

一、任务分工

1、具体分工

通过爬虫实现疫情数据表实时更新、数据库存储过程设计、Contactme 模块编写、ppt 制作及展示：苑华莹

后台的搭建与完善、前台成员信息动态展示页面、需求文档与设计文档的编写：陈燕

疫情新闻模块编写、标题滚动效果及实时链接的实现、模板连接、项目整合、个人作业下载模块编写：郑惠麟

通过爬虫实现经济数据表实时更新、绘制随数据库更新的折线图、用户手册与部署文档的编写：张明悦

疫情数据可视化、谣言模块编写：郑惠麟、苑华莹

需求文档编写、数据库设计：苑华莹、张明悦、郑惠麟、陈燕

2、Tower 截图

05月13日 周三				
✔ 掌握yii框架的文件架构，能够在合适位置展示模板...	清单外任务		普通	5月13日，郑惠麟
✔ 掌握yii框架的文件架构，能够在合适位置展示模板...	清单外任务		普通	5月13日，张明悦
05月12日 周二				
✔ 掌握yii框架的文件架构，能够在合适位置展示模板...	清单外任务		普通	5月12日，陈燕
✔ 实时更新后台数据库	清单外任务		普通	5月12日，苑华莹
✔ 掌握yii框架的文件架构，能够在合适位置展示模板...	清单外任务		普通	5月12日，苑华莹
05月10日 周日				
✔ 疫情展示模块 🔗 新窗口打开	清单外任务		普通	5月10日，苑华莹
05月02日 周六				
✔ 通过axure或其他软件为团队设计一个前后台模板	清单外任务		普通	5月2日，郑惠麟
✔ 股票信息py爬取	清单外任务		普通	5月2日，苑华莹
✔ 学习如何通过github进行多人开发，下次例会给夫...	清单外任务		普通	5月2日，苑华莹
✔ 搭建团队yii框架，学习如何多人使用yii框架	清单外任务		普通	5月2日，烟沉东西
05月01日 周五				
✔ 学习爬虫，获得股票数据资源	清单外任务		普通	5月1日，苑华莹

[互联网数据库大作业](#)

今天 周日	任务清单	标签	优先级	完成时间
<div>✓ 页面布局排版</div>	清单外任务		普通	12 分钟前, 陈燕
<div>✓ 后台功能的搭建与完善</div>	清单外任务		普通	5 小时前, 苑华莹
<div>✓ 新闻展示</div>	清单外任务		普通	5 小时前, 苑华莹
<div>✓ 折线图</div>	清单外任务		普通	5 小时前, 苑华莹
昨天 周六				
<div>✓ 静态显示经济图线</div>	清单外任务		普通	昨天, 苑华莹
06月08日 周一				
<div>✓ 实时滚动条展示新闻标题</div>	清单外任务		普通	6月8日, 郑惠麟
<div>✓ 表格展示疫情数据</div> <div>🔗 新窗口打开</div>	清单外任务		普通	6月8日, 郑惠麟
05月29日 周五				
<div>✓ 用疫情数据实时更新数据库</div>	清单外任务		普通	5月29日, 苑华莹
05月26日 周二				
<div>✓ 前后台衔接</div>	清单外任务		普通	5月26日, 陈燕
05月20日 周三				
<div>✓ 作业模块与部署文档</div>	清单外任务		普通	5月20日, 郑惠麟

