Министерство науки и высшего образования Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

По дисциплине: «Веб-технологии»

Лабораторная работа № 1

«Исследование архитектуры MVC приложения и возможностей обработки данных HTML-форм на стороне сервера с использованием языка PHP»

Выполнила

ст. гр. ИС/б-17-2-о

Кайда Е.Н.

Проверил:

Овчинников А.Л.

Севастополь

2020

1. Цель работы

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.

1. Постановка задачи

Реструктурировать код персонального сайта на основе шаблона MVC.

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.

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.

Реализовать класс FormValidation, выполняющий валидацию данных форм, передаваемых на сторону сервера.

С использованием разработанного класса реализовать валидацию форм «Контакт» и «Тест по дисциплине «Экология» ».

Реализовать дочерний класс CustomFormValidation от класса FormValidation, дополнив его возможностью выполнения специализированной проверки формы «Тест по дисциплине» на стороне сервера.

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.

1. внешний вид переработанных Web-страниц

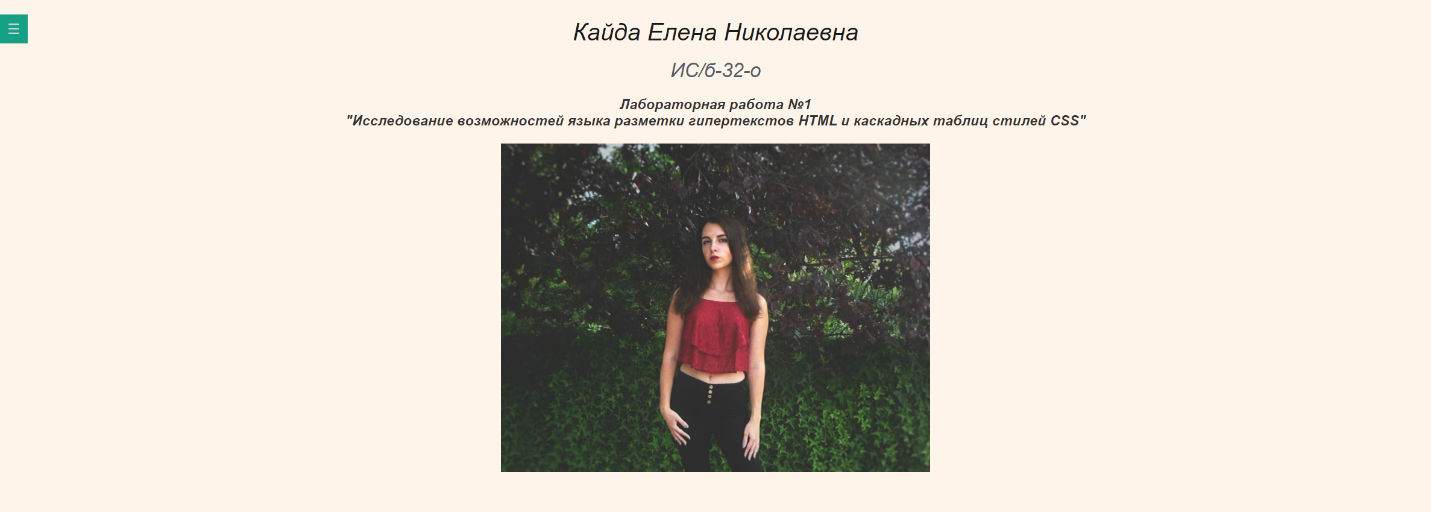


Рисунок 1.1 – Страница «Главная»

