

14. Бұлттық қойманың «Қолданушы әрекеттері» атты функционалдық мүмкіндігін әзірлеу

Жоспар:

- 14.1 Деректер базасында «history» атты кестесін құрастыру
- 14.2 Құрастырылған «history» кестесі үшін сәйкес әрекеттер моделін қалыптастыру және оның GRUD генерациясы
- 14.3 «Select2» және «Export Menu» виджеттерін орнату
- 14.4 Құрастырылған қолданушы әрекеттері моделін және оның сүзгілеу моделін өзгерту
- 14.5 «Қолданушы әрекеттерінің тізімі» бетін өзгерту
- 14.6 Қолданушы әрекеттерінің контроллерін өзгерту
- 14.7 Қолданушы әрекеттерін тіркейтін модульді әзірлеу

Мақсаты: Қолданушы әрекеттеріне байланысты беттер мен контроллерді өзгертуді және әрекеттерді автоматты түрде базада тіркейтін модульді әзірлеуді үйрену.

Кілтті сөздер: деректер базасы, кестені құру, қолданушы әрекеттері, модель, контроллер, CRUD, беттерді өзгерту, bootstrap модулі.

14.1 Деректер базасында «history» атты кестесін құрастыру

Веб-қосымшаның мәзір жолында орналасқан «Өзгертулер тарихы» деген пунктінде жүйеде тіркелген қолданушылардың әрекеттері (жүйеге кіру, жүйеден шығу, қолданушыны қосу, өзгерту және өшіру, файлды қосу, өзгерту және өшіру) бойынша жазбалары көрсетілетін болады. Бұл жазбаларды сақтау үшін деректер базасында сәйкес кестені құру керек. Сол себепті алдымен «oblako» атты құрастырылған деректер базасында жаңадан «history» атты кестесін құрастыруымыз керек. Ол үшін Open Server веб-серверінің мәзір жолына кіріп, «Дополнительно» → «MySQL менеджер» пункттерін таңдаймыз да, HeidiSQL бағдарламалық құралын іске қосып, құрылымдары төмендегі сурет 14.1-ге сәйкес «history» атты кестесін құруымыз қажет.

#	Имя	Тип данных	Длина/Значения	Беззнаковое	Разрешить NULL	Zerofill	По умолчанию
1	id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	user_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет значения по умолчанию
3	action	VARCHAR	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
4	comment	VARCHAR	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
5	created_at	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
6	updated_at	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

Сурет 14.1. Құрастырылатын «history» кестесінің құрылымы

Құрастырылатын «history» кестесінің CREATE кодының үлгісі төмендегі кодтар жолында көрсетілген:

```
CREATE TABLE `history` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `user_id` INT(11) NOT NULL,
  `action` VARCHAR(255) NULL DEFAULT NULL COMMENT '1 - вход в
систему, 2 - выход из системы, 3 - создание папки, 4 - удаление
папки, 5 - создание файла',
  `comment` VARCHAR(255) NULL DEFAULT NULL,
  `created_at` INT(11) NULL DEFAULT NULL,
  `updated_at` INT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  INDEX `user` (`user_id`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=326
;
```

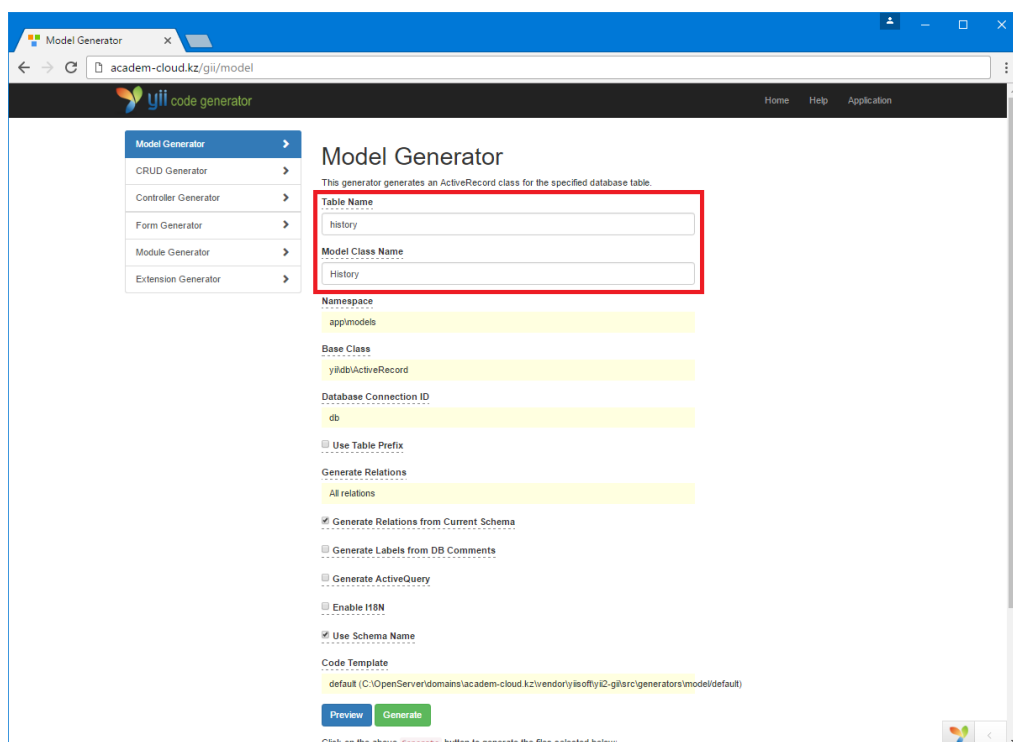
Бұл сәйкес кодтар жолын көшіріп алып, жаңа кестені құрар кезде соңғы «Запрос» вкладкасына өтіп, ондағы өріске қойып, «Выполнить SQL...» батырмасын шерту керек немесе F9 ыстық пернесін басу керек. Нәтижесінде кестелердің құрылымын қолдан құрастырмай, аталған сұраныстарды орындай отырып, өзімізге қажетті кестенің құрылымын жылдам уақыт арасында құрастыра аламыз.

