13. «Каталогтар» және «Файлдар» атты модельдерін толықтыру және индекс файлын әзірлеу

Жоспар:

13.1 «GridView» және «ActiveForm» виджеттерін орнату

13.2 Каталогты құру және файлдарды жүктеу бойынша қосымша модельдерді құру

13.3 Каталог пен файлдар модельдері бойынша ортақ контроллерді құру

13.4 Индекс файлын әзірлеу және өзгерту

13.5 «Создание папки» бетін құрастыру

13.6 «Загрузка файла» және «Просмотр файла» беттерін құру

13.7 Жүктелген файлмен бөлісу функциясын әзірлеу

Мақсаты: Каталогтар және файлдармен жұмыс істеуге арналған негізгі беттерді құрастыру бойынша нұсқаулықтармен танысу және әзірлеуді үйрену, «Каталогтар» және «Файлдар» атты модельдерін толықтыру, жаңа контроллерді құру және индекс файлын әзірлеу бойынша жұмыстарын қарастыру.

Кілтті сөздер: калатогты құру, файлды жүктеу, жүктелген файлды көру, файлмен бөлісу, GridView виджеті, ActiveForm виджеті, модель, контроллер, индекс файлы.

13.1 «GridView» және «ActiveForm» виджеттерін орнату

Бұлттық қоймада құрылған каталогтар және жүктелген файлдармен жұмыс істеу үшін жобаның функционалдылығын одан әрі байыту мақсатында әдемі виджеттердің бірі болып табылатын кесте түріндегі kartik-тің «GridView» виджетін және сол автордың «ActiveForm» атты форма виджетін орнату керек.

Ең алдымен «GridView» виджетін орнату үшін Open Server вебсерверіндегі Composer менеджерін іске қосып, келесі командаларды орындау қажет [29]:

```
cd C:\OpenServer\domains\academ-cloud.kz
composer require kartik-v/yii2-grid "dev-master"
```

Бұдан кейін келесі команда арқылы «ActiveForm» виджетін орнатамыз:

Сотрозег менеджеріндегі жоғарыда аталған виджеттерді орнату процедураларының кескіні төмендегі сурет 13.1-де көрсетілген.



Сурет 13.1. Composer менеджері арқылы «GridView» және «ActiveForm» виджеттерін орнату

Аталған виджеттер орнатылып болған соң, «GridView» виджетін кейіннен модуль ретінде қолданатындықтан, оны жобаның конфигурациялық файлында көрсетіп кеткен жөн. Сол себепті C:\OpenServer\domains\academ-cloud.kz\config каталогына өтіп, ондағы web.php файлын ашамыз да, келесі кодты \$config атты бөлімінің кез келген бөлігіне орналастырамыз:

```
'modules' => [
    'gridview' => ['class' => 'kartik\grid\Module'],
],
```

13.2 Каталогты құру және файлдарды жүктеу бойынша қосымша модельдерді құру

Алдыңғы тарауда Gii модулімен «Каталог» және «Файл» атты екi модельді құрастырып, оларды өзгерткен болатынбыз. Аталған модельдерге жаңа каталогты құру және таңдалған каталогқа файлды жүктеу бойынша қосымша функциялары бар ішкі модельдерді құруымыз қажет. Алдымен жаңа каталогты құру бойынша 0jcsvif модельді құрастырайық. Ол үшін жаңадан FolderForm.php файлын C:\OpenServer\domains\academ-cloud.kz\models\forms каталогында құрастырамыз да, келесі кодтар жолына сәйкес енгіземіз:

```
<?php
    namespace app\models\forms;
    use app\models\Folder;
    use yii\base\Model;
    use Yii;
    class FolderForm extends Model
        public $name;
        public $parent id;
        private $ folder;
        public function construct (Folder $folder, array $config
= [])
        {
            $this-> folder = $folder;
            parent:: construct($config);
        }
        public function rules()
        {
            return [
                ['name', 'trim'],
                ['name', 'required'],
                ['name', 'string', 'max' => 255],
                ['parent id', 'default'],
                ['parent id', 'integer'],
                ['parent_id', 'exist', 'targetClass' =>
Folder::className(), 'targetAttribute' => 'id'],
            ];
        }
        public function save()
        {
            if (!$this->validate()) {
                return false;
            }
```

```
$this->folder->name = $this->name;
        $this->folder->owner id = Yii::$app->user->id;
        $this->folder->parent id = $this->parent id;
        return $this->folder->save();
    }
   public function attributeLabels()
    {
        return [
            'name' => 'Название',
        ];
    }
   public function getFolder()
    {
        return $this-> folder;
    }
}
```