3、github 部分截图



Create update.php

 yhy-2000 committed 3 days ago

Create view.php

 yhy-2000 committed 3 days ago

Create index.php

 yhy-2000 committed 3 days ago


Create create.php

 yhy-2000 committed 3 days ago

Rename _form to _form.php

 yhy-2000 committed 3 days ago

Create _search.php

 yhy-2000 committed 3 days ago

Create _form

 yhy-2000 committed 3 days ago

Add files via upload

 carinanan committed 19 minutes ago

Add files via upload

 carinanan committed 19 minutes ago

Add files via upload

 carinanan committed 7 hours ago

Add files via upload

 carinanan committed 7 hours ago

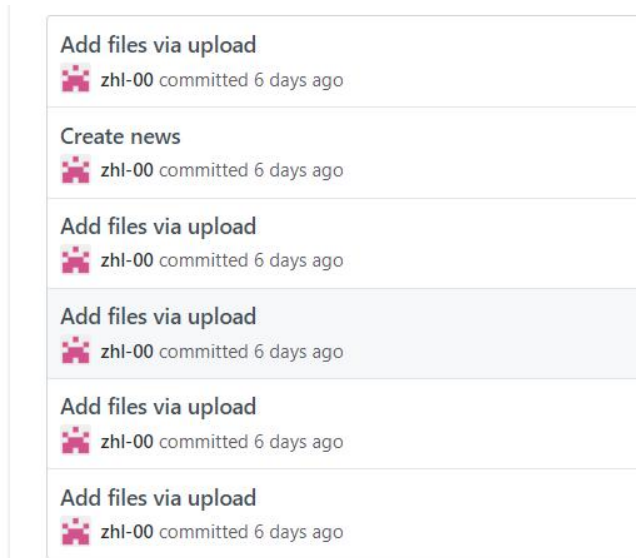
Add files via upload

 carinanan committed 7 hours ago

 Commits on Jun 12, 2020

Create 张明悦

 carinanan committed 2 days ago



二、代码部分

前台：

1、前台模板连接

1.1、创建一个资源类作为前台网页布局中新的 layout 文件

编辑 yii/frontend/assets/AppAsset.php

在其中进行补充：

```
public $js = [
```

引用模板中 index.html 文件中引用的所有.js 文件

```
];
```

```
public $css = [
```

```
// 'css/site.css', (注释 yii 中原有内容)
```

引用模板中 index.html 文件中引用的所有.css 文件

];

```
class AppAsset extends AssetBundle
{
    public $basePath = '@webroot';
    public $baseUrl = '@web/statics';
    public $css = [
        // 'css/site.css',
        'assets/vendor/bootstrap/css/bootstrap.css' ,
        'assets/vendor/font-awesome/css/font-awesome.css' ,
        'assets/vendor/magnific-popup/magnific-popup.css' ,
        'assets/vendor/bootstrap-datepicker/css/datepicker3.css' ,

        //<!-- Specific Page Vendor CSS -->
        'assets/vendor/jquery-ui/css/ui-lightness/jquery-ui-1.10.4.custom.css' ,
        'assets/vendor/bootstrap-multiselect/bootstrap-multiselect.css' ,
        'assets/vendor/morris/morris.css' ,

        //<!-- Theme CSS -->
        'assets/stylesheets/theme.css' ,

        //<!-- Skin CSS -->
        'assets/stylesheets/skins/default.css' ,

        //<!-- Theme Custom CSS -->
        'assets/stylesheets/theme-custom.css',
    ];
    public $js = [
        //<!-- Head Libs -->
        'assets/vendor/modernizr/modernizr.js',

        //<!-- Vendor -->
        'assets/vendor/jquery/jquery.js',
        'assets/vendor/jquery-browser-mobile/jquery.browser.mobile.js',
        'assets/vendor/bootstrap/js/bootstrap.js',
        'assets/vendor/nanoscroll/nanoscroll.js',
        'assets/vendor/bootstrap-datepicker/js/bootstrap-datepicker.js',
        'assets/vendor/magnific-popup/magnific-popup.js',
        'assets/vendor/jquery-placeholder/jquery.placeholder.js',
    ];
}
```

1.2、编辑 yii/frontend/views/layout/main_layout.php

其中内容为原模板 index.html 的主体部分

```

AppAsset::register($this);
?>
<?php $this->beginPage() ?>
<!DOCTYPE html>
<html lang="<?= Yii::$app->language ?>">
<head>
    <meta name="renderer" content="webkit">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta charset="<?= Yii::$app->charset ?>">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <?= Html::csrfMetaTags() ?>
    <title><?= Html::encode($this->title) ?></title>
    <?php $this->head() ?>
</head>
<body>
<?php $this->beginBody() ?>

    <!-- start: header -->
    <header class="header">
        <div class="logo-container">
            <a href=".." class="logo">
                
            </a>
            <div class="visible-xs toggle-sidebar-left" data-toggle-class="sidebar-left-opened" data-target="html"
                data-fire-event="sidebar-left-opened">
                <i class="fa fa-bars" aria-label="Toggle sidebar"></i>
            </div>
        </div>

        <!-- start: search & user box -->
        <div class="header-right">

            <span class="separator"></span>

```

并且为了实现后续内容的嵌套，在 main_layout.php 的合适位置加入

```

<!--Begin:Content-->
<section id="content" class="table-layout animated fadeIn">
    <?= $content ?>
</section>
<!-- End: Content -->

```

在<head></head>中加入下列语句，让视图确定网页的 head 位置，以引入 css 文件

```

<title><?= Html::encode($
<?php $this->head() ?>