14.2 Құрастырылған «history» кестесі үшін сәйкес әрекеттер моделін қалыптастыру және оның GRUD генерациясы

Жобаланып отырған history кестесінің құрылымы құрастырылған соң Gii модулі арқылы жобаның сәйкес әрекеттер моделін қалыптастыруымыз керек. Ол үшін браузерде <http://academ-cloud.kz/gii> сілтемесіне өтіп, код генераторының басты бетіндегі

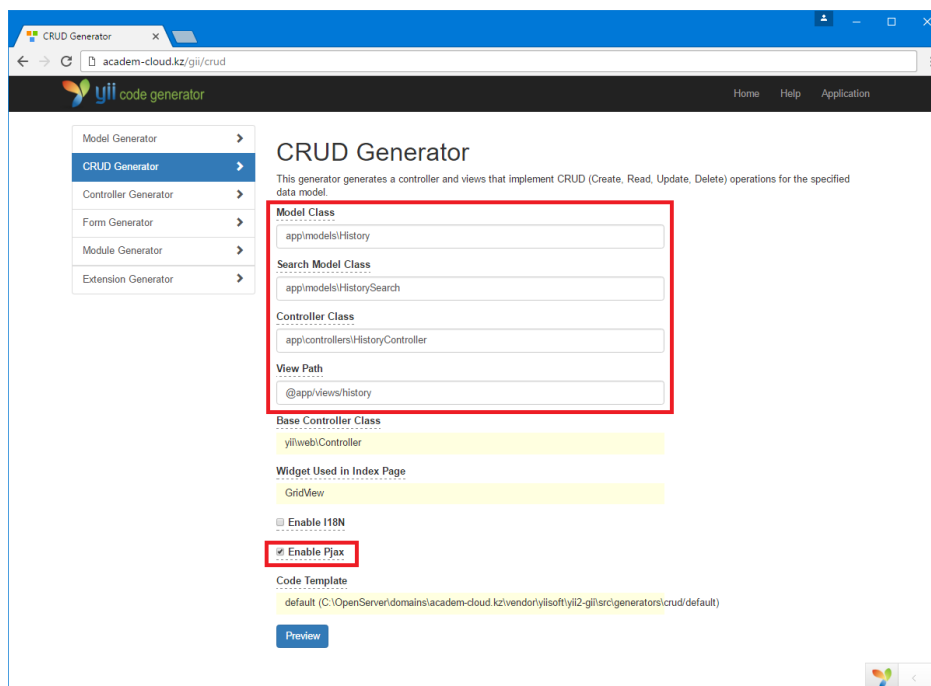
«Model Generator» бөліміндегі «Start» батырмасын шертеміз. Ашылған бөлімде төмендегі сурет 14.2-ге сәйкес «Table Name» өрісінде «history» деп, ал «Model Class Name» өрісінде «History» деп көрсетіп, алдымен беттің төменгі бөлігінде орналасқан «Preview» батырмасын, содан кейін төменде пайда болған «Generate» батырмасын шертеміз.

Нәтижесінде модельдер жатқан C:\OpenServer\domains\academ-cloud.kz\models каталогында жаңадан History.php файлы қалыптасатын болады.



Сурет 14.2. Әрекеттер моделін генерациялау

Әрекеттер моделін генерациялаған соң Gii модулінің GRUD Generator атты екінші бөліміне көшеміз. Онда әрекеттер моделінің GRUD генерациясын жүзеге асыруымыз керек, ол үшін төмендегі сурет 14.3-те көрсетілгендей, «Model Class» өрісінде құрастырылған әрекеттер моделінің сілтемесін «app\models\History» деп көрсетеміз, «View Path» өрісінде «@app/views/history», «Search Model Class» өрісінде «app\models\HistorySearch», «Controller Class» өрісінде «app\controllers\HistoryController» деп толтырып, «Enable Pjax» айырып-қосқышында белгішені орнатамыз да, алдымен «Preview» батырмасын, содан кейін «Generate» батырмасын шертеміз.



Сурет 14.3. Әрекеттер моделінің GRUD генерациясы

Нәтижесінде жобаның models каталогында HistorySearch.php файлы, controllers каталогында HistoryController.php файлы, сонымен қатар views\history каталогында _form.php, _search.php, create.php, index.php, update.php және view.php атты алты файлы автоматты түрде қалыптастырылатын болады.

14.3 «Select2» және «Export Menu» виджеттерін орнату

Бұлттық қоймада қолданушы әрекеттері бойынша деректермен жұмыс істеу үшін жобаның функционалдылығын байыту мақсатында әдемі виджеттердің бірі болып табылатын таңдалмалы тізім түріндегі kartik-тің «Select2» виджетін және сол автордың кестедегі деректерді әр түрлі форматқа автоматты түрде экспорттайтын «Export Menu» атты форма виджетін орнату керек.

