

流浪 NK：实现文档

团队成员

姜天嘉

闫钊辉

张继盛

王天鹏

项目任务分配：

姜天嘉：实现了前台后台界面的设计，实现了前台数据分析板块，实现了后台作业提交界面，权限管理界面，成员简介管理界面

闫钊辉：实现了前台全局概况板块

王天鹏：实现了前台博客界面板块

张继盛：实现了前台新闻界面板块

Github 提交记录

```
commit 66ee1fe231cc363dc5a88a758d84879ffb882545 (HEAD -> master, origin/master)
```

```
Author: G2 OptJiang20 <2499184084@qq.com>
```

```
Date: Sat Feb 11 01:52:41 2023 +0800
```

```
add vendor
```

```
commit 1bcf7817b50bdf4bfe64b3983afde8a3a7f8f9dd
```

```
Author: G2 OptJiang20 <2499184084@qq.com>
```

```
Date: Fri Feb 10 17:14:55 2023 +0800
```

```
final project
```

```
commit 481ff7064585550ccad7e07d9c2d3aba37efa80f
```

```
Author: unknown <1341210299@qq.com>
```

```
Date: Fri Feb 10 15:06:56 2023 +0800
```

```
115
```

```
commit 13dd2bd2dd7daecd6689d06e289cafef2cc295c3
```

Merge: 79788c7 c13e4cb
Author: 王天鹏 <2664004187@qq.com>
Date: Fri Feb 10 07:21:24 2023 +0800

Merge branch 'master' of <https://github.com/JiangSio/Internet-Database-Project>

拉取远程分支。

commit 79788c77f21a03bc03e1482d5b2683c95acdb860
Author: 王天鹏 <2664004187@qq.com>
Date: Fri Feb 10 07:19:06 2023 +0800

”完善新闻展示模块“

commit c13e4cb69647dbfc071f9ba1d0281aad1271e27e
Author: G2 OptJiang20 <2499184084@qq.com>
Date: Thu Feb 9 00:05:27 2023 +0800

jtj+zjs

commit dd70b285ac8036fe4cd045e3ef1c4fcbc4608936
Author: unknown <1341210299@qq.com>
Date: Wed Feb 8 19:57:04 2023 +0800

init

commit 10c4feb323339469ed5388972c1ff8b9d3f921c8
Author: 2012615 <2536196890@qq.com>
Date: Wed Feb 8 15:24:27 2023 +0800

yii2022 1.0

commit c560aa48b618fd472b2c90a38f92ded92a57e270
Author: 2012615 <2536196890@qq.com>
Date: Wed Feb 8 15:21:36 2023 +0800

2

commit 82de1ff2b6a29986c9f2fb3a5a3977b12d0cb469
Author: 2012615 <2536196890@qq.com>
Date: Wed Feb 8 15:21:08 2023 +0800

1

commit be3dd7f781b208405bde8f116d2d2b5ba0f70fa4

Merge: 615f977 46c0adb

Author: 2012615 <2536196890@qq.com>

Date: Wed Feb 8 15:20:22 2023 +0800

Merge branch 'master' of <https://github.com/JiangSio/Internet-Database-Project>

commit 615f977adfd2a01354aef1cc26a9ae4b9a76bcf9

Author: 2012615 <2536196890@qq.com>

Date: Wed Feb 8 15:16:21 2023 +0800

yii2.0 advanced 1.0

commit 46c0adb369919e28e2cbbe7ea3a56321ea9f9617

Author: G2 OptJiang20 <2499184084@qq.com>

Date: Mon Feb 6 15:35:27 2023 +0800

前台后台首页添加

commit 8201e41b3151a1e11542a1f5375b774f9a955708

Author: G2 OptJiang20 <2499184084@qq.com>

Date: Mon Feb 6 15:13:16 2023 +0800

前端后台首页设计

commit 1e202b366a6b80901584595bf15aa1956faf2a94

Author: 2012615 <2536196890@qq.com>

Date: Sat Feb 4 16:51:10 2023 +0800

yii2更新模板

commit 54e2d8fc03850cc9a72cbb63b064a2890d560291

Merge: e48d637 793efcd

Author: 2012615 <2536196890@qq.com>

Date: Sat Feb 4 16:43:03 2023 +0800

Merge branch 'master' of <https://github.com/JiangSio/Internet-Database-Project>

commit e48d637bd49bfbac5547bac069fa75426485374a

Author: 2012615 <2536196890@qq.com>

Date: Sat Feb 4 16:40:39 2023 +0800

2

commit b5e9beeb36d7ae561e5083d9fe4d61b249f3b4f0

Author: 2012615 <2536196890@qq.com>

Date: Sat Feb 4 16:37:04 2023 +0800

更新模板yii2

commit 793efcd4147ed0a67a6696105cfb5d3571a9876f

Author: G2 OptJiang20 <2499184084@qq.com>

Date: Wed Feb 1 23:54:30 2023 +0800

中国与俄乌外贸分析

commit ee6d6d8619edd5a4f03fc0474c87b9c160500516

Author: G2 OptJiang20 <2499184084@qq.com>

Date: Wed Feb 1 21:04:16 2023 +0800

油价分析界面

commit 2c371df00ecfa852e0fbbd447d5007960374df25

Author: 2012615 <2536196890@qq.com>

Date: Tue Jan 31 17:34:56 2023 +0800

yii2

commit 9297b375f1ba8507882dd1e2bb9b144c051dbc8a

Author: 2012615 <2536196890@qq.com>

Date: Tue Jan 31 15:47:27 2023 +0800

9

commit 56b1ab843dcfa843fce65d180c737929e42f491d

Author: 2012615 <2536196890@qq.com>

Date: Tue Jan 31 15:46:48 2023 +0800

8

commit d62cb975b3757e8238daec63612d1431bfac4df4

Author: 2012615 <2536196890@qq.com>

Date: Tue Jan 31 15:46:23 2023 +0800

7

commit 66cf524d8bb18f7c54c461ae81d52a9101ae04d8
Author: 2012615 <2536196890@qq.com>
Date: Tue Jan 31 15:41:48 2023 +0800

担当

commit a8900c2f76e97601c80bbffa6d4628af21706caa
Author: 2012615 <2536196890@qq.com>
Date: Tue Jan 31 15:40:19 2023 +0800

6

commit fb3f0b196280970b54a09d234bc44386225bcf48
Author: 2012615 <2536196890@qq.com>
Date: Tue Jan 31 15:39:45 2023 +0800

5

commit 485e6f715279541cea3c760bfc846cb56f5200de
Author: 2012615 <2536196890@qq.com>
Date: Tue Jan 31 15:39:02 2023 +0800

4

commit bc35fabd33822c348e14a7b468c9faae958308e3
Author: 2012615 <2536196890@qq.com>
Date: Tue Jan 31 15:37:58 2023 +0800

3

commit eb873cacff41ff9eb7a905d3929aa82a67b8def6
Author: 2012615 <2536196890@qq.com>
Date: Tue Jan 31 15:34:14 2023 +0800

2

commit 2be5aef4d5dafc94580f2781b7fa61dbfc183e0a
Author: 2012615 <2536196890@qq.com>
Date: Tue Jan 31 15:33:24 2023 +0800

1

commit 3849719f8c4d4e58e489ae30690eaa70a00d979c

Author: 2012615 <2536196890@qq.com>
Date: Mon Jan 30 18:50:18 2023 +0800

2

commit cab9fa1aa6c3c194fee8197e95bc5461bd4f091b
Author: 2012615 <2536196890@qq.com>
Date: Mon Jan 30 18:46:47 2023 +0800

1

commit 05921f957b38afb8c9c5e27312c8f2016cd68ff3
Author: 2012615 <2536196890@qq.com>
Date: Mon Jan 30 18:41:46 2023 +0800

1

commit 8051de209e2ba46f19f82abd7263681948c8524e
Author: 2012615 <2536196890@qq.com>
Date: Mon Jan 30 18:38:26 2023 +0800

1

commit 45a2b1ed357459685657ce55ad3548d5c8886a92
Author: 王天鹏 <2664004187@qq.com>
Date: Mon Jan 30 14:59:37 2023 +0800

