

Russian-Ukraine War

实现文档



团队：NKU-HLPY

队长：韩佳迅（2012682）

成员：林语盈（2012174）

彭钰婷（2013631）

姚鑫（2014089）

目录

Russian-Ukraine War	1
第一部分 分工与记录	3
一、分工	3
二、记录	4
(一) git 提交	4
(二) Tower 分工	5
第二部分 代码实现	5
一、导入 Yii 框架	5
二、导入模板	6
(一) 前台模板的导入	6
(二) 后台模板的导入	8
三、登录、注册、登出界面的实现	11
(一) 登录界面	11
(二) 注册界面	12
(三) 登出接口	13
四、 前台界面的实现	13
(一) 主页	13
(二) “国际视野” 界面	15
(三) “战争实况” 界面	17
(四) “战争影响” 界面	22
(五) “关于团队” 界面	23
(六) “网友评论” 界面	24
五、 后台界面的实现	25
(一) 主页	25
(二) 团队管理和用户管理	26
(三) 数据管理	27
(四) 作业下载	28
六、一键部署的实现	30

第一部分 分工与记录

一、分工

韩佳迅（组长）

1. 完成后台模板的导入、用户管理、作业下载、部分数据管理页面
2. 完成前台主页、国际视野页面、对经济、能源的影响页面
3. 完成了对应于数据表 countriesview、energy、economics、ourteammems、teamhomework、personhomework、user 的 MVC 结构
4. 负责文档：实现文档、项目展示 PPT

林语盈

1. 完成前台对国际关系的影响的页面
2. 完成了对应于数据表 internalrelation 的 MVC 结构，以及相应的后台数据管理
3. 完成了爬虫，对 news 数据表的数据爬取
4. 实现一键部署功能
5. 负责文档：实现文档、部署文档

彭钰婷

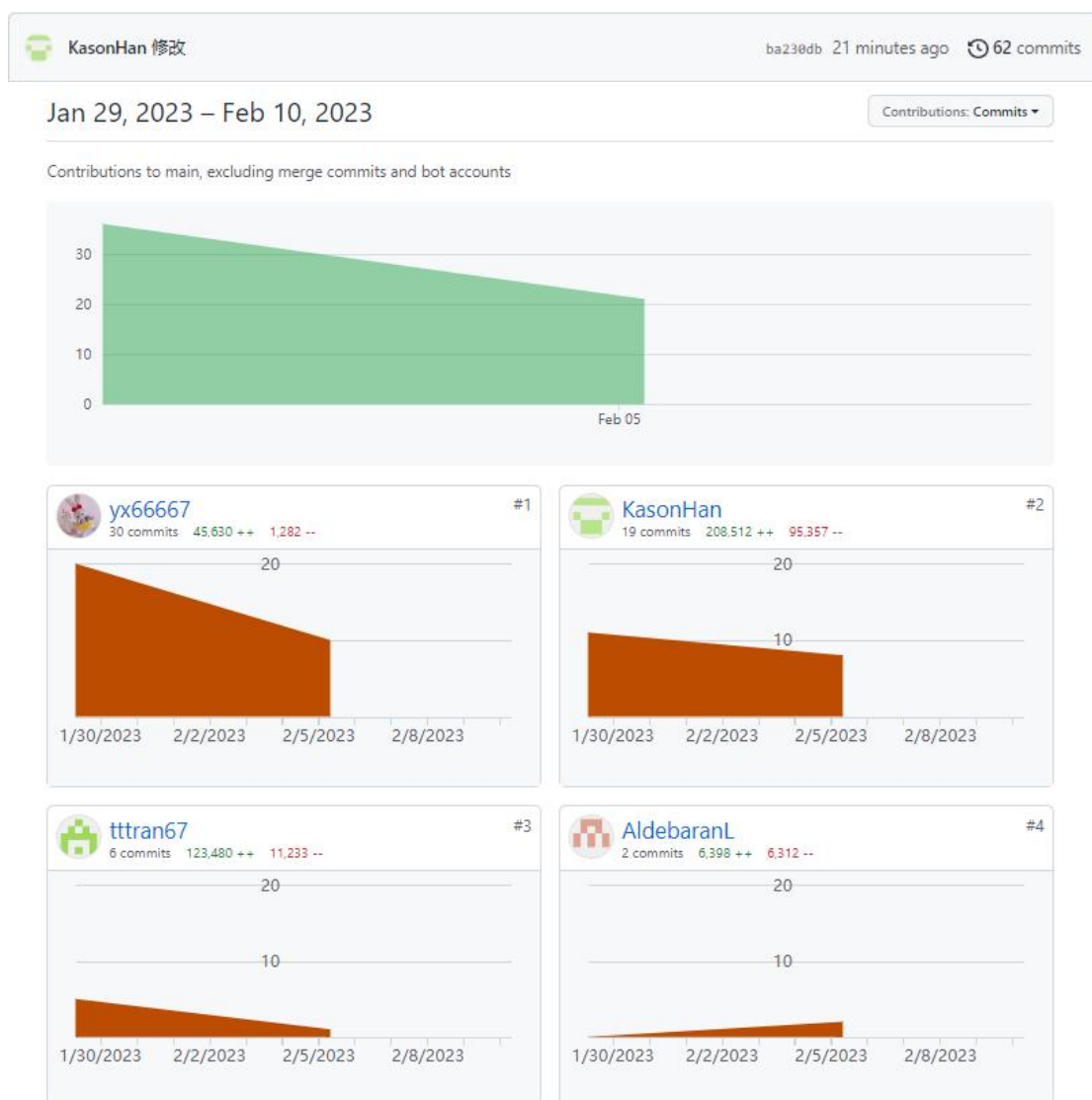
1. 完成前台的战争实况中的新闻、时间线页面
2. 完成了对应于数据表 news 的 MVC 结构
3. 完成了爬虫，对 news 数据表的数据爬取
4. 负责文档：实现文档、用户手册

