**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт кибернетики

Кафедра информатики и проектирования систем

Направление «Информатика и вычислительная техника»

Отчет по лабораторной работе №4 по дисциплине

**«Современные платформы программирования»**

Разработка многостраничного веб-сайта по шаблону MVC

Выполнил:

Студент группы 8ВМ81 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Т. Хусаинова

Проверил:

Доцент ОИТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Н. Попов

Томск 2018

# Задание

Разработать многостраничный web-сайт на языке PHP с использованием фреймворка (в данном случае – Yii2), шаблона проектирования и технологии MVC.

Необходимо:

* Реализовать веб-сайт со следующими страницами: административная страница для удаления файлов и их описаний с сайта; пользовательская страница для загрузки файлов и их описаний на сайт; пользовательская страница для поиска файла по ключевым словам описания и скачивания файла.
* Защитить все административные страницы сайта парольным доступом.
* Дополнительно создать на сайте страницу с описанием контакта на разработчика сайта («Об авторе» / «Контакты» и т.п.) и страницу с описанием принципа работы сайта («О сайте»).
* Реализовать на web-сайте возможность протоколирования действия пользователя (нажатия на ссылки, заполнения полей) в файле (текстовом, xml, json) и (при потребности) отображать их перечень на странице приложения в табличном виде (дата, время, действие).

**Yii 2.0**

Yii – это высокопроизводительный компонентный PHP фреймворк, предназначенный для быстрой разработки современных веб-приложений. Слово Yii в китайском языке означает «простой и эволюционирующий». Также Yii может расшифровываться как акроним **Yes It Is**!

Yii приложения организованы согласно шаблону проектирования модель-представление-контроллер (MVC). Модели представляют собой данные, бизнес логику и бизнес правила; представления отвечают за отображение информации, в том числе и на основе данных, полученных из моделей; контроллеры принимают входные данные от пользователя и преобразовывают их в понятный для моделей формат и команды, а также отвечают за отображение нужного представления.

Кроме MVC, Yii приложения также имеют следующие сущности:

* входные скрипты: это PHP скрипты, которые доступны напрямую конечному пользователю приложения. Они ответственны за запуск и обработку входящего запроса;
* приложения: это глобально доступные объекты, которые осуществляют корректную работу различных компонентов приложения и их координацию для обработки запроса;
* компоненты приложения: это объекты, зарегистрированные в приложении и предоставляющие различные возможности для обработки текущего запроса;
* модули: это самодостаточные пакеты, которые включают в себя полностью все средства для MVC. Приложение может быть организованно с помощью нескольких модулей;
* фильтры: это код, который должен быть выполнен до и после обработки запроса контроллерами;
* виджеты: это объекты, которые могут включать в себя представления. Они могут содержать различную логику и быть использованы в различных представлениях.

Ниже на диаграмме представлена структурная схема приложения:

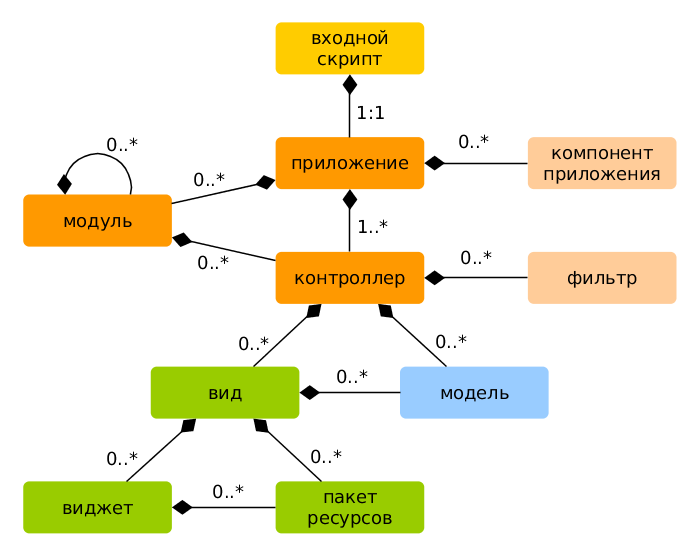


Рисунок 1 – Схема взаимодействия компонентов Yii 2.0

# Листинг

**Model**

<?php

namespace app\models;

use yii\db\ActiveRecord;

class Post extends ActiveRecord

{

  public static function tableName()

  {

      return 'post';

  }

  public $upload;

  public function rules()

  {

      return [

          [['desc'], 'required'],

          [['desc'], 'string', 'max' => 255],

          [['upload'], 'file'],

      ];

  }

}

**Controller**

<?php

namespace app\controllers;

use Yii;

use yii\filters\AccessControl;

use yii\web\Controller;

use yii\filters\VerbFilter;

use yii\web\UploadedFile;

use app\models\LoginForm;

use app\models\ContactForm;

use app\models\Post;

class SiteController extends Controller

{

*/\*\**

*\* {@inheritdoc}*

*\*/*

public function behaviors()

