <input class="radio-input" type="radio" name="q\_2" value=""/>1977  
 </p>  
 </div>  
  
 <div class="tooltip" data-tooltip-text="Третий вопрос">  
 <p>3. Что нужно делать при пожаре?</p>  
 <select name="q\_3">  
 <option value="">Вызвать пожарных</option>  
 <option value="">Залезть под кровать</option>  
 </select>  
 </div>  
  
 <button  
 id="sendBtn"  
 class="main-btn btn-show-modal"  
 data-modal-text="Отправить данные?"  
 data-btn-callback="checkTestForm"  
 type="submit"  
 name="button"  
 >  
 Отправить  
 </button>  
 <button  
 id="resetBtn"  
 class="main-btn btn-show-modal"  
 data-modal-text="Стереть данные?"  
 data-btn-callback="resetForm"  
 type="reset"  
 name="button"  
 >  
 Очистить форму  
 </button>  
 </form>  
 <div class="notification">  
 <?php  
 if( isset($errors) ):  
 foreach ($errors as $error):  
 ?>  
 <div class="notification\_\_item notification\_\_item\_red">  
 <?= $error; ?>  
 </div>  
 <?php  
 endforeach;  
 if( count($errors) == 0 ):  
 ?>  
 <div class="notification\_\_item notification\_\_item\_green">  
 Данные успешно отправлены  
 </div>  
 <?php endif;endif; ?>  
 </div>  
 </div>  
</main>

Файл .htaccess

RewriteEngine On  
RewriteCond %{REQUEST\_FILENAME} !-f  
RewriteCond %{REQUEST\_FILENAME} !-d  
RewriteRule ^(.\*)$ index.php