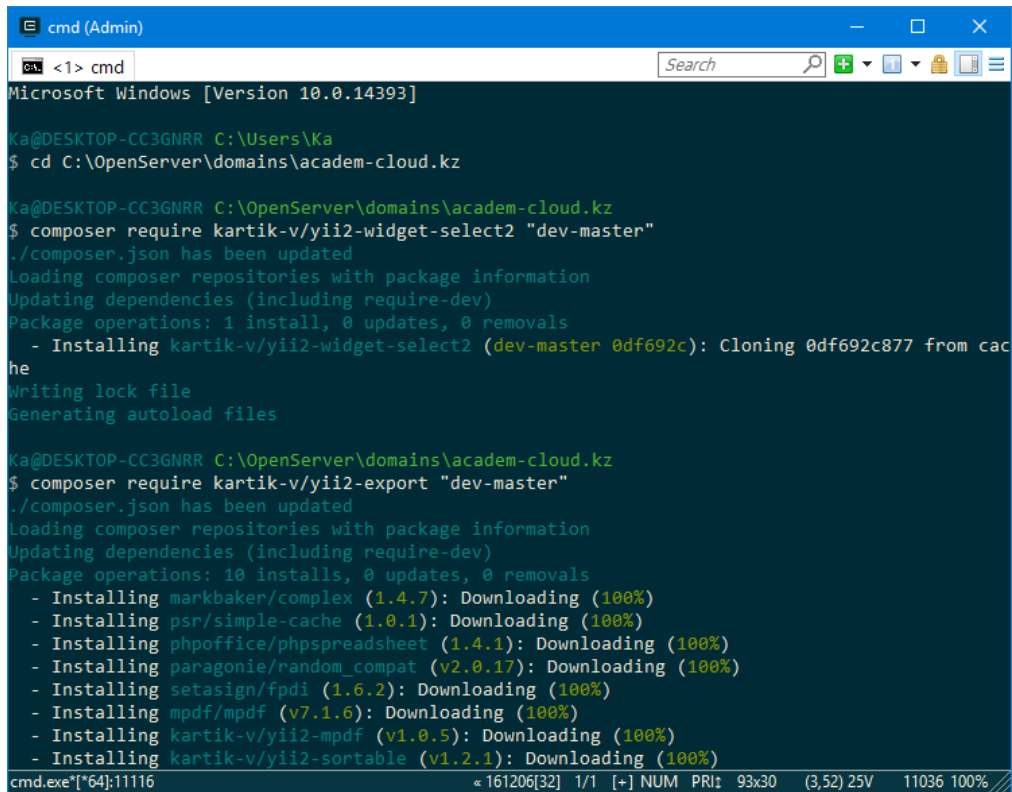
Ең алдымен «Select2» виджетін орнату үшін Open Server веб-серверіндегі Composer менеджерін іске қосып, келесі командаларды орындау қажет [30]:

```
cd C:\OpenServer\domains\academ-cloud.kz
composer require kartik-v/yii2-widget-select2 "dev-master"
```

Бұдан кейін келесі команда арқылы «Export Menu» виджетін орнатамыз [31]:

```
composer require kartik-v/yii2-export "dev-master"
```

Composer менеджеріндегі жоғарыда аталған виджеттерді орнату процедураларының кескіні төмендегі сурет 14.4-те көрсетілген.



```
cmd (Admin)
cmd
Microsoft Windows [Version 10.0.14393]
Ka@DESKTOP-CC3GNRR C:\Users\Ka
$ cd C:\OpenServer\domains\academ-cloud.kz
Ka@DESKTOP-CC3GNRR C:\OpenServer\domains\academ-cloud.kz
$ composer require kartik-v/yii2-widget-select2 "dev-master"
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
 - Installing kartik-v/yii2-widget-select2 (dev-master 0df692c): Cloning 0df692c877 from cac
he
Writing lock file
Generating autoload files
Ka@DESKTOP-CC3GNRR C:\OpenServer\domains\academ-cloud.kz
$ composer require kartik-v/yii2-export "dev-master"
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 10 installs, 0 updates, 0 removals
 - Installing markbaker/complex (1.4.7): Downloading (100%)
 - Installing psr/simple-cache (1.0.1): Downloading (100%)
 - Installing phpoffice/phpspreadsheet (1.4.1): Downloading (100%)
 - Installing paragonie/random_compat (v2.0.17): Downloading (100%)
 - Installing setasign/fpdf (1.6.2): Downloading (100%)
 - Installing mpdf/mpdf (v7.1.6): Downloading (100%)
 - Installing kartik-v/yii2-mpdf (v1.0.5): Downloading (100%)
 - Installing kartik-v/yii2-sortable (v1.2.1): Downloading (100%)
cmd.exe[*64]:11116 161206[32] 1/1 [+] NUM PRI: 93x30 (3,52) 25V 11036 100%
```

Сурет 14.4. Composer менеджері арқылы «Select2» және «Export Menu» виджеттерін орнату

14.4 Құрастырылған қолданушы әрекеттері моделін және оның сүзгілеу моделін өзгерту

Жоғарыда Gii модулімен автоматты түрде құрастырылған C:\OpenServer\domains\academ-cloud.kz\models каталогында жатқан қолданушы әрекеттері (History.php) моделін өзгерту қажеттілігі туындайды. Ол үшін аталған модель файлы ашып, жарияланып отырған айнымалылар тізімін келесі кодқа сәйкес жаңартайық:

```
/**
 * @property integer $id
 * @property integer $user_id
 * @property integer $created_at
 * @property string $action
 * @property string $comment
```

```
*/
```