王天鹏的第一次提交，配置yii2框架完成。

commit a87ae8dced3ea3833320f777a88a60878ec4964b
Author: G2 OptJiang20 <2499184084@qq.com>
Date: Mon Jan 30 14:39:37 2023 +0800

init yii2 advanced

commit e58123fd3ee24486cb6e25c39fe2a10325f6fab6
Author: G2 OptJiang20 <2499184084@qq.com>
Date: Sat Jan 28 21:02:41 2023 +0800

init workspace

commit a091550367bfd0b420404ca3b22d46d3769bcd66 (origin/main)
Author: JiangSio <102448598+JiangSio@users.noreply.github.com>
Date: Sat Jan 28 20:50:37 2023 +0800

Initial commit

实现文档@王天鹏

实现文档

——王天鹏 2013013

新闻列表页面

页面展示



俄乌战争最新局势报道



苏纳克宣称英国将支持乌克兰“确保在今年取得胜利”

由记者杜智 于2023年2月8日20:21报道。

【环球网报道】据法新社8日最新消息，英国首相苏纳克表示……

READ MORE >



德国总理强调对待俄乌冲突的三项原则

由记者甄佳闻 于2023年2月8日20:30报道。

当地时间7日，德国联邦总理朔尔茨在联邦议院接受访问……

READ MORE >



乌克兰总统泽连斯基在英国议会发表演讲

实时热搜榜

 1.德国总理强调对待俄乌冲突的三项原则

阅读人数：2546人

 2.波兰总理声称：打败俄罗斯是波兰存在的意义

阅读人数：2341人

 3.荷兰首相谈“对乌提供战机”：西方绝不能

在战机问题上冒险与俄罗斯对抗

阅读人数：1687人

 4.俄罗斯独有的BMPT战车，是“终结者”还是“鸡肋”？



泽连斯基抵达欧盟总部布鲁塞尔，冯德莱恩和米歇尔发推晒图欢迎

由记者李福宏 于2023年2月9日18:21报道。

【环球网快讯】欧盟委员会主席冯德莱恩和欧洲理事会主席米歇尔当地时间9日.....

[READ MORE >](#)



美传奇记者曝光北溪管道爆炸过程 俄媒：希望他不会突然死亡

由记者张鑫 于2023年2月9日18:39报道。

俄罗斯时政专栏作家萨夫丘克2月9日在俄新社发表了一篇题为《北溪管道被恐怖主义国家炸毁》的文章.....

[READ MORE >](#)



泽连斯基突访英国，五大细节意味深长

由记者李力 于2023年2月9日18:54报道。

2月8日，就在全世界舆论都在关注土耳其（包括地震和袭击叙利亚）时，CNN突然爆出一条消息，泽连斯基访问英国.....

[READ MORE >](#)

功能实现

- 1.展示图片、新闻以及摘要
- 2.更新新闻的报道日期、时间、报道人
- 3.通过 read more 实现新闻详情页跳转
- 4.实现分页展示功能

代码展示

涉及表 author、news。

视图层

```
<section class="top-image" style="background:url('css/images/1.jpg') no-repeat top center;">
  <h1>俄乌战争最新局势报道</h1>
</section>

<section id="bottom">
  <section class="shell">
```



```
<section id="content" class="news-feed two_third">

    <div class="one_fourth">
        
        </div>

        <div class="three_fourth last">
            <h4 class="title"><a href="./index.php?r=article/index0#">
                <?php echo $title[0]['title']; ?>
            </a></h4>

            <div class="post-meta">
                由记者<strong><?php echo
$authorName[0]['authorName']; ?></a></strong> 于<strong><?php echo
$doneDate[0]['doneDate']; ?></strong><strong><?php echo
$doneTime[0]['doneTime']; ?></strong>报道。
            </div>

            <p><?php echo $abstract[0]['abstract']; ?></p>

            <p><a href="./index.php?r=article/index0#"
class="continue">Read More</a></p>
        </div>

    <div class="cl"></div>

    <div class="one_fourth">
        
        </div>

        <div class="three_fourth last">
            <h4 class="title"><a href="./index.php?r=article/index1#">
                <?php echo $title[1]['title']; ?>
            </a></h4>

            <div class="post-meta">
                由记者<strong><?php echo
$authorName[1]['authorName']; ?></a></strong> 于<strong><?php echo
$doneDate[1]['doneDate']; ?></strong><strong><?php echo
```

```
$doneTime[1]['doneTime']; ?></strong>报道。
    </div>

    <p><?php echo $abstract[1]['abstract']; ?></p>

    <p><a href="/index.php?r=article/index1#"
class="continue">Read More</a></p>
</div>

<div class="cl"></div>

<div class="one_fourth">
    
</div>

<div class="three_fourth last">
    <h4 class="title"><a href="/index.php?r=article/index2#">
        <?php echo $title[2]['title']; ?>
    </a></h4>

    <div class="post-meta">
        由记者<strong><?php echo
$authorName[2]['authorName']; ?></a></strong> 于<strong><?php echo
$doneDate[2]['doneDate']; ?></strong><strong><?php echo
$doneTime[2]['doneTime']; ?></strong>报道。
    </div>

    <p><?php echo $abstract[2]['abstract']; ?></p>

    <p><a href="/index.php?r=article/index2#"
class="continue">Read More</a></p>
</div>

<div class="cl"></div>

<div class="one_fourth">
    
</div>

<div class="three_fourth last">
```

```

        <h4 class="title"><a href="./index.php?r=article/index3#">
            <?php echo $title[3]['title']; ?>
        </a></h4>

        <div class="post-meta">
            由记者<strong><?php echo
$authorName[3]['authorName']; ?></a></strong> 于<strong><?php echo
$doneDate[3]['doneDate']; ?></strong><strong><?php echo
$doneTime[3]['doneTime']; ?></strong>报道。
        </div>

        <p><?php echo $abstract[3]['abstract']; ?></p>

        <p><a href="./index.php?r=article/index3#"
class="continue">Read More</a></p>
    </div>

    <div class="cl"></div>

    <div class="one_fourth">
        
    </div>

    <div class="three_fourth last">
        <h4 class="title"><a href="./index.php?r=article/index4#">
            <?php echo $title[4]['title']; ?>
        </a></h4>

        <div class="post-meta">
            由记者<strong><?php echo
$authorName[4]['authorName']; ?></a></strong> 于<strong><?php echo
$doneDate[4]['doneDate']; ?></strong><strong><?php echo
$doneTime[4]['doneTime']; ?></strong>报道。
        </div>

        <p><?php echo $abstract[4]['abstract']; ?></p>

        <p><a href="./index.php?r=article/index4#"
class="continue">Read More</a></p>
    </div>

    <div class="cl"></div>

```

```

        <div class="one_fourth">
            
        </div>

        <div class="three_fourth last">
            <h4 class="title"><a href="/index.php?r=article/index5#">
                <?php echo $title[5]['title']; ?>
            </a></h4>

            <div class="post-meta">
                由记者<strong><?php echo
$authorName[5]['authorName']; ?></a></strong> 于<strong><?php echo
$doneDate[5]['doneDate']; ?></strong><strong><?php echo
$doneTime[5]['doneTime']; ?></strong>报道。
            </div>

            <p><?php echo $abstract[5]['abstract']; ?></p>

            <p><a href="/index.php?r=article/index5#"
class="continue">Read More</a></p>
        </div>

        <div class="cl"></div>

        <div id="pagination">
            <span class="current">1</span>
            <a href="/index.php?r=news/index1#">2</a>
            <a href="/index.php?r=news/index2#">3</a>
        </div>

    </section>

```

模型层

```

<?php
namespace app\models;

use yii\db\ActiveRecord;

```

```
class news extends ActiveRecord
{
}
```

控制器层

```
public function actionIndex()
{
    $sql1="select title from news limit 0,6";
    $r1=news::findBysql($sql1)->all();

    $sql2="select abstract from news limit 0,6";
    $r2=news::findBysql($sql2)->all();

    $sql3="select authorName from author limit 0,6";
    $r3=author::findBysql($sql3)->all();

    $sql4="select doneDate from author limit 0,6";
    $r4=author::findBysql($sql4)->all();

    $sql5="select doneTime from author limit 0,6";
    $r5=author::findBysql($sql5)->all();

    $sql6="select title from reader order by number desc";
    $r6=reader::findBysql($sql6)->all();

    $sql7="select number from reader order by number desc";
    $r7=reader::findBysql($sql7)->all();

    $data=[ 'title'=>$r1,
            'abstract'=>$r2,
            'authorName'=>$r3,
            'doneDate'=>$r4,
            'doneTime'=>$r5,
            'bestArticle'=>$r6,
            'readNumber'=>$r7,
            ];

    return $this->renderPartial('index',$data);
}
```