Бұдан кейін ағымдағы каталогта таңдалған каталогқа жаңа файлдарды жүктеу бойынша функциялары бар UploadFileForm.php атты жаңа файлды құрастырып, онда төменде көрсетілгендей, кодтарды енгізетін боламыз:

```
<?php
   namespace app\models\forms;
   use app\models\File;
   use Yii;
   use yii\base\Model;
   use yii\helpers\ArrayHelper;
   use yii\helpers\FileHelper;
   use yii\web\UploadedFile;
   class UploadFileForm extends Model
    {
       public $folder_id;
       public $files;
       public $ fileModel;
       public function construct(File $fileModel, array
config = [])
        {
            $this-> fileModel = $fileModel;
           parent:: construct($config);
        }
       public function rules()
        {
            return [
```

```
['folder id', 'integer'],
                ['files', 'safe'],
                ['files', 'file', 'skipOnEmpty' => false,
'maxSize' => 10 * 1024 * 1024, 'maxFiles' => 0],
                Γ
                    'files',
                    'validateTotalSize',
                    'skipOnEmpty' => false,
                    'params' => [
                         'maxTotalSize' => 200 * 1024 * 1024,
                    1
                ],
            1;
        }
        public function validateTotalSize($attribute, $params,
$validator)
        {
            $totalSize = File::find()->where(['owner id' =>
Yii::$app->user->id])->sum('size');
            $totalSize = array sum(ArrayHelper::getColumn($this-
>files, 'size')) + $totalSize;
            if ($totalSize > $params['maxTotalSize']) {
                $this->addError($attribute, 'Превышен
максимальный размер файлов на диске. ');
            }
        }
        public function upload()
            if (!$this->validate()) {
                return false;
            }
            $file = $this->files[0];
            $dirPath = '@app/uploads/users/' . Yii::$app->user-
>id . '/' . Yii::$app->security->generateRandomString(2);
            $fileName = Yii::$app->security-
>generateRandomString(10) . '.' . $file->extension;
            FileHelper::createDirectory(Yii::getAlias($dirPath));
            $file->saveAs(Yii::getAlias($dirPath) . '/' .
$fileName);
            $this->fileModel->name = $file->name;
            $this->fileModel->owner id = Yii::$app->user->id;
            $this->fileModel->path = $dirPath . '/' . $fileName;
            $this->fileModel->size = $file->size;
            $this->fileModel->folder id = $this->folder id;
            return $this->fileModel->save();
        }
        public function getFileModel()
            return $this-> fileModel;
```

```
}
public function attributeLabels()
{
    return [
        'files' => 'Файлы',
    ];
}
```

13.3 Каталог пен файлдар модельдері бойынша ортақ контроллерді құру

Жоғары тарауда Gii модулімен сәйкес «Каталог» пен «Файл» модельдері үшін өзінің жеке контроллерлерін автоматты түрде генерациялаған болатынбыз. Аталған FolderController.php және FileController.php файлдары жобаның C:\OpenServer\domains\academ-cloud.kz\controllers атты каталогында құрастырып отырған веб-қосымшамызда жатыр, алайда біз каталогты құру мен файлдарды жүктеу функцияларын бір бетте оларды асыратындықтан, бір контроллермен жүзеге басқарылатындай етіп жасауымыз керек, сол себепті аталған екі контроллердағы кейбір функцияларды пайдана отырып, өзіміздің DriveController.php атты контроллерімізді құрастыратын боламыз. Ол үшін жоғарыдағы каталогта жаңадан DriveController.php файлын құрып, келесі кодтарды тереміз:

```
<?php
```