```

在<body></body>中加入下列语句，让视图确定网页的 body 位置，以在最后引入 js 文件

1.3、将模板具体应用至前台网页布局

在 frontend/controllers/sitecontroller.php 中 class sitecontroller 中声明全局变量 public \$layout = "main_layout";

```

    */
class SiteController extends Controller
{
    /**
     * {@inheritdoc}
     */
    public $layout = "main_layout";
}

```

1.4、引入所需 js 文件

新建一个 statics 文件夹，将模板的所有 js 文件等拷在这个文件夹下，然后将这个文件夹拷贝到 yii/frontend/web 目录下(如果出现问题,检查 AppAsset.php 中的 `public $baseUrl = '@web/statics';`)

2、前台登录和注册页面布局更改

具体实现步骤与上述前台模板连接大体一致，在最终具体应用时在 frontend/controllers/sitecontroller.php 中 sitecontroller 类中的 function actionLogin 和 function actionSignup 中修改全局变量 \$layout 的值，如下图所示：

```

public function actionLogin()
{
    $this->layout="login_layout";
    if (!$this->user->isGuest) {
        // ...
    }
}

public function actionSignup()
{
    $this->layout="signup_layout";
    $model = new SignupForm();
    // ...
}

```

3、成员信息展示

3.1、使用 layui 来搭建信息展示模块，在 assets 下添加 LayuiAsset.php 引入静态资源

3.2、为了能直接通过后台修改前台的成员信息展示，需要修改 MemberController.php，获取数据表的值


```

public function actionIndex(){
    $model= Member::find()->asArray()->all();
    $name = ArrayHelper::getColumn($model,'name');
    $stu_id=ArrayHelper::getColumn($model,'stu_id');
    $gender=ArrayHelper::getColumn($model,'gender');
    $college=ArrayHelper::getColumn($model,'college');
    $email=ArrayHelper::getColumn($model,'email');
    $message=ArrayHelper::getColumn($model,'message');
    return $this->render('index',[
        'name'=>$name,
        'stu_id'=>$stu_id,
        'gender'=>$gender,
        'college'=>$college,
        'email'=>$email,
        'message'=>$message,
    ]);
}

```

修改 views\member\index.php

```

<?php $i=0 ?>
<?php foreach ($name as $value):?>
<div class="layui-container">
    <div class="pull-right" style="position: absolute; left: 650px; top: 10px;">
        
    </div>
    <br>
    <div class="layui-row">
        <div class="layui-col-md3" style="position: relative; left: 100px;">
            <i class="fa fa-user fa-2x">姓名</i>
        </div>
    </div>
    <div class="layui-row">
        <div class="layui-col-md3" style="position: relative; left: 100px;">
            <p class="h4"><?= $name[$i] ?></p>
        </div>
    </div>
    <br>
    <div class="layui-row">
        <div class="layui-col-md3" style="position: relative; left: 100px;">
            <i class="fa fa-info-circle fa-2x">学号</i>
        </div>
    </div>
    <div class="layui-row">
        <div class="layui-col-md3" style="position: relative; left: 100px;">
            <p class="h4"><?= $stu_id[$i] ?></p>
        </div>
    </div>
    <br>
    <div class="layui-row">
        <div class="layui-col-md3" style="position: relative; left: 100px;">
            <i class="fa fa-superpowers fa-2x">座右铭</i>
        </div>
    </div>
    <div class="layui-row">
        <div class="layui-col-md10" style="position: relative; left: 100px;">
            <p class="h4"><?= $message[$i] ?></p>
        </div>
    </div>
</div>
</div>

```

4、疫情数据更新部分：

通过 python 爬虫，每隔十分钟从 API 网站上爬取数据：

```
def run(self):
    while True:
        self.listener()
        time.sleep(600)
```

```
while True:
    request = requests.get(url='https://lab.isaacclin.cn/nCoV/api/' + collections.get(collection))
    if request.status_code == 200:
        current_data = request.json()
        break
    else:
        time.sleep(1)
        continue
```

执行 sql 语句，将 json 文件里的数据导入 mysql 数据库，分别对 DXYArea、DXYNews、DXYRumors、China 数据表进行更新

如果数据存在更新，则将 `current_data` 列表传给 `mysql_dumper` 函数

```
# 爬虫爬取信息完毕，下面开始导入数据库
if static_data != current_data: # 数据存在更新
    self.json_dumper(collection=table, content=current_data)
    changed_files.append('json/' + table + '.json')
    self.mysql_dumper(table, current_data)
```