热搜榜界面

页面展示

实时热搜榜

- 

1.德国总理强调对待俄乌冲突的三项原则

阅读人数：2546人
- 

2.波兰总理声称：打败俄罗斯是波兰存在的意义

阅读人数：2341人
- 

3.荷兰首相谈“对乌提供战机”：西方绝不能
能在战机问题上冒险与俄罗斯对抗

阅读人数：1687人
- 

4.俄罗斯独有的BMPT战车，是“终结者”还是“鸡肋”？

阅读人数：1476人
- 

5.诞生于车臣的巷战神器 俄军“终结者”战车的列装之路充满坎坷

阅读人数：1354人

功能实现

- 1.实现新闻热度实时更新
- 2.展示新闻阅读人数
- 3.根据热度排序并连接新闻详情页面

代码展示

涉及表 reader

视图层

```
<aside class="one_third last">
<ul class="widgets">
    <li class="upcoming-widget">
        <h4>实时热搜榜</h4>

        <section class="container">
            <ul class="slides">
                <li><a href="./index.php?r=article/index1#"><span
class="icon"></span>
<strong>1.<?php echo $bestArticle[0]['title'] ?></strong></a></li>
                <span class="right">阅读人数: <?php echo
$readNumber[0]['number'] ?>人</span><br>
                <li><a href="./index.php?r=article/index7#"><span
class="icon"></span>
<strong>2.<?php echo $bestArticle[1]['title'] ?></strong></a></li>
                <span class="right">阅读人数: <?php echo
$readNumber[1]['number'] ?>人</span><br>
                <li><a
href="./index.php?r=article/index12#"><span class="icon"></span> <strong>3.<?php echo
$bestArticle[2]['title'] ?></strong></a></li>
                <span class="right">阅读人数: <?php echo
$readNumber[2]['number'] ?>人</span><br>
                <li><a
href="./index.php?r=article/index15#"><span class="icon"></span> <strong>4.<?php echo
$bestArticle[3]['title'] ?></strong></a></li>
                <span class="right">阅读人数: <?php echo
$readNumber[3]['number'] ?>人</span><br>
                <li><a
href="./index.php?r=article/index17#"><span class="icon"></span> <strong>5.<?php echo
$bestArticle[4]['title'] ?></strong></a></li>
```

```

                <span class="right">阅读人数: <?php echo
$readNumber[4]['number']?>人</span><br>

                </ul>
            </section>
        </li>

    </ul>
</aside>

```

模型层

```

<?php
namespace app\models;

use yii\db\ActiveRecord;

class reader extends ActiveRecord
{
}

```

控制器层

```

$sql6="select title from reader order by number desc";
$r6=reader::findBySql($sql6)->all();

$sql7="select number from reader order by number desc";
$r7=reader::findBySql($sql7)->all();

$data=[ 'title'=>$r1,
        'abstract'=>$r2,
        'authorName'=>$r3,
        'doneDate'=>$r4,
        'doneTime'=>$r5,
        'bestArticle'=>$r6,
        'readNumber'=>$r7,
    ];

return $this->renderPartial('index',$data);

```


新闻详情界面

页面展示



波兰总理声称：打败俄罗斯是波兰存在的意义

波兰总理莫拉维茨基2月8日在接受意大利《晚报》采访时表示，“如果所有国家都像波兰一样，以同样的速度和决心对俄罗斯做出反应，那么我们将更接近胜利。”《晚报》记者问：德国终于批准向乌克兰供应其豹2坦克，并授权其他国家也可以这样故。在美国和英国之后，柏林是向基辅提供军援最多的国家。然而您说在欧洲，特别是在中欧和东欧，由于德国对乌克兰的政策，人们对德国的不信任度越来越高。这是为什么？莫拉维茨基表示，“迟到总比没有好。我感谢德国对乌克兰的帮助，但我们也需要保持正确的比例。柏林所宣布的援助比实际提供的要多。我们不会仅仅通过声明来捍卫乌克兰。即使德国向乌克兰提供它所声称的数量，那也只是沧海一粟。我们要对最强大的欧洲国家抱有更多期望。”《晚报》记者问：有分析认为，波兰想借俄乌冲突之机，将欧洲的重心东移，打造新的地缘政治平衡，由华沙掌舵，取代传统的法德领导欧盟。是这样吗？莫拉维茨基表示，“我们认为俄乌冲突首先是对波兰和整个欧洲的生存威胁。如果俄罗斯赢了，所有地缘政治分析都可以抛开。波兰没有选择它在地图上的位置，但完全理解这个位置所带来的责任。打败俄罗斯既是波兰存在的意义，也是欧洲国家的意义。”

功能实现

涉及表 news

- 1.实现新闻详情的报道
- 2.实现多篇新闻间的相互跳转

代码展示

视图层

```

    <section class="top-image"
style="background:url('css/images/1.jpg') no-repeat top center;">
    <h1>俄乌战争最新局势报道</h1>
</section>

<section id="bottom">
    <section class="shell">

        <section id="content" class="two_third">

            <h1> <?php echo $title[0]['title']; ?></h1>
            <p> <?php echo $text[0]['text']; ?></p>

            <div class="comments">

                <h4>评论</h4>

```

模型层

```

<?php
namespace app\models;

use yii\db\ActiveRecord;

class news extends ActiveRecord
{
}

```

控制器层

```

public function actionIndex0()
{

    $sql1="select title from news where id=0";
    $r1=news::findBysql($sql1)->all();

    $sql2="select text from news where id=0";
    $r2=news::findBysql($sql2)->all();

    $sql3="select name from review order by rand() limit 4";
    $r3=review::findBysql($sql3)->all();

    $sql4="select content from review order by rand() limit 4";
    $r4=review::findBysql($sql4)->all();

    $sql5="select time from review order by rand() limit 4";
    $r5=review::findBysql($sql5)->all();
}

```

```

$sql6="select title from reader order by number desc";
$r6=reader::findBysql($sql6)->all();

$sql7="select number from reader order by number desc";
$r7=reader::findBysql($sql7)->all();

$data=[ 'title'=>$r1,
        'text'=>$r2,
        'name'=>$r3,
        'content'=>$r4,
        'time'=>$r5,
        'bestArticle'=>$r6,
        'readNumber'=>$r7,
];
return $this->renderPartial('index0',$data);
}

```

评论界面

页面展示

评论



来自张跃进:
打死俄罗斯

发表于2023年2月10日



来自范德强的评论:
开始有正义的记者曝光美国和西方的虚伪了

发表于2023年2月9日



来自王杰的评论:
为了国家也算是鞠躬尽瘁了，当然，有人希望他是“汪精卫”式的人物也不奇怪

5. 诞生于车臣的越战神鹰 俄军“终结者”战车的列装之路充满坎坷

阅读人数: 1354人

功能实现

- 1.实现用户信息的展示
- 2.实现用户评论内容的展示
- 3.根据评论时间排序展示评论

代码展示

视图层

```
<h4>评论</h4>

<div class="one_fifth">
    
</div>

<div class="four_fifth last">

    <p><strong>来自<?php echo $name[0]['name']; ?>的评论:
</strong><br />
        <?php echo $content[0]['content']; ?>
    </p>

    <h6>发表于<?php echo $time[0]['time']; ?></h6>

</div>

<div class="c1"></div>

<div class="one_fifth">
    
</div>

<div class="four_fifth last">

    <p><strong>来自<?php echo $name[1]['name']; ?>的评论:
</strong><br />
        <?php echo $content[1]['content']; ?>
    </p>

    <h6>发表于<?php echo $time[1]['time']; ?></h6>

</div>
```

```

<div class="c1"></div>

<div class="one_fifth">
    
</div>

<div class="four_fifth last">

    <p><strong>来自<?php echo $name[2]['name']; ?>的评论:
</strong><br />
        <?php echo $content[2]['content']; ?>
    </p>

    <h6>发表于<?php echo $time[2]['time']; ?></h6>

</div>

<div class="c1"></div>

<div class="one_fifth">
    
</div>

<div class="four_fifth last">

    <p><strong>来自<?php echo $name[3]['name']; ?>的评论:
</strong><br />
        <?php echo $content[3]['content']; ?>
    </p>

    <h6>发表于<?php echo $time[3]['time']; ?></h6>

</div>

<div class="c1"></div>