}

```
namespace app\controllers;
use app\models\File;
use app\models\Folder;
use app\models\forms\FolderForm;
use app\models\forms\UploadFileForm;
use Yii;
use yii\data\ArrayDataProvider;
use yii\filters\AccessControl;
use yii\filters\VerbFilter;
use yii\helpers\ArrayHelper;
use yii\web\Controller;
use yii\web\ForbiddenHttpException;
use yii\web\NotFoundHttpException;
use yii\web\UploadedFile;
class DriveController extends Controller
{
```

```
public function behaviors()
        {
            return [
                 'access' => [
                     'class' => AccessControl::className(),
                     'rules' => [
                         [
                             'allow' => true,
                             'roles' => ['@'],
                         ],
                    ],
                ],
                 'verbs' => [
                     'class' => VerbFilter::className(),
                     'actions' => [
                         'share-file' => ['POST'],
                         'shield-file' => ['POST'],
                         'delete-file' => ['POST'],
                    ],
                ],
            ];
        }
        public function actionIndex($folder id = null)
        {
            $folderModel = Folder::findOne($folder id);
            $folderModels = Folder::find()
                ->where([
                     'owner id' => Yii::$app->user->getId(),
                     'parent id' =>
ArrayHelper::getValue($folderModel, 'id'),
                ])
                ->all();
            $fileModels = File::find()
                ->where([
                     'owner_id' => Yii::$app->user->getId(),
                     'folder id' =>
ArrayHelper::getValue($folderModel, 'id'),
                ])
                ->all();
            $dataProvider = new ArrayDataProvider([
                'allModels' => array merge($folderModels,
$fileModels),
                'pagination' => false,
            ]);
            return $this->render('index', [
                'dataProvider' => $dataProvider,
                'folderModel' => $folderModel,
            ]);
        }
        public function actionCreateFolder($parent id = null)
        {
```

```
$folderModel = new Folder();
            $folderForm = new FolderForm($folderModel);
            $folderForm->parent id = $parent id;
            if ($folderForm->load(Yii::$app->request->post()) &&
$folderForm->save()) {
                return $this->redirect([
                    'index',
                    'folder id' => $folderForm->parent id,
                ]);
            }
            return $this->render('create-folder', [
                'folderForm' => $folderForm,
            1);
        }
        public function actionDeleteFolder($id)
            $folderModel = Folder::findOne($id);
            if ($folderModel === null) {
                throw new NotFoundHttpException();
            if ($folderModel->owner id !== Yii::$app->user->id) {
                throw new ForbiddenHttpException();
            $parent id = $folderModel->parent id;
            $folderModel->delete();
            return $this->redirect([
                'index',
                'folder id' => $parent id,
            ]);
        }
        public function actionUploadFile($folder id = null)
        {
            $fileModel = new File();
            $uploadFileForm = new UploadFileForm($fileModel);
            $uploadFileForm->folder id = $folder id;
            if (Yii::$app->request->isPost) {
                $uploadFileForm->files =
UploadedFile::getInstances($uploadFileForm, 'files');
                if ($uploadFileForm->upload()) {
                    return $this->asJson([]);
                }
                return $this->asJson(['error' => $uploadFileForm-
>getFirstError('files')]);
            }
            return $this->render('upload-file', [
                'uploadFileForm' => $uploadFileForm,
            ]);
        }
        public function actionDownloadFile($id)
        {
```

```
$fileModel = File::findOne($id);
            if ($fileModel === null) {
                throw new NotFoundHttpException();
        }
            if ($fileModel->owner id !== Yii::$app->user->id &&
!$fileModel->is shared) {
                throw new ForbiddenHttpException();
            $fileModel->updateCounters(['download count' => 1]);
            $path = Yii::getAlias($fileModel->path);
            return Yii::$app->response->sendFile($path,
$fileModel->name);
        }
       public function actionPreviewFile($id)
            $fileModel = File::findOne($id);
            if ($fileModel === null) {
                throw new NotFoundHttpException();
            }
            if ($fileModel->owner id !== Yii::$app->user->id &&
!$fileModel->is shared) {
                throw new ForbiddenHttpException();
            }
            $fileModel->updateCounters(['view count' => 1]);
            return $this->render('preview-file', [
                'fileModel' => $fileModel,
            ]);
        }
       public function actionDeleteFile($id)
        {
            $fileModel = File::findOne($id);
            if ($fileModel === null) {
                throw new NotFoundHttpException();
            }
            if ($fileModel->owner id !== Yii::$app->user->id) {
                throw new ForbiddenHttpException();
            }
            $fileModel->delete();
            return $this->redirect([
                'index',
                'folder id' => $fileModel->folder id,
            ]);
        }
   }
```

Қажетті функцияларды көрсетіп болған соң алдында құрастырылған FolderController.php және FileController.php файлдарын өшіруге болады, себебі бұл контроллерлермен ештеңе

басқарылмайды. Бұл файлдардың қажеті болмаған соң Gii модуліндегі GRUD генераторымен автоматты түрде құрастырылған C:\OpenServer\ domains\academ-cloud.kz\views каталогындағы «folder» және «file» каталогтарын да өшіруге болады.

Жоғарыда құрастырылған DriveController.php атты контроллерда каталогты құру және өшіру, файлды бұлтқа жүктеу, көру, бұлттан жүктеу, сонымен қатар файлды өшіру сияқты функцияларды көруге болады.

13.4 Индекс файлын әзірлеу және өзгерту

Каталог пен файлдар бойынша ортақ контроллерді құрастырып болған соң олардың ортақ бір функциялары бар бетін кескіндейтін индекс файлын құру қажеттілігі туындайды.