函数体内对地名进行判断，如果这个地名之前被记录过，则调用 UpdateProc 存储过程：

程序名称	updateChinaProc					
类型	PROCEDURE					
参数	方向	名字	类型	长度/值	选项	
	↑	IN	currer	IN		删除
	↑	IN	confir	IN		删除
	↑	IN	suspe	IN		删除
	↑	IN	cured	IN		删除
	↑	IN	deadC	IN		删除
	↑	IN	cityNa	IN		删除
添加参数						
定义	<pre> 1 BEGIN 2 UPDATE china 3 SET china.confirmedCount = confirmedCount1 4 WHERE china.cityname=cityname; 5 6 UPDATE china 7 SET china.currentConfirmedCount = currentConfirmedCount1 </pre>					

更新该地名对应的 count 属性；否则，新增一条完整记录

```
else:#之前没有出现过就插入
    sql2=~insert into dycarea values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)~
    value2 = (
        item["locationId"], item["continentName"], item["continentEnglishName"], item["countryName"],
        item["countryEnglishName"], item["countryFullName"], item["provinceName"],
        item["provinceEnglishName"],
        item["provinceShortName"], item["currentConfirmedCount"], item["confirmedCount"],
        item["suspectedCount"],
        item["curedCount"], item["deadCount"], item["comment"], item["cities"],
        item["updateTime"])
self.cursor.execute(sql2, value2)
self.db.commit()
```

5、疫情全球数据展示

5.1、用 gii 在 common/models 下生成 Area.php，对应于数据表 dxyarea。

5.2、在 frontend/controllers 下编写 AreaController.php，借助对应的 model 从对应的数据表中获取数据信息，并且通过 yii\data\Pagination 完成分页操作，最终将获取的数据渲染至对应的视图层。

```
namespace frontend\controllers;

use Yii;
use yii\web\Controller;
use common\models\Area;
use yii\helpers\ArrayHelper;
use yii\data\Pagination;
use yii\grid\GridView;
use yii\db\ActiveRecord;

class AreaController extends Controller{
    public $layout = "main_layout";
    public function actionIndex() {
        $query = Area::find();
        $countQuery = clone $query;
        $pageSize = 15;
        $pages = new Pagination(['totalCount' => $countQuery->count(), 'pageSize' => $pageSize]);
        $models = $query->offset($pages->offset)
            ->limit($pages->limit)
            ->all();
        return $this->render('index', [
            'models' => $models,
            'pages' => $pages,
        ]);
    }
}
```

5.3、在 frontend/views 下创建文件夹 area，并编写文件 index.php，借助 foreach 方法将对应获取的数据以表格的形式循环展示，并结合 LinkPager 完成分页操作

```

use yii\helpers\Html;
use yii\widgets\LinkPager;
use yii\widgets;

$this->title = Yii::t('app', '全球数据');
$this->params['breadcrumbs'][] = $this->title;
?>

<div align="center">
    <h1><?= Html::encode($this->title) ?></h1>
    <table width="880" border="0" align="center">
        <tr align="center" height="50">
            <td width="143" bgcolor="#f0ffff">countryName</td>
            <td width="185" bgcolor="#ffb6c1">currentConfirmedCount</td>
            <td width="134" bgcolor="#ff6347">confirmedCount</td>
            <td width="147" bgcolor="#e6e6fa">suspectedCount</td>
            <td width="114" bgcolor="#f0ffff">curedCount</td>
            <td width="131" bgcolor="#f5f5dc">deadCount</td>
        </tr>
        <?php foreach ($models as $model):?>
            <tr align="center">
                <td><?= "<li>".$model['countryName']."</li>" ?>
                <td><?= "<li>".$model['currentConfirmedCount']."</li>" ?>
                <td><?= "<li>".$model['confirmedCount']."</li>" ?>
                <td><?= "<li>".$model['suspectedCount']."</li>" ?>
                <td><?= "<li>".$model['curedCount']."</li>" ?>
                <td><?= "<li>".$model['deadCount']."</li>" ?>
            </tr>
        <?php endforeach ?>
    </table>
</div>
<?= LinkPager::widget([
    'pagination' => $pages,
    'nextPageLabel' => '下一页',
    'prevPageLabel' => '上一页',
    'firstPageLabel' => '首页',
    'lastPageLabel' => '尾页',
]);
?>