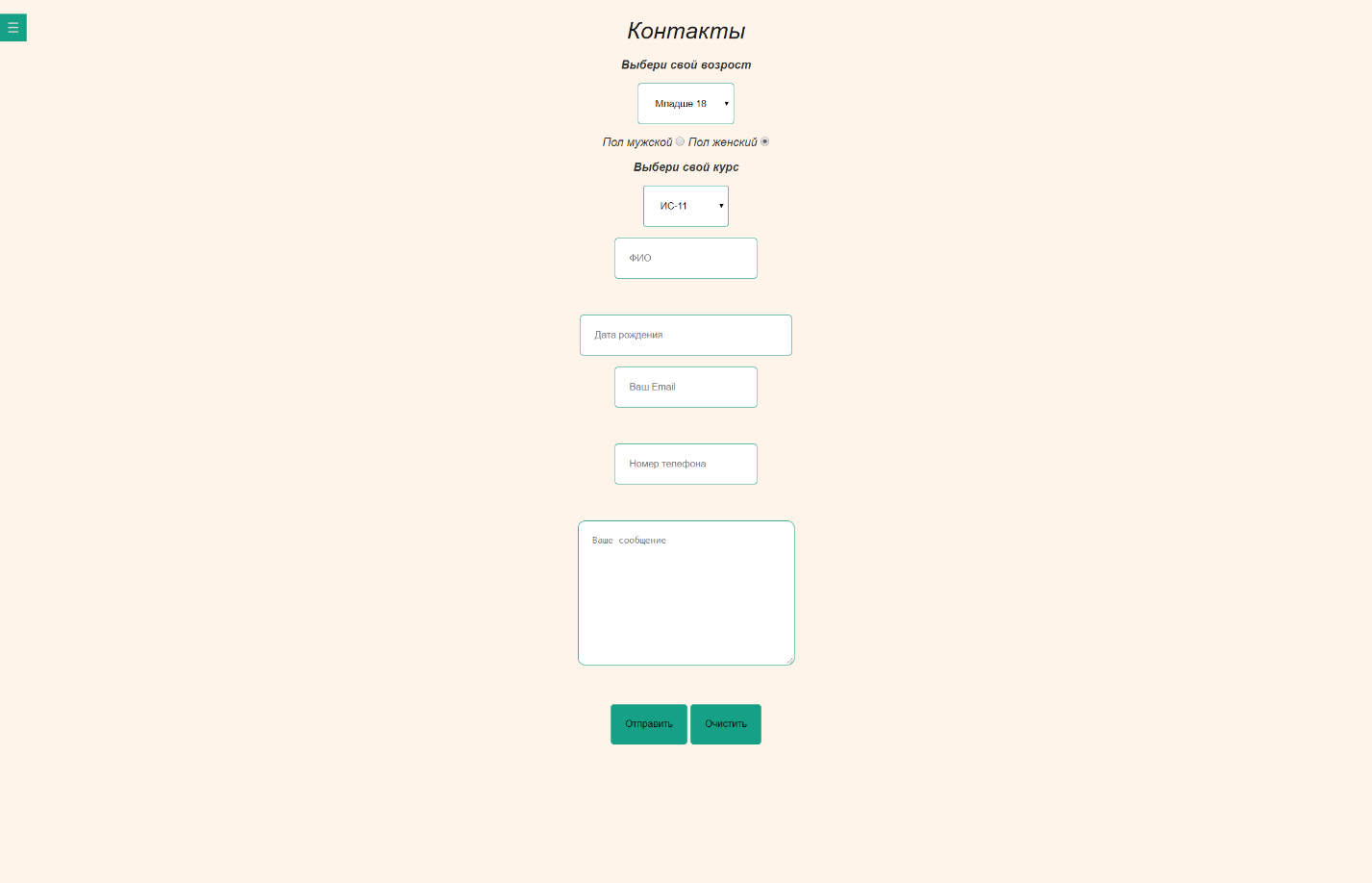


Рисунок 1.2 – Страница «Контакты»



Рисунок 1.3 – Страница «Мои интересы»

