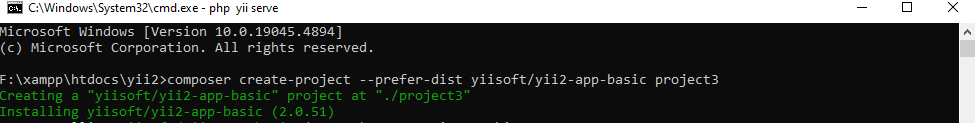
**Yii2 Assignment**

1. **Install yii2 via composer :-** composer create-project --prefer-dist yiisoft/yii2-app-basic project3

****

**Run yii2 :** php yii serve

**Clear cache :** - php yii cache/flush-all

1. **Create user table with name, email, password, role etc.:-**

* php yii migrate/create create\_user\_table

In migration : table is created (F:\xampp\htdocs\yii2\project3\migrations\m240924\_055920\_create\_user\_table.php):-

<?php

use yii\db\Migration;

/\*\*

 \* Handles the creation of table `{{%user}}`.

 \*/

class m240924\_055920\_create\_user\_table extends Migration

{

    /\*\*

     \* {@inheritdoc}

     \*/

    public function safeUp()

    {

        $this->createTable('user', [

            'id' => $this->primaryKey(),

            'name' => $this->string()->notNull(),

            'email' => $this->string()->notNull()->unique(),

            'password' => $this->string()->notNull(),

            'role' => $this->string()->notNull()->defaultValue('user'),

            'created\_at' => $this->integer()->notNull(),

            'updated\_at' => $this->integer()->notNull(),

        ]);

    }

    public function safeDown()

    {

        $this->dropTable('user');

    }

}

**Apply the Migrations:-** php yii migrate

1. **Create table project, with title, description etc.**

**Create table :-** php yii migrate/create create\_project\_table

*In migration : table is created*

*F:\xampp\htdocs\yii2\project3\migrations\m240924\_055934\_create\_project\_table.php*

<?php

use yii\db\Migration;

/\*\*

 \* Handles the creation of table `{{%project}}`.

 \*/

class m240924\_055934\_create\_project\_table extends Migration

{

    /\*\*

     \* {@inheritdoc}

     \*/

    public function safeUp()

    {

        $this->createTable('project', [

            'id' => $this->primaryKey(),

            'title' => $this->string()->notNull(),

            'description' => $this->text()->notNull(),

            'created\_at' => $this->integer()->notNull(),

            'updated\_at' => $this->integer()->notNull(),

        ]);

        // Add foreign key for user\_id

        $this->addForeignKey(

            'fk-project-user\_id',

            'project',

            'user\_id',

            'user',

            'id',

            'CASCADE'

        );

    }

    public function safeDown()

    {

        // Drop foreign key

        $this->dropForeignKey('fk-project-user\_id', 'project');

        $this->dropTable('project');

    }

}

**Apply the Migrations:-** php yii migrate

1. **Create  Login and Signup(With Different Roles . i.e admin ,user , manager)**

Step 1: Update User Model

F:\xampp\htdocs\yii2\project3\models\User.php

<?php

namespace app\models;

use Yii;

use yii\db\ActiveRecord;

class User extends ActiveRecord

{

    public static function tableName()

    {

        return 'user'; // Adjust the table name as necessary

    }

    public function rules()

    {

        return [

            [['name', 'email', 'role'], 'required'],

            ['email', 'email'],

            [['name', 'email', 'role'], 'string', 'max' => 255],

        ];

    }

}

**Step 2: Create Signup Form** :- Create a file named SignupForm.php in the models directory:

F:\xampp\htdocs\yii2\project3\models\SignupForm.php

<?php

    namespace app\models;

    use Yii;

    use yii\base\Model;

    use app\models\User; // Ensure this is at the top of your SignupForm model

    class SignupForm extends Model

    {

        public $name;

        public $email;

        public $password;

        public $role;

        public function rules()

        {

            return [

                [['name', 'email', 'password', 'role'], 'required'],

                ['email', 'email'],

                ['email', 'unique', 'targetClass' => User::class, 'message' => 'This email address has already been taken.'],

                ['role', 'in', 'range' => ['admin', 'user', 'manager']],

                ['password', 'string', 'min' => 6],

            ];