Сонымен қатар жүйеде қолданушының жаңа әрекеті тіркелген кезінде және оны оның өзгерісін көру мақсатында оның уақыты көрсетілетіндей етіп, келесі кодқа сәйкес жаңа behaviors (тәртіп) функциясын қосамыз:

```
public function behaviors()
{
    return [
        [
            'class' => TimestampBehavior::className(),
            'attributes' => [
                ActiveRecord::EVENT_BEFORE_INSERT =>
['created_at'],
            ],
        ],
    ];
}
```

Аталған функцияда фреймворктың кірістірілген класс экземпляры қолданылғандықтан, файлдың жоғарғы жағында төмендегідей бұл кластар экземплярына сілтемесін орнату керек:

```
use yii\behaviors\TimestampBehavior;
```

Автоматты түрде құрастырылған rules() функциясының құрылымын және attributeLabels() функциясында реквизиттердің атауларын келесі кодтарда көрсетілгендей түрде өзгертеміз:

```
public function rules()
{
    return [
        [['user_id'], 'required'],
        [['user_id', 'created_at'], 'integer'],
        [['action', 'comment'], 'string', 'max' => 255],
    ];
}

public function attributeLabels()
{
    return [
        'id' => 'ID',
        'user_id' => 'Пользователь',
        'created_at' => 'Время',
        'action' => 'Действие',
        'comment' => 'Комментарий',
    ];
}
```

```
    ];  
}
```

Сонымен қатар ағымдағы модель файлына келесі көрсетілген қосымша функцияларды қосамыз:

```
public function getUser()  
{  
    return $this->hasOne(User::className(), ['id' =>  
'user_id']);  
}  
  
public static function actionLabels()  
{  
    return [  
        'login' => 'Авторизация',  
        'logout' => 'Выход',  
        'user_create' => 'Добавление пользователя',  
        'user_update' => 'Редактирование пользователя',  
        'user_delete' => 'Удаление пользователя',  
        'file_create' => 'Загрузка файла',  
        'file_update' => 'Редактирование файла',  
        'file_delete' => 'Удаление файла',  
    ];  
}
```

Қолданушы әрекеттері моделімен қатар Gii модулімен автоматты түрде құрастырылған C:\OpenServer\domains\academ-cloud.kz\models каталогында жатқан қолданушы әрекеттерінің сүзгілеу моделін (HistorySearch.php) өзгерту керек. Ол үшін аталған файлды ашып, оның құрылымын келесі кодқа сәйкес жаңарту қажет:

```
<?php  
  
namespace app\models;  
  
use app\models\History;  
use app\validators\DateRangeSeparatorValidator;  
use yii\base\Model;  
use yii\data\ActiveDataProvider;  
  
class HistorySearch extends Model  
{  
    public $created_at_range;  
    public $created_at_from;  
    public $created_at_from_ts;  
    public $created_at_to;  
    public $created_at_to_ts;  
    public $user_id;
```

```

public $user_ids;
public $action;
public $comment;

public function rules()
{
    return [
        ['user_id', 'integer'],
        [
            'created_at_range',
            DateRangeSeparatorValidator::className(),
            'startAttribute' => 'created_at_from',
            'endAttribute' => 'created_at_to',
            'startTimestampAttribute' =>
'created_at_from_ts',
            'endTimestampAttribute' =>
'created_at_to_ts',
            'interval' =>
DateRangeSeparatorValidator::INTERVAL_DAY,
        ],
        ['action', 'string'],
        ['comment', 'string'],
    ];
}

public function search($params)
{
    $query = History::find();
    $dataProvider = new ActiveDataProvider([
        'query' => $query,
        'sort' => [
            'defaultOrder' => [
                'created_at' => SORT_DESC,
            ],
        ],
        'pagination' => [
            'pageSize' => 50,
        ],
    ]);
    $this->load($params);
    if (!$this->validate()) {
        return $dataProvider;
    }
    $query->andWhere(['>=' , 'created_at', $this->
>created_at_from_ts]);
    $query->andWhere(['<', 'created_at', $this->
>created_at_to_ts]);
    $query->andWhere(['user_id' => $this->
>user_id]);
    $query->andWhere(['like', 'action', $this->
>action]);
    $query->andWhere(['like', 'comment', $this->
>comment]);
}