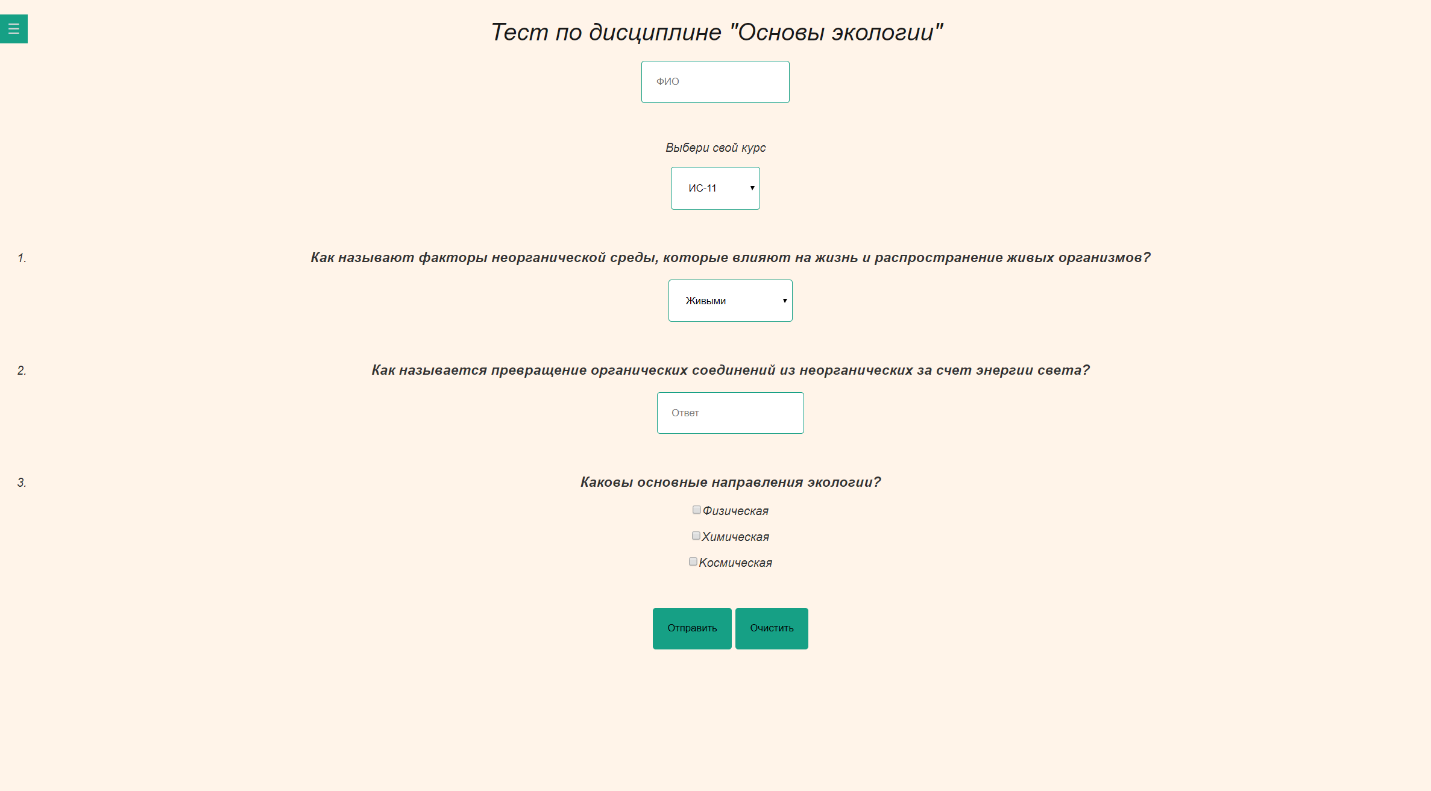


Рисунок 1.4 – Страница «Тест»

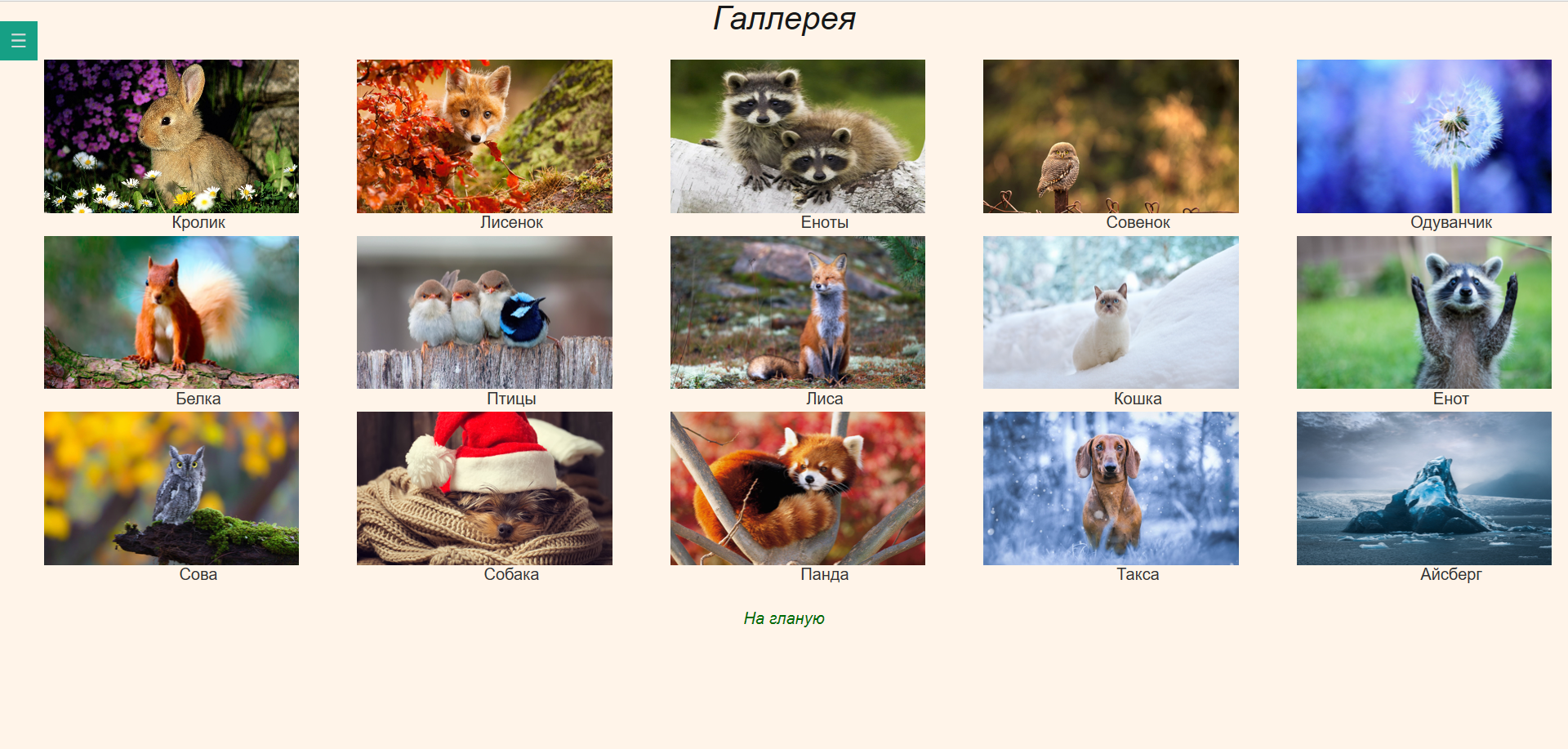


Рисунок 1.5 – Страница «Фотоальбом»

1. Листинг PHP скриптов и переписанных html страниц

Index.php

<?php

use app\core\Router;

    spl\_autoload\_register(function($class) {

        $path = str\_replace('\\', '/', $class.'.php');

        if (file\_exists($path)) {

            require $path;