姚鑫




























1. 完成前台模板的导入，前台导航栏和主页
2. 完成前台关于团队、网友评论页面
3. 完成了对应于数据表 comment 的 MVC 结构
4. 负责文档：实现文档、需求文档、设计文档

二、记录

(一) git 提交



（二）Tower 分工

互联网数据库开发					   添加成员	
看板 列表 日历 时间线 文件 文档 进展 回收站						
添加任务					保存视图	<input checked="" type="checkbox"/> 所有状态
					筛选	↑↓ 排序
					按源单分组	任务字段
					默认	
任务标题	优先级	截止时间	负责人	完成百分比	+	
✓ Yii框架导入、初始化	普通	2月01日		100%		
✓ 前台模板导入	普通	2月02日	 郑鑫	100%		
✓ 后台模板导入	普通	2月02日	 HJX	100%		
✓ 后台登录、退出界面	普通	2月02日	 HJX	100%		
✓ 前台成员介绍界面	普通	2月03日	 郑鑫	100%		
✓ 前台登录、注册、退出界面	普通	2月03日	 HJX	100%		
✓ MVC结构——团队成员	普通	2月03日	 HJX	100%		
✓ MVC结构——前台新闻界面	普通	2月04日	 PyI	100%		
✓ 前台影响、国际关系界面	普通	2月05日	 林进盈	100%		
✓ 前台主页面	普通	2月04日	 HJX	100%		
✓ 前台导航栏、主页部分内容	普通	2月05日	 郑鑫	100%		
✓ 前台影响、经济、能源界面	普通	2月05日	 HJX	100%		
✓ 前台国际视野界面	普通	本周二	 HJX	100%		
✓ 后台国际关系管理	普通	本周二	 林进盈	100%		
✓ 前台网表评论界面	普通	本周二	 郑鑫	100%		
✓ 爬虫文件完善	普通	本周二	 PyI	100%		
:: ✓ 后台所有数据管理的界面	普通	本周二	 HJX	100%		
✓ 爬虫代码	普通	本周三	 林进盈	100%		
✓ MVC结构——前台时间线界面	普通	本周三	 PyI	100%		
✓ 部署代码	普通	昨天	 林进盈	100%		
✓ 后台所有作业下载的界面	普通	昨天	 HJX	100%		
✓ 实现文档	普通	今天		100%		
✓ 用户文档	普通	今天	 PyI	100%		
✓ 部署文档	普通	今天	 林进盈	100%		
✓ 设计文档、需求文档	普通	今天	 郑鑫	100%		
✓ 项目展示PPT	普通	今天	 HJX	100%		

第二部分 代码实现

一、导入 Yii 框架

注：本组选择在 xampp 环境下导入并运行 Yii 框架。

1. 下载归档文件安装包，选择高级应用模板。

（下载地址 <https://www.yiichina.com/download>）

2. 将项目压缩包解压到 xampp/htdocs 文件夹

3. 在项目文件夹下执行./init.bat 命令，选择 development 版本安装。（若出现 php.exe 不可运行的问题，则将 init.bat 中的 PHP_COMMAND，改为 xampp 安装文件夹中的 php 文件夹的 php.exe 的

绝对路径。)

4. 打开/common/config/main-local.php, 填入数据库账号密码。并在 mysql 中创建一个名为 yii2advanced 的数据库。

```
<?php
return [
    'components' => [
        'db' => [
            'class' => 'yii\db\Connection',
            'dsn' => 'mysql:host=localhost;dbname=yii2advanced',
            'username' => 'root',
            'password' => '123456',
            'charset' => 'utf8',
```

5. 在项目文件夹下执行 php yii migrate 命令, 会自动创建相关表。
6. 打开前台 http://localhost:8081/yii_project/frontend/web/ (修改为自己端口号), 则成功进入前台。

Congratulations!

You have successfully created your Yii-powered application.

