

4. Authentication, registration

Цель работы

Реализовать авторизацию и регистрацию пользователей

Критерии оценки

5 - авторизация проходит успешно; регистрация проходит успешно; код содержит все необходимые комментарии;

4 - авторизация проходит успешно; регистрация проходит успешно; код не содержит комментариев ;

3 - есть ошибки при регистрации или авторизации

Отчет

1. Ссылка на репозиторий GitHub.
2. Ссылка на видео или отчет со скриншотами, демонстрирующими работу авторизации и регистрации

Создание таблицы

Создаем миграцию

```
yii migrate/create create_user_table
```

Открываем файл с миграцией `/migration/m000000_000000_create_user_table.php` и добавляем код создания миграции

```
<?php

use yii\db\Migration;

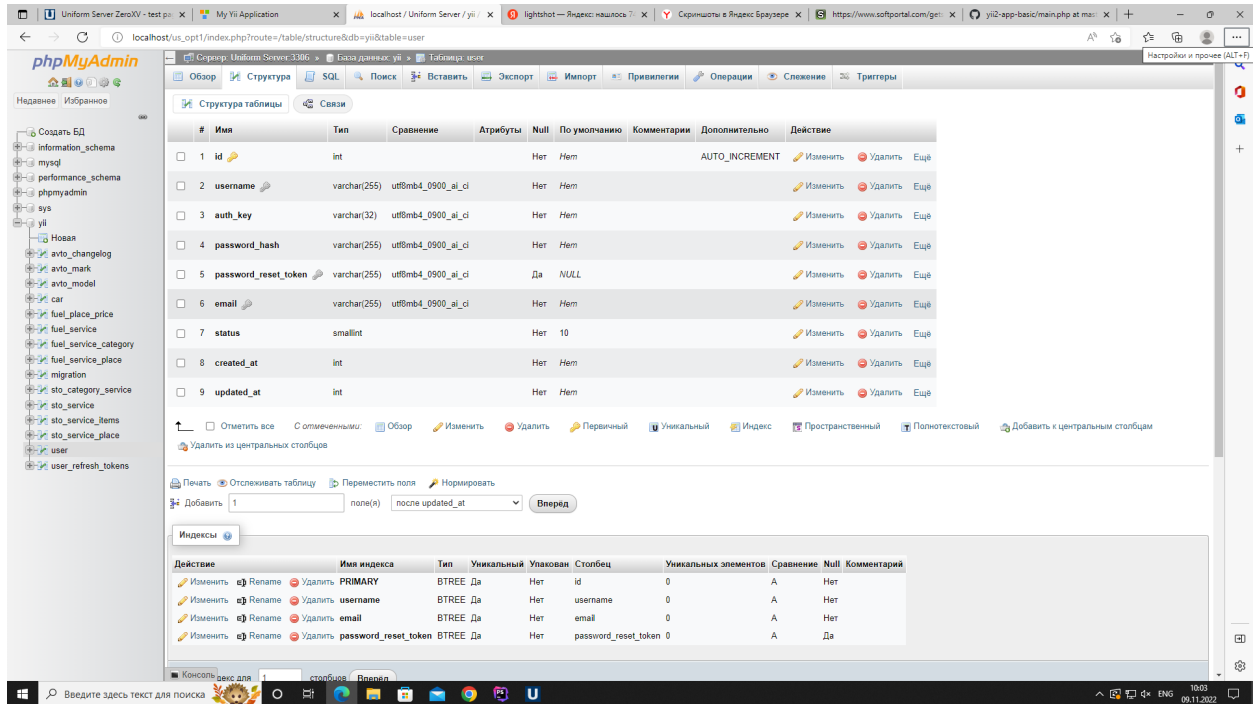
/**
 * Handles the creation of table `{{%user}}`.
 */
class m221109_070023_create_user_table extends Migration
{
    public function up()
    {
        $this->createTable('user', [
            'id' => $this->primaryKey(),
            'username' => $this->string()->notNull()->unique(),
            'auth_key' => $this->string(32)->notNull(),
            'password_hash' => $this->string()->notNull(),
            'password_reset_token' => $this->string()->unique(),
            'email' => $this->string()->notNull()->unique(),
            'status' => $this->smallInteger()->notNull()->defaultValue(10),
            'created_at' => $this->integer()->notNull(),
            'updated_at' => $this->integer()->notNull(),
        ]);
    }

    public function down()
    {
        $this->dropTable('user');
    }
}
```