        }

        public function signup()

        {

            if ($this->validate()) {

                $user = new User();

                $user->name = $this->name;

                $user->email = $this->email;

                $user->password = Yii::$app->security->generatePasswordHash($this->password);

                $user->role = $this->role;

                $user->created\_at = time();

                $user->updated\_at = time();

                return $user->save() ? $user : null;

            }

            return null;

        }

    }

?>

**Step 3: Create Login Form**

Create a file named LoginForm.php in the models directory:

F:\xampp\htdocs\yii2\project3\models\LoginForm.php

<?php

namespace app\models;

use Yii;

use yii\base\Model;

class LoginForm extends Model

{

    public $email;

    public $password;

    public function rules()

    {

        return [

            [['email', 'password'], 'required'],

            ['email', 'email'],

        ];

    }

    public function login()

    {

        if ($this->validate()) {

            return Yii::$app->user->login(User::findByEmail($this->email), 3600 \* 24 \* 30); // 30 days

        }

        return false;

    }

}

?>

**Step 4: Create Signup and Login Actions in Controller**

create a SiteController.php in the controllers directory

F:\xampp\htdocs\yii2\project3\controllers\SiteController.php

<?php

namespace app\controllers;

use Yii;

use yii\filters\AccessControl;

use yii\web\Controller;

use yii\web\Response;

use yii\filters\VerbFilter;

use app\models\LoginForm;

use app\models\SignupForm;

use app\models\ContactForm;

use app\models\User;

use app\models\UserSearch;

use app\models\Project;

class SiteController extends Controller

{

    public function actionDashboard()

    {

        // Get the currently logged-in user

        $user = Yii::$app->user->identity;

        if ($user) {

            // Check the user's role

            if ($user->role === 'admin') {

                $users = User::find()->all(); // Fetch all users for admin

                return $this->render('dashboard-admin', [

                    'users' => $users,

                ]);

            } elseif ($user->role === 'manager') {

                $projects = Project::find()->all(); // Fetch all projects for manager

                return $this->render('dashboard-manager', [

                    'projects' => $projects,

                ]);

            }

        }

        return $this->redirect(['site/login']); // Redirect if not logged in

    }

    public function actionSignup()

    {

        $model = new SignupForm();

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

            return $this->redirect(['site/login']);

        }

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

            'model' => $model,

        ]);

    }

    /\*\*

     \* {@inheritdoc}

     \*/

    public function behaviors()

    {

        return [

            'access' => [

                'class' => AccessControl::class,

                'only' => ['logout'],

                'rules' => [

                    [

                        'actions' => ['logout'],

                        'allow' => true,

                        'roles' => ['@'],

                    ],

                ],

            ],

            'verbs' => [

                'class' => VerbFilter::class,

                'actions' => [

                    'logout' => ['post'],

                ],

            ],

        ];

    }

    /\*\*

     \* {@inheritdoc}

     \*/

    public function actions()

    {

        return [

            'error' => [

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

            ],

            'captcha' => [

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

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

            ],

        ];

    }

    /\*\*

     \* Displays homepage.

     \*

     \* @return string

     \*/

    public function actionIndex()

    {

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

    }

    /\*\*

     \* Login action.

     \*

     \* @return Response|string

     \*/

    public function actionLogin()

    {

        $model = new LoginForm();

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

            return $this->goBack();

        }

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

            'model' => $model,

        ]);

    }

    /\*\*

     \* Logout action.

     \*

     \* @return Response

     \*/

    public function actionLogout()

    {

        Yii::$app->user->logout();

        return $this->goHome();

    }

    /\*\*

     \* Displays contact page.

     \*

     \* @return Response|string

     \*/

    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,

        ]);

    }

    /\*\*

     \* Displays about page.