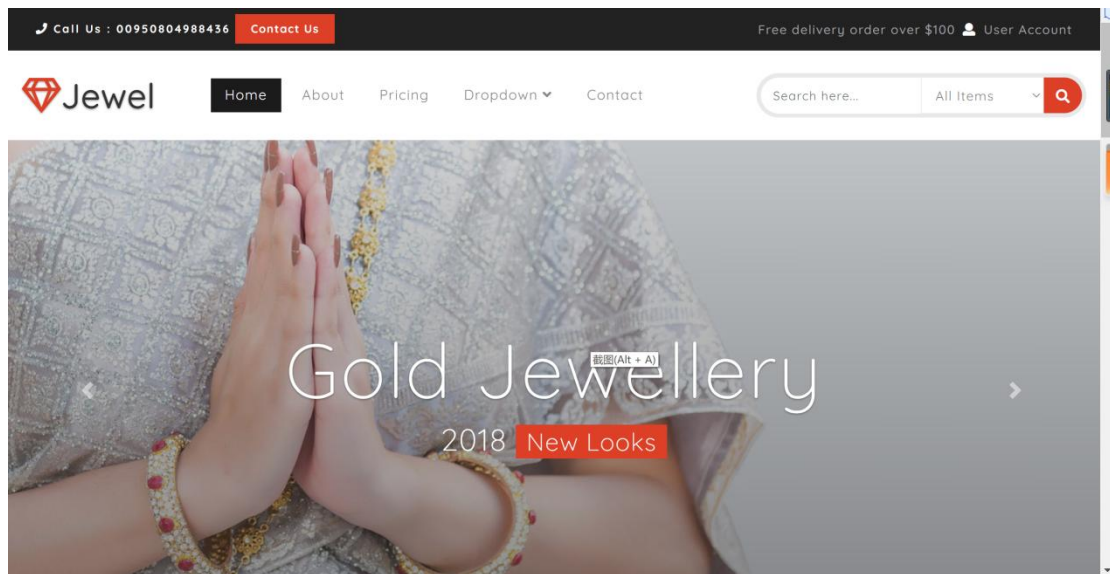
Get started with Yii

7. 在前台注册, 注册后数据库的 user 表多了一条记录, 将这条记录的状态 status 为 9, 改为 10。
8. 打开后台 http://localhost:8081/yii_project/backend/web/ (修改为自己端口号) 即可用注册信息登录。

二、导入模板

(一) 前台模板的导入

1. 在网上找到喜欢的前台模板, 并下载。



2. 打开其中的 `intex.html` 文件，按 `F12`，将不用的部分都删除，最终只剩下导航栏部分（包括原文件的 `header` 部分、`model` 部分、`script` 部分）。

3. 在 `frontend/views/layouts` 中新建文件 `main_layout.php`，将中间的框架代码换成 `intex.html` 文件所抠出来的前台模板框架代码。

在 `header` 后加入以下代码，为了之后每个页面的渲染。

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

4. 将模板所需要的 `css`、`js`、`image` 等文件夹放到 `/frontend/web` 文件夹下，命名文件夹为 `statics`。

5. 修改 `main_layout.php` 中引用资源样式文件的路径。并修改 `AppAsset.php` 文件，将原来 `baseUrl` 改成将 `web/statics`，将模板需要的 `css`、`js` 文件放在其中，注释掉原来 `yii` 框架的 `depends` 部分。

```

public $baseUrl = '@web/statics';
public $css = [
    //'css/site.css',
    'css/bootstrap.css',
    'css/style.css',
    'newscss/style.css',
    'newscss/font-awesome.min.css',
    'newscss/bootstrap.min.css',
    'css/prettyPhoto.css',
    'css/owl.carousel.css',
    'css/owl.theme.css',
    'css/fontawesome-all.css'
];
public $js = [
    'js/jquery-2.2.3.min.js',
    'js/bootstrap.js',
    'js/owl.carousel.js',
    'js/move-top.js',
    'js/easing.js',
    'js/jquery-1.7.2.js',
    'js/jquery.quicksand.js',
    'js/script.js',
    'js/jquery.prettyPhoto.js',
    'js/newsjs/bootstrap.min.js',
    'js/newsjs/jquery.min.js',
    'js/newsjs/main.js'
];
public $depends = [
    //'yii\web\YiiAsset',
    //'yii\bootstrap\BootstrapAsset',

```

6. 在 frontend\controllers\SiteController.php 文件中，设置全局变量 layout 为我们新建的 main_layout.php。

```

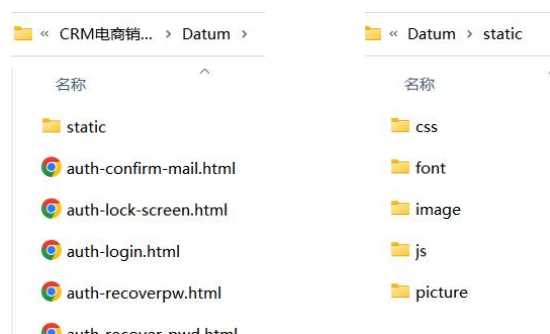
class SiteController extends Controller
{
    /**
     * {@inheritdoc}
     */
    public $layout = "main_layout";