  {

      return [

          'access' => [

              'class' => AccessControl::*className*(),

              'only' => ['logout'],

              'rules' => [

                  [

                      'actions' => ['logout'],

                      'allow' => true,

                      'roles' => ['@'],

                  ],

              ],

          ],

          'verbs' => [

              'class' => VerbFilter::*className*(),

              'actions' => [

                  'logout' => ['post'],

              ],

          ],

      ];

  }

*/\*\**

*\* {@inheritdoc}*

*\*/*

public function actions()

  {

      return [

          'error' => [

              'class' => 'yii\web\ErrorAction',

          ],

          'captcha' => [

              'class' => 'yii\captcha\CaptchaAction',

              'fixedVerifyCode' => *YII\_ENV\_TEST* ? 'testme' : null,

          ],

      ];

  }

  public function actionLogin()

  {

      if (!Yii::*$app*->user->isGuest) {

          return $this->goHome();

      }

      $model = new LoginForm();

      if ($model->load(Yii::*$app*->request->post()) && $model->login()) {

          return $this->goBack();

      }

      $model->password = '';

      return $this->render('login', [

          'model' => $model,

      ]);

  }

  public function actionLogout()

  {

      Yii::*$app*->user->logout();

      return $this->goHome();

  }

  public function actionContact()

  {

      $model = new ContactForm();

      if ($model->load(Yii::*$app*->request->post()) && $model->contact(Yii::*$app*->params['adminEmail'])) {

          Yii::*$app*->session->setFlash('contactFormSubmitted');

          return $this->refresh();

      }

      return $this->render('contact', [

          'model' => $model,

      ]);

  }

  public function actionAbout()

  {

      return $this->render('about');

  }

  public function actionIndex()

  {

      $model = new Post();

      $this->handlePostSave($model);

      return $this->render('index', [

          'model' => $model,

      ]);

  }

  protected function handlePostSave(Post $model)

  {

      if ($model->load(Yii::*$app*->request->post())) {

          $model->upload = UploadedFile::*getInstance*($model, 'upload');

          if ($model->validate()) {

              if ($model->upload) {

                  $filePath = $model->upload->baseName . '.' . $model->upload->extension;

                  if ($model->upload->saveAs('uploads/' . $filePath)) {

                      $model->file = $filePath;

                      Yii::*info*('File upload', 'custom');

                  }

              }

              if ($model->save(false)) {

                  return $this->refresh();

              }

          }

      }

  }

  public function actionDownload($id)

  {

      $model = Post::*find*()->where(['id' => $id ])->one();

      $fileName = $model->file;

      $file = Yii::*getAlias*('uploads/' . $fileName);

      if (*file\_exists*($file)) {

          Yii::*info*('File download', 'custom');

          return Yii::*$app*->response->sendFile($file);

      } else {

          return $this->render('error');

      }

  }

  public function actionRemove($id)

  {

      $model = Post::*find*()->where(['id' => $id ])->one();

      if ($model != null) {

          $model->delete();

          Yii::*info*('File deletion', 'custom');

      } else {

          return $this->render('error');

      }

      return $this->render('delete', ['model' => $model]);

  }

  public function actionDelete()

  {

      $model = new Post();

      return $this->render('delete', [

          'model' => $model,

      ]);

  }

  public function actionSearch() {

      if (Yii::*$app*->request->get('search')){

          $search = Yii::*$app*->request->get('search');

          $posts = Post::*find*()->where(['like', 'desc', $search])->all();

      } else {

          $posts = Post::*find*()->all();

      }

      Yii::*info*('File search by request', 'custom');

      return $this->render('search', ['posts' => $posts]);

  }

}

**View (Index)**

<?php

use yii\helpers\Html;

use yii\widgets\ActiveForm;

use app\models\Post;

$this->title = 'Загрузка файлов';

$model = new Post();

$posts = Post::*find*()->all();

?>

<div class="site-index card border-secondary">

  <div class="card-header"><h3><?= Html::*encode*($this->title) ?></h3></div>

  <?php $form = ActiveForm::*begin*(['options' => ['enctype' => 'multipart/form-data']]) ?>

  <div class="body-content card-body">

      <div class="form-group row">

          <label class="col-sm-4 col-form-label">Файл</label>

          <label class="btn btn-outline-primary custom-button">

              <?= $form->field($model, 'upload', ['template' => "{label}\n{input}"])->label('Выбрать файл')->fileInput(['class' => 'hidden']) ?>

          </label>

      </div>

      <div class="form-group row">

          <label class="col-sm-4 col-form-label">Описание</label>

          <div class="col-sm-8">

              <?= $form->field($model, 'desc')->textInput(['class' => 'form-control'])->label(false) ?>