```

模型层

```

<?php
namespace app\models;

use yii\db\ActiveRecord;

class review extends ActiveRecord

```

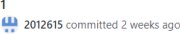


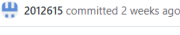


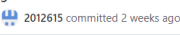


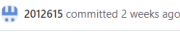


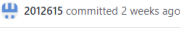


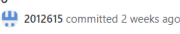


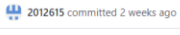


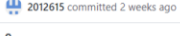


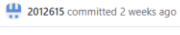


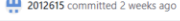


```
{  
}
```

控制器层

```
$sql3="select name from review order by rand() limit 4";  
$r3=review::findBysql($sql3)->all();  
  
$sql4="select content from review order by rand() limit 4";  
$r4=review::findBysql($sql4)->all();  
  
$data=[ 'title'=>$r1,  
        'text'=>$r2,  
        'name'=>$r3,  
        'content'=>$r4,  
        'time'=>$r5,  
        'bestArticle'=>$r6,  
        'readNumber'=>$r7,  
];  
return $this->renderPartial('index0',$data);
```

实现文档@张继盛

Github 提交记录

Commits on Jan 31, 2023		
1		 2be5aef 
2		 eb873ca 
3		 bc35fab 
4		 485e6f7 
5		 fb3f0b1 
6		 a8900c2 
-		
7		 d62cb97 
8		 56b1ab8 
9		 9297b37 
yii2		 2c371df 

Commits on Feb 8, 2023		
yii2.0 advanced 1.0	615f977	<>
2012615 committed 4 days ago		
Merge branch 'master' of https://github.com/jiangSio/Internet-Databas...	be3dd7f	<>
2012615 committed 4 days ago		
1	82de1ff	<>
2012615 committed 4 days ago		
2	c560aa4	<>
2012615 committed 4 days ago		
yii2022 1.0	18c4feb	<>
2012615 committed 4 days ago		

项目细节

俄乌历史矛盾@张继盛主界面



搜索功能

losthost/project/yjs/yjs2022/frontend/index.php?post%3Fdetail&id=338&title=俄罗斯与乌克兰为什么总是有矛盾？

俄乌历史矛盾关于我们联系我们

迷失的脚步

首页文章列表俄罗斯与乌克兰为什么总是有矛盾？(二)

俄罗斯与乌克兰为什么总是有矛盾？(二)1970-08-23 04:30:08中国网友

俄乌局势演变，乌克兰问题或将长期化和复杂化

随着乌克兰紧张局势的急剧升级，俄罗斯总统普京21日签署总统令，承认顿涅茨克和卢甘斯克独立，并签署了与顿涅茨克和卢甘斯克的友好合作互助条约。24日，普京发表紧急电视讲话，决定在顿巴斯地区进行特别军事行动，此举标志着俄军开进顿巴斯地区开展军事部署，是对乌克兰内战决裂强力支持。对俄罗斯来说，由于此前已经承认顿涅茨克和卢甘斯克的独立，俄罗斯应请求出兵“合理合法”，对乌克兰发声，俄罗斯如果在顿巴斯地区开展军事行动，将不得不使乌克兰的主权和领土完整。因此，普京的决定对于乌克兰局势的发展无疑产生了重要影响。随着乌克兰国际地位逐渐宣布与俄罗斯断交，乌克兰紧张局势将进一步恶化。

普京的决策表明以和平方式缓和乌克兰紧张局势并按明斯克协议解决乌克兰问题的道路已行不通。因此，俄罗斯再次决定了对顿巴斯，在承认顿涅茨克和卢甘斯克独立后并向顿巴斯地区开展军事部署，这是俄乌关系进一步恶化的标志性行为。

俄乌关系演变到现在的局面，从原理上说是上世纪90年代初期地缘政治发生根本改变的结果”。东欧剧变、苏联解体、美国为首的西方在重塑欧洲秩序的过程中，并未考虑到俄罗斯的安全诉求，不断推行北约东扩的“双管齐下”，压缩俄罗斯的战略空间。三十年代，乌克兰与北约在战略层面处于对立的高层。

2013年乌克兰发生“颜色革命”后，乌克兰采取了亲西方的外交政策，由此引发了俄罗斯的战略焦虑。2014年，随着俄罗斯“收复”克里米亚，俄乌的紧张关系加剧。由于乌克兰东部地区俄罗斯族人众多，顿涅茨克和卢甘斯克这两个共和国实际上处于被占领的状态。这使俄罗斯有了借乌克兰问题与西方讨价还价的资本”。2015年2月，乌克兰冲突双方达成《明斯克协议》，大规模武装冲突得到控制，但小范围的冲突时有发生。

2017年4月，乌克兰总统泽连斯基呼吁修改《明斯克协议》，加强美国在结束乌克兰冲突中的作用，引起了俄罗斯的强烈关注。进入2022年，西方媒体声称俄罗斯在俄乌边境地区集结兵力，有入侵之嫌，俄罗斯则称乌克兰违反协议。在此背景下，乌克兰局势依然紧张。

随着普京宣布展开特别军事行动，乌克兰宣布全面进入战时状态，普京则表示没有“侵略”乌克兰的计划。乌克兰局势引起国际社会的高度关注，其发展前景充满不确定性。

一是俄罗斯与乌克兰是否会发生全面战争？从各方的表态看，俄罗斯与乌克兰发生全面战争的主动权在乌克兰而非俄罗斯。首先，俄罗斯的目标有限，即通过采取军事行动打击由乌克兰在顿巴斯的军事力量，维持顿涅茨克和卢甘斯克的独立，并非全面占领乌克兰。因此，俄罗斯不会对乌克兰宣战。对乌克兰来说，俄罗斯的相关行动是对乌克兰本土主权的侵犯，在宣布与俄罗斯断交之后，是置战事于不顾还是要求美国和其他西方国家提供军事援助，而美国宣布无义务与俄罗斯断交。在没有美国以北约成员出兵干预的前提下，乌克兰难以与俄罗斯进行全面战争。

二是顿涅茨克和卢甘斯克是否会成为下一个克里米亚？由于俄罗斯已经承认顿涅茨克和卢甘斯克独立，如果俄罗斯军队能够快速控制顿巴斯地区，不排除俄罗斯会对它们完全吞并的可能性。一旦俄罗斯完全占领了卢甘斯克和顿涅茨克，乌克兰将面临更大的压力。