Применяем миграцию

```
yii migrate
```

Должна создаваться таблица



Правим модель /models/User.php

```
<?php

namespace app\models;

use Yii;
use yii\base\NotSupportedException;
use yii\behaviors\TimestampBehavior;
use yii\db\ActiveRecord;
use yii\web\IdentityInterface;

/**
 * User model
 */
class User extends ActiveRecord implements IdentityInterface
{
    /**
     * @property integer $id
     * @property string $username
     * @property string $password_hash
     * @property string $password_reset_token
     * @property string $email
     * @property string $auth_key
     * @property integer $status
     * @property integer $created_at
     * @property integer $updated_at
     * @property string $password write-only password
     */

    const STATUS_DELETED = 0;
    const STATUS_ACTIVE = 10;
}
```

```

/**
 * @inheritdoc
 */
public static function tableName()
{
    return '{{%user}}';
}

/**
 * @inheritdoc
 */
public function behaviors()
{
    return [
        TimestampBehavior::className(),
    ];
}

/**
 * @inheritdoc
 */
public function rules()
{
    return [
        ['status', 'default', 'value' => self::STATUS_ACTIVE],
        ['status', 'in', 'range' => [self::STATUS_ACTIVE, self::STATUS_DELETED]],
    ];
}

/**
 * @inheritdoc
 */
public static function findIdentity($id)
{
    return static::findOne(['id' => $id, 'status' => self::STATUS_ACTIVE]);
}

/**
 * @inheritdoc
 */
public static function findIdentityByAccessToken($token, $type = null)
{
    throw new NotSupportedException('"findIdentityByAccessToken" is not implemented.');
```

```

}

/**
 * Finds user by username
 *
 * @param string $username
 * @return static|null
 */
public static function findByUsername($username)
{
    return static::findOne(['username' => $username, 'status' => self::STATUS_ACTIVE]);
}

/**
 * @inheritdoc
 */
public function getId()
{
    return $this->getPrimaryKey();
}

/**
 * @inheritdoc
 */
public function getAuthKey()
{
    return $this->auth_key;
}

/**
```

```

        * @inheritdoc
        */
        public function validateAuthKey($authKey)
        {
            return $this->getAuthKey() === $authKey;
        }

        /**
         * Validates password
         *
         * @param string $password password to validate
         * @return bool if password provided is valid for current user
         */
        public function validatePassword($password)
        {
            return Yii::$app->security->validatePassword($password, $this->password_hash);
        }

        /**
         * Generates password hash from password and sets it to the model
         *
         * @param string $password
         */
        public function setPassword($password)
        {
            $this->password_hash = Yii::$app->security->generatePasswordHash($password);
        }

        /**
         * Generates "remember me" authentication key
         */
        public function generateAuthKey()
        {
            $this->auth_key = Yii::$app->security->generateRandomString();
        }
    }
}

```

Форма регистрации

Создадим модель для форм регистрации, для этого необходимо создать файл `/models/SignupForm.php` со следующим содержимым:

```

<?php

namespace app\models;

use Yii;
use yii\base\Model;

/**
 * Signup form
 */
class SignupForm extends Model
{
    public $username;
    public $email;
    public $password;

    /**
     * @inheritdoc
     */
    public function rules()
    {
        return [
            ['username', 'trim'],
            ['username', 'required'],
            ['username', 'unique', 'targetClass' => '\app\models\User', 'message' => 'This username has already been taken.'],
            ['username', 'string', 'min' => 2, 'max' => 255],

```

```

        ['email', 'trim'],
        ['email', 'required'],
        ['email', 'email'],
        ['email', 'string', 'max' => 255],
        ['email', 'unique', 'targetClass' => '\app\models\User', 'message' => 'This email address has already been taken.'],
        ['password', 'required'],
        ['password', 'string', 'min' => 6],
    ];
}

/**
 * Signs user up.
 *
 * @return User|null the saved model or null if saving fails
 */
public function signup()
{
    if (!$this->validate()) {
        return null;
    }

    $user = new User();
    $user->username = $this->username;
    $user->email = $this->email;
    $user->setPassword($this->password);
    $user->generateAuthKey();
    return $user->save() ? $user : null;
}
}

```

Далее в `/controllers/SiteController.php` добавляем экшен **signup**

```

public function actionSignup()
{
    $model = new SignupForm();

    if ($model->load(Yii::$app->request->post())) {
        if ($user = $model->signup()) {
            if (Yii::$app->getUser()->login($user)) {
                return $this->goHome();
            }
        }
    }

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

```

Добавляем представление для регистрации, файл `/views/site/signup.php` со следующим содержимым:

```

<?php

use yii\helpers\Html;
use yii\bootstrap5\ActiveForm;

$this->title = 'Signup';
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="site-signup">
    <h1><?= Html::encode($this->title) ?></h1>
    <p>Please fill out the following fields to signup:</p>
    <div class="row">
        <div class="col-lg-5">

```

```

<?php $form = ActiveForm::begin(['id' => 'form-signup']); ?>
<?= $form->field($model, 'username')->textInput(['autofocus' => true]) ?>
<?= $form->field($model, 'email') ?>
<?= $form->field($model, 'password')->passwordInput() ?>
<div class="form-group">
    <?= Html::submitButton('Signup', ['class' => 'btn btn-primary', 'name' => 'signup-button']) ?>
</div>
<?php ActiveForm::end(); ?>

</div>
</div>
</div>

```

И последнее, что необходимо сделать - это добавить регистрацию в меню. Открываем файл **/views/layouts/main.php** находим виджет меню и добавляем пункт регистрации

```
['label' => 'Signup', 'url' => ['/site/signup']],
```

В результате получится форма регистрации

The screenshot displays a web browser window with the address bar showing `localhost/auto-control-yii2/web/index.php?r=site%2Fsignup`. The page features a dark navigation bar at the top with the text "My Application" and links for "Home", "About", "Contact", "Signup", and "Login". The main content area is titled "Signup" and includes the instruction "Please fill out the following fields to signup:". There are three input fields: "Username" with the value "admin" and a green checkmark, "Email" with the value "admin@admin.ru" and a green checkmark, and "Password" which is masked with dots and has a red error message below it stating "Password should contain at least 6 characters.". A blue "Signup" button is positioned below the password field. The browser's taskbar at the bottom shows various application icons and the system clock indicating 10:15 on 09.11.2022.