          </div>

      </div>

      <div class="col-sm-offset-4">

          <?= Html::*submitButton*('Загрузить', ['class' => 'btn btn-primary single-button mb60']) ?>

      </div>

      <div class="col-sm-offset-4">

          <?php if(Yii::*$app*->session->hasFlash('success')): ?>

              <div class="alert alert-success alert-dismissible single-button mb60" role="alert">

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

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

                  </button>

                  <?php echo Yii::*$app*->session->getFlash("success"); ?>

              </div>

          <?php endif;?>

          <?php if(Yii::*$app*->session->hasFlash('error')): ?>

              <div class="alert alert-danger alert-dismissible single-button mb60" role="alert">

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

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

                  </button>

                  <?php echo Yii::*$app*->session->getFlash("error"); ?>

              </div>

          <?php endif;?>

      </div>

      <table class="table">

          <thead>

          <tr>

              <th class="col-md-4">Наименование</th>

              <th class="col-md-6">Описание</th>

              <th class="col-md-2">Действие</th>

          </tr>

          </thead>

          <tbody>

          <?foreach ($posts as $post):?>

              <tr>

                  <td class="hidden"><?=$post->id?></td>

                  <td><?=$post->file?></td>

                  <td><?=$post->desc?></td>

                  <td><?= Html::*a*('Скачать', ['download', 'id'=>$post->id]) ?></td>

              </tr>

          <?endforeach;?>

          </tbody>

      </table>

  </div>

  <?php ActiveForm::*end*() ?>

</div>

**View (Search)**

<?php

use yii\helpers\Html;

use yii\widgets\ActiveForm;

$this->title = 'Поиск файлов';

?>

<div class="site-search card border-secondary">

  <div class="card-header"><h3><?= Html::*encode*($this->title) ?></h3></div>

  <?php $form = ActiveForm::*begin*([

          'action' => ['search'],

          'method' => 'get',

  ]) ?>

  <div class="body-content card-body">

      <div class="form-group row">

          <label class="col-sm-4 col-form-label">Описание</label>

          <div class="col-sm-8">

              <?= Html::*textInput*('search','', ['class' => 'form-control']) ?>

          </div>

      </div>

      <div class="col-sm-offset-4">

          <?= Html::*submitButton*('Найти', ['class' => 'btn btn-primary single-button mb60']) ?>

      </div>

      <table class="table">

          <thead>

              <tr>

                  <th class="col-md-4">Наименование</th>

                  <th class="col-md-6">Описание</th>

                  <th class="col-md-2">Действие</th>

              </tr>

          </thead>

          <tbody>

          <?php foreach ($posts as $post):?>

              <tr>

                  <td class="hidden"><?=$post->id?></td>

                  <td><?=$post->file?></td>

                  <td><?=$post->desc?></td>

                  <td><?= Html::*a*('Скачать', ['download', 'id'=>$post->id]) ?></td>

              </tr>

          <?php endforeach;?>

          </tbody>

      </table>

  </div>

  <?php ActiveForm::*end*() ?>

</div>

**View (Delete)**

<?php

use yii\helpers\Html;

use yii\widgets\ActiveForm;

use app\models\Post;

$this->title = 'Удаление файлов';

$posts = Post::*find*()->all();