查找文章

按标题

搜索

标签云

俄乌冲突原因战争民族顿涅斯克

最新动态

近期战争

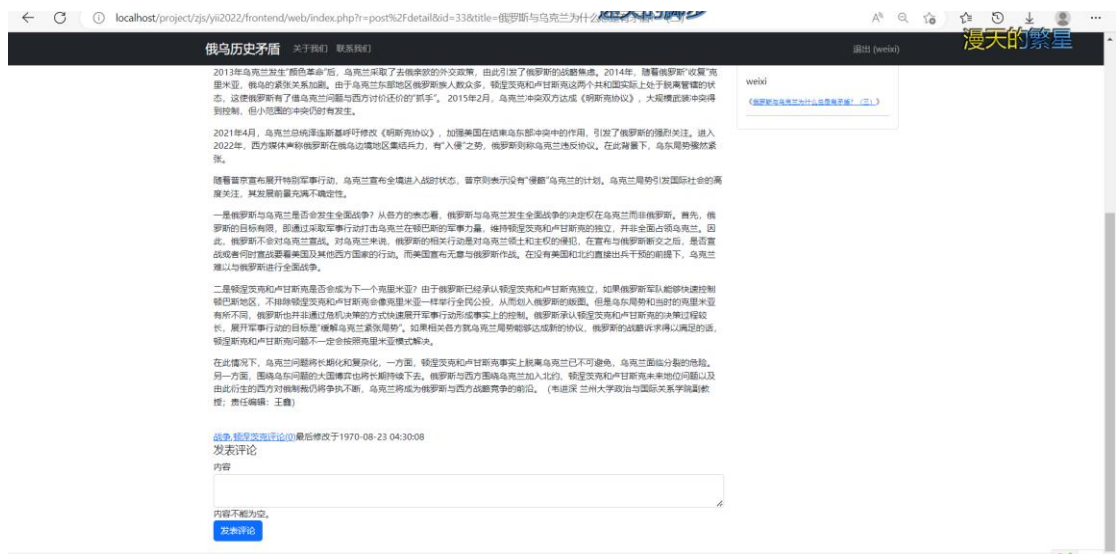
weiki

《乌克兰与俄罗斯的历史恩怨与现状》

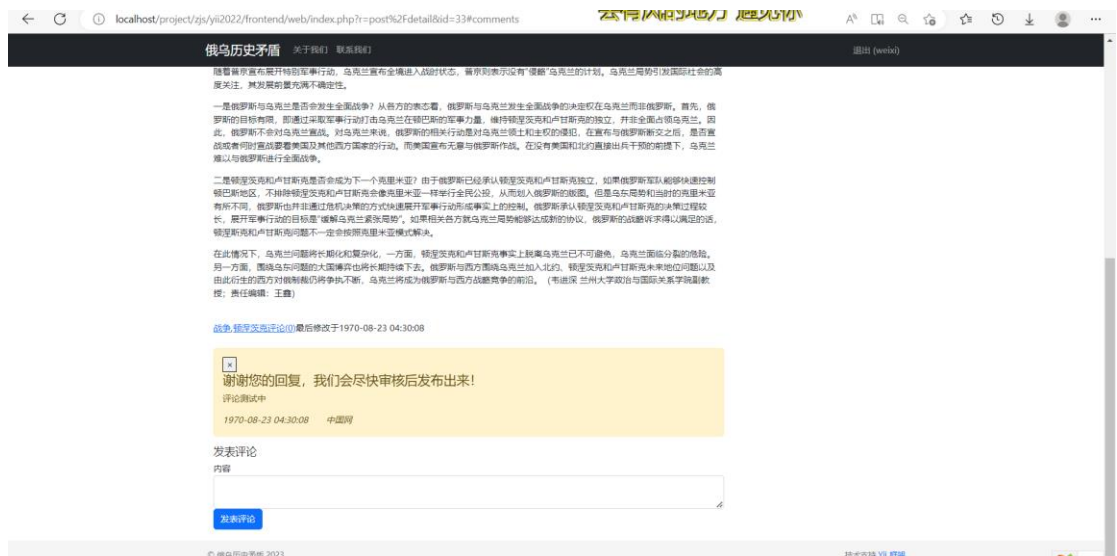
和平来之不易，更应该珍惜

weiki

《俄罗斯与乌克兰为什么总是有矛盾？(一)》



可发表评论，提交后经后台审核通过可展示



代码实现

Index.php

```
<?php

use yii\widgets\ListView;
use yii\data\ActiveDataProvider;
use common\models\Post;
use yii\helpers\Html;
use yii\helpers\Url;
use yii\grid\ActionColumn;
use yii\grid\GridView;
use frontend\components\TagsCloudWidget;
use frontend\components\RctReplyWidget;

/** @var yii\web\View $this */
/** @var common\models\PostSearch $searchModel */
```

```

/** @var yii\data\ActiveDataProvider $dataProvider */

?>

<div class="container">
    <div class="row">
        <div class="col-md-9">
            <ol class="breadcrumb">
                <li><a href="{? = Yii::$app->homeUrl; ?}">首页</a></li>
                <li>文章列表</li>
            </ol>
            <?=ListView::widget([
                'id'=>'postlist',
                'dataProvider'=>$dataProvider,
                'itemView'=>'_listitem',
                'layout'=>'{items} {pager}',
                'pager'=>[
                    'maxButtonCount'=>10,
                    'nextPageLabel'=>Yii::t('app','下一页'),
                    'prevPageLabel'=>Yii::t('app','上一页'),
                ],
            ])?>

        </div>

        <div class="col-md-3">
            <div class="searchbox">
                <ul class="list-group">
                    <li class="list-group-item">
                        <span class="glyphicon glyphicon-search" aria-
hidden="true"></span>查找文章
                    </li>
                    <li class="list-group-item">
                        <form class="form-inline" action="index.php?r=post/index"
id="w0" method="get">
                            <div class="form-group">
                                <input type="text" class="form-control"
name="PostSearch[title]" id="w0input" placeholder="按标题">
                            </div>
                            <button type="submit" class="btn btn-default">搜索
                        </button>
                        </form>
                    </li>
                </ul>

```

```

        </div>
        <div class="tagcloudbox">
            <ul class="list-group">
                <li class="list-group-item">
                    <span class="glyphicon glyphicon-search" aria-
hidden="true"></span>标签云
                </li>
                <li class="list-group-item">
                    <?= TagsCloudWidget::widget(['tags'=>$tags])?>
                </li>
            </ul>
        </div>
        <div class="commentbox">
            <ul class="list-group">
                <li class="list-group-item">
                    <span class="glyphicon glyphicon-search" aria-
hidden="true"></span>最新回复
                </li>
                <li class="list-group-item">
                    <?=
RctReplyWidget::widget(['recentComments'=>$recentComments])?>
                </li>
            </ul>
        </div>

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

```

Post.php

```

<?php

namespace common\models;

use Yii;
use yii\helpers\Html;
/**
 * This is the model class for table "post".
 *
 * @property int $id
 * @property string $title
 * @property string $content
 * @property string|null $tags

```

```

* @property int $status
* @property int|null $create_time
* @property int|null $update_time
* @property int $author_id
*
* @property Adminuser $author
* @property Comment[] $comments
* @property Poststatus $status0
*/
class Post extends \yii\db\ActiveRecord
{
    private $_oldTags;

    /**
     * {@inheritdoc}
     */

    public static function tableName()
    {
        return 'post';
    }

    /**
     * {@inheritdoc}
     */

    public function rules()
    {
        return [
            [['title', 'content', 'status', 'author_id'], 'required'],
            [['content', 'tags'], 'string'],
            [['status', 'create_time', 'update_time', 'author_id'], 'integer'],
            [['title'], 'string', 'max' => 128],
            [['author_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
Adminuser::class, 'targetAttribute' => ['author_id' => 'id']],
            [['status'], 'exist', 'skipOnError' => true, 'targetClass' =>
Poststatus::class, 'targetAttribute' => ['status' => 'id']],
        ];
    }

    /**
     * {@inheritdoc}
     */

    public function attributeLabels()
    {
        return [
            'id' => 'ID',

```

```

        'title' => '标题',
        'content' => '内容',
        'tags' => '标签',
        'status' => '状态',
        'create_time' => '创建时间',
        'update_time' => '修改时间',
        'author_id' => '作者',
    ];
}

/**
 * Gets query for [[Author]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getAuthor()
{
    return $this->hasOne(Adminuser::class, ['id' => 'author_id']);
}

/**
 * Gets query for [[Comments]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getComments()
{
    return $this->hasMany(Comment::class, ['post_id' => 'id']);
}

public function getActiveComments()
{
    return $this->hasMany(Comment::class, ['post_id' =>
'id'])->where('status=:status',[':status'=>2])->orderBy('id DESC');
}

/**
 * Gets query for [[Status0]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getStatus0()
{
    return $this->hasOne(Poststatus::class, ['id' => 'status']);
}

```

```

public function beforeSave($insert)
{
    if(parent::beforeSave($insert))
    {
        if($insert)
        {
            $this->create_time=time();
            $this->update_time=time();
        }
        else
        {
            $this->update_time=time();
        }
        return true;
    }
    else
    {
        return false;
    }
}

public function afterFind()
{
    parent::afterFind();
    $this->_oldTags = $this->tags;
}

public function afterSave($insert, $changedAttributes)
{
    parent::afterSave($insert,$changedAttributes);
    Tag::updateFrequency($this->_oldTags,$this->tags);
}

public function afterDelete()
{
    parent::afterDelete();
    Tag::updateFrequency($this->tags,'');
}

public function getUrl()
{
    return Yii::$app->urlManager->createUrl(
        ['post/detail','id'=>$this->id,'title'=>$this->title]);
}

public function getBeginning($length=288)
{
    $tmpStr=strip_tags($this->content);

```

```

        $tmpLen=mb_strlen($tmpStr);

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

    public function getTagLinks()
    {
        $links=array();
        foreach(Tag::string2array($this->tags) as $tag)
        {
            $links[]=Html::a(Html::encode($tag),array('post/index','PostSearch[tags]
'=>$tag));
        }
        return $links;
    }

    public function getCommentCount()
    {
        return Comment::find()->where(['post_id'=>$this->id,'status'=>2])->count();
    }
}