     \*

     \* @return string

     \*/

    public function actionAbout()

    {

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

    }

}

**Step 5: Create Views for Signup and Login:**

F:\xampp\htdocs\yii2\project3\views\site\signup.php

<?php

use yii\helpers\Html;

use yii\widgets\ActiveForm;

$this->title = 'Signup';

?>

<h1><?= Html::encode($this->title) ?></h1>

<div class="site-signup">

    <p>Please fill out the following fields to signup:</p>

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

    <?= $form->field($model, 'name')->textInput(['autofocus' => true]) ?>

    <?= $form->field($model, 'email') ?>

    <?= $form->field($model, 'password')->passwordInput() ?>

    <?= $form->field($model, 'role')->dropDownList(['admin' => 'Admin', 'manager' => 'Manager', 'user' => 'User']) ?>

    <div class="form-group">

        <?= Html::submitButton('Signup', ['class' => 'btn btn-primary', 'name' => 'signup-button']) ?>

    </div>

    <?php ActiveForm::end(); ?>

</div>

Create a view file for the login form: views/site/login.php:

F:\xampp\htdocs\yii2\project3\views\site\login.php

<?php

use yii\helpers\Html;

use yii\widgets\ActiveForm;

$this->title = 'Login';

?>

<h1><?= Html::encode($this->title) ?></h1>

<div class="site-login">

    <p>Please fill out the following fields to login:</p>

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

    <?= $form->field($model, 'email')->textInput(['autofocus' => true]) ?>

    <?= $form->field($model, 'password')->passwordInput() ?>

    <div class="form-group">

        <?= Html::submitButton('Login', ['class' => 'btn btn-primary', 'name' => 'login-button']) ?>

    </div>

    <?php ActiveForm::end(); ?>

</div>

**Start the Yii2 Development Server**: Navigate to your project directory in your terminal and run the following command:

* *php yii serve*

**Open Your Web Browser**.

[*http://localhost:8080/site/signup*](http://localhost:8080/site/signup)

**Ensure Routing is Configured:-**

***F:\xampp\htdocs\yii2\project3\config\web.php***

<?php

$params = require \_\_DIR\_\_ . '/params.php';

$db = require \_\_DIR\_\_ . '/db.php';

$config = [

    'id' => 'basic',

    'basePath' => dirname(\_\_DIR\_\_),

    'bootstrap' => ['log'],

    'aliases' => [

        '@bower' => '@vendor/bower-asset',

        '@npm'   => '@vendor/npm-asset',

    ],

    'components' => [

        'request' => [

            // !!! insert a secret key in the following (if it is empty) - this is required by cookie validation

            'cookieValidationKey' => 'hgtfSjrM5B2a8qfb16PjY-pjm\_kMuqsH',

        ],

        'cache' => [

            'class' => 'yii\caching\FileCache',

        ],

        'user' => [

            'identityClass' => 'app\models\User',

            'enableAutoLogin' => true,

        ],

        'errorHandler' => [

            'errorAction' => 'site/error',

        ],

        'mailer' => [

            'class' => \yii\symfonymailer\Mailer::class,

            'viewPath' => '@app/mail',

            // send all mails to a file by default.

            'useFileTransport' => true,

        ],

        'log' => [

            'traceLevel' => YII\_DEBUG ? 3 : 0,

            'targets' => [

                [

                    'class' => 'yii\log\FileTarget',

                    'levels' => ['error', 'warning'],

                ],

            ],

        ],

        'db' => $db,

        'urlManager' => [

            'enablePrettyUrl' => true,

            'showScriptName' => false,

            'rules' => [

                'project' => 'project/index',

        'project/create' => 'project/create',

        'project/<id:\d+>' => 'project/view',

        'project/update/<id:\d+>' => 'project/update',

        'project/delete/<id:\d+>' => 'project/delete',

        'user' => 'user/index',  // Route for user index

        'user/create' => 'user/create',  // Route for creating a user

        'user/update/<id:\d+>' => 'user/update',  // Route for updating a user

        'user/view/<id:\d+>' => 'user/view',  // Route for viewing a user

        'user/delete/<id:\d+>' => 'user/delete',  // Route for deleting a user

            ],

        ],

    ],

    'params' => $params,

];

if (YII\_ENV\_DEV) {

    // configuration adjustments for 'dev' environment

    $config['bootstrap'][] = 'debug';

    $config['modules']['debug'] = [

        'class' => 'yii\debug\Module',

        // uncomment the following to add your IP if you are not connecting from localhost.

        //'allowedIPs' => ['127.0.0.1', '::1'],

    ];

    $config['bootstrap'][] = 'gii';

    $config['modules']['gii'] = [

        'class' => 'yii\gii\Module',

        // uncomment the following to add your IP if you are not connecting from localhost.