```

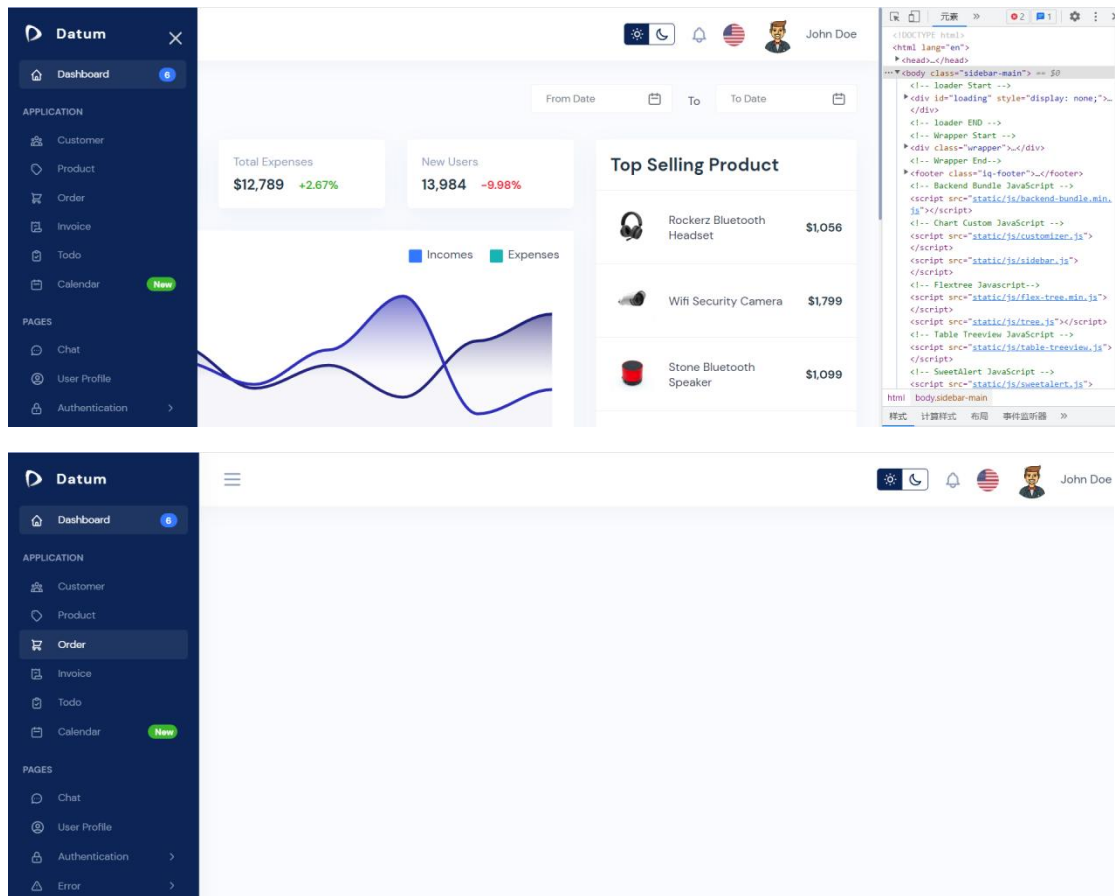
7. 此时，再打开前台 localhost:8081/yii_project/backend/web/，则可看到模板已经成功导入。

（二）后台模板的导入

1. 在网上找到喜欢的后台模板，并下载



2. 打开其中的 `intex.html` 文件，按 `F12`，将不用的部分都删除，最终只剩下我们需要的框架部分。



3. 在 `backend/views/layouts` 中新建文件 `main_layout.php`，将 `main.php` 的文件头复制到 `main_layout.php`，将中间的框架代码换成上一步所抠出来的后台模板框架代码。

4. 在框架的合适部位插入如下代码，使得框架得以“镂空”，显示 `site` 文件夹下的内容。

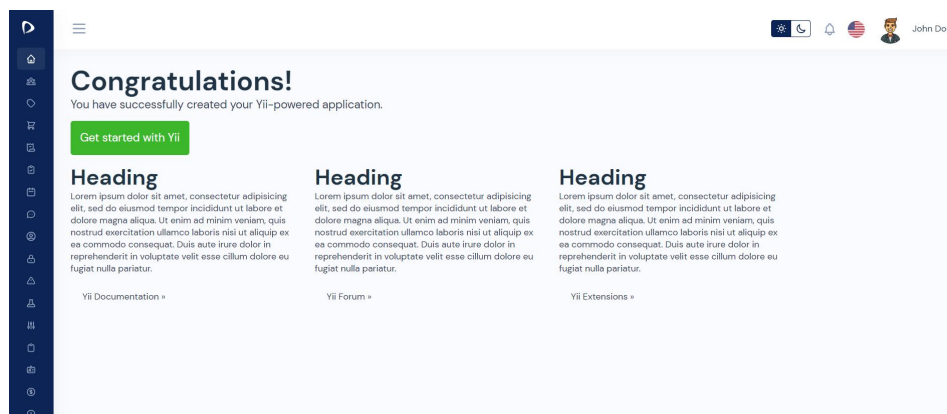
```
<div class="row">
  <div class="col-lg-15 col-md-15 col-sm-15 col-xs-15">
    <div class="container">
      <?= Breadcrumbs::widget([
        'links' => isset($this->params['breadcrumbs']) ? $this->params['breadcrumbs'] : [],
      ]) ?>
      <?= Alert::widget() ?>
      <?= $content ?>
    </div>
  </div>
</div>
```

5. 将模板所需要的 css、js、image 等文件夹放到/backend/web 文件夹下，修改 main_layout.php 中引用资源样式文件的路径，并修改 AppAsset.php 文件。

6. 在 backend\controllers\SiteController.php 文件中，设置变量 layout 为我们新建的 main_layout.php。