```

6、新闻标题滚动及数据展示

6.1、用 gii 在 common/models 下生成 News.php，对应于数据表 dxynews。

6.2、在 frontend/controllers 下编写 NewsController.php，借助对应的 model 从对应的数据表中获取数据信息，并且通过 yii\data\Pagination 完成分页操作，最终将获取的数据渲染至对应的视图层。


```

namespace frontend\controllers;

use Yii;
use yii\web\Controller;
use common\models\News;
use yii\helpers\ArrayHelper;
use yii\data\Pagination;
use yii\grid\GridView;
use yii\db\ActiveRecord;

class NewsController extends Controller{
    public $layout = "main_layout";
    public function actionIndex(){
        $model= News::find()->asArray()->all();
        $title = ArrayHelper::getColumn($model,'title');
        $source=ArrayHelper::getColumn($model,'sourceTrl');
        $summary=ArrayHelper::getColumn($model,'summary');
        $infosource=ArrayHelper::getColumn($model,'infoSource');

        $query = News::find();
        $countQuery = clone $query;
        $pageSize = 3;
        $pages = new Pagination(['totalCount' => $countQuery->count(),'pageSize' => $pageSize]);
        $models = $query->offset($pages->offset)
            ->limit($pages->limit)
            ->all();
        return $this->render('index',[
            'models'=>$models,
            'pages'=>$pages,
            'source'=>$source,
            'summary'=>$summary,
            'infosource'=>$infosource,
            'title'=>$title]);
    }
}

```

6.3、在 frontend/views 下创建文件夹 news，并编写文件 index.php，借助 foreach 方法将对应获取的新闻数据以表格的形式循环展示，结合 LinkPager 完成分页操作。并通过 css3 中的@keyframes 完成新闻标题的动态垂直滚动效果。使用百分比形式将动画总时长 33s 进行了划分，并规定了在指定时间区间内做向上移动的“距离”。同时利用从数据表中获取到的 source 属性，将滚动的新闻标题链接到对应的原新闻网页，实现新闻的全面浏览。

```

<style>
ul,li{ margin:0; padding:0; }
li{ list-style:none; }
.container{ display:inline-block; background: #ffffff; padding:10px; border:1px solid #ccc; font-family:"Microsoft YaHei"; }
.container .li-box{ height:28px; overflow:hidden; }
.container ul{ position:relative; animation: moveUp 33s infinite; -webkit-animation: moveUp 33s infinite; }
@keyframes moveUp
{
0% {top:0px;}
10% {top:0px;}
20% {top:-20px;}
30% {top:-40px;}
40% {top:-60px;}
50% {top:-80px;}
60% {top:-100px;}
70% {top:-120px;}
80% {top:-140px;}
90% {top:-160px;}
100% {top:-180px;}
}
</p> <p> @-webkit-keyframes moveUp /*Safari and Chrome*/
{
0% {top:0px;}
10% {top:0px;}
20% {top:-20px;}
30% {top:-40px;}
40% {top:-60px;}
50% {top:-80px;}
60% {top:-100px;}
70% {top:-120px;}
80% {top:-140px;}
90% {top:-160px;}
100% {top:-180px;}
}
}
.container li{ line-height:1.8; color:#666; }
</style>

<body>
<h1 align="center"><? = Html::encode($this->title) ?></h1>
<div class="container">
<div class="li-box">
<ul>
<?php $i=0 ?>
<?php foreach ($title as $value): ?>
<? = "<a href=$source[$i]>".$title[$i]."</a>" ?>
<br>
<?php $i=$i+1 ?>
<?php endforeach?>
</ul>
</div>
</div>
<div align="center">
<table width="1000" border="0" align="center">
<?php $i=0 ?>
<?php foreach ($models as $value):?>
<tr align="center">
<td><? = "<h3>".$value['title'].</h3>" ?><br><? = $value['summary'] ?><br><div align="right"><? = $value['infoSource'] ?></div></td>
</tr>
<?php $i=$i+1?>
<?php endforeach ?>
</table>
</div>
</body>
</html>
<? = LinkPager::widget([
'pagination' => $pages,
'nextPageLabel' => '下一页',
'prevPageLabel' => '上一页',
'firstPageLabel' => '首页',
'lastPageLabel' => '尾页',
]);
?>