Контроллердің атауы «Drive» сөзінен басталғандықтан, жобаның C:\OpenServer\domains\academ-cloud.kz\views каталогына көшіп, «drive» атты каталогын құрамыз да, ондағы индекс файлына келесі кодтарды тереміз:

<?php

```
use app\models\File;
use app\models\Folder;
use kartik\grid\ActionColumn;
use kartik\grid\GridView;
use yii\bootstrap\Html;
use yii\data\DataProviderInterface;
use yii\helpers\ArrayHelper;
use yii\helpers\Url;
use yii\web\View;
/* @var View $this */
/* @var DataProviderInterface $dataProvider */
/* @var Folder $folderModel */
$this->title = 'Moй диск';
if ($folderModel !== null) {
    $this->params['breadcrumbs'][] = [
        'label' => $this->title,
        'url' => ['index'],
    ];
} else {
    $this->params['breadcrumbs'][] = $this->title;
if ($folderModel !== null) {
```

```
$folderModelTmp = $folderModel;
        $breadcrumbs = [];
        do {
            $breadcrumbs[] = [
                'label' => $folderModelTmp->name,
                'url' => ['index', 'folder id' =>
$folderModelTmp->id],
            ];
            $folderModelTmp = $folderModelTmp->parent;
        } while ($folderModelTmp !== null);
        $breadcrumbs = array reverse($breadcrumbs);
        $this->params['breadcrumbs'] = array merge($this-
>params['breadcrumbs'], $breadcrumbs);
    ?>
    <div class="box box-default color-palette-box">
        <div class="box-body">
            <div class="row">
                <div class="col-md-12">
                     <div>
                         <div class="pull-left">
                             <?= Html::a('Создать папку', [
                                 'create-folder',
                                 'parent id' =>
ArrayHelper::getValue($folderModel, 'id'),
                             ], [
                                 'class' => 'btn btn-primary',
                             1) ?>
                             <?= Html::a('Загрузить файл', [
                                 'upload-file',
                                 'folder id' =>
ArrayHelper::getValue($folderModel, 'id'),
                             ], [
                                 'class' => 'btn btn-success'
                             ]) ?>
                         </div>
                         <div class="clearfix"></div></div>
                     </div>
                         ...Кестені орналастыру
                </div>
            </div>
        </div>
    </div>
```

Терген кодымызды тексеру үшін браузерде <u>http://academcloud.kz/drive/index</u> сілтемесін тереміз немесе басты мәзір жолының (жоғары тарауларда бұл мәзір пунктінде бұл индекс файлына алдынала сілтеме жасалынды) «Менің дискім» пунктін таңдаймыз, нәтижесінде сурет 13.2-де көрсетілгендей, бұлттағы каталогтармен және файлдармен жұмыс істеу бойынша индекс файлының қаңқасы кескінделетін болады.

| •• Мой диск × | _ | ▲ – □ × | | |
|--|-------------------------------|-------------------|--|--|
| ← → C [] academ-cloud.kz/drive/index : | | | | |
| Academ Cloud | | 😂 admin | | |
| eadmin admin | Мой диск | Главная > Мойдиск | | |
| 42) Мойдиск | Создать папку Загрузить файл | | | |
| Пользователи | | | | |
| Э История | | | | |
| 0 нас | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | Облачные технологии © 2017 г. | 🌱 - < | | |

Сурет 13.2. Индекс файлының қаңқасы

Егер жоғарыда көрсетілген кодтар жолына назар аударатын болсаңыз, бұл жерде каталог пен файл бойынша модельдер, алдынала орнатылған виджеттер қолданылған. Бірақ құрылған каталогтар мен ондағы файлдар тізімін көрсететін, сондай-ақ DriveController.php контроллерындағы қарастырылған функциялары бар кестенің кескіні жоқ. Сол себепті кодтың арасында кесте кодының орналасу жерінде осы тарауда алдын-ала орнатылған GridView виджетін қолдана отырып, мына кодтар жолдарын тереміз:

```
'style' => 'cursor: pointer;', ];
            }
            if ($model instanceof File) {
                return [
                     'id' => $model['id'],
                     'ondblclick' => 'document.location.href="' .
Url::to([
                             'preview-file',
                             'id' => $model->id
                         ]). '"',
                     'style' => 'cursor: pointer;', ];
            }
            return [];
        },
        'hover' => true,
        'columns' => [
            Γ
                 'attribute' => 'name',
                 'label' => 'Название',
            ],
            Γ
                 'label' => 'Pasmep',
                 'format' => ['shortSize', 2],
                 'value' => function ($model) {
                     if ($model instanceof File) {
                         return $model->size;
                     }
                     return null;
                 },
                 'options' => [
                     'width' => 100,
                ],
            ],
            ſ
                 'attribute' => 'created_at',
                 'format' => 'datetime',
                 'label' => 'Время',
                 'options' => [
                     'width' => 150,
                ],
            ],
            Γ
                 'class' => ActionColumn::className(),
                 'options' => ['width' => 125],
                 'template' => '{download-file} {delete-folder}
{share-file} {shield-file} {delete-file}',
                 'visibleButtons' => [
                     'delete-folder' => function ($model, $key,
$index) {
                         if ($model instanceof Folder) {
                             return true;
                         return false;
```

```
},
                     'download-file' => function ($model, $key,
$index) {
                         if ($model instanceof File) {
                             return true;
                         }
                         return false;
                     },
                     'delete-file' => function ($model, $key,
$index) {
                         if ($model instanceof File) {
                             return true;
                         }
                         return false;
                     },
                ],
                 'buttons' => [
                     'delete-folder' => function ($url, $model,
$key) {
                         if ($model instanceof Folder) {
                             return Html::a('<button type="button"</pre>
class="btn btn-danger btn-sm" title="Удалить"><span
class="glyphicon glyphicon-trash"
style="color:white;"></span></button>', [
                                  'delete-folder',
                                  'id' => $model->id,
                             ], [
                                  'data' => [
                                      'confirm' => 'Дейтвительно
удалить папку и все файлы в ней?',
                                      'method' => 'post',
                                  ], ]);
                         }
                         return null;
                     },
                     'download-file' => function ($url, $model,
$key) {
                         if ($model instanceof File) {
                             return Html::a('<button type="button"</pre>
class="btn btn-primary btn-sm" title="Скачать">>>span class="fa
fa-cloud-download" style="color:white;"></span></button>', [
                                  'download-file',
                                  'id' => $model->id,
                             1);
                         }
                         return null;
                     },
                     'delete-file' => function ($url, $model,
$key) {
                         if ($model instanceof File) {
                             return Html::a('<button type="button"</pre>
class="btn btn-danger btn-sm" title="Удалить">>>span
```