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

7. 此时，再打开后台 localhost:8081/yii_project/backend/web/，则可看到模板已经成功导入。

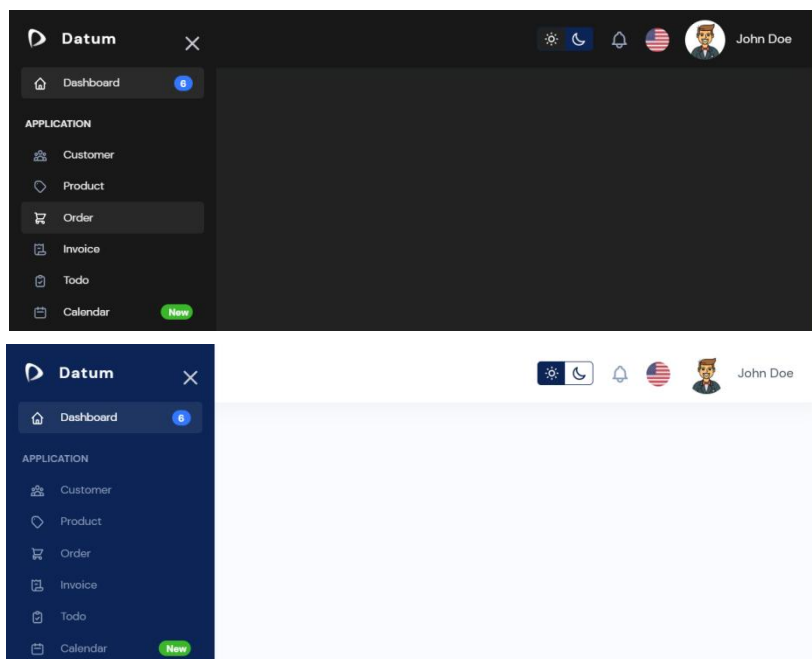


8. 我们的后台模板支持黑夜和白昼两种模式，用户可自定义切换。其实现上，在代码中用 stroke="currentColor" 定义不同的模式，根据当前所选的模式，使用不同的颜色。

```

<div class="change-mode"> == $0
  <div class="custom-control custom-switch custom-switch-icon custom-control-inline"> {flex}
    <div class="custom-switch-inner">
      <p class="mb-0"> </p>
      <input type="checkbox" class="custom-control-input" id="dark-mode" data-active="false">
      <label class="custom-control-label" for="dark-mode" data-mode="toggle">
        ::before
        <span class="switch-icon-right">
          <svg xmlns="http://www.w3.org/2000/svg" id="h-moon" height="20" width="20" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor">
            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M20.354 15.354A9 9 0 018.646 3.646 9.003 9.003 0 0012 21a9.003 9.003 0 008.354-5.646z">
            </path>
          </svg>
        </span>
        <span class="switch-icon-left">
          <svg xmlns="http://www.w3.org/2000/svg" id="h-sun" height="20" width="20" class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor">
            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M12 3v1m0 16v1m9-9h-1M4 12h3m15.364 6.364l-.707-.707M6.343 6.343l-.707-.707M12 21a9.003 9.003 0 008.354-5.646z">
            </path>
          </svg>
        </span>
        ::after
      </label>
    </div>
  </div>
</div>

```



三、登录、注册、登出界面的实现

（一）登录界面

1. 由于我们的登录、注册界面和后台是同一个模板，因此可以用同一份资源样式文件夹，不需重新导入。
2. 将登录界面的模板代码复制到\backend\views\site\login.php（前台也是），替换其中的主体部分。

3. 在登录模板的输入部分，用如下代码进行替换。使得其能够正确执行相关动作并提交表单。

后台：在输入框的最下面添加跳转接口，让其能够直接从登录界面跳转到前台界面。

前台：添加跳转到注册界面的接口。

```
<div class="col-lg-12">
  <?php $form = ActiveForm::begin(['id' => 'login-form']); ?>

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

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

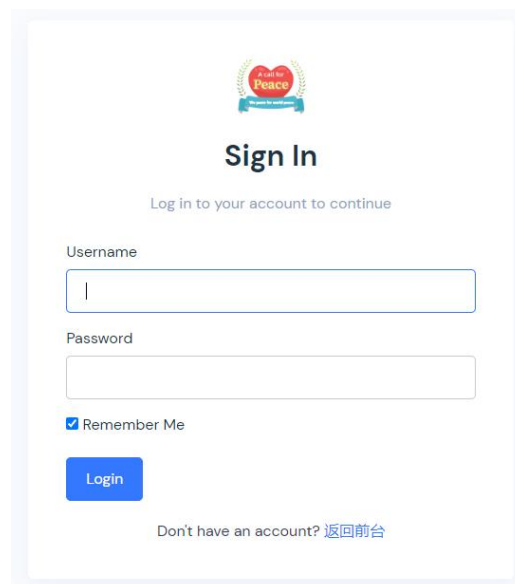
  <?= $form->field($model, 'rememberMe')->checkbox() ?>

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

  <?php ActiveForm::end(); ?>

  <div class="col-lg-12 mt-3">
    <p class="mb-0 text-center">Don't have an account? <a href="../../frontend/web">返回前台</a></p>
  </div>
</div>
```

4. 实现界面如下：



（二）注册界面

注册与登录界面的实现方法相同（在 `signup.php` 文件中完成），不在

此赘述。

（三）登出接口

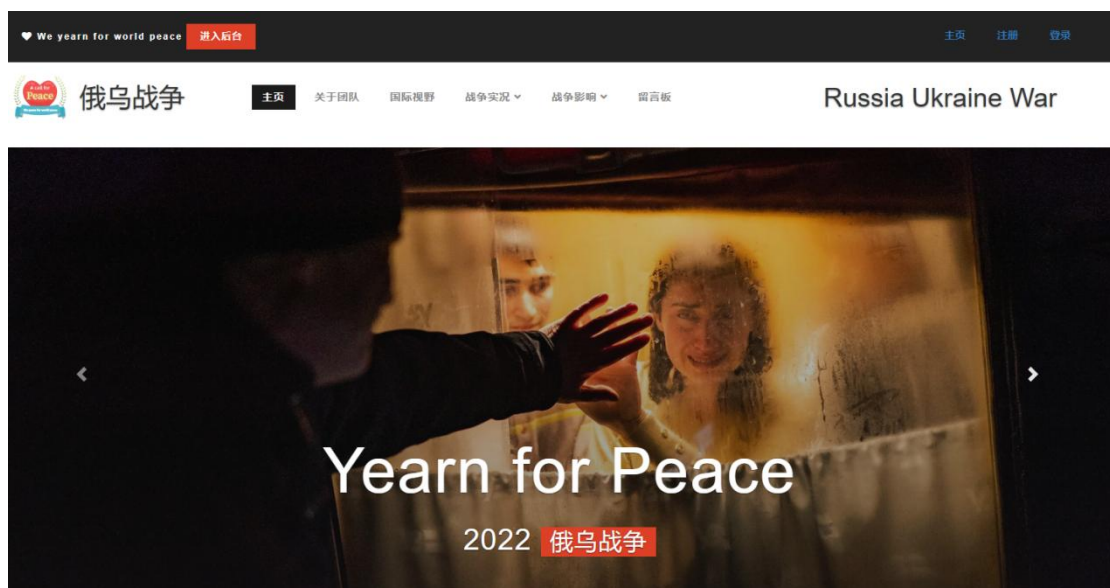
在 main_layout 中添加如下代码，在导航栏中实现登出功能。

```
<form action="<?php echo Yii::$app->getHomeUrl(); ?>?r=site%2Flogout" method="post">
  <input type="hidden" name="_csrf-backend" value="<?= Yii::$app->request->csrfToken ?>">
  <button type="submit" class="btn btn-link logout"> logout</button>
</form>
```