```



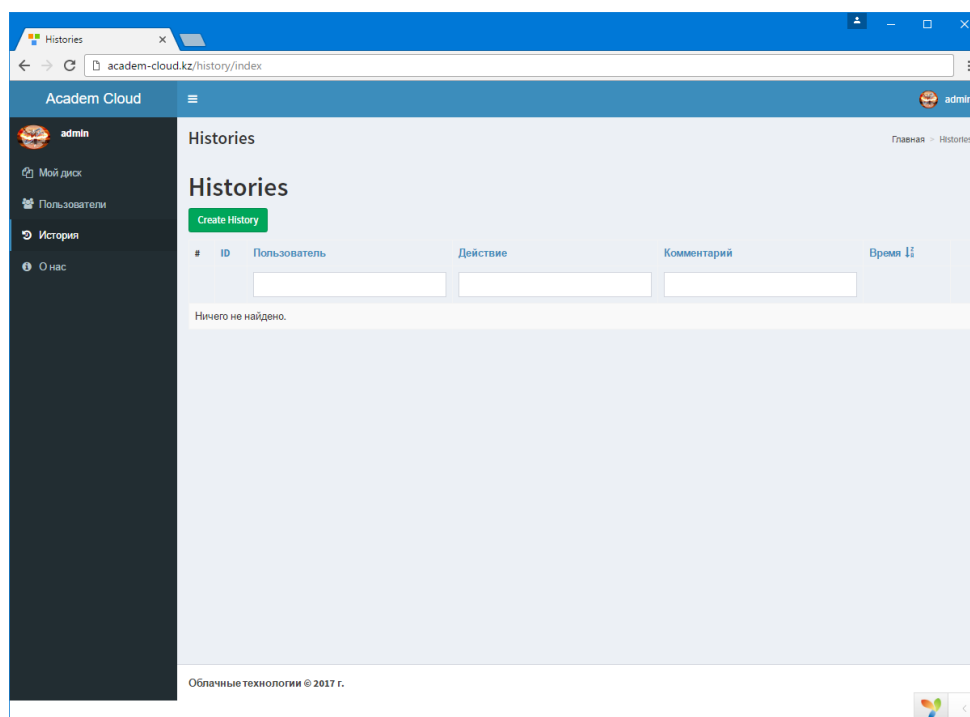
```

        return $dataProvider;
    }

    public function formName()
    {
        return '';
    }
}

```

Енгізілген өзгерістерден кейін сурет 14.5-ке сәйкес браузерде <http://academ-cloud.kz/history/index> сілтемесін тере отырып немесе мәзір жолында «Өзгертулер тарихы» пунктін таңдап, жасалған өзгерістерден кейінгі қолданушы әрекеттері бетінің жұмыс бабын тексеруге болады.



Сурет 14.5. «Қолданушы әрекеттерінің тізімі» бетінің көрінісі

Ағымдағы файлда алдын-ала құрастырылған және қолданушылар бойынша сүзгілеу моделінде де кездесетін жобаның `app\validators` каталогында жатқан арнайы сүзгілеу параметрлері енгізілген `DateRangeSeparatorValidator` валидатор файлы қолданылғанын көруге болады.

Сүзгілеу моделінде енгізілген өзгерістерді көру үшін `C:\OpenServer\domains\academ-cloud.kz\views\history` каталогындағы `index.php` файлын өзгерту керек.

14.5 «Қолданушы әрекеттерінің тізімі» бетін өзгерту

Жоғарыда көрсетілген сурет 14.5-тегі «Қолданушы әрекеттерінің тізімі» атты беттің құрылымын өзімізге қажетті кескінге өзгерту керек. Ол үшін index.php файлын (аталған файл C:\OpenServer\domains\ academ-cloud.kz\views\history каталогында орналасқан) ашып, ондағы барлық кодтарды өшіріп, төмендегі кодқа сәйкес алдын-ала қолданылатын кітапханалар мен виджеттердің сілтемелерін (соның ішінде ағымдағы модель үшін алдын-ала орнатылған Select2 және ExportMenu виджеттері) көрсете отырып, беттің қаңқасын әзірлейміз:

```
<?php

use app\models\History;
use app\models\HistorySearch;
use app\models\User;
use kartik\daterange\DateRangePicker;
use kartik\export\ExportMenu;
use kartik\select2\Select2;
use yii\helpers\ArrayHelper;
use yii\helpers\Html;
use yii\widgets\Pjax;

/* @var $this yii\web\View */
/* @var $searchModel HistorySearch */
/* @var $dataProvider yii\data\ActiveDataProvider */

$this->title = 'Список историй';
$this->params['breadcrumbs'][] = $this->title;

$columns = [
    [
        'attribute' => 'user_id',
        'format' => 'raw',
        'value' => 'user.username',
        'width' => '150px',
    ],
    [
        'attribute' => 'created_at',
        'format' => 'datetime',
        'width' => '150px',
    ],
    'action',
    'comment',
];
?>
<div class="box box-default color-palette-box">
    <div class="box-body">
```

```

        <div class="row">
            <div class="col-md-12">
                <!-- Батырмалар мен кесте деректерін
орналастыру -->
            </div>
        </div>
    </div>
</div>

```

Кодтың комментарий ретінде көрсетілген «батырмалар мен кесте деректерін орналастыру» жолына алдымен ExportMenu виджетін, сонымен қатар «Очистить таблицу» атты батырмасын келесі кодтар жолына сәйкес орналастырамыз [31]:

```

<div class="pull-right">
    <?= ExportMenu::widget([
        'dataProvider' => $dataProvider,
        'columns' => $columns,
        'target' => ExportMenu::TARGET_SELF,
        'showConfirmAlert' => false,
        'fontAwesome' => true,
        'exportConfig' => [
            ExportMenu::FORMAT_TEXT => false,
            ExportMenu::FORMAT_PDF => false,
            ExportMenu::FORMAT_HTML => false,
            ExportMenu::FORMAT_CSV => false,
        ],
        'columnSelectorOptions' => [
            'class' => 'btn btn-default',
        ],
        'dropdownOptions' => [
            'class' => 'btn btn-default',
            'menuOptions' => [
                'class' => 'pull-right',
            ],
        ],
    ]); ?>
    <div class="btn-group" role="group">
        <?= Html::a('Очистить таблицу', ['clear'], [
            'class' => 'btn btn-primary',
            'data' => [
                'confirm' => 'Вы уверены что хотите полностью
очистить таблицу?',
                'method' => 'post',
            ],
        ]) ?>
    </div>
</div>
<div class="clearfix"></div>