```

7、经济模块

7.1、首先通过 python 代码爬取到雪球网最近 30 天的股票资讯，将其导入数据库 stock 和 stock_volume 中，在前台展示出该股票的有关收盘价和成交量的折线图。

```

<h2>
    收盘价:
    <strong>
        <select class="form-control" id="salesSelector">
            <?php foreach ($data as $v) {?>
                <option value="<?php echo $v['value']>"><?php echo $v['name']></option>
            <?php }?>
        </select>
    </strong>
</h2>

<div id="salesSelectorItems" class="chart-data-selector-items mt-sm">
    <!-- Plot: Sales JSOF Admin -->
    <?php foreach ($data as $k=>$v) {?>
        <div class="chart chart-sm" data-sales-rel="<?php echo $v['value']>" id="<?php echo $v['value']>" class="chart-hidden"></div>
        <script>
            var <?php echo $v['value']>=<?php echo md5($v['value'])>; = [ <?php echo json_encode($v['data'])> ];
        </script>
    <?php }?>

```

7.2、通过下拉框选择不同的股票种类展示，分别对应数据库的控制器 stock 和 stockvolume，定义 value1 和 value2 分别储存数据（此处选取收盘价折线图作为事例）

```

$( '#salesSelectorWrapper' ).addClass('ready');

<?php foreach ($data as $k=>$v) {?>
    var <?php echo $v['value1']>= $.plot("#<?php echo $v['value1']>", <?php echo $v['value1']>?<?php echo md5($v['value1'])>, {
        series: {
            lines: {
                show: true,
                lineWidth: 2
            },
            points: {
                show: true
            },
            shadowSize: 0
        },
        grid: {
            hoverable: true,
            clickable: true,
            borderColor: 'rgba(0,0,0,0.1)',
            borderWidth: 1,
            labelMargin: 15,
            backgroundColor: 'transparent'
        },
        yaxis: {
            min: 0,
            color: 'rgba(0,0,0,0.1)'
        },
        xaxis: {
            mode: 'categories',
            color: 'rgba(0,0,0,0)'
        },
        legend: {
            show: false
        }
    }
}

```

```

1  <?php
2
3  /*...*/
4
5
6
7
8  namespace common\models;
9
10 use ...
11
12
13
14
15
16 class Stock extends ActiveRecord
17 {
18     /**
19      * @inheritdoc
20      */
21     public static function tableName()
22     {
23         return '{{stock}}';
24     }
25 }

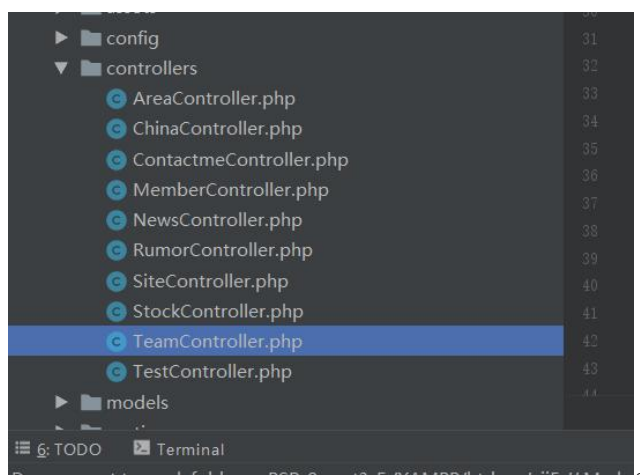
```

7.3、在前台侧边栏中定义 Controller 实现数据库对应读取

```
public function actionEconomic() {
    $data1 = array();
    $result = Stock::find()->select( columns: 'name')->groupBy( columns: "name")->limit( limit: 30)->all();
    $i=1;
    foreach ($result as $v)
    {
        $temp = Stock::find()->where(['name' => $v->name])->select( columns: 'day,price,volume')->all();
        $tempVolume = StockVolume::find()->where(['name' => $v->name])->select( columns: 'day,price,volume')->all();
        $result_temp['name'] = $v->name;
        $result_temp['value1'] = "a".fletDashSalesData.$i;
        $result_temp['value2'] = "b".fletDashSalesData.$i;

        $data1_echart = []
        $data2_echart = [];
    }
}
```