四、前台界面的实现

（一）主页

1. 将主页的显示图片和文字换成相应的主题。



2. 在介绍部分实现了图片的滚动：鼠标未移入图片区域时，图片横向滚动；鼠标移入后，图片静止。实现代码如下。

俄罗斯乌克兰冲突



2022年2月15日，随着俄罗斯国防部高调宣布撤回部分部署在俄乌边境，此前正在参与大规模军事演习的陆上部队，从2021年10月起持续至2022年2月的乌克兰东部危机，似乎呈现出了降温降温的趋势。

2022年2月17日，乌克兰地区局势恶化，乌克兰和当地民间武装相互指责对方在接触线地带发动挑衅性袭击。2月18日，乌克兰民间武装宣布，因存在乌克兰发起军事行动的危险，自即日起向俄罗斯大规模撤离中流散当地居民。2月21日晚，俄罗斯总统普京签署命令，承认乌克兰东部的“顿涅茨克人民共和国”和“卢甘斯克人民共和国”。

2022年2月24日，乌克兰管理部门宣布关闭全国领空，乌克兰总统泽连斯基表示，乌克兰全境将进入战时状态，首都基辅地铁免费开放，地铁站将作为防空洞使用；俄军开始对乌克兰东部部队和其他地区的军事指挥中心、机场进行炮击。乌克兰国民卫队司令部被摧毁。2月24日，乌克兰宣布与俄罗斯断交。2月24日，乌克兰边防部队称俄军突入基辅地区。当地时间2月24日，乌克兰基辅市政府发出防空警报，通知所有人立即前往民防避难所避难。当地时间26日，乌克兰基辅市长宣布，该市地铁转为避难所，不再提供运输服务。3月2日，乌克兰已经关闭基辅俄罗斯圣彼得堡的总领馆。13日凌晨，俄军对利沃夫州瓦西洛瓦斯基训练场的空袭共造成9人死亡，37人受伤。5月3日，俄军摧毁一处美国援乌克兰火库，内部存放导弹和无人机。

>> 了解更多

```
<script>
var speed=18//速度数值越大速度越慢
butong_net_left2.innerHTML=butong_net_left1.innerHTML
function Marquee3(){
    if(butong_net_left2.offsetwidth-butong_net_left.scrollLeft<=0)
        butong_net_left.scrollLeft-=butong_net_left1.offsetwidth
    else{
        butong_net_left.scrollLeft++
    }
}
var MyMar3=setInterval(Marquee3,speed)
butong_net_left.onmouseover=function() {clearInterval(MyMar3)}
butong_net_left.onmouseout=function() {MyMar3=setInterval(Marquee3,speed)}
</script>
```

3. 在热门问答区域实现选项卡和视频。点击卡片可下拉显示内容。

热门问答

相关视频



中国对于俄乌冲突的立场和态度? ∨

- ◆ 中国坚决支持联合国宪章，支持国家的主权独立和领土完整。
- ◆ 这个事件的发展有复杂的历史原因。
- ◆ 一定要尽快劝和促谈、结束战争。
- ◆ 希望在乌克兰危机之后建立一个更加均衡、更加可持续的欧洲安全体系。

中国对冲突的应对? ∨

俄罗斯面临的困境? ∨

乌克兰所处的态势? ∨

```
<h2 class="mb-0" id="headingOne">
  <button class="btn btn-link" type="button" data-toggle="collapse"
    data-target="#collapseOne" aria-expanded="true" aria-controls="collapseOne" style="font-family: 华文新魏;font-size:18px" >
    <b>中国对于俄乌冲突的立场和态度? </b><i class="fa fa-chevron-down"></i>
  </button>
</h2>

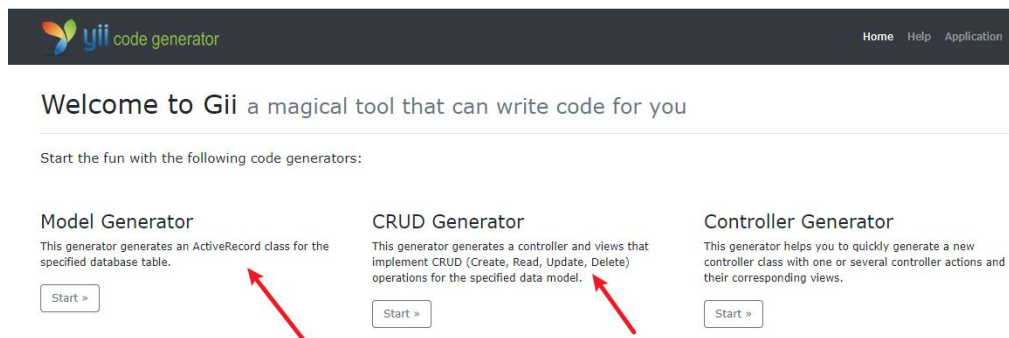
<div id="collapseOne" class="collapse show" aria-labelledby="headingOne"
  data-parent="#accordion 1">
  <div class="accordion-body">
    &diam; 中国坚决支持联合国宪章，支持国家的主权独立和领土完整。
    <p></p>
    &diam; 这个事件的发展有复杂的历史原因。
    <p></p>
    &diam; 一定要尽快劝和促谈、结束战争。
    <p></p>
    &diam; 希望在乌克兰危机之后建立一个更加均衡、更加可持续的欧洲安全体系。
  </div>
</div>
```