        }

    });

    session\_start();

    $router=new Router;

    $router->run();

.htaccess

AddDefaultCharset utf-8

RewriteEngine On

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteRule ^(.\*)$ index.php

routes.php

<?php

return [

    '' => [

        'controller' => 'main',

        'action' => 'index',

    ],

    'aboutMe' => [

        'controller' => 'me',

        'action' => 'about',

    ],

    'interest' => [

        'controller' => 'me',

        'action' => 'interest',

    ],

    'schedule' => [

        'controller' => 'study',

        'action' => 'schedule',

    ],

    'fotos' => [

        'controller' => 'gallery',

        'action' => 'fotos',

    ],

    'test' => [

        'controller' => 'study',

        'action' => 'test',

    ],

    'contacts' => [

        'controller' => 'me',

        'action' => 'contacts',

    ],

];

Controller.php

<?php

namespace app\core;

use app\core\View;

abstract class Controller{

public $route;

public $view;

public $model;

    public function \_\_construct($route){

        $this->route = $route;

        $this->model = $this->autoloadModel($route['controller']);

        $this->data = $this->model->loadModel();      //загрузка данных для определенного action

        $this->view = new View($this->route);

    }

    public function autoloadModel($name){

         $path = 'app\models\\'.ucfirst($name);      //ucfirst — Преобразует первый символ строки в верхний регистр

        $fn = $path.'.php';

        if (file\_exists($fn)) {

             require $fn;

        }

           //создание определенной (дочерней) модели. Параметром передается action

        return new $path($this->route['action']);

    }

}

Model.php

<?php  
namespace Core;  
abstract class Model{

     public $method;

     public $validator;

     //запоминаем action

     function \_\_construct($action){

          $this->method = $action."Model";

    }

    //загружаем необходимый метод в зависмиости от action: action+"Model" = actionModel();

    public function loadModel(){

        $method = $this->method;

        return $this->$method();

    }

}

Router.php

<?php

namespace app\core;

use app\core\View;

class Router{

    protected $routes = [];

    protected $params = [];

    public function \_\_construct()

    {

        $arr = require('app/config/routes.php');

        foreach ($arr as $key=>$val){

            $this -> add($key, $val);

        }

    }

    public function add($route, $param){

        $route = '#^'.$route.'$#';

        $this -> routes[$route] = $param;

    }

    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';      //ucfirst — Преобразует первый символ строки в верхний регистр

            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);

        }

    }

}

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);

        // extract — Импортирует переменные из массива в текущую таблицу символов.

        // Эта функция рассматривает ключи массива в качестве имен переменных, а их значения - в качестве значений этих переменных.

        // Для каждой пары ключ/значение будет создана переменная в текущей таблице символов

        $view = 'app/views/'.$this->path.'.php';

        if (file\_exists('app/views/'.$this->path.'.php')){

            // ob\_start();     //включение буффера

            //require аналогично include, за исключением того, что при ошибке он также выдаст фатальную ошибку уровня E\_COMPILE\_ERROR.

            // require $view;      //считывание основного кода страницы(body) в буффер

            // $content = ob\_get\_clean();      //заносим все из буффера в переменную content

            require 'app/views/layouts/'.$this->layout.'.php';      //подключение шаблона default

        }else{

            echo 'View not found: '.$this->path;

        }

    }

    public static function errorCode($code){

        http\_response\_code($code);      //http\_response\_code — Получает или устанавливает код ответа HTTP

        require 'app/views/errors/'.$code.'.php';

        exit;

    }

    public function redirect($url){

        header('location: '.$url);

        exit;

    }

    public function message($status, $message) {

        exit(json\_encode(['status' => $status, 'message' => $message]));

    }

    public function location($url) {

        exit(json\_encode(['url' => $url]));

    }

}

MainController.php

<?php

namespace app\controllers;

use app\core\Controller;

class MainController extends Controller{

    public function indexAction(){

        $vars = [

            'args' => $this->data,

        ];

           $this->view->render('Главная страница', $vars);

    }

}

MeController.php

<?php

namespace app\controllers;

use app\core\Controller;

use app\models\validators\ValidContacts;

class MeController extends Controller{

    public function  aboutAction(){

        $this->view->render('Обо мне');

    }

    public function interestAction(){

        $vars = [

            'names' => $this->data,

        ];

        $this->view->render('Мои интересы', $vars);

    }

    public function contactsAction(){

        $this->view->render('Контакты', $this->data);

    }

}

GalleryController.php

<?php

namespace app\controllers;

use app\core\Controller;

class GalleryController extends Controller{

    public function  fotosAction(){

        $vars = [

            'args' => $this->data,

        ];

        $this->view->render('Галерея', $vars);

    }

}

StudyController.php

<?php

namespace app\controllers;

use app\core\Controller;

use app\models\validators\ValidTest;

class StudyController extends Controller{

    public function  scheduleAction(){

        $this->view->render('Расписание');

    }

    public function  testAction(){

        $this->view->render('Тест', $this->data);

    }

}

Validator.php

<?php

namespace app\models\validators;