        //'allowedIPs' => ['127.0.0.1', '::1'],

    ];

}

return $config;

**Step 3: Check Database Configuration**

Make sure your database connection is properly configured in your config/db.php file:

php

<?php

return [

    'class' => 'yii\db\Connection',

    'dsn' => 'mysql:host=localhost;dbname=project3',

    'username' => 'root',

    'password' => '',

    'charset' => 'utf8',

    // Schema cache options (for production environment)

    //'enableSchemaCache' => true,

    //'schemaCacheDuration' => 60,

    //'schemaCache' => 'cache',

];

1. ***Make CRUD's for project table.***

Create Project model

F:\xampp\htdocs\yii2\project3\models\Project.php

<?php

    namespace app\models;

    use yii\db\ActiveRecord;

    class Project extends ActiveRecord

    {

        public static function tableName()

        {

            return 'project'; // Name of your project table

        }

        public function rules()

        {

            return [

                [['title', 'description'], 'required'],

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

                ['description', 'string'],

            ];

        }

    }

?>

Generate a controller for the Project model:

F:\xampp\htdocs\yii2\project3\controllers\ProjectController.php

<?php

    namespace app\controllers;

    use Yii;

    use app\models\Project;

    use yii\web\Controller;

    use yii\web\NotFoundHttpException;

    use yii\data\ActiveDataProvider;

    class ProjectController extends Controller

    {

        public function actionIndex()

        {

            $dataProvider = new ActiveDataProvider([

                'query' => Project::find(),

            ]);

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

                'dataProvider' => $dataProvider,

            ]);

        }

        public function actionView($id)

        {

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

                'model' => $this->findModel($id),

            ]);

        }

        public function actionCreate()

        {

            $model = new Project();

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

                $model->created\_at = time();

                $model->updated\_at = time();

                if ($model->save()) {

                    return $this->redirect(['view', 'id' => $model->id]);

                }

            }

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

                'model' => $model,

            ]);

        }

        public function actionUpdate($id)

        {

            $model = $this->findModel($id);

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

                $model->updated\_at = time();

                if ($model->save()) {

                    return $this->redirect(['view', 'id' => $model->id]);

                }

            }

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

                'model' => $model,

            ]);

        }

        public function actionDelete($id)

        {

            $this->findModel($id)->delete();

            return $this->redirect(['index']);

        }

        protected function findModel($id)

        {

            if (($model = Project::findOne($id)) !== null) {

                return $model;

            }

            throw new NotFoundHttpException('The requested page does not exist.');

        }

    }

?>

**Create Views for CRUD Operations**

create the corresponding views for the CRUD operations in the views/project directory.

F:\xampp\htdocs\yii2\project3\views\project\index.php

<?php

use yii\helpers\Html;

use yii\grid\GridView;

$this->title = 'Projects';

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

?>

<div class="project-index">

    <h1><?= Html::encode($this->title) ?></h1>

    <p>

        <?= Html::a('Create Project', ['create'], ['class' => 'btn btn-success']) ?>

    </p>

    <?= GridView::widget([

        'dataProvider' => $dataProvider,

        'columns' => [

            ['class' => 'yii\grid\SerialColumn'],

            'id',

            'title',

            'description:ntext',

            'created\_at',

            'updated\_at',

            ['class' => 'yii\grid\ActionColumn'],

        ],

    ]); ?>

</div>

F:\xampp\htdocs\yii2\project3\views\project\create.php

<?php

use yii\helpers\Html;

use yii\widgets\ActiveForm;

$this->title = 'Create Project';

$this->params['breadcrumbs'][] = ['label' => 'Projects', 'url' => ['index']];

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

?>

<div class="project-create">

    <h1><?= Html::encode($this->title) ?></h1>

    <div class="project-form">

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

        <?= $form->field($model, 'title')->textInput(['maxlength' => true]) ?>

        <?= $form->field($model, 'description')->textarea(['rows' => 6]) ?>

        <div class="form-group">

            <?= Html::submitButton('Create', ['class' => 'btn btn-success']) ?>

        </div>

        <?php ActiveForm::end(); ?>

    </div>

</div>

F:\xampp\htdocs\yii2\project3\views\project\update.php

<?php

use yii\helpers\Html;

use yii\widgets\ActiveForm;

$this->title = 'Update Project: ' . $model->title;