GridView виджетін қолданып, браузердегі бетті жаңартатын болсақ, төмендегі сурет 13.3-те көрсетілгендей, каталогтар мен файлдардың тізімі болатын кесте кескінін көруге болады.



Сурет 13.3. Жаңартылған индекс файлының кескіні

13.5 Бұлтта каталог құру бетін құрастыру

Жоғарыдағы бөлімде сурет 13.3-те көрсетілгендей, каталогтармен және файлдармен жұмыс істеуге арналған негізгі индекс файлын әзірлеген болатынбыз. Аталған беттің жоғарғы бөлігінде «Бума жасау» және «Файлды жүктеу» атты арнайы батырмалары бар, index.php файлының кодын ашатын болсақ, аталған батырмаларды шерткен кезде сәйкес «create-folder» және «upload-file» беттері ашылатынын көруге болады.

Сол себепті index.php файлы жатқан каталогта (біздің жағдайда C:\OpenServer\domains\academ-cloud.kz\views\drive) жаңадан createfolder.php және _folder-form.php файлдарын құру керек. Бұл файлдардың атаулары алдын-ала тағайындалған болатын, оны DriveController.php контроллерінің actionCreateFolder функциясынан көруге болады.

Алдымен create-folder.php файлын ашып, келесі кодты тереміз (көрші жатқан C:\OpenServer\domains\academ-cloud.kz\views\user каталогында create.php файлынан кодты көшіріп алып, өзімізге қажетті өзгертеміз):

<?php

```
use app\models\forms\FolderForm;
use yii\web\View;
/* @var View $this */
/* @var FolderForm $folderForm */
$this->title = 'Создание папки';
?>
<div class="box box-default color-palette-box">
    <div class="box-body">
         <div class="row">
             <div class="col-md-12">
                  <?= $this->render('_folder-form', [
    'folderForm' => $folderForm,
                  ]) ?>
             </div>
         </div>
    </div>
</div>
```

Содан кейін _folder-form.php файлын ашып, келесі кодты тереміз (бастапқы кодты C:\OpenServer\domains\academ-cloud.kz\views\user каталогында _form.php файлынан көшіріп алып, қажетімізше өзгертеміз):

<?php

use app\models\forms\FolderForm;

```
use kartik\form\ActiveForm;
use yii\bootstrap\Html;
use yii\web\View;
/* @var View $this */
/* @var FolderForm $folderForm */
?>
<div class=" folder-form">
    <?php $form = ActiveForm::begin(); ?>
    <?= Html::activeHiddenInput($folderForm, 'parent id') ?>
    <?= $form->field($folderForm, 'name')->textInput() ?>
    <div class="form-group">
        <?= Html::submitButton('Сохранить', [
            'class' => 'btn btn-success',
        ]) ?>
    </div>
    <?php ActiveForm::end(); ?>
</div>
```

Нәтижесінде индекс файлының «Создать папку» батырмасын шерткен кезде жаңа құрастырылған «Создание папки» бетінің кескіні төмендегі сурет 13.4-ке сәйкес болады. Оның атауын өз еркімізше енгізіп, төмендегі «Сохранить» батырмасы арқылы деректер базасында сақталуын және кестеде көрсетілуін тексеруге болады.