?>

<div class="site-search card border-secondary">

  <div class="card-header"><h3><?= Html::*encode*($this->title) ?></h3></div>

  <?php $form = ActiveForm::*begin*() ?>

  <div class="body-content card-body">

      <table class="table">

          <thead>

          <tr>

              <th class="col-md-4">Наименование</th>

              <th class="col-md-6">Описание</th>

              <th class="col-md-2">Действие</th>

          </tr>

          </thead>

          <tbody>

          <?foreach ($posts as $post):?>

              <tr>

                  <td class="hidden"><?=$post->id?></td>

                  <td><?=$post->file?></td>

                  <td><?=$post->desc?></td>

                  <td><?= Html::*a*('Удалить', ['remove', 'id'=>$post->id]) ?></td>

              </tr>

          <?endforeach;?>

          </tbody>

      </table>

  </div>

  <?php ActiveForm::*end*() ?>

</div>

# Результаты работы

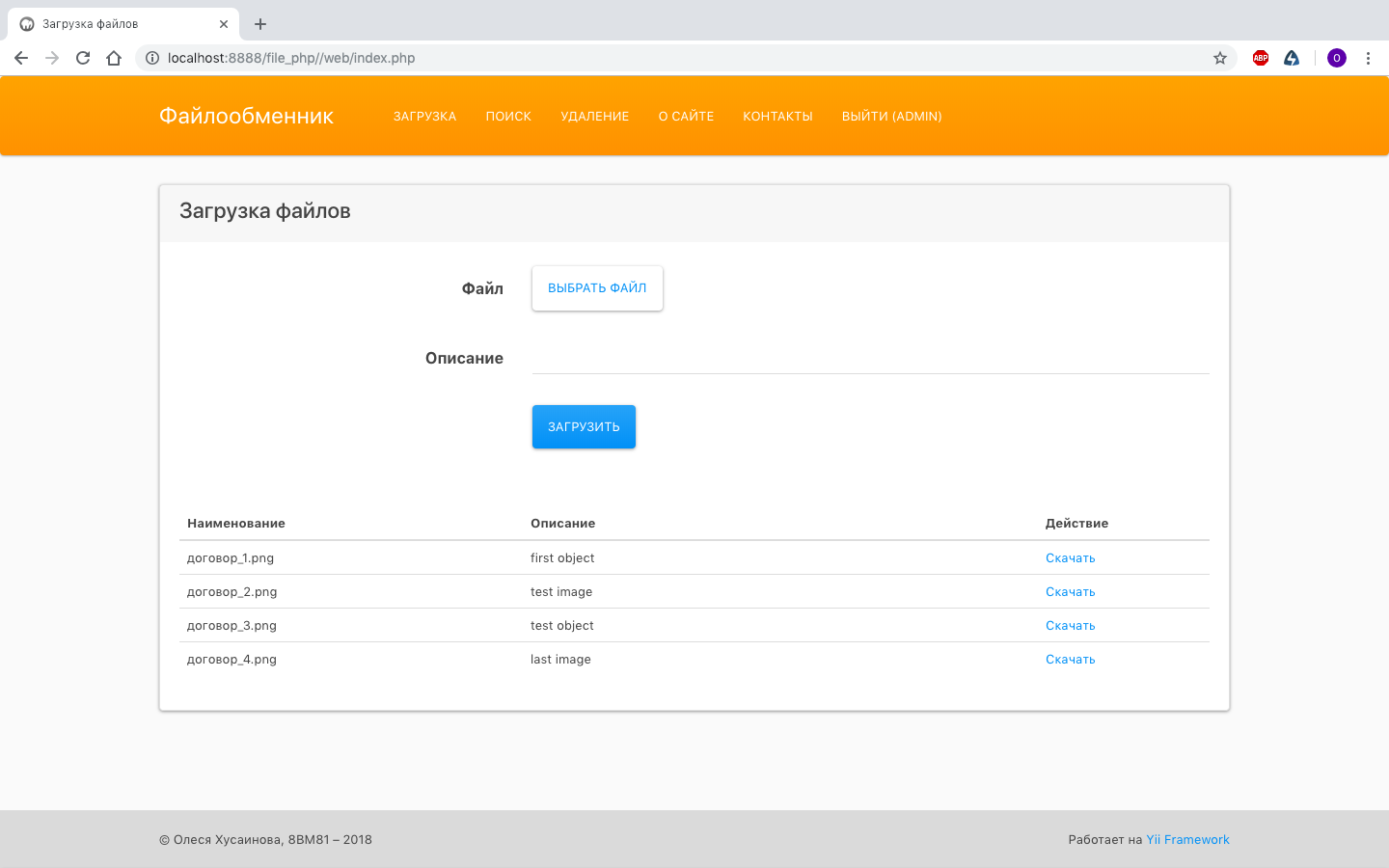


Рисунок 1 – Загрузка и скачивание файлов

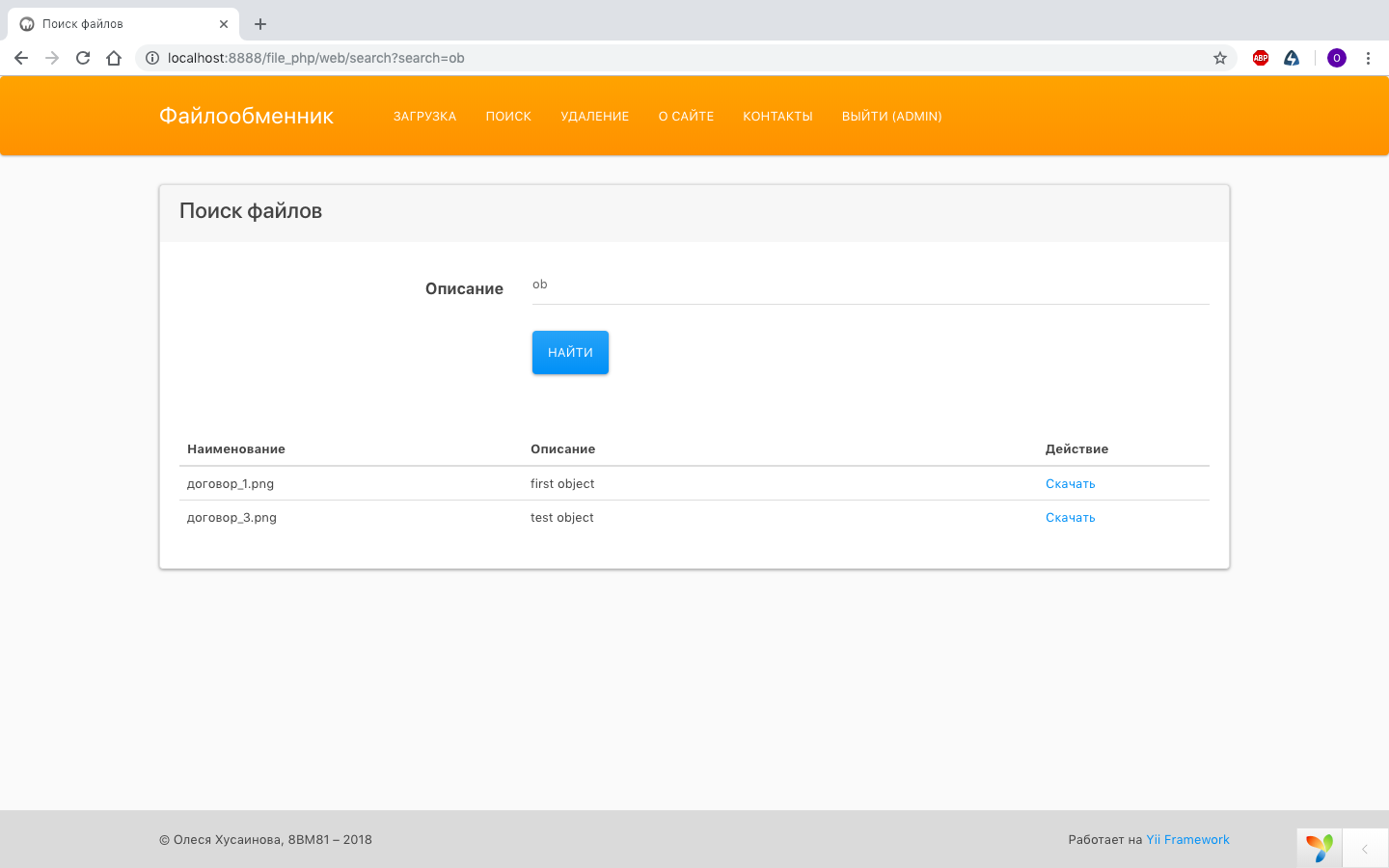


Рисунок 2 – Поиск файлов