$this->params['breadcrumbs'][] = ['label' => 'Projects', 'url' => ['index']];

$this->params['breadcrumbs'][] = ['label' => $model->title, 'url' => ['view', 'id' => $model->id]];

$this->params['breadcrumbs'][] = 'Update';

?>

<div class="project-update">

    <h1><?= Html::encode($this->title) ?></h1>

    <div class="project-form">

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

        <?= $form->field($model, 'title')->textInput(['maxlength' => true]) ?>

        <?= $form->field($model, 'description')->textarea(['rows' => 6]) ?>

        <div class="form-group">

            <?= Html::submitButton('Update', ['class' => 'btn btn-primary']) ?>

        </div>

        <?php ActiveForm::end(); ?>

    </div>

</div>

F:\xampp\htdocs\yii2\project3\views\project\view.php

<?php

use yii\helpers\Html;

$this->title = $model->title;

$this->params['breadcrumbs'][] = ['label' => 'Projects', 'url' => ['index']];

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

?>

<div class="project-view">

    <h1><?= Html::encode($this->title) ?></h1>

    <p>

        <?= Html::a('Update', ['update', 'id' => $model->id], ['class' => 'btn btn-primary']) ?>

        <?= Html::a('Delete', ['delete', 'id' => $model->id], [

            'class' => 'btn btn-danger',

            'data' => [

                'confirm' => 'Are you sure you want to delete this item?',

                'method' => 'post',

            ],

        ]) ?>

    </p>

    <table class="table table-bordered table-striped">

        <tr>

            <th><?= Html::encode($model->getAttributeLabel('id')) ?></th>

            <td><?= $model->id ?></td>

        </tr>

        <tr>

            <th><?= Html::encode($model->getAttributeLabel('title')) ?></th>

            <td><?= $model->title ?></td>

        </tr>

        <tr>

            <th><?= Html::encode($model->getAttributeLabel('description')) ?></th>

            <td><?= $model->description ?></td>

        </tr>

        <tr>

            <th><?= Html::encode($model->getAttributeLabel('created\_at')) ?></th>

            <td><?= date('Y-m-d H:i:s', $model->created\_at) ?></td>

        </tr>

        <tr>

            <th><?= Html::encode($model->getAttributeLabel('updated\_at')) ?></th>

            <td><?= date('Y-m-d H:i:s', $model->updated\_at) ?></td>

        </tr>

    </table>

</div>

1. **Dashboad showing list of users for admin role only while manager can see list projects :**

Create UserController.php

F:\xampp\htdocs\yii2\project3\controllers\UserController.php

<?php

namespace app\controllers;

use Yii;

use app\models\User;

use app\models\UserSearch;

use yii\web\Controller;

use yii\web\NotFoundHttpException;

use yii\web\ForbiddenHttpException;

class UserController extends Controller

{

    public function actionIndex()

    {

        $searchModel = new UserSearch();

        $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

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

            'searchModel' => $searchModel,

            'dataProvider' => $dataProvider,

        ]);

    }

    public function actionView($id)

    {

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

            'model' => $this->findModel($id),

        ]);

    }

    public function actionCreate()

    {

        $model = new User();

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

            return $this->redirect(['view', 'id' => $model->id]);

        }

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

            'model' => $model,

        ]);

    }

    public function actionUpdate($id)

    {

        $model = $this->findModel($id);

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

            return $this->redirect(['view', 'id' => $model->id]);

        }

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

            'model' => $model,

        ]);

    }

    public function actionDelete($id)

    {

        $this->findModel($id)->delete();

        return $this->redirect(['index']);

    }

    protected function findModel($id)

    {

        if (($model = User::findOne($id)) !== null) {

            return $model;

        }

        throw new NotFoundHttpException('The requested page does not exist.');

    }

}

**Create the Views**

**F:\xampp\htdocs\yii2\project3\views\user\create.php**

<?php

use yii\helpers\Html;

use yii\widgets\ActiveForm;

$this->title = 'Create User';

$this->params['breadcrumbs'][] = ['label' => 'Users', 'url' => ['index']];