Сурет 13.4. «Создание папки» бетінің кескіні

13.6 Бұлтқа файлдарды жүктеу және қарап шығуды ұйымдастыру

Каталогты құру бетін әзірлеген соң индекс файлындағы «Загрузить файл» батырмасын шерткен кезде ашылатын «uploadfile» бетін құрастыру керек. Ол үшін index.php файлы жатқан каталогта жаңадан upload-file.php және _upload-file-form.php файлдарын құру керек. Бұл файлдардың атаулары да алдын-ала тағайындалған болатын, оны DriveController.php контроллерінің аctionUploadFile функциясынан көруге болады.

Алдымен upload-file.php файлын ашып, төмендегі кодты тереміз (қажет кодты көрші жатқан create-folder.php файлынан көшіріп алып, өзімізге қажетті өзгертеміз):

<?php

```
use app\models\forms\UploadFileForm;
use yii\web\View;
/* @var View $this */
/* @var UploadFileForm $uploadFileForm */
$this->title = 'Загрузка файла';
?>
<div class="box box-default color-palette-box">
    <div class="box-body">
        <div class="row">
            <div class="col-md-12">
                <?= $this->render('_upload-file-form', [
                'uploadFileForm' => $uploadFileForm,
                ]) ?>
            </div>
        </div>
    </div>
</div>
```

Содан кейін _upload-file-form.php файлын ашып, келесі кодты орналастырамыз (бастапқы кодты ағымдағы каталогта жатқан _folder-form.php файлынан көшіріп алып, өзімізге қажетті өзгертеміз):

<?php

```
use app\models\forms\UploadFileForm;
   use kartik\file\FileInput;
   use kartik\form\ActiveForm;
   use yii\bootstrap\Html;
   use yii\helpers\Url;
   use yii\web\JsExpression;
   use yii\web\View;
   /* @var View $this */
   /* @var UploadFileForm $uploadFileForm */
   ?>
   <?php $form = ActiveForm::begin(['options' => ['enctype' =>
'multipart/form-data']]); ?>
   <?= Html::activeHiddenInput($uploadFileForm, 'folder id') ?>
   <?= $form->field($uploadFileForm, 'files[]')-
>widget(FileInput::classname(), [
        'options' => [
            'multiple' => true,
        ],
        'language' => 'ru',
        'pluginOptions' => [
            'previewFileType' => 'any',
            'uploadUrl' => Url::to(['upload-file', 'folder id' =>
$uploadFileForm->folder id]),
            'maxFileSize' => 10 * 1024 * 1024,
        ],
        'pluginEvents' => [
            'filebatchselected' => new
JsExpression('function(event, files) {
                        $(this).fileinput("upload");
                    }'),
        ],
   ]) ?>
   <?php ActiveForm::end(); ?>
```

Жоғары тарауда алдын-ала орнатылған kartik-тiң FileInput виджетi қолданылып отырғанына назар аударуға болады. Аталған файлдарды құрастырған соң индекс файлының «Загрузить файл» батырмасын шерткен кезде жаңа құрастырылған «Загрузка файла» бетiнiң кескiнi төмендегi сурет 13.5-ке сәйкес болады. Беттiң төмендегi бөлiгiнде орналасқан «Выбрать...» батырмасы арқылы қажеттi файлдарды жүктеуге болады, жүктелген файлдар орналасқан каталогқа көшiрiлетiн болады.



Сурет 13.5. «Загрузка файла» бетінің кескіні

Қажетті файлдарды жүктеген соң веб-қосымшаның «Менің дискім» мәзір пунктін шерту керек, нәтижесінде төмендегі сурет 13.6-да көрсетілгендей, жоғарыда құрылған каталог пен жүктелген файлдардың тізімі бар кестені көре аламыз.

| Мой диск × | Δ | | 2 | – – × | | |
|-------------------------------------|-------------------------------|-----------|---------------------|-------------------|--|--|
| ← → C D academ-cloud.kz/drive/index | | | | | | |
| Academ Cloud | | | | 😂 admin | | |
| edmin admin | Мой диск | | | Главная > Мойдиск | | |
| 4∄ Мойдиск | Создать папку Загрузить файл | | | | | |
| 嶜 Пользователи | Название | Размер | Время | Действия | | |
| Э История | Каталог №1 | | 19.10.2018 00:30:38 | Ê | | |
| 6) Онас | 01.png | 29.34 КиБ | 19.10.2018 01:07:04 | | | |
| | 1.png | 19.76 КиБ | 19.10.2018 01:07:04 | | | |
| | 02.png | 41.61 КиБ | 19.10.2018 01:07:04 | a î | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | Облачные технологии © 2017 г. | | | ? | | |