// use app\core\Controller;

class Validator {

    protected $rules = [ ];

    protected $errors = [];

    protected $findErrors = [];

    //preg\_match — Выполняет проверку на соответствие регулярному выражению

    function \_\_construct($rules, $errors){

        foreach($rules as $key=>$val){

            $this->setRule($key, $val);

        }

        foreach($errors as $key=>$val){

            $this->setError($key, $val);

        }

    }

    public function Validate($post\_array){

        foreach($this->rules as $field => $rule){

            if (array\_key\_exists($field, $post\_array)){

                $method = "check\_".$field;

                $this->findErrors[$field] = $this->$method($post\_array[$field]);

            }

        }

    }

    public function setRule($field\_name, $validator\_name){

        $this->rules[$field\_name] = $validator\_name;

    }

    public function setError($field\_name, $error\_name){

        $this->errors[$field\_name] = $error\_name;

    }

    public function checkErrors(){

        foreach ($this->findErrors as $key=>$val){

            if ( $val == "" || $val == null) continue;

            else return false;

        }

        return true;

    }

    public function getErrors(){

        return $this->findErrors;

    }

    public function getRules(){

        return $this->rules;

    }

    public function getRule($key){

        if (array\_key\_exists($key, $this->rules)) return $this->rules[$key];

        else return false;

    }

    public function getError($key){

        if (array\_key\_exists($key, $this->findErrors)) return $this->findErrors[$key];

        else return false;

    }

    public function check\_name($p1) {

        $res = $this->isNotEmpty($p1);

        if (!$res) return $this->errors['empty'];

        if (preg\_match('/^[А-я]{1,10}[\s]{1,3}[А-я]{1,15}[\s]{1,3}[А-я]{1,15}+$/u', $p1)) return null;

        return $this->errors['name'];

    }

    public function check\_course($p1) {

        if (preg\_match('/^[0-9]{1,2}+$/', $p1)) return null;

        return $this->errors['empty'];

    }

    public function isNotEmpty($data){

        if ($data != "" && $data != null) return true;

        return false;

    }

    public function isInteger($data){

        if (is\_int($data)) return null;

        return $this->errors['notValidData'];

    }

    public function isLess($data, $value){

        if (preg\_match('/^[0-9]{3,30}+$/', $data))

            if ($data >= $value) return null;

        return $this->errors['notValidData'];

    }

    public function isGreater($data, $value){

        if (preg\_match('/^[0-9]{3,30}+$/', $data))

            if ($data <= $value) return null;

        return $this->errors['notValidData'];

    }

}

ValidContacts.php

<?php

namespace app\models\validators;

use app\models\validators\Validator;

class ValidContacts extends Validator {

    public function check\_email($p1) {

        $res = $this->isNotEmpty($p1);

        if (!$res) return $this->errors['empty'];

        if (preg\_match('/^([A-z0-9\_\.-]+)@([A-z0-9\_\.-]+)\.([A-z\.]{2,6})$/', $p1)) return null;

        return $this->errors['email'];

    }

    public function check\_data($p1) {

        $res = $this->isNotEmpty($p1);

        if (!$res) return $this->errors['empty'];

        if (preg\_match('/^(0|1)[0-2]{1}[\/]{1}[0-3]{1}[0-9]{1}[\/]{1}[0-9]{4}+$/', $p1)) return null;

        return $this->errors['notValidData'];

    }

    public function check\_phone($p1) {

        $res = $this->isNotEmpty($p1);

        if (!$res) return $this->errors['empty'];

        if (preg\_match('/^[+]{1}[37]{1}[0-9]{10}+$/', $p1)) return null;

        else return $this->errors['phone'];

    }

    public function check\_message($p1) {

        $res = $this->isNotEmpty($p1);

        if (!$res) return $this->errors['empty'];

        return null;

    }

    public function check\_age($p1) {

        if (preg\_match('/^[1-3]{1,1}+$/', $p1)) return null;

        return $this->errors['age'];

    }

}

ValidTest.php

<?php

namespace app\models\validators;

use app\models\validators\Validator;

class ValidTest extends Validator {

    public function check\_question1($value){

        if (preg\_match('/^[1-6]{1,1}+$/', $value)){

            if ($value == 2) return null;

            else return $this->errors['incorrectly'];

        } else return $this->errors['emptyAnswer'];

    }

    public function check\_question2($value){

        if ($value != "" && $value != null){

            if ($value == "Фотосинтез" || $value == "фотосинтез") return null;

            else return $this->errors['incorrectly'];

        }else return $this->errors['emptyAnswer'];

    }

    public function check\_question3($value){

        if (!$value) return $this->errors['emptyAnswer'];

        if (count($value)==2 && $value[0]==1 && $value[1]==2) return null;

        return $this->errors['incorrectly'];

    }

    public function Validate($post\_array){

        foreach($this->rules as $field => $rule){

            if (array\_key\_exists($field, $post\_array)){

                $method = "check\_".$field;

                $this->findErrors[$field] = $this->$method($post\_array[$field]);

            }

        }

        if (!array\_key\_exists('question3', $post\_array))  $this->findErrors['question3'] = $this->check\_question3(null);