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

?>

<div class="user-create">

    <h1><?= Html::encode($this->title) ?></h1>

    <div class="user-form">

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

        <?= $form->field($model, 'name')->textInput(['maxlength' => true]) ?>

        <?= $form->field($model, 'email')->textInput(['maxlength' => true]) ?>

        <?= $form->field($model, 'password')->textInput(['maxlength' => true]) ?>

        <?= $form->field($model, 'role')->textInput(['maxlength' => true]) ?>

        <div class="form-group">

            <?= Html::submitButton('Create', ['class' => 'btn btn-success']) ?>

        </div>

        <?php ActiveForm::end(); ?>

    </div>

</div>

**F:\xampp\htdocs\yii2\project3\views\user\index.php**

<?php

use yii\helpers\Html;

use yii\grid\GridView;

$this->title = 'Users';

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

?>

<div class="user-index">

    <h1><?= Html::encode($this->title) ?></h1>

    <!-- <p>

        <?= Html::a('Create User', ['create'], ['class' => 'btn btn-success']) ?>

    </p> -->

    <?= GridView::widget([

        'dataProvider' => $dataProvider,

        'filterModel' => $searchModel,

        'columns' => [

            ['class' => 'yii\grid\SerialColumn'],

            'name',

            'email:email',

            'role',

            ['class' => 'yii\grid\ActionColumn'],

        ],

    ]); ?>

</div>

**F:\xampp\htdocs\yii2\project3\views\user\update.php**

<?php

use yii\helpers\Html;

use yii\widgets\ActiveForm;

$this->title = 'Update User: ' . $model->name;

$this->params['breadcrumbs'][] = ['label' => 'Users', 'url' => ['index']];

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

?>

<div class="user-update">

    <h1><?= Html::encode($this->title) ?></h1>

    <div class="user-form">

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

        <?= $form->field($model, 'name')->textInput(['maxlength' => true]) ?>

        <?= $form->field($model, 'email')->textInput(['maxlength' => true]) ?>

        <?= $form->field($model, 'role')->textInput(['maxlength' => true]) ?>

        <div class="form-group">

            <?= Html::submitButton('Update', ['class' => 'btn btn-primary']) ?>

        </div>

        <?php ActiveForm::end(); ?>

    </div>

</div>

**F:\xampp\htdocs\yii2\project3\views\user\view.php**

<?php

use yii\helpers\Html;

$this->title = $model->name;

$this->params['breadcrumbs'][] = ['label' => 'Users', 'url' => ['index']];

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

?>

<div class="user-view">

    <h1><?= Html::encode($this->title) ?></h1>

    <p>

        <?= Html::a('Update', ['update', 'id' => $model->id], ['class' => 'btn btn-primary']) ?>

        <?= Html::a('Delete', ['delete', 'id' => $model->id], [

            'class' => 'btn btn-danger',

            'data' => [

                'confirm' => 'Are you sure you want to delete this user?',

                'method' => 'post',

            ],

        ]) ?>

    </p>

    <ul>

        <li><strong>Name:</strong> <?= Html::encode($model->name) ?></li>

        <li><strong>Email:</strong> <?= Html::encode($model->email) ?></li>

        <li><strong>Role:</strong> <?= Html::encode($model->role) ?></li>

    </ul>

</div>

**create a UserSearch model to handle search/filtering in the index page.**

**F:\xampp\htdocs\yii2\project3\models\UserSearch.php**

<?php

namespace app\models;

use yii\base\Model;

use yii\data\ActiveDataProvider;

class UserSearch extends User

{

    public function rules()

    {

        return [

            [['name', 'email', 'role'], 'safe'],

        ];

    }

    public function search($params)

    {

        $query = User::find();

        $dataProvider = new ActiveDataProvider([

            'query' => $query,

        ]);

        $this->load($params);

        if (!$this->validate()) {

            return $dataProvider;

        }

        $query->andFilterWhere(['like', 'name', $this->name])