```

Tag.php

```

<?php

namespace common\models;

use Yii;
use yii\base\Model;

/**
 * This is the model class for table "tag".
 *
 * @property int $id
 * @property string $name
 * @property int|null $frequency
 */
class Tag extends \yii\db\ActiveRecord
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'tag';
    }
}

```

```

}

/**
 * {@inheritdoc}
 */
public function rules()
{
    return [
        [['name'], 'required'],
        [['frequency'], 'integer'],
        [['name'], 'string', 'max' => 128],
    ];
}

/**
 * {@inheritdoc}
 */
public function attributeLabels()
{
    return [
        'id' => 'ID',
        'name' => 'Name',
        'frequency' => 'Frequency',
    ];
}

public static function string2array($tags)
{
    return preg_split('/\s*,\s*/', trim($tags), -1, PREG_SPLIT_NO_EMPTY);
}

public static function array2string($tags)
{
    return implode(',', $tags);
}

public static function addTags($tags)
{
    if(empty($tags)) return ;
    foreach ($tags as $name)
    {
        $aTag = Tag::find()->where(['name'=>$name])->one();
        $aTagCount=Tag::find()->where(['name'=>$name])->count();

        if(!$aTagCount)
        {
            $tag = new Tag;

```



```

        $tag->name= $name;
        $tag->frequency=1;
        $tag->save();
    }
    else
    {
        $aTag->frequency += 1;
        $aTag->save();
    }
}

}

public static function removeTags($tags)
{
    if(empty($tags))return ;

    foreach($tags as $name)
    {
        $aTag = Tag::find()->where(['name'=>$name])->one();
        $aTagCount=Tag::find()->where(['name'=>$name])->count();
        if($aTagCount)
        {
            if($aTagCount && $aTag->frequency<=1)
            {
                $aTag->delete();
            }
            else
            {
                $aTag->frequency -= 1;
                $aTag->save();
            }
        }
    }
}

}

public static function updateFrequency($oldTags,$newTags)
{
    if(!empty($oldTags)||!empty($newTags))
    {
        $oldTagsArray= self::string2array($oldTags);
        $newTagsArray= self::string2array($newTags);

        self::addTags(array_values(array_diff($newTagsArray,$oldTagsArray)));
        self::removeTags(array_values(array_diff($oldTagsArray,$newTagsArray)));
    }
}

```

```

    }
}
public static function findTagWeights($limit=20)
{
    $tag_size_level = 5;

    $models=Tag::find()->orderBy('frequency desc')->limit($limit)->all();
    $total=Tag::find()->limit($limit)->count();

    $stepper=ceil($total/$tag_size_level);

    $tags=array();
    $counter=1;

    if($total>0)
    {
        foreach ($models as $model)
        {
            $weight=ceil($counter/$stepper)+1;
            $tags[$model->name]=$weight;
            $counter++;
        }
        ksort($tags);
    }
    return $tags;
}
}

```

PostController.php

```

<?php

namespace frontend\controllers;

use Yii;
use common\models\Post;
use common\models\PostSearch;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;
use yii\filters\AccessControl;

```

```

use common\models\Tag;
use common\models\Comment;
use common\models\User;
use yii\rest\Serializer;

/**
 * PostController implements the CRUD actions for Post model.
 */
class PostController extends Controller
{
    public $added=0; //0 代表还没有新回复
    /**
     * @inheritdoc
     */
    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['POST'],
                ],
            ],

            'access' =>[
                'class' => AccessControl::className(),
                'rules' =>
                [
                    [
                        'actions' => ['index'],
                        'allow' => true,
                        'roles' => ['?'],
                    ],
                    [
                        'actions' => ['index', 'detail'],
                        'allow' => true,
                        'roles' => ['@'],
                    ],
                ],
            ],

            'pageCache'=>[
                'class'=>'yii\filters\PageCache',
            ],
        ];
    }
}

```

```

        'only'=>['index'],
        'duration'=>600,
        'variations'=>[
            Yii::$app->request->get('page'),
            Yii::$app->request->get('PostSearch'),
        ],
        'dependency'=>[
            'class'=>'yii\caching\DbDependency',
            'sql'=>'select count(id) from post',
        ],
    ],

    'httpCache'=>[
        'class'=>'yii\filters\HttpCache',
        'only'=>['detail'],
        'lastModified'=>function ($action,$params){
            $q = new \yii\db\Query();
            return $q->from('post')->max('update_time');
        },
        'etagSeed'=>function ($action,$params) {
            $post = $this->findModel(Yii::$app->request->get('id'));
            return serialize([$post->title,$post->content]);
        },

        'cacheControlHeader' => 'public,max-age=600',

    ],

];

}

/**
 * Lists all Post models.
 * @return mixed
 */
public function actionIndex()
{
    $tags=Tag::findTagWeights();
    $recentComments=Comment::findRecentComments();

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

```

```

        return $this->render('index', [
            'searchModel' => $searchModel,
            'dataProvider' => $dataProvider,
            'tags'=>$tags,
            'recentComments'=>$recentComments,
        ]);
    }

    /**
     * Displays a single Post model.
     * @param integer $id
     * @return mixed
     */
    public function actionView($id)
    {
        return $this->render('view', [
            'model' => $this->findModel($id),
        ]);
    }

    /**
     * Creates a new Post model.
     * If creation is successful, the browser will be redirected to the 'view' page.
     * @return mixed
     */
    public function actionCreate()
    {
        $model = new Post();

        if ($model->load(Yii::$app->request->post()) && $model->save()) {
            return $this->redirect(['view', 'id' => $model->id]);
        } else {
            return $this->render('create', [
                'model' => $model,
            ]);
        }
    }

    /**
     * Updates an existing Post model.
     * If update is successful, the browser will be redirected to the 'view' page.
     * @param integer $id
     * @return mixed
     */

```

```

public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($model->load(Yii::$app->request->post()) && $model->save()) {
        return $this->redirect(['view', 'id' => $model->id]);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}

/**
 * Deletes an existing Post model.
 * If deletion is successful, the browser will be redirected to the 'index'
page.
 * @param integer $id
 * @return mixed
 */
public function actionDelete($id)
{
    $this->findModel($id)->delete();

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

/**
 * Finds the Post model based on its primary key value.
 * If the model is not found, a 404 HTTP exception will be thrown.
 * @param integer $id
 * @return Post the loaded model
 * @throws NotFoundHttpException if the model cannot be found
 */
protected function findModel($id)
{
    if (($model = Post::findOne($id)) !== null) {
        return $model;
    } else {
        throw new NotFoundHttpException('The requested page does not exist.');
    }
}

public function actionDetail($id)

```

```

{
    //step1. 准备数据模型
    $model = $this->findModel($id);
    $tags=Tag::findTagWeights();
    $recentComments=Comment::findRecentComments();

    $userMe = User::findOne(Yii::$app->user->id);
    $commentModel = new Comment();
    $commentModel->email = $userMe->email;
    $commentModel->userid = $userMe->id;

    //step2. 当评论提交时，处理评论
    if($commentModel->load(Yii::$app->request->post()))
    {
        $commentModel->status = 1; //新评论默认状态为 pending
        $commentModel->post_id = $id;
        if($commentModel->save())
        {
            $this->added=1;
        }
    }

    //step3. 传数据给视图渲染

    return $this->render('detail',[
        'model'=>$model,
        'tags'=>$tags,
        'recentComments'=>$recentComments,
        'commentModel'=>$commentModel,
        'added'=>$this->added,
    ]);
}