    }

}

Models\Main.php

<?php

namespace app\models;

use app\core\Model;

class Main extends Model{

    public function \_\_construct($action){

        parent::\_\_construct($action);

    }

    public function indexModel(){

    }

}

Models\Gallery.php

<?php

namespace app\models;

use app\core\Model;

class Gallery extends Model{

    public $path = "public/assets/img/";

    public $jpg = ".jpg";

    public $titles = ["Кролик","Лисенок","Еноты","Совенок","Одуванчик","Белка","Птицы",

    "Лиса","Кошка","Енот","Сова","Собака","Панда","Такса","Айсберг"];

    public function \_\_construct($action){

        parent::\_\_construct($action);

    }

    public function fotosModel(){

        for($i = 0; $i < count($this->titles); $i++){

            $result[$i] = [

                "id"=> $i+1,

                'src'=> $this->path.($i+1).$this->jpg,

                "alt"=> $this->titles[$i],

                "titles"=> $this->titles[$i],

                "figcaption"=> $this->titles[$i],

            ];

        }

        return $result;

    }

}

Models\Me.php

<?php

namespace app\models;

use app\core\Model;

use app\models\validators\ValidContacts;

class Me extends Model{

    public $path;

    function \_\_construct($action){

        parent::\_\_construct($action);

    }

    public function aboutModel(){

    }

    public function contactsModel(){

        $rules = [

            'name' => "Пример: Иванов Иван Иванович",

            'email' => "Пример: example33@ety.ru",

            'data' => "Нажмитe на поле ввода для выбора даты",

            'phone' => "Пример: +71234567890 / +31234567890",

            'message' => "Текст вашего сообщения",

        ];

        $errors = [

            'name' => "Имя должно состоять из трех слов и может содержать только русские буквы длиной до 30 символов",

            'email' => "E-mail указан не верно",

            'age' => "Выберите свой возраст",

            'emptyData' => "Введите дату рождения",

            'empty' => "Поле не должно быть пустым",

            'notValidData' => "Неверная дата",

            'phone' => "Неверный номер телефона",

        ];

        $this->validator = new ValidContacts($rules, $errors);

        return  [

            'valid' => $this->validator,

            'rules' => $this->validator->getRules(),

            'errors' => $this->validator->getErrors(),

            'values'=> $this->validator->getErrors(),

        ];

    }

    public function interestModel(){

        $result = [

            [

                'src' => "book",

                'name' => 'Любимая книга',

                'description' => "Представьте себе ровное, как стол место, вымощенное камнем. Вокруг него высоченные стены. За стенами - Лабиринт.",

            ],

            [

                'src' => "film",

                'name' => 'Любимый фильм',

                'description' => "Преуспевающий бизнесмен получает в наследство шестерых пингвинов и буквально влюбляется в них.",

            ],

            [

                'src' => "sport",

                'name' => 'Любимый вид спорта',

                'description' => "Есть разнообразное количество видов бега, как двигательной активности, которой занимаются профессиональные атлеты и многие люди, заботящиеся о здоровье и физической форме."

            ]

        ];

        return $result;

    }

}

Models\Study.php

<?php

namespace app\models;

use app\core\Model;

use app\models\validators\ValidTest;

class Study extends Model{

    public $method;

    function \_\_construct($action){

        parent::\_\_construct($action);

    }

    public function scheduleModel(){

    }

    public function testModel(){

        $rules = [

            'name' => "Пример: Иванов Иван Иванович",

            'question1' => "Выберите правильный ответ",

            'question2' => "Введите ответ",

            'question3' => "Выберите правильные ответы",

        ];

        $errors = [

            'name' => "Имя должно состоять из трех слов и может содержать только русские буквы длиной до 30 символов",

            'empty' => "Поле не должно быть пустым",

            'emptyAnswer' => "Введите ответ",

            'incorrectly' => "Не верно",

        ];

        $this->validator = new ValidTest($rules, $errors);

        return [

            'valid' => $this->validator,

            'rules' => $this->validator->getRules(),

            'errors' => $this->validator->getErrors(),

        ];

    }

}

Views\layouts\default.php

<!DOCTYPE html>

<html>

<head>

    <!-- <title><?php echo $title; ?></title> -->

    <link rel="stylesheet" href="public/assets/css/style.css">

    <link rel="stylesheet" href="public/assets/css/foto.css">

    <link rel="stylesheet" href="public/assets/css/menu.css">

    <link rel="stylesheet" href="public/assets/css/study.css">

    <script src="public/scripts/submenu.js"></script>

    <script src="public/scripts/check.js"></script>

    <script src="public/scripts/jquery-2.2.3.js"></script>

    <script src="public/scripts/JQcontacts.js"></script>

    <script src="public/scripts/JQModalWindow.js"></script>

    <script src="public/scripts/bootstrap/js/bootstrap.js"></script>

    <script src="public/scripts/calendar.js"></script>

    <script src="public/scripts/fotos.js"></script>

    <script src="public/scripts/interests.js"></script>

</head>

<body>

<input type="checkbox" id="nav-toggle" hidden>

    <nav class="nav">

        <label for="nav-toggle" class="nav-toggle" onclick></label>

        <h2 class="logo">

            <a>Little\_coon\_</a>

        </h2>

        <ul>

            <li><a href="/">Главная</a>

            <li><a href="/aboutMe">Обо мне</a>

            <li><a href="/interest">Мои интересы</a>

            <li><a href="/schedule">Учеба</a>

            <li><a href="/fotos">Фотоальбом</a>

            <li><a href="/test">Тест</a>

            <li><a href="/contacts">Контакты</a>

        </ul>

    </nav>

<?php require $view; ?>

</body>

</html>

Views\gallery\fotos.php

<section>

    <h1>Галлерея</h1>

    <table class="foto">

        <?php

        $i=0;