```

Бұдан кейін бұл кодтар жолынан кейін келесі кодтар жолына сәйкес қолданушы әрекеттерінің тізімін көрсететін кестені орналастырамыз [30]:

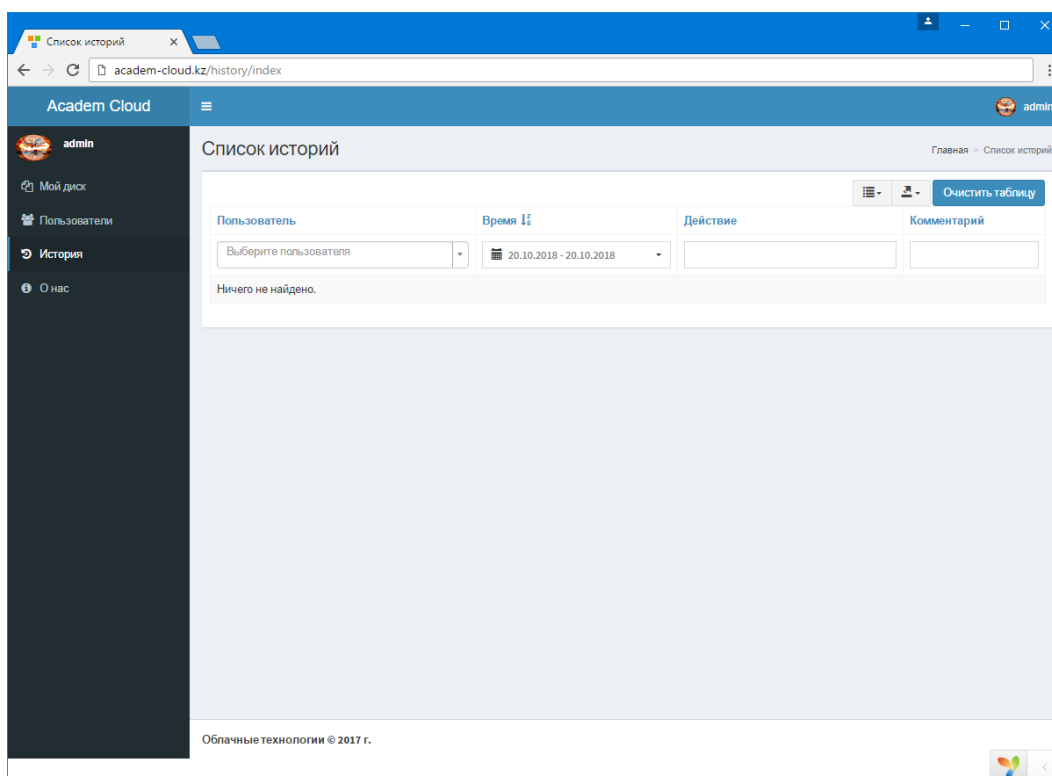
```
<?php Pjax::begin(); ?>
<?=\kartik\grid\GridView::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,
    'columns' => [
        [
            'attribute' => 'user_id',
            'filter' => Select2::widget([
                'name' => 'user_id',
                'model' => $searchModel,
                'data' => ArrayHelper::map(User::find()-
>orderBy(['username' => SORT_ASC])->all(), 'id',
                'username'),
                'value' => $searchModel->user_id,
                'options' => [
                    'multiple' => false,
                    'prompt' => 'Выберите пользователя',
                ],
                'showToggleAll' => false,
                'pluginOptions' => [
                    'allowClear' => true,
                    'minimumInputLength' => 0,
                ],
            ]),
            'value' => 'user.username',
            'options' => [
                'width' => 300
            ],
        ],
        [
            'attribute' => 'created_at',
            'format' => 'datetime',
            'filter' => DateRangePicker::widget([
                'attribute' => 'created_at_range',
                'defaultPresetValueOptions' => ['style' =>
'display:none'],
                'model' => $searchModel,
                'presetDropdown' => true,
                'pluginOptions' => [
                    'showDropdowns' => true,
                    'alwaysShowCalendars' => true,
                    'locale' => [
                        'format' => 'DD.MM.YYYY',
                    ],
                ],
            ]),
            'options' => [
                'width' => 220
            ],
        ],
    ],
]);
```

```

    ],
    ],
    [
        'attribute' => 'action',
        'value' => function (History $historyModel) {
            return
ArrayHelper::getValue(History::actionLabels(), $historyModel->action);
        }
    ],
    'comment',
],
]); ?>
<?php Pjax::end(); ?>

```

Нәтижесінде «Қолданушы әрекеттерінің тізімі» бетінің кескіні төмендегі сурет 14.6-да көрсетілгендей өзгертіледі. Беттің жоғарғы жағында кестедегі барлық деректерін Microsoft Excel форматына экспорттайтын арнайы батырмаларының, «Очистить таблицу» батырмасының, сонымен қатар сүзгілеу моделінде көрсетілген параметрлерге сәйкес сүзгілеу панелі бар кестенің пайда болғанын көруге болады.



Сурет 14.6. Қолданушы әрекеттерінің тізімі бетінің көрінісі

C:\OpenServer\domains\academ-cloud.kz\views\history каталогында жатқан index.php файлынан басқа Gii модулімен автоматты түрде генерацияланған файлдарды өшіруге болады, себебі бұл деректер қолданушының белгілі бір әрекеттерінен автоматты түрде пайда болатындықтан, оларды құрудың және өзгертудің қажеті жоқ.