7.4、定义 TeamController，实现团队成员页面完善和侧边栏跳转。



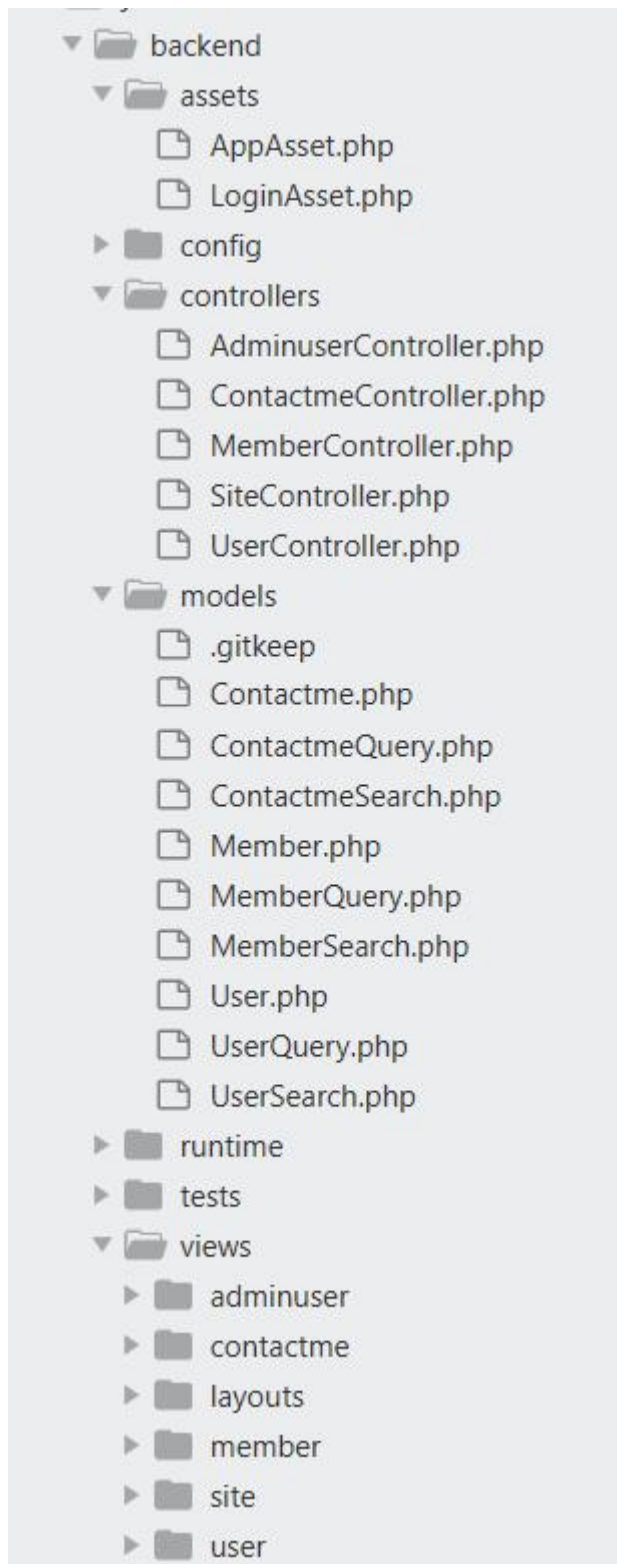
8、联系我们：

调用前台的 gii 生成 contactme 表对应的 model 以及 crud 模板，以这个模板为基础框架，删除用户“查看数据表”的功能，并对页面进行美工

stock 的表格展示部分用类似的方法实现，在表头增加了每一个属性的查询入口，方便用户查询相应信息

后台：

目录结构



1、登录

后台登录时必须为管理员身份才能登录，用一张 adminuser 数据表存储管理员用户。

1.1、将用户登录验证转换到 adminuser 表

```
'user' => [
    'identityClass' => 'common\models\Adminuser',
    'bootstrap\BootstrapAsset' => 'bootstrap\BootstrapAsset',
]
```

1.2 登录页面引入资源

```
namespace backend\assets;

use yii\web\AssetBundle;

/**
 * Main backend application asset bundle.
 */
class LoginAsset extends AssetBundle
{
    public $basePath = '@webroot';
    public $baseUrl = '@web';
    public $css = [
        'statics/css/font-awesome/css/font-awesome.min.css',
        'statics/css/layout.css',
    ];
    public $js = [
    ];
    public $depends = [
        'yii\web\YiiAsset',
        'yii\bootstrap\BootstrapAsset',
    ];
}
```