        foreach ($args as $val):?>

            <td><img

                id=<?php echo $val['id']; ?>

                class="inTable"

                src=<?php echo $val['src']; ?>

                alt=<?php echo $val['alt']; ?>

                title=<?php echo $val['titles']; ?>

                onClick=bigFotoDiv(this) >

                <figcaption> <?php echo $val['figcaption']; ?> </figcaption></td>

            <?php $i++;

                if($i % 5 == 0):?> <tr>

            <?php endif;?>

        <?php endforeach; ?>

    </table>

    <p> <a href="/">На гланую</a></p>

</section>

Views\main\index.php

<section>

        <h1>Кайда Елена Николаевна</h1>

        <h2>ИС/б-32-о</h2>

        <h3>Лабораторная работа №1

        <br>"Исследование возможностей языка разметки гипертекстов HTML и каскадных таблиц стилей CSS"</h3>

        <img src="public/assets/img/me.jpg" width="30%" height="40%">

</section>

Views\me\interest.php

<section>

        <h1>Мои интересы</h1>

        <ol class="ol" >

        <?php foreach($names as $val): ?>

                <li>

                        <a href= <?php echo "#".$val['src'] ?> > <?php echo $val['name'] ?> </a>

                </li>

        <?php endforeach; ?>

        </ol>

        <table>

                <?php foreach($names as $val): ?>

                <th id= <?php echo $val['src'] ?> colspan="2" >

                        <h2><?php echo $val['name'] ?> </h2></th>

                <tr>

                        <td CLASS=data><img src=<?php echo "public/assets/img/".$val['src'].".jpg" ?> class="leftImg"></td>

                        <td CLASS=data>

                                <?php echo $val['description'] ?>

                        </td>

                </tr>

                <?php endforeach; ?>

        </table>

</section>

Views\me\contacts.php

<section>

    <h1>Контакты</h1>

    <?php

    $values;

    if ($\_SERVER["REQUEST\_METHOD"] == "POST"){

        $valid->Validate($\_POST);

        $errors = $valid->errorsGet;

        if (!$valid->checkErrors($errors)) $values = $\_POST;

    }

    ?>

    <!-- <form  action="<?php echo $\_SERVER["PHP\_SELF"]?>" method=POST id="contact">  -->

    <form  action="" method=POST id="contact" > <!--onchange="checkSubmit()"-->

        <p><strong>Выбери свой возрост</strong></p>

        <p><select id="age" name="age" >

            <option value="1">Младше 18</option>

            <option value="2">18 - 25</option>

            <option value="3">Старше 25</option>

            </select>

        <p>Пол мужской <input name="pol" id="pol1" type="radio" value="m" >

        Пол женский <input name="pol" id="pol2" type="radio" value="w" checked></p>

        <p><strong>Выбери свой курс</strong></p>

        <SELECT id="gr" name="course">

            <OPTGROUP label="1 курс">

                <OPTION value="1">ИС-11

                <OPTION value="2">ИС-12

                <OPTION value="3">ИС-13

            </OPTGROUP>

            <OPTGROUP label="2 курс">

                <OPTION value="4">ИС-21

                <OPTION value="5">ИС-22

                <OPTION value="6">ИС-23

            </OPTGROUP>

            <OPTGROUP label="3 курс">

                <OPTION value="7">ИС-31

                <OPTION value="8">ИС-32

            </OPTGROUP>

            <OPTGROUP label="4 курс">

                <OPTION value="9">ИС-41

                <OPTION value="10">ИС-42

            </OPTGROUP>

            <OPTGROUP LABEL="5 курс">

                <OPTION value="11">ИС-51

                <OPTION value="12">ИС-52

            </OPTGROUP>

            <OPTGROUP LABEL="Магистры">

                <OPTION  value="13">Maг-51

            </OPTGROUP>

        </SELECT>

        <p><pre><?php echo $errors['course'] ?></pre></p>

        <p><input type="text" id="FIO" name="name" placeholder="ФИО" data-toggle="popover" onblur="validate(this)"

             value="<?php if (array\_key\_exists('name', $\_POST)) echo $values['name'] ?>" data-content="<?php echo $rules['name'] ?>"><br><pre><?php echo $errors['name'] ?></pre><br></p>

        <div id="calendar" >

            <input class="notFocus" id="data"  data-toggle="popover" data-content="<?php echo $rules['data'] ?>"

                value="<?php if (array\_key\_exists('data', $values)) echo $values['data'] ?>" name="data" type="text" size="35"  onfocus="show(this)" oninput="writeDate()" placeholder="Дата рождения"><pre><?php echo $errors['data'] ?><br></pre>