（二）“国际视野”界面

1. 在数据库中创建数据表 `countriesview`，字段包括：国家 `country`、人名 `name`、观点 `view`、图片 `picture`、序号 `num_id`。

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
1	<u>country</u>	varchar(100)	utf8mb4_general_ci		否	无	国家		修改 删除 更多
2	<u>name</u>	varchar(100)	utf8mb4_general_ci		否	无	人名		修改 删除 更多
3	<u>view</u>	text	utf8mb4_general_ci		否	无	观点		修改 删除 更多
4	<u>picture</u>	varchar(100)	utf8mb4_general_ci		否	无	图片		修改 删除 更多
5	<u>num_id</u>	int			否	无	序号	AUTO_INCREMENT	修改 删除 更多

2. 借助 Yii 框架的 gii 工具在前台生成 MVC 结构。其中模型（Countriesview）映射到表 `countriesview`；视图（`views/countriesview`）展示数据表的相关内容；控制器（CountriesviewController）管理模型和视图中的通信。



3. 在控制器中，获取数据表的所有行，并以字段 `num_id` 排序。

```
<?php

namespace frontend\controllers;

use Yii;
use frontend\models\Countriesview;
use frontend\models\CountriesviewSearch;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;
use yii\data\Pagination;
use yii\db\ActiveRecord;
use yii\data\Sort;
$Countriesview = Countriesview::find() -> orderBy('num_id') -> all();
```

4. 在控制器中创建动作，`index` 操作调用了活动记录 `Countriesview`

的 `find()` 方法,去生成查询语句并从 `Countriesview` 表中取回所有数据。
为了限定每个请求所返回的数据行数,查询在 `yii\data\Pagination` 对象的帮助下进行分页。

```
public function actionIndex()
{
    $searchModel = new CountriesviewSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

    $query = Countriesview::find();
    $pagination = new Pagination([
        'totalCount' => $query->count(),
        'pageSize' => 10
    ]);
    $Countriesview = $query->orderBy('num_id')
        ->offset($pagination->offset)
        ->limit($pagination->limit)
        ->all();

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
        'Countriesview' => $Countriesview,
        'pagination' => $pagination
    ]);
}
```

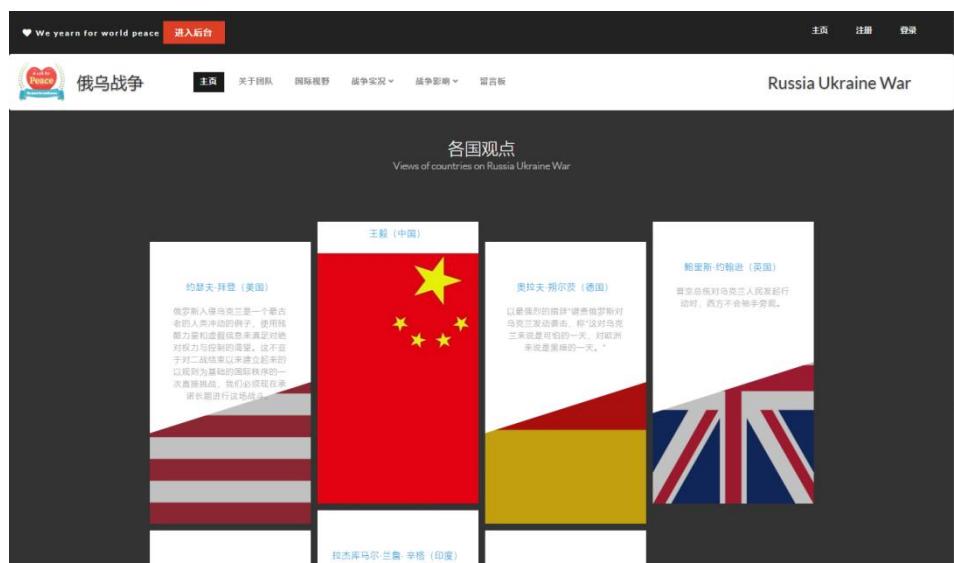
5. 将 `/frontend/views/countriesview/index.php` 的主体部分替换我们的模板主体代码。然后使用 `php` 循环,取出数据并展示其各个字段值。

```
<?php foreach ($Countriesview as $view) : ?>

<a href="#" data-path-hover="m 180,34.57627 -180,0 L 0,0 180,0 z">
    <figure>
        
        <svg viewBox="0 0 180 320" preserveAspectRatio="none"><path d="M 180,160 0,218 0,0 180,0 z"></pa
        <figcaption>
            <h2><?=$view->name ?><?=$view->country?></h2>
            <p><?=$view->view ?></p>
        </figcaption>
    </figure>
</a>

<?php endforeach; ?>
```