```

Detail.php

```
<?php
```



```
</div>

</div>

<br>

<div class="content">
    <?= HTMLPurifier::process($model->content)?>
</div>

<br>

<div class="nav">
    <span class="glyphicon glyphicon-tag" aria-hidden="true"></span>
    <?= implode(' ', $model->tagLinks);?>
    <br>
    <?= Html::a("评论
({$model->commentCount})", $model->url . '#comments');?>
    最后修改于<?= date('Y-m-d H:i:s', $model->update_time);?>
</div>
</div>

<div id="comments">

    <?php if($added) {?>
        <br>
        <div class="alert alert-warning alert-dismissible" role="alert">
            <button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>

            <h4>感谢您的回复，我们会尽快审核后发布出来! </h4>

            <p><?= nl2br($commentModel->content);?></p>
            <span class="glyphicon glyphicon-time" aria-
hidden="true"></span><em><?= date('Y-m-d
H:i:s', $model->create_time). "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&";?></em>
            <span class="glyphicon glyphicon-user" aria-
hidden="true"></span><em><?= Html::encode($model->author->nickname);?></em>
        </div>
    <?php }?>

    <?php if($model->commentCount>=1) :?>

    <h5><?= $model->commentCount.'条评论';?></h5>
```

```

        <?= $this->render('_comment',array(
            'post'=>$model,
            'comments'=>$model->activeComments,
        ));?>
    <?php endif;?>

    <h5>发表评论</h5>
    <?php
    $commentModel =new Comment();
    echo $this->render('_guestform',array(
        'id'=>$model->id,
        'commentModel'=>$commentModel,
    ));?>

</div>

</div>

<div class="col-md-3">
    <div class="searchbox">
        <ul class="list-group">
            <li class="list-group-item">
                <span class="glyphicon glyphicon-search" aria-
hidden="true"></span> 查找文章
            </li>
            <li class="list-group-item">
                <form class="form-inline" action="index.php?r=post/index"
id="w0" method="get">
                    <div class="form-group">
                        <input type="text" class="form-control"
name="PostSearch[title]" id="w0input" placeholder="按标题">
                    </div>
                    <button type="submit" class="btn btn-default">搜索</button>
                </form>
            </li>
        </ul>
    </div>

    <div class="tagcloudbox">
        <ul class="list-group">
            <li class="list-group-item">

```

```

        <span class="glyphicon glyphicon-tags" aria-hidden="true"></span>
    标签云
        </li>
        <li class="list-group-item">
            <?= TagsCloudWidget::widget(['tags'=>$tags])?>
        </li>
    </ul>
</div>

    <div class="commentbox">
        <ul class="list-group">
            <li class="list-group-item">
                <span class="glyphicon glyphicon-comment" aria-
    hidden="true"></span> 最新回复
            </li>
            <li class="list-group-item">
                <?= RctReplyWidget::widget(['recentComments'=>$recentComments])?>
            </li>
        </ul>
    </div>

</div>

</div>
</div>

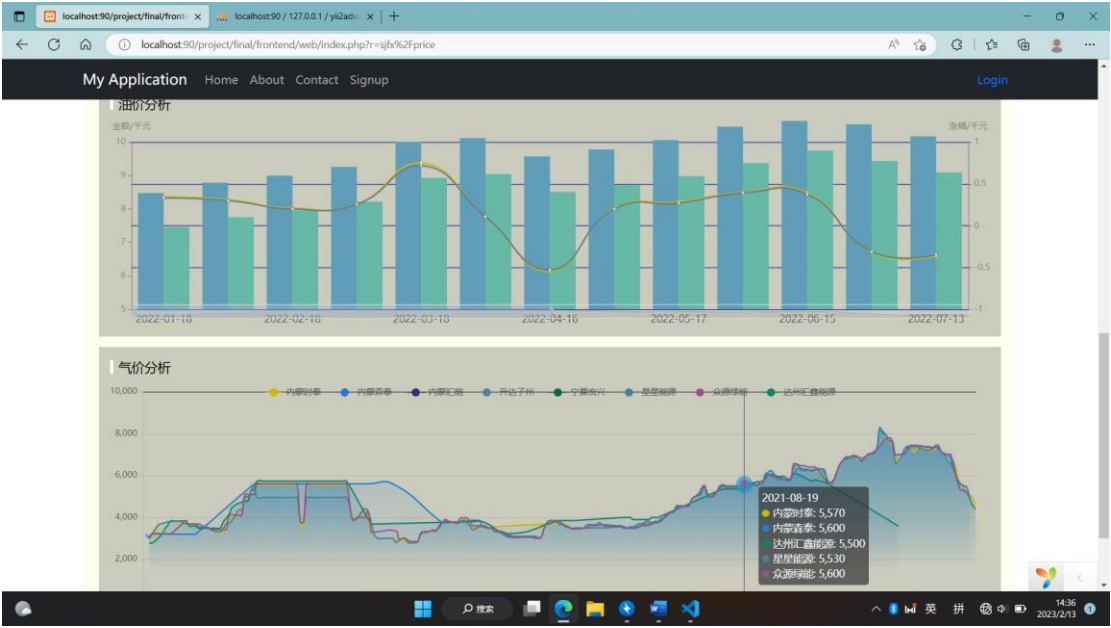
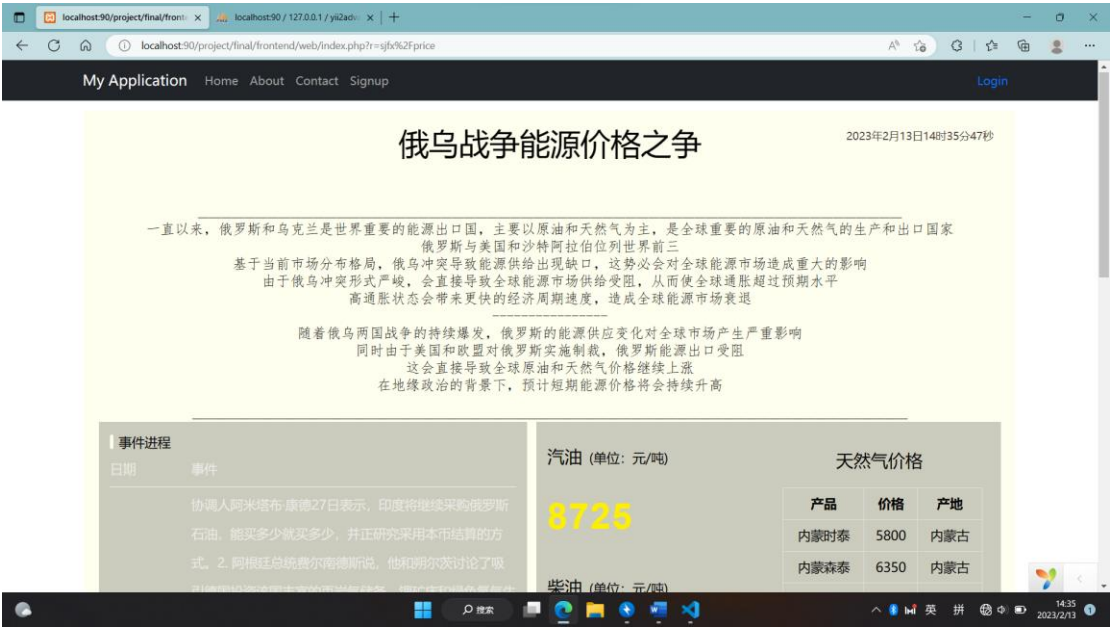
```