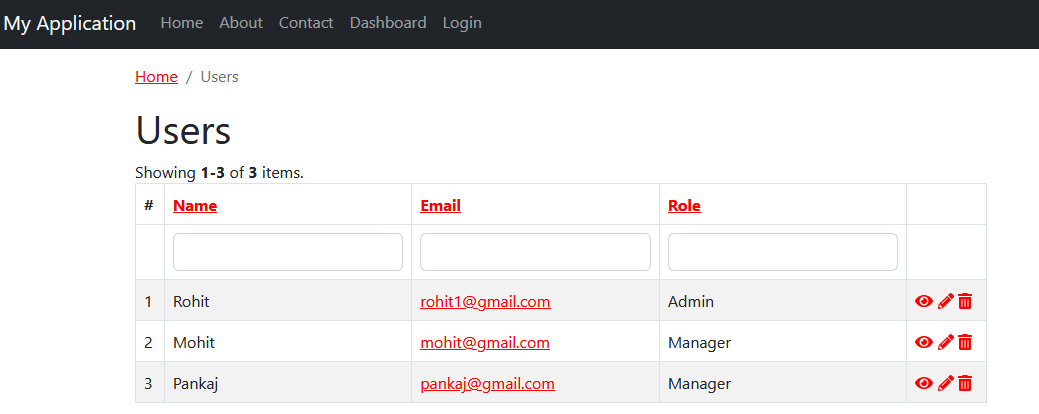
              ->andFilterWhere(['like', 'email', $this->email])

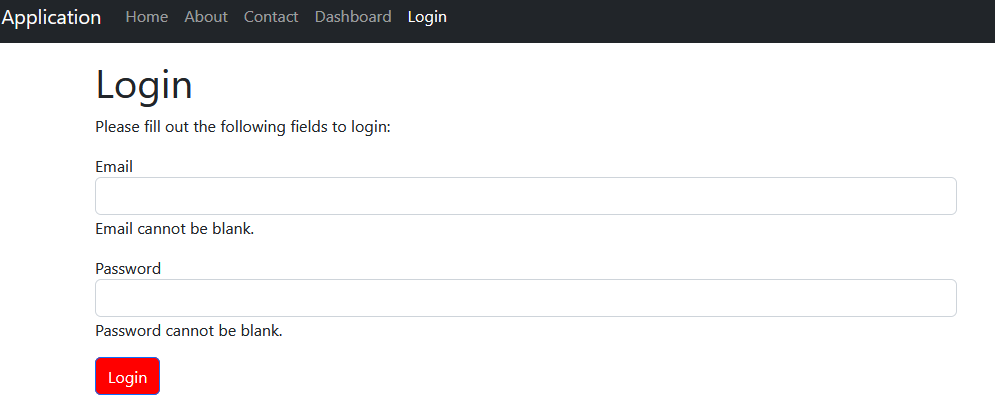
              ->andFilterWhere(['like', 'role', $this->role]);

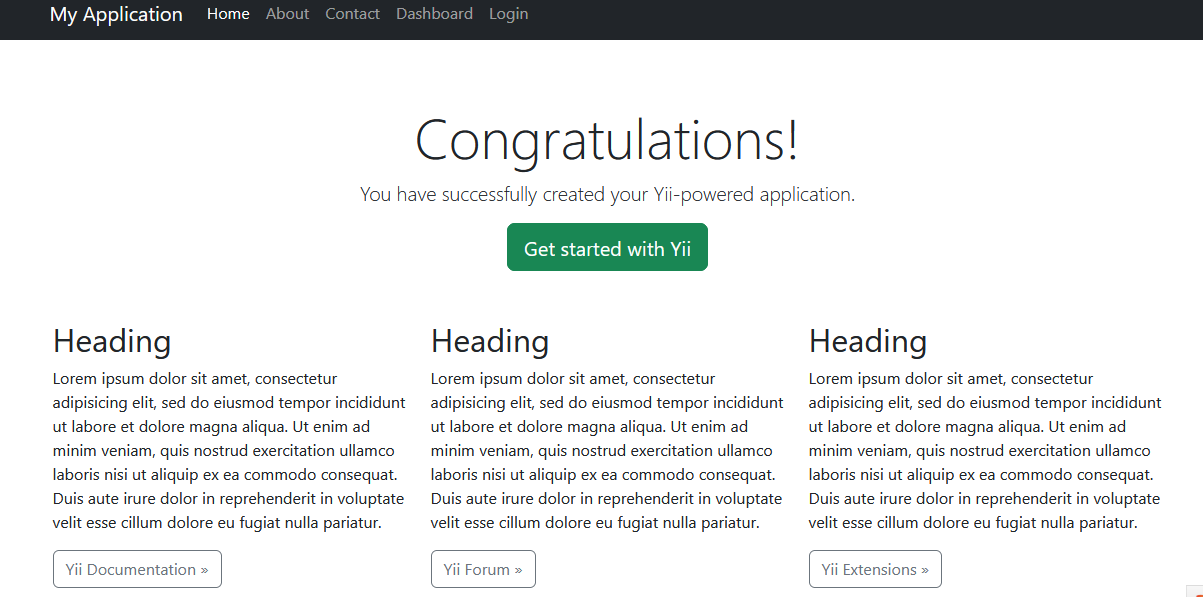
        return $dataProvider;