14.6 Қолданушы әрекеттерінің контроллерін өзгерту

C:\OpenServer\domains\academ-cloud.kz\views\history каталогында жатқан қажеті жоқ файлдар өшірілген соң C:\OpenServer\domains\academ-cloud.kz\controllers каталогына көшіп, HistoryController.php файлын ашамыз да, ондағы actionIndex функциясынан басқа төмендегі action функцияларын да жоямыз:

```
public function actionView($id)
public function actionCreate()
public function actionUpdate($id)
public function actionDelete($id)
protected function findModel($id)
```

«Қолданушы әрекеттерінің тізімі» бетінде «Очистить таблицу» батырмасы бойынша оның кодына көз жүгіртсек, ['clear'] жолы бар екенін көреміз. Бұл батырманы шерткен кезде HistoryController.php контроллерінің арнайы clear атты action функциясымен орындалады дегенді білдіреді, сол себепті контроллерді ашып, келесі action функциясын қосамыз:

```
public function actionClear()
{
    History::deleteAll();
    Yii::$app->session->setFlash('success', 'Таблица "Список историй" полностью очищена. ');
    return $this->redirect(['index']);
}
```

Нәтижесінде аталған батырманы шерткен кезде кестедегі барлық деректер өшірілетін болады.

14.7 Қолданушы әрекеттерін тіркейтін модульді әзірлеу

Жоғарыда айтылғандай, «Өзгертулер тарихы» атты беттегі кесте деректері автоматты түрде қолданушының әрекеттерінен қалыптасатынын айтып кеткенбіз. Сол себепті бұл функционалды жүзеге асыру үшін келесі қадамдарды орындау керек.

Жобада алдымен жаңа «modules» атты каталогын, содан сол каталогтың ішінде жаңа «history» каталогын құрамыз да, онда жаңа HistoryModule.php атты файлына келесі кодты орналастырамыз:

```
<?php

namespace app\modules\history;

use yii\base\Module;

class HistoryModule extends Module
{
}
```

Сонымен қатар ағымдағы каталогта жаңа HistoryBootstrap.php файлын құрып, оған келесі кодтар жолын тереміз:

```
<?php

namespace app\modules\history;

use app\models\File;
use app\models\History;
use app\models\User;
use Yii;
use yii\base\BootstrapInterface;
use yii\base\Event;
use yii\web\Application;
use yii\web\User as UserComponent;
use yii\web\UserEvent;

class HistoryBootstrap implements BootstrapInterface
{
    public function bootstrap($app)
    {
        $this->registerEvents($app);
    }

    public function registerEvents(Application $app)
    {
        // Жүйеге кіру
        Event::on(UserComponent::className(),
        UserComponent::EVENT_AFTER_LOGIN, function (UserEvent $event) {
            $historyModel = new History();
        });
    }
}
```

```

        $historyModel->user_id = $event->identity-
>getId();
        $historyModel->action = 'login';
        $historyModel->comment = Yii::$app->request-
>userIP;
        if (!$historyModel->save()) {
            Yii::error($historyModel->getErrors());
        }
    });
    // Жүйеден шығу
    Event::on(UserComponent::className(),
UserComponent::EVENT_AFTER_LOGOUT, function (UserEvent $event) {
    $historyModel = new History();
    $historyModel->user_id = $event->identity-
>getId();
    $historyModel->action = 'logout';
    $historyModel->comment = Yii::$app->request-
>userIP;
    if (!$historyModel->save()) {
        Yii::error($historyModel->getErrors());
    }
});
    // Жаңа қолданушыны құру
    Event::on(User::className(),
User::EVENT_AFTER_INSERT, function (Event $event) {
    $historyModel = new History();
    $historyModel->user_id = Yii::$app->user-
>getId();
    $historyModel->action = 'user_create';
    if (!$historyModel->save()) {
        Yii::error($historyModel->getErrors());
    }
});
    // Қолданушы деректерін өзгерту
    Event::on(User::className(),
User::EVENT_AFTER_UPDATE, function (Event $event) {
    $historyModel = new History();
    $historyModel->user_id = Yii::$app->user-
>getId();
    $historyModel->action = 'user_update';
    if (!$historyModel->save()) {
        Yii::error($historyModel->getErrors());
    }
});
    // Қолданушыны өшіру
    Event::on(User::className(),
User::EVENT_AFTER_DELETE, function (Event $event) {
    $historyModel = new History();
    $historyModel->user_id = Yii::$app->user-
>getId();
    $historyModel->action = 'user_delete';
    if (!$historyModel->save()) {
        Yii::error($historyModel->getErrors());
    }
});