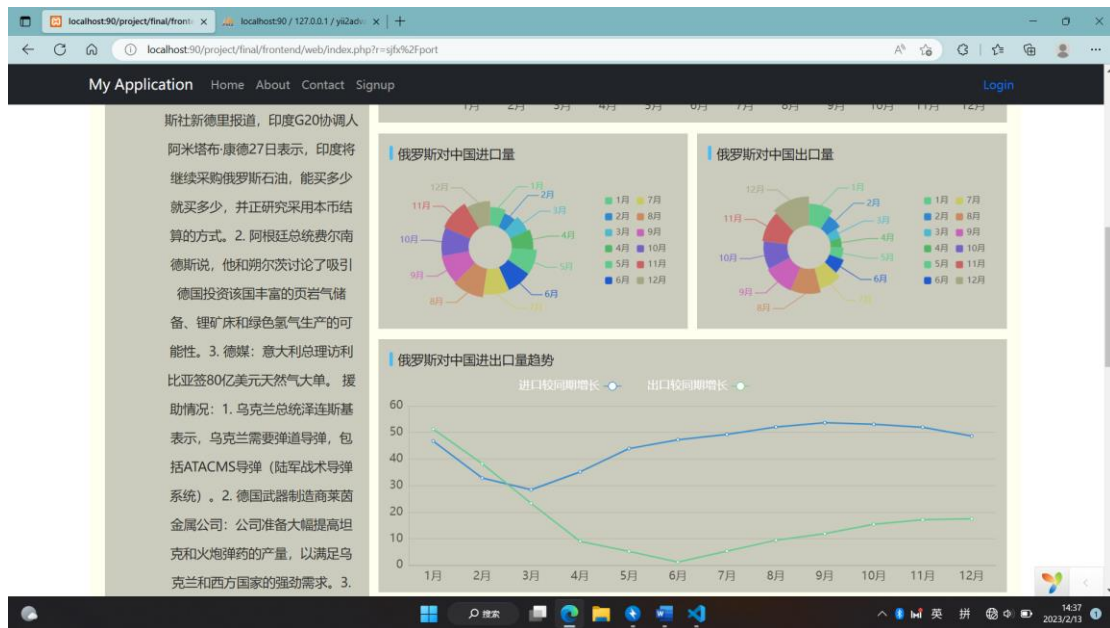
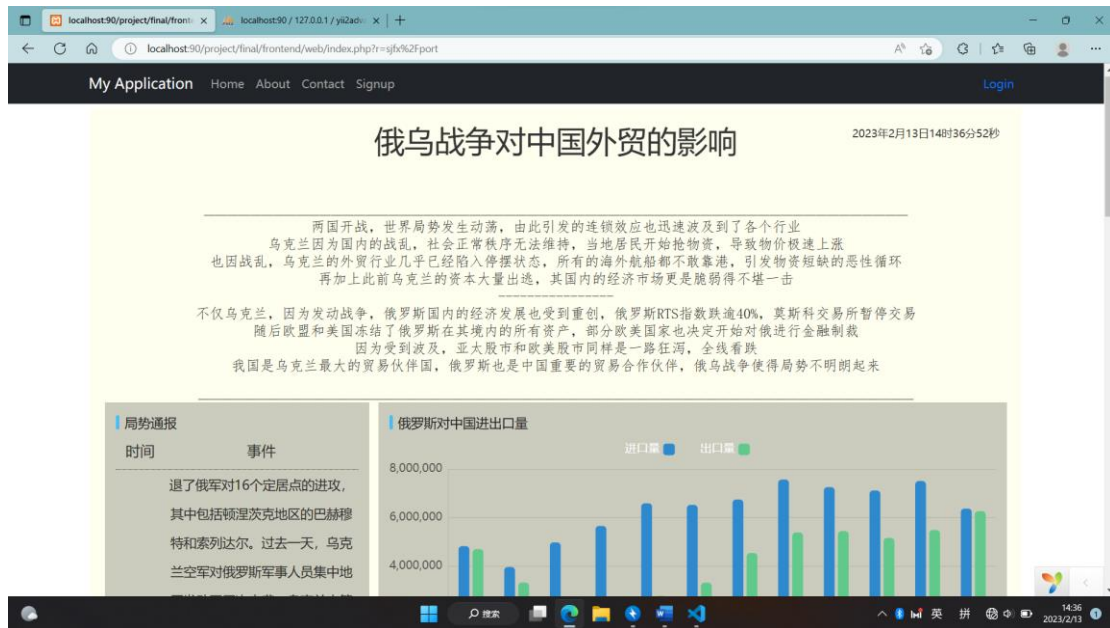
实现文档@姜天嘉

模板设计

前台设计采用的模板为 Mate 前台模板
 后台设计采用的模板为 AdminLTE 后台模板
 数据分析界面采用的的大数据模板

界面展示





功能设计

登录功能

在 yii2 框架特有的登录功能上进行改进, 增加了 admin 项用户判断用户是否为管理员用户, 并且在管理员用户登录状态下显示后台入口

```
<li class="nav-item">
    <?php
    if(!Yii::$app->user->isGuest)
    {
```

```

        $user=User::find()->where(['username'=>Yii::$app->user
->identity->username])->one();
        if($user->admin==1)
        {
            echo '<a class="nav-link page-scroll"
href="/project/jtj/yii2022/backend/web/index.php">Backend</a>';
        }
    }
    ?>
</li>

```

倒计时模块

首页中有计算战争持续时间模块，利用异步模式触发页面变化

```

<script>
    var t = setTimeout(time, 0); /*开始运行*/
    function time(){
        clearTimeout(t); /*清除定时器*/
        var date_start = new Date(2022,1,24).getTime();
        var date_now = new Date().getTime();
        var long = Math.floor((date_now-date_start)/1000);
        var day=Math.floor(long/86400);
        var hour=Math.floor((long%86400)/3600);
        var minuet=Math.floor((long%3600)/60);
        var second=(long%60);
        document.getElementById("day").innerHTML =day;
        document.getElementById("hour").innerHTML =hour;
        document.getElementById("minuet").innerHTML =minuet;
        document.getElementById("second").innerHTML =second;
        t = setTimeout(time, 1000);
    }

</script>

```

数据分析模块

爬取数据

利用爬虫直接从一些网站上爬取数据（官方网站或者一些数据整理的网站）

将数据库的数据读取

```

use frontend\models\SjfxImportexport;
$query=SjfxImportexport::find();
$datas=$query->orderBy('month')->all();

```

将数据映射到网页上

```

data:[
    <?php

```

```

        $str = '';
        foreach($datas as $data){
            if($data->country=="俄罗斯联邦"){
                $str =
$str.'{value:'.$data->in_m.",name:\"".$data->month."月\"},";
            }
        }
        echo $str;
    ?>
]

```

大致来说网页都是用这种方式实现的数据库访问

给过长数据表添加可伸缩横向拉伸条，在表的 option 中添加

```

dataZoom : [
    {
        type: 'slider',
        show: true,
        height:20,
        start: 0,
        end: 50,
        xAxisIndex: [0],
    },
],

```

直接利用 echat 中的库实现

实现数据库的提交

在后台界面有时候需要进行数据库信息的提交，这里采用一种 javascript 直接提交+php 链接解析方式实现

```

<div class="card-body">
<form action="index.php" method="get" id="jtjclick">
    <div class="input-group mb-3">
        <div class="input-group-prepend">
            <span class="input-group-text"><i class="fas fa-envelope"></i></span>
            <span class="input-group-text">@姜天嘉</span>
        </div>
        <input type="text" class="form-control" id="jtj" name="jtj">
        <span class="input-group-append">
            <button type="button" class="btn btn-info btn-flat"
onclick='document.getElementById("jtjclick").submit();'>Go!</button>
        </span>
    </div>
</form>

```

在控制器中实现数据提交至数据库并刷新

```

public function actionIndex()
{
    $request = Yii::$app->request;
    $get1 = $request->get("jtj");
    if($get1!=null){
        $record=Developer::find()->where(['name'=>'姜天嘉']->one();
        $record->text=$get1;
        $record->save();
        $this->redirect(array('site/index'));
    }
    $get2 = $request->get("yzh");
    if($get2!=null){
        $record=Developer::find()->where(['name'=>'闫钊辉']->one();
        $record->text=$get2;
        $record->save();
        $this->redirect(array('site/index'));
    }
    $get3 = $request->get("wtp");
    if($get3!=null){
        $record=Developer::find()->where(['name'=>'王天鹏']->one();
        $record->text=$get3;
        $record->save();
        $this->redirect(array('site/index'));
    }
    $get4 = $request->get("zjs");
    if($get4!=null){
        $record=Developer::find()->where(['name'=>'张继盛']->one();
        $record->text=$get4;
        $record->save();
        $this->redirect(array('site/index'));
    }
    return $this->render('index');
}

```

实现文档@闫钊辉

数据库实现

数据库的实现根据 localhost 进入数据库，数据库就是我们所设计的三个数据库，详见设计文档。

其中的数据来自网页中的自己寻找，数据详实

网页的设计

网页的设计首先要根据自己的选择的模板进行设计，下面是我选择的模板：



根据我自己的设计改良

```
<?php
namespace frontend\controllers;

use yii\web\Controller;
use yii\data\Pagination;

class XwController extends Controller
{
    public function actionIndex()
    {
        return $this->render('view: 'index');
    }
}
```

先实现一个控制器

```

$n1 = News::findOne( condition: '1');
$d1=$n1->news_name;
$b1=$n1->news_web;

$n2 = News::findOne( condition: '2');
$d2=$n2->news_name;
$b2=$n2->news_web;

$n3 = News::findOne( condition: '3');
$d3=$n3->news_name;
$b3=$n3->news_web;

$n4 = News::findOne( condition: '4');
$d4=$n4->news_name;
$b4=$n4->news_web;

```

实现一下寻找数据库里面的数据

之后就是根据网页来进行修改。