SiteController 中引入布局

```
$this->layout = "login.php";
```

修改 views\site 中的 login.php 和 index.php 文件

```

16 $this->title = '登录';
17 $this->params['breadcrumbs'][] = $this->title;
18 ?>
19 <div class="sign-overlay"></div>
20 <div class="signpanel"></div>
21
22 <div class="panel signin">
23   <div class="panel-heading">
24     <h4 class="panel-title">欢迎登陆系统后台</h4>
25   </div>
26   <div class="panel-body">
27     <?php $form = ActiveForm::begin(['id' => 'login-form']); ?>
28
29     <?= $form->field($model, 'username',[
30       'inputOptions'=>[
31         'placeholder'=>'请输入登录账号',
32       ],
33       'inputTemplate'=>'<div class="input-group">
34         <span class="input-group-addon"><i class="fa fa-user"></i></span><input>
35         </div>',
36     ]->textInput(['autofocus' => true])->label(false) ?>
37
38     <?= $form->field($model, 'password',[
39       'inputOptions'=>[
40         'placeholder'=>'请输入登录密码',
41       ],
42       'inputTemplate'=>
43         '<div class="input-group">
44           <span class="input-group-addon"><i class="fa fa-lock"></i></span><input>
45           </div>',
46     ]->passwordInput()->label(false) ?>
47
48     <!--div><a href="#" class="forgot">忘记密码</a></div-->
49
50
51   <div class="form-group">
52     <?= Html::submitButton('登录', ['class' => 'btn btn-primary btn-success btn-block', 'name' => 'login-button']) ?>
53   </div>
54
55   <?php ActiveForm::end(); ?>

```

1.3、登录验证功能

common\models 里的 Adminuser.php 文件使用 User.php 文件里的用户验证函数。

1.4、退出登录

退出登录后跳转到登录页面

```

16 <div class="profile-info" data-lock-name="John Doe" data-lock-email="johndoe@JSOFT.com">
17   <span class="role">管理员</span>
18   <p>
19     <?php echo $user = Yii::$app->user->identity->username; ?>
20   </p>
21 </div>
22
23 <i class="fa custom-caret"></i>
24 </a>
25
26 <div class="dropdown-menu">
27   <ul class="list-unstyled">
28     <li class="divider"></li>
29     <li>
30       <?= $menuItems[] = 'li'
31         . Html::beginForm(['/site/logout'], 'post')
32         . Html::submitButton(
33           '退出登录',
34           ['class' => 'btn btn-link logout']
35         )
36         . Html::endForm()
37         . '</li>'; ?>

```

2、用户管理和成员管理

使用 gii 生成的 MVC，更换布局为模板布局，可进行修改、添加、查看、删除操作。

3、留言管理

3.1、Model\Contactme.php 添加获取路径函数和显示部分留言函数

```
public function getUrl()
{
    return Yii::$app->urlManager->createUrl(
        ['contactme/view', 'id'=>$this->id, 'name'=>$this->name]);
}

public function getBeginning($length=50)
{
    $tmpStr = strip_tags($this->feedback);
    $tmpLen = mb_strlen($tmpStr);

    $tmpStr = mb_substr($tmpStr,0,$length,'utf-8');
    return $tmpStr.($tmpLen>$length?'...:');
}
```

3.2、在 index 页面使用列表形式而不是表格形式去展示用户的留言，展示的信息有每条留言的留言人姓名、留言时间以及留言的前段，点击留言人姓名可以跳转到 view.php 查看留言的详细信息。

```
<div class="row">
    <div class="col-md-8">
        <ol class="breadcrumb">
            <li><a href="{?=$app->homeUrl;?}">Home</a></li>
            <li>留言列表</li>
        </ol>

        <?= ListView::widget([
            'id'=>'messageList',
            'dataProvider'=>$dataProvider,
            'itemView'=>'_listitem',
            'layout'=>'{items}{pager}',
            'pager'=>[
                'maxButtonCount' => 5, //最多显示几个分页按钮
                'firstPageLabel' => '首页',
                'prevPageLabel' => '上一页',
                'nextPageLabel' => '下一页',
                'lastPageLabel' => '尾页'
            ],
        ])?>
    </div>
```

```
<div class="contactme">  
  <div class="name">  
    <h2><span class="glyphicon glyphicon-user" aria-hidden="true"><a href="?= $model->:url;"><?= Html::encode($model->name);  
?></a></span></h2>  
    <div class="content_time">  
      <span class="glyphicon glyphicon-time" aria-hidden="true"></span><em><?= Html::encode($model->content_time)."  
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&";?></em>  
    </div>  
  </div>  
  
  <br>  
  <div class="content">  
    <?= $model->beginning;?>  
  </div>  
</div>
```