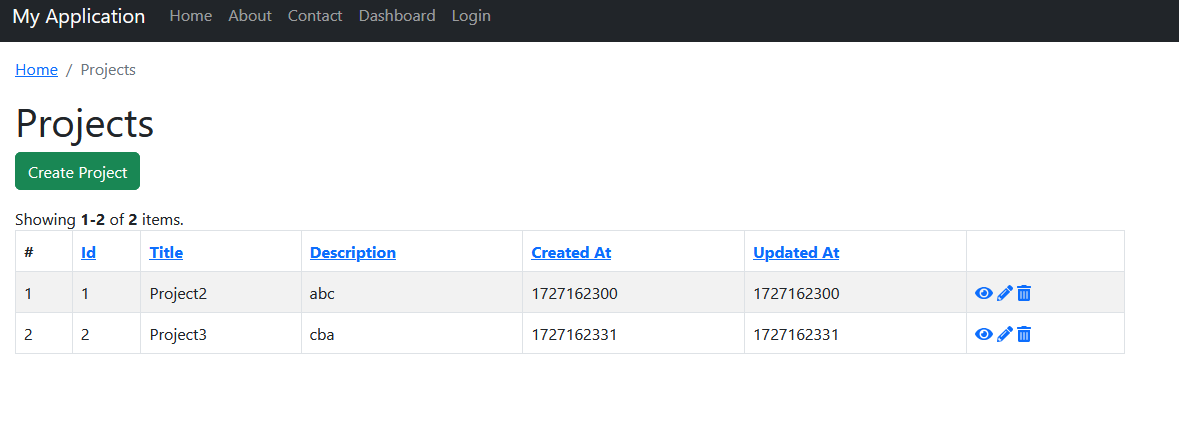
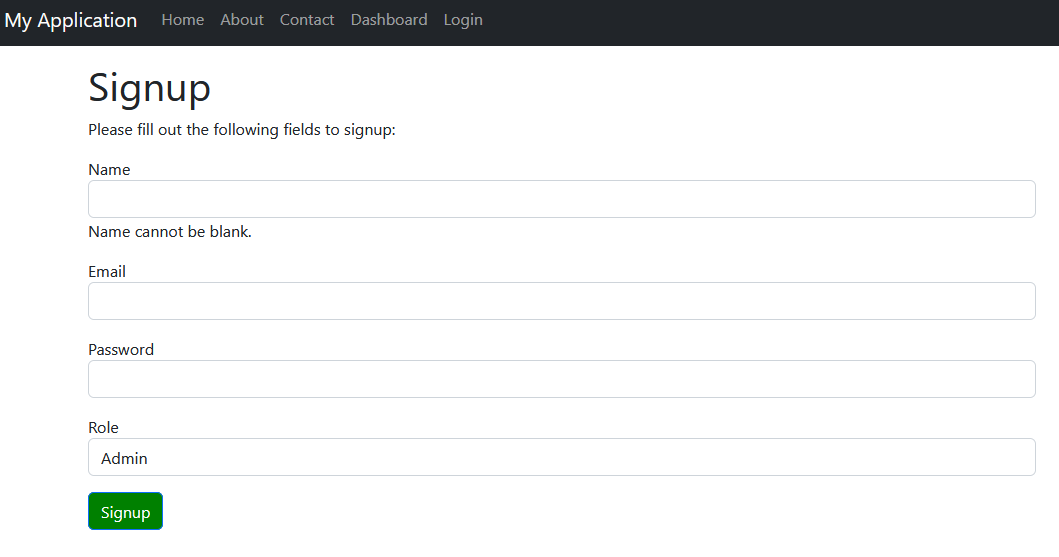
    }

}

****

****

****

****