```



```

    }
    });
    // Жаңа файлды жүктеу
    Event::on(File::className(),
File::EVENT_AFTER_INSERT, function (Event $event) {
    /** @var File $fileModel */
    $fileModel = $event->sender;
    $historyModel = new History();
    $historyModel->user_id = Yii::$app->user-
>getId();

    $historyModel->action = 'file_create';
    $historyModel->comment = "{$fileModel->name} (" .
Yii::$app->formatter->asShortSize($fileModel->size) . ')';
    if (!$historyModel->save()) {
        Yii::error($historyModel->getErrors());
    }
    });
    // Файлды өзгерту
    Event::on(File::className(),
File::EVENT_AFTER_UPDATE, function (Event $event) {
    /** @var File $fileModel */
    $fileModel = $event->sender;
    $historyModel = new History();
    $historyModel->user_id = Yii::$app->user-
>getId();

    $historyModel->action = 'file_update';
    $historyModel->comment = "{$fileModel->name} (" .
Yii::$app->formatter->asShortSize($fileModel->size) . ')';
    if (!$historyModel->save()) {
        Yii::error($historyModel->getErrors());
    }
    });
    // Файлды өшіру
    Event::on(File::className(),
File::EVENT_AFTER_DELETE, function (Event $event) {
    /** @var File $fileModel */
    $fileModel = $event->sender;
    $historyModel = new History();
    $historyModel->user_id = Yii::$app->user-
>getId();

    $historyModel->action = 'file_delete';
    $historyModel->comment = "{$fileModel->name} (" .
Yii::$app->formatter->asShortSize($fileModel->size) . ')';
    if (!$historyModel->save()) {
        Yii::error($historyModel->getErrors());
    }
    });
}
}
}

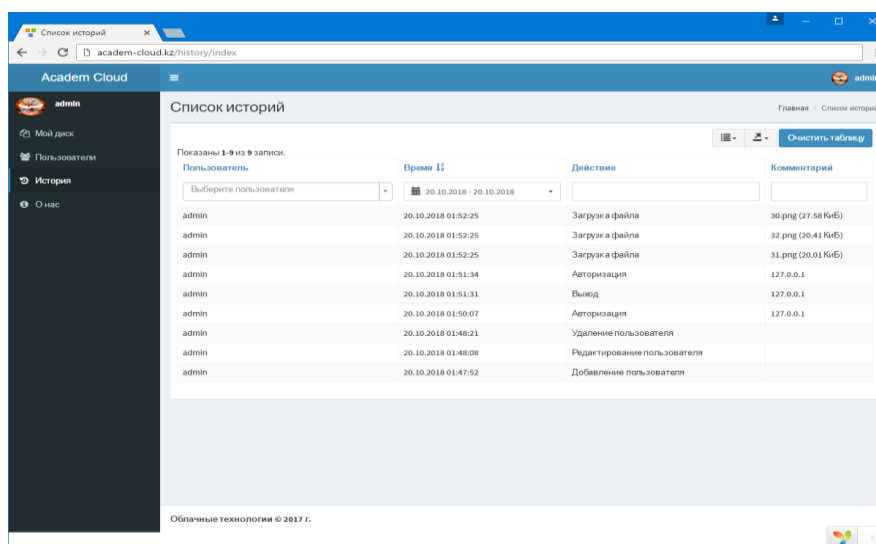
```

Ағымдағы файлда қолданушы әрекеттерін бақылап, оны деректер базасында тіркейтін арнайы функциялары бар екенін

аңғаруға болады. Ағымдағы модуль дұрыс жұмыс атқаруы үшін оны жоба конфигурациясында көрсетіп кету керек, ол үшін төмендегі кодқа сәйкес C:\OpenServer\domains\academ-cloud.kz\config каталогында жатқан web.php файлындағы \$config бөлімінің bootstrap пунктін өзгерту қажет:

```
'bootstrap' => [  
    'log',  
    'app\modules\history\HistoryBootstrap',  
],
```

Нәтижесінде бұл модульдің арқасында қолданушы жүйеге кірген кезде немесе жүйеден шыққан кезде, жүйе администраторымен жаңа қолданушыны қосқанда, өзгерткенде немесе өшіргенде, сонымен қатар бұлтқа файлдарды жүктеген кезде автоматты түрде әрекеттер базаға тіркеліп көрсетілетін болады. Аталған әрекеттерді орындап, базада тіркелетінін өзімізге тексеруге болады, нәтижесінде төмендегі сурет 14.7-де көрсетілгендей, «Список историй» бетіндегі кесте деректері қолданушы әрекеттеріне сәйкес толтырылатын болады. Ал беттің жоғарғы оң жақ бетінде тұрған «Очистить таблицу» батырмасын шерткен кезде кестедегі барлық деректердің өшірілетініне көзімізді жеткізуге болады.



Пользователь	Время	Действие	Комментарий
admin	20.10.2018 01:52:25	Загрузка файла	30.png (27.58 КиБ)
admin	20.10.2018 01:52:25	Загрузка файла	32.png (20.41 КиБ)
admin	20.10.2018 01:52:25	Загрузка файла	31.png (20.01 КиБ)
admin	20.10.2018 01:51:34	Авторизация	127.0.0.1
admin	20.10.2018 01:51:31	Выход	127.0.0.1
admin	20.10.2018 01:50:07	Авторизация	127.0.0.1
admin	20.10.2018 01:48:21	Удаление пользователя	
admin	20.10.2018 01:48:08	Редактирование пользователя	
admin	20.10.2018 01:47:52	Добавление пользователя	

Сурет 14.7. Қолданушы әрекеттерінің тізімі бетінің көрінісі

Бақылау сұрақтары:

1. Модельді Gii модулі арқылы қалыптастырмастан бұрын ең алдымен базада кестені құру керек?

2. Ағымдағы модульде Select2 виджеті арқылы қандай мүмкіндіктерге ие боламыз?
3. Export Menu виджеті арқылы деректерді қандай форматтарға экспорттай аламыз?
4. Кестедегі деректерді тазарту функциясын қандай қадамдармен жүзеге асырдық?
5. Не себепті қолданушы әрекеттерін тіркейтін модульді жеке модуль ретінде әзірледік?