        </div>

        <p><input type="text" id="mail" name="email" onblur="validate(this)" placeholder="Ваш Еmail"

            value="<?php if (array\_key\_exists('email', $values)) echo $values['email'] ?>" data-toggle="popover" data-content="<?php echo $rules['email'] ?>"><pre><?php echo $errors['email'] ?></pre><br></p>

        <p><input type="text"  id="mobile" name="phone" onblur="validate(this)" placeholder="Номер телефона"

            value="<?php if (array\_key\_exists('phone', $values)) echo $values['phone'] ?>"  data-toggle="popover" data-content="<?php echo $rules['phone'] ?>"><pre><?php echo $errors['phone'] ?></pre><br></p></p>

        <p><textarea id="message" name="message" onblur="validate(this)" data-toggle="popover" placeholder="Ваше сообщение"

            value="<?php if (array\_key\_exists('message', $values)) echo $values['message'] ?>" data-content="<?php echo $rules['message'] ?>"></textarea><br><pre><?php echo $errors['message'] ?></pre> <br></p>

        <input type="submit"  id="submit" value="Отправить" >

        <input type="reset" id="reset" value="Очистить" >

    </form>

    <div id="modalWindow">

        <div id="modalDiv">

            <p>Вы действительно хотите это сделать?</p>

            <input type="button" class="close" id="yes" value="yes">

            <input type="button" class="close" id="no" value="no">

        </div>

    </div>

</section>

Views\study\test.php

<section>

    <h1> Тест по дисциплине "Основы экологии"</h1>

    <?php

    // var\_dump($\_POST);

    $value;

    if ($\_SERVER["REQUEST\_METHOD"] == "POST"){

        $valid->Validate($\_POST);

        if (!$valid->checkErrors()) $value = $\_POST;

        $errors = $valid->errorsGet;

        // var\_dump($errors);

    }

    ?>

    <!-- <form onsubmit="return check(foundT());" id="contact">  -->

    <form action="" method=POST id="contact">

        <p><input id="FIO" type="text" name="name" placeholder="ФИО" data-toggle="popover" onblur="validate(this)" data-content="<?php echo $rules['name'] ?>"><br><pre><?php echo $errors['name'] ?></pre><br></p>

        <p>Выбери свой курс</p>

        <p><SELECT id="gr" name="course">

            <OPTGROUP label="1 курс">

                <OPTION value="1">ИС-11

                <OPTION value="2">ИС-12

                <OPTION value="3">ИС-13

            </OPTGROUP>

            <OPTGROUP label="2 курс">

                <OPTION value="4">ИС-21

                <OPTION value="5">ИС-22

                <OPTION value="6">ИС-23

            </OPTGROUP>

            <OPTGROUP label="3 курс">

                <OPTION value="7">ИС-31

                <OPTION value="8">ИС-32

            </OPTGROUP>

            <OPTGROUP label="4 курс">

                <OPTION value="9">ИС-41

                <OPTION value="10">ИС-42

            </OPTGROUP>

            <OPTGROUP LABEL="5 курс">

                <OPTION value="11">ИС-51

                <OPTION value="12">ИС-52

            </OPTGROUP>

            <OPTGROUP LABEL="Магистры">

                <OPTION  value="13">Maг-51

            </OPTGROUP>

        </SELECT>

        <pre><?php echo $errors['course'] ?></pre><br></p>

        <ol class=test>

            <li><h3>Как называют факторы неорганической среды, которые влияют на жизнь и распространение живых организмов?</h3></li>

            <p>

                <select id="test1" name="question1">

                    <option value="1">Живыми</option>

                    <option value="2">Абиотическими</option>

                    <option value="3">Антропогенными</option>

                    <option value="4">Биотическими</option>

                    <option value="5">Лимитирующие</option>

                    <option value="6">Неживые</option>

                    <!-- 2 -->

                </select>

                <pre><?php echo $errors['question1'] ?></pre><br>

            </p>

            <li><h3> Как называется превращение органических соединений из неорганических за счет энергии света?</h3></li>

            <!-- фотосинтез -->

            <p><input id="test2" type="text" name="question2" placeholder="Ответ"> <pre><?php echo $errors['question2'] ?></pre><br></p>

            <li><h3>Каковы основные направления экологии?</h3></li>

            <p>

                <p><input id="c1" type="checkbox" name="question3[]" value="1">Физическая</p>

                <p> <input id="c2" type="checkbox" name="question3[]" value="2">Xимическая</p>

                <p> <input id="c3" type="checkbox" name="question3[]" value="3">Kосмическая</p>

                <pre><?php if (array\_key\_exists('question3', $errors)) echo $errors['question3'] ?></pre><br>

            </p>

            <!-- 1 2 -->

            <input type="submit" id="submit"  value="Отправить" >

            <input  type="reset" value="Очистить">

        </ol>

    </form>

</section>

1. Вывод

Изучил основы синтаксиса PHP и принципы функционирования MVC приложения на стороне сервера. А также приобретены практические навыки использования языка PHP для генерации HTML-кода и обработки HTML-форм в MVC приложении.