Сурет 13.6. Құрылған каталог пен жүктелген файлдардың тізімі бар «Мой диск» беті

файлды өшіру, файлды жүктеу Каталог бойынша пен функционал алдын-ала DriveController.php файлында әзірленген еді. Егер кестедегі каталог жолын тышқан батырмасымен екі рет шертетін болсақ, сол каталогқа көшетін боламыз. Ал жүктелген файлды екі рет шертетін болсақ, онда қатені көрсететін беттің жүктелгенін көре аламыз, яғни бұл жағдайда қосымша жүктелген файлдың параметрлерін көруге мүмкіндік беретін preview-file.php файлын құрастыру қажеттілігі туындайды (бұл функционал да DriveController.php файлында actionPreviewFile алдын-ала функциясында әзірленген), ол үшін ағымдағы каталогқа previewfile.php файлын құрып, келесі кодты тереміз:

<?php

```
use app\models\File;
use yii\bootstrap\Html;
use vii\web\View;
use yii\widgets\DetailView;
/* @var View $this */
/* @var File $fileModel */
$this->title = $fileModel->name;
?>
<div class="box box-default color-palette-box">
    <div class="box-body">
        <div class="row">
            <div class="col-md-12">
                 <?= DetailView::widget([</pre>
                     'model' => $fileModel,
                     'attributes' => [
                         'name',
                         'size:shortSize',
                         'view count',
                         'download count',
                     ],
                 ]) ?>
                 <?= Html::a('Скачать файл', [
                     'download-file',
                     'id' => $fileModel->id,
                 ], [
                     'class' => 'btn btn-primary',
```

```
]) ?>
</div>
</div>
</div>
</div>
```

Нәтижесінде кестедегі жүктелген кез келген файлды екі рет шерткен кезде ағымдағы файлдың параметрлерін (атап айтқанда файлдың атауы, оның мөлшері, көру және жүктеу саны) көруге мүмкіндік беретін төмендегі сурет 13.7-ге сәйкес бет ашылатынын көреміз. Сонымен қатар беттің төменгі бөлігінде орналасқан «Скачать файл» батырмасы арқылы ағымдағы файлды бұлттан жүктеуге болады.

| 🚏 01.png 🛛 🗙 | | ▲ – □ × |
|--|-------------------------------|------------|
| igstarrow igstarro | /drive/preview-file?id=5 | : |
| Academ Cloud | | 😂 admin |
| admin admin | 01.png | |
| 4∄ Мойдиск | Название | 01.png |
| 嶜 Пользователи | Размер | 29.34 КиБ |
| Э История | Просмотров | 1 |
| O нас | Загрузок | 0 |
| | Скачать файл | |
| | | |
| | Облачные технологии © 2017 г. | * < |



13.7 Жүктелген файлмен бөлісу функциясын әзірлеу

Жоғарыда көрсетілген сурет 13.7-ге сәйкес бұлтқа жүктелген файлдар тізімі бар бетте сол файлмен бөлісу функциясын әзірлеу керек. Ол үшін басқарылатын DriveController.php файлына (аталған файл C:\OpenServer\domains\academ-cloud.kz\controllers каталогында жатыр) төменде көрсетілген екі функцияны қосу қажет:

```
public function actionShareFile($id)
{
```

```
$fileModel = File::findOne($id);
        if ($fileModel === null) {
            throw new NotFoundHttpException();
        }
        if ($fileModel->owner id !== Yii::$app->user->id) {
            throw new ForbiddenHttpException();
        }
        $fileModel->is shared = true;
        if (!$fileModel->save()) {
            throw new ModelValidationException($fileModel);
        }
        $url = Yii::$app->urlManager-
>createAbsoluteUrl(['/drive/download-file', 'id' => $fileModel-
>id]);
        Yii::$app->session->addFlash('success', "Ссылка на
ckayuBaHue: $url");
        return $this->redirect([
            'index',
            'folder id' => $fileModel->folder id,
        ]);
    }
   public function actionShieldFile($id)
    {
        $fileModel = File::findOne($id);
        if ($fileModel === null) {
            throw new NotFoundHttpException();
        }
        if ($fileModel->owner id !== Yii::$app->user->id) {
            throw new ForbiddenHttpException();
        }
        $fileModel->is shared = false;
        if (!$fileModel->save()) {
            throw new ModelValidationException($fileModel);
        }
        return $this->redirect([
            'index',
            'folder id' => $fileModel->folder id,
        ]);
    }
```

Аталған функцияда ModelValidationException атты валидация моделі көрсетілген, сол себепті аталған валидация моделін құруымыз керек. Ол үшін жобада жаңа «exceptions» атты каталогын құрып, онда жаңа ModelValidationException.php файлын құрып, келесідей кодты орналастырамыз:

<?php

namespace app\exceptions;

```
use yii\base\Exception;
use yii\base\Model;
class ModelValidationException extends Exception
{
    public function getName()
    {
        return 'ModelValidationException';
    }
    public function __construct($model, $code = 0, Exception
$previous = null)
    {
        parent::__construct(print_r($model->getErrors(),
    true));
    }
}
```

Содан кейін қайта DriveController.php файлына көшіп, оның жоғарғы жағына құрылған ModelValidationException.php файлына сілтемесін келесі кодқа сәйкес көрсету керек:

use app\exceptions\ModelValidationException;

DriveController.php файлында көрсетілген actionShareFile және actionShieldFile функцияларына сәйкес C:\OpenServer\domains\academ-cloud.kz\views\drive каталогында жатқан index.php файлына (GridView виджетіндегі ActionColumn класының visibleButtons параметріне download-file функциясынан кейін) келесі кодтарды қосамыз:

```
'share-file' => function ($model, $key, $index) {
    if ($model instanceof File) {
        return !$model->is_shared;
    }
    return false;
},
'shield-file' => function ($model, $key, $index) {
    if ($model instanceof File) {
        return $model->is_shared;
    }
    return false;
},
```

Бұдан кейін ағымдағы index.php файлының GridView виджетіндегі ActionColumn класының buttons параметріне «download-file» функциясынан кейін келесі кодтарды тереміз:

```
'share-file' => function ($url, $model, $key) {
        if ($model instanceof File) {
            return Html::a('<button type="button" class="btn btn-
success btn-sm" title="Поделится">>>span class="fa fa-share-alt"
style="color:white;"></span></button>', [
                'share-file',
                'id' => $model->id,
            ], [
                'data' => [
                     'confirm' => 'Хотите поделиться данным
файлом?',
                    'method' => 'post',
                ],
            ]);
        }
        return null;
    },
    'shield-file' => function ($url, $model, $key) {
        if ($model instanceof File) {
            return Html::a('<i class="fa fa-shield"></i>', [
                'shield-file',
                'id' => $model->id,
            ], [
                'data' => [
                     'confirm' => 'Закрыть доступ к данному
файлу?',
                    'method' => 'post',
                ],
            ]);
        }
        return null;
},
```

Нәтижесінде браузерде index.php файлын жаңартқан кезде (вебқосымшаның басты мәзіріндегі «Менің дискім» пунктін таңдаған кезде немесе браузерде <u>http://academ-cloud.kz/drive/index</u> сілтемесін терген жағдайда) файлмен бөлісу және бөліскен файлға сілтемені жабу атты қосымша екі батырмасы пайда болғанын көруге болады. Төмендегі сурет 13.8-де кескінделгендей, тізімдегі кез келген файлдың «Бөлісу» атты батырмасын шерткен кезде беттің жоғарғы бетінде ағымдағы файлды бұлттан жүктеу бойынша сілтемесі көрсетілетін болады.

Жоғарыда көрсетілген сілтемені көшіріп алып, оны браузердің жаңа бөлімінде көрсететін болса, сол файлды жүктеуге болатынын көруге болады. Егер ағымдағы файлдың сілтемесін жабу үшін жаңадан пайда болған «Закрыть ссылку» батырмасын шерту керек. Егер құрылған каталогты өшіретін болсақ, ондағы барлық файлдар да өшіретін болады. Әуел баста файлдарды жүктеген кезде жобада жаңадан «uploads» атты каталог құрылатын болады да, онда жүктелген барлық файлдар көшірілетін болады.

| Мойдиск х | L drive linder | | | ▲ – □ × |
|---|--|-----------|---------------------|-------------------|
| Academ Cloud | = | | | 😂 admin |
| admin | Мой диск | | | Главная > Мойдиск |
| 42 Мойдиск | ✓ Ссылка на скачивание: http://academ-cloud.kz:81/drive/download-file?id=5 | | | × |
| 嶜 Пользователи | | | | |
| Октория Окас | Создать палку Загрузить файл Название | Размер | Время | Действия |
| | Katanor №1 | | 19.10.2018 00:30:38 | â |
| | 01.png | 29.34 КиБ | 19.10.2018 01:07:04 | |
| | 1.png | 19.76 КиБ | 19.10.2018 01:07:04 | a < i |
| | 02.png | 41.61 КиБ | 19.10.2018 01:07:04 | <u>م</u> |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | Облачные технологии © 2017 г. | | | ? < |

Сурет 13.8. Жүктелген файлмен бөлісу және сілтемені алып тастау функциялары бар «Менің дискім» бетінің көрінісі

Бақылау сұрақтары:

1. «GridView» және «ActiveForm» виджеттері қалай орнатылады?

2. Каталогты құру және файлдарды жүктеу бойынша қосымша модельдер қалай құрылады?

3. Каталог пен файлдар модельдері бойынша ортақ контроллерді құру қалай ұйымдастырылады?