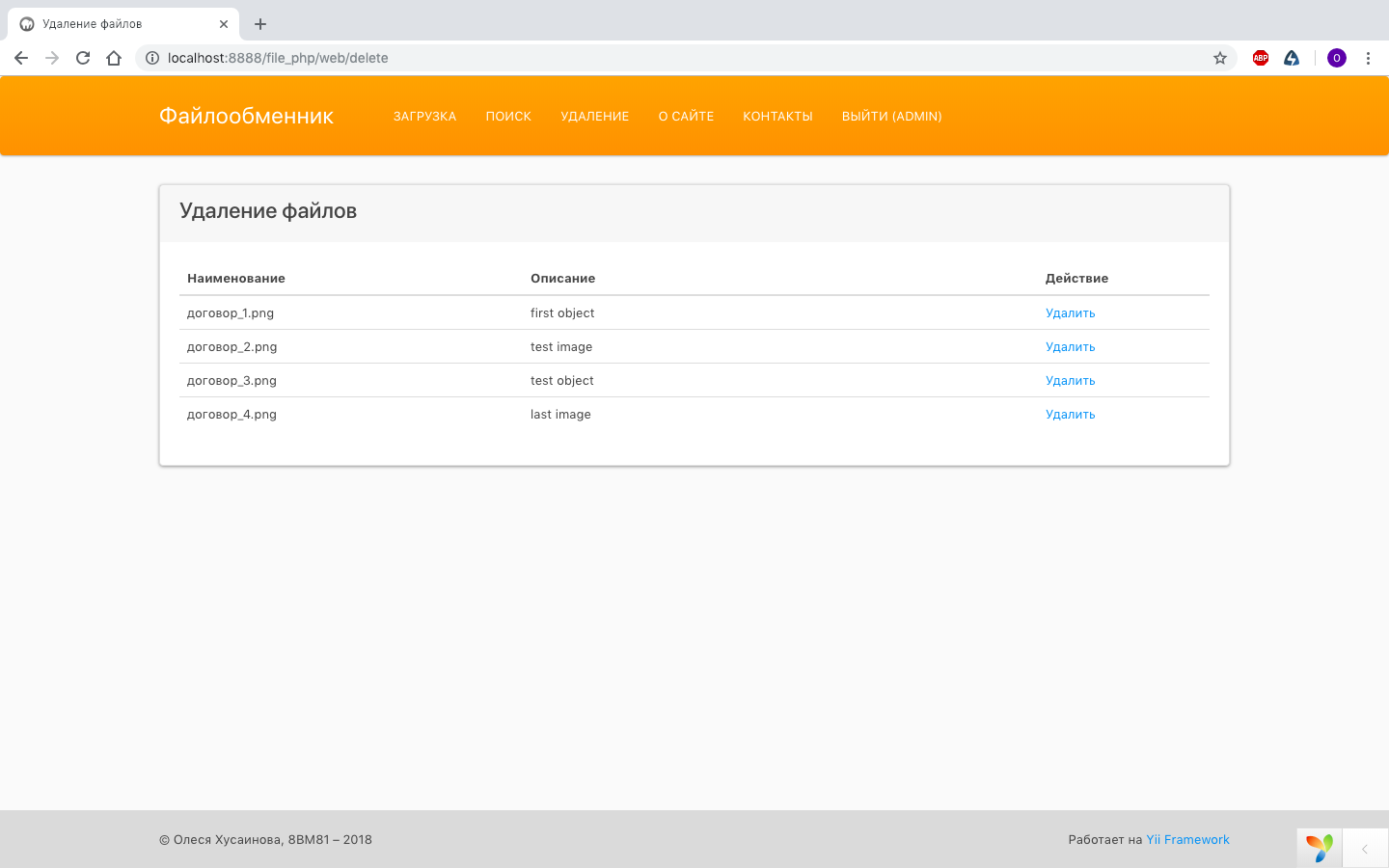


Рисунок 3 – Удаление файлов

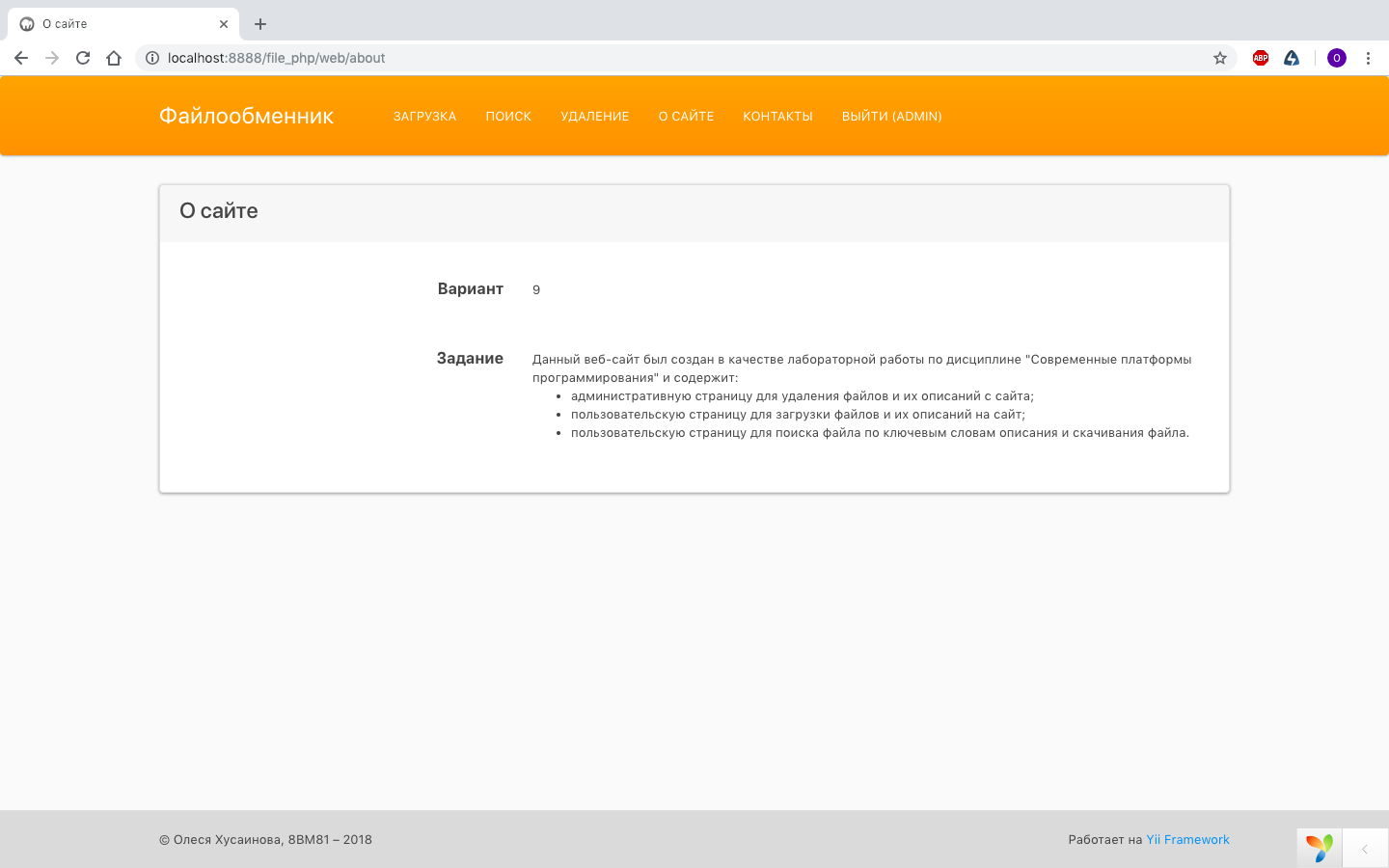


Рисунок 4 – О сайте

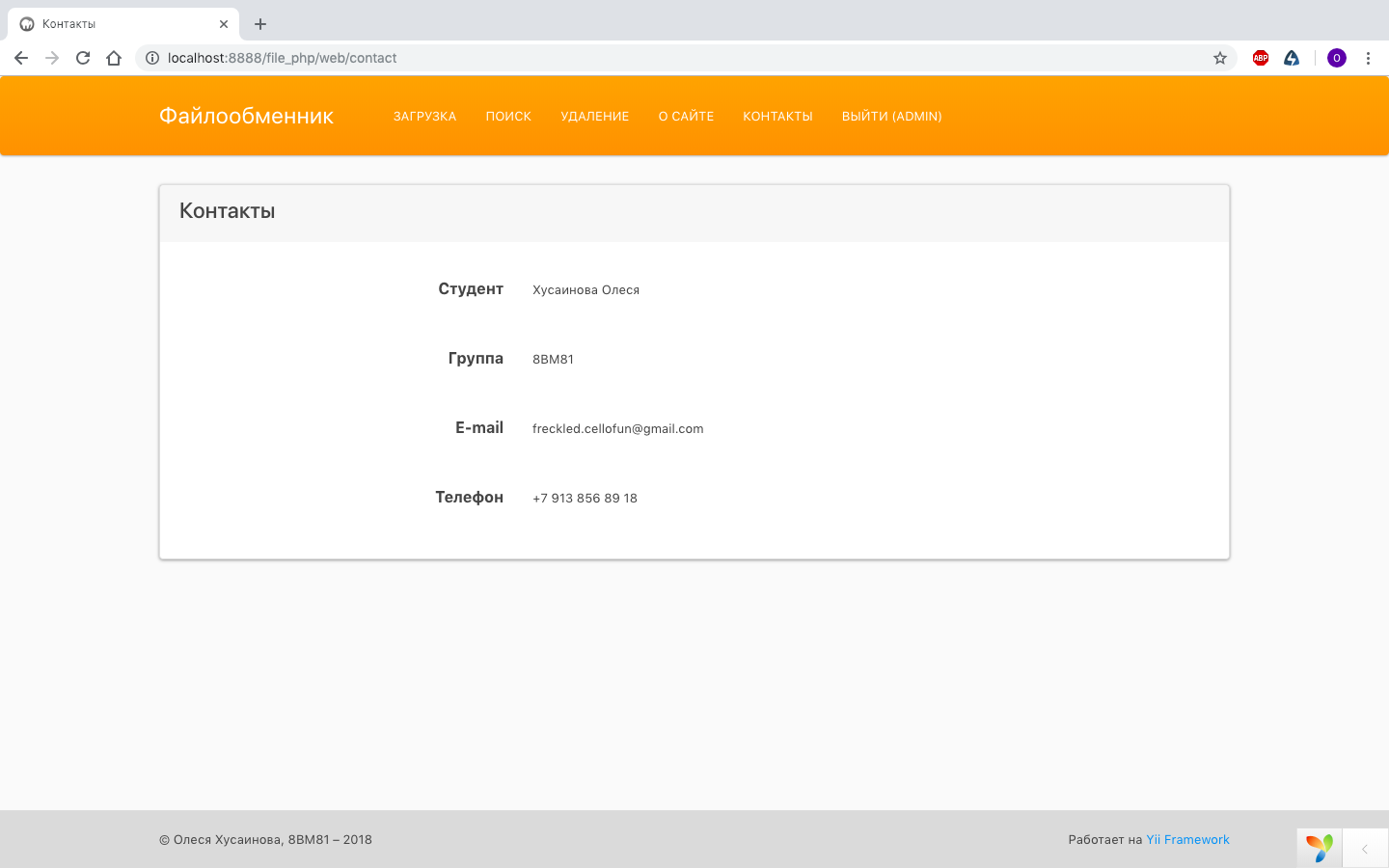


Рисунок 5 – Контакты

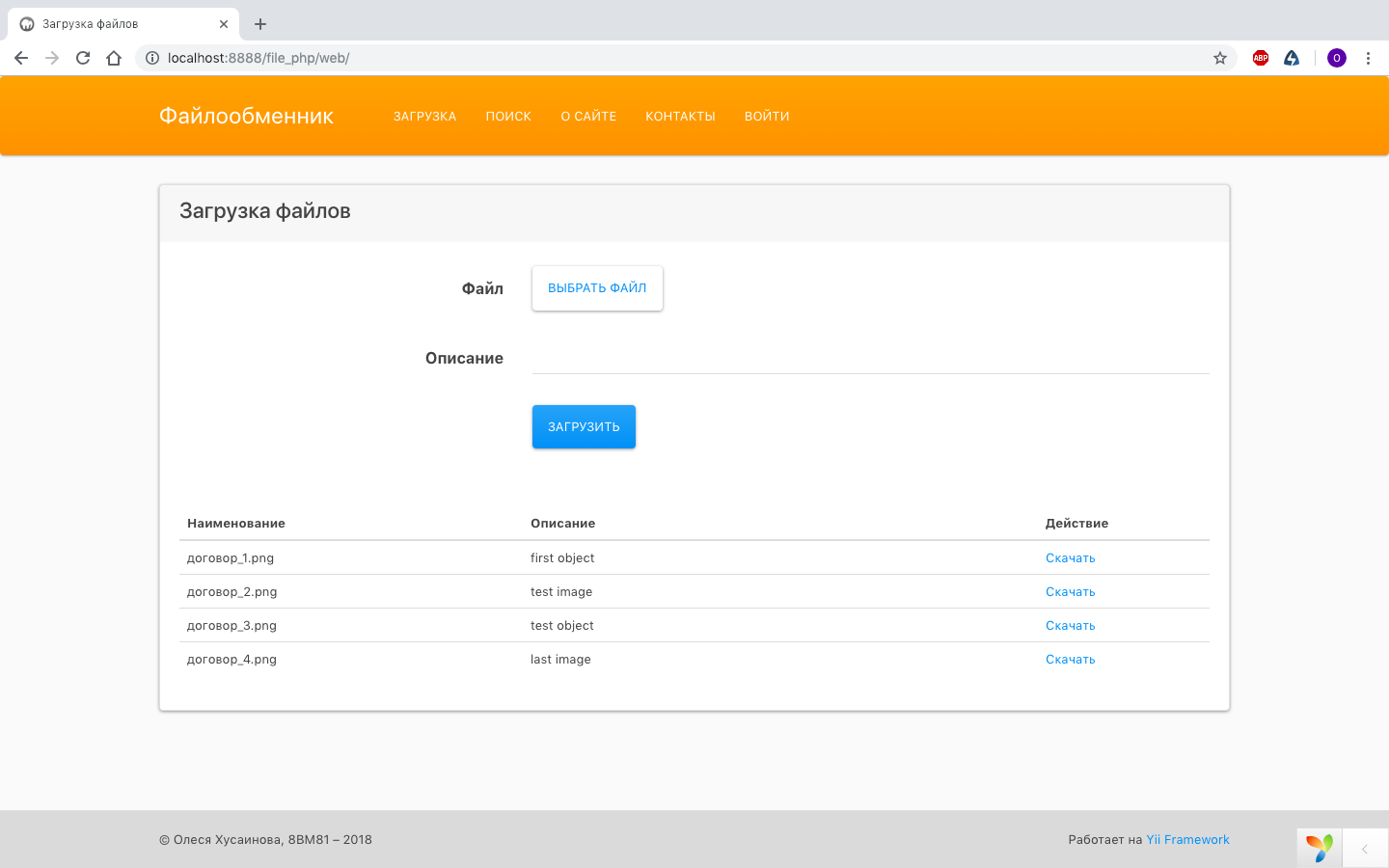


Рисунок 6 – Страницы, доступные для обычных пользователей

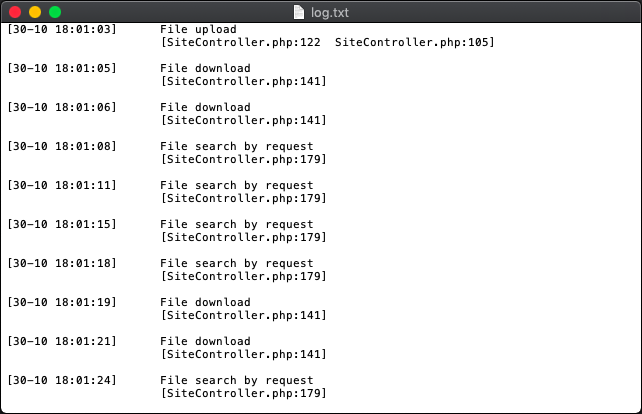


Рисунок 7 – Логирование действий

# Выводы

На основе фреймворка Yii 2.0 был реализован веб-сайт со следующими страницами:

* административная страница для удаления файлов и их описаний с сайта;
* пользовательская страница для загрузки файлов и их описаний на сайт;
* пользовательская страница для поиска файла по ключевым словам описания и скачивания файла.

Все основные действия пользователя логируются в файл.