6. 页面最终显示效果如下,卡片上的标题使用的是数据表中的 `name` 和 `country` 字段,文字使用的是 `view` 字段,图片使用的 `picture` 字段。
并且卡片支持动态显示功能,当鼠标停留在卡片上时,会显示整个底层图片。



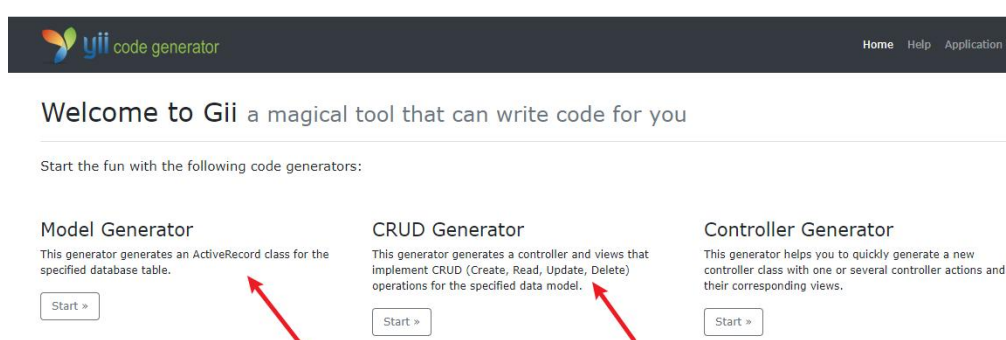
(三) “战争实况”界面

1、新闻页面

1. 在数据库中创建数据表 **news**，字段包括：新闻名 **name**、发布时间 **date**、新闻链接 **url**、序号 **num_id**。

	name 新闻标题	date 发布时间	url 链接	num_id 序号
<input type="checkbox"/>	编辑 复制 删除 俄防长：美西方向乌提供重型武器将导致局势朝着不可预测方向升级	2023-02-07 22:18:58	http://www.chinanews.com.cn/gj/2023/02-07/9949306....	1
<input type="checkbox"/>	编辑 复制 删除 俄乌局势进展：俄称普京与格罗西本周末安排会面 泽连斯基召开最高统帅部会议	2023-02-07 21:11:28	http://www.chinanews.com.cn/gj/2023/02-07/9949259....	2

2. 借助 Yii 框架的 gii 工具在前台生成 MVC 结构。其中模型(**News**)映射到表 **news**；视图 (**views/news**) 展示数据表的相关内容；控制器 (**NewsController**) 管理模型和视图中的通信。



3. 在控制器中，获取数据表的所有行，并以字段 **date** 倒序排序。

```
$news = $query-> orderBy('date DESC');
```

4. 在控制器中创建动作，**index** 操作调用了活动记录 **News** 的 **find()** 方法，去生成查询语句并从 **news** 表中取回所有数据。为了限定每个请求所返回的数据行数，查询在 **yii\data\Pagination** 对象的帮助下进行分页。

```
public function actionIndex()
{
    $query = News::find();
    $pagination = new Pagination([
        'totalCount' => $query-> count(),
        'pageSize' => 200
    ]);
    $news = $query-> orderBy('date DESC')
        -> offset($pagination-> offset)
        -> limit($pagination-> limit)
        -> all();
    $searchModel = new NewsSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

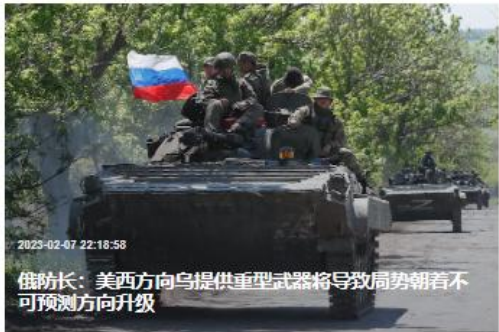
    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
        'news' => $news,
        'pagination' => $pagination
    ]);
}
```

5. 将 **/frontend/views/news/index.php** 的主体部分替换我们的模板主体代码。然后使用 **php** 循环，取出数据并展示其各个字段值。

```
<div class="col-md-4">
    <div class="post">
        <a class="post-img" href="<?=$new->url ?>"></a>
        <div class="post-body">
            <div class="post-meta">
                <?php
                if($cnt4color > 4) {$cnt4color=1;}
                ?>
                <a class="post-category cat-<?=$cnt4color++>" href="<?=$new->url ?>">news<?=$cnt4label++ ?></a>
                <span class="post-date"><?=$new->date ?></span>
            </div>
            <h3 class="post-title"><a href="<?=$new->url ?>"><?=$new->name ?></a></h3>
        </div>
    </div>
</div>
```

6. 页面最终显示效果如下，卡片上的标题使用的是数据表中的 **name**

字段，时间使用的是 `date` 字段，链接使用的是 `url` 字段。鼠标点击图片/标题/标签后即可跳转到新闻内容页面。



Recent Posts



俄乌局势进展：俄称普京与格罗西本周末安排会面 泽连斯基召开最高统帅部会议



局势紧张，换不换防长，泽连斯基犯难了



非法移民和难民问题让欧盟左右为难（环球热点）



2、时间线页面

1. 爬取新闻：

最新新闻通过 Python 爬虫实现。解析中国网搜索的前几条新闻，并获取其 `url`、发布时间和新闻标题，更新 `news` 数据库，再通过 `news` 数据库加载到前台的时间线界面，进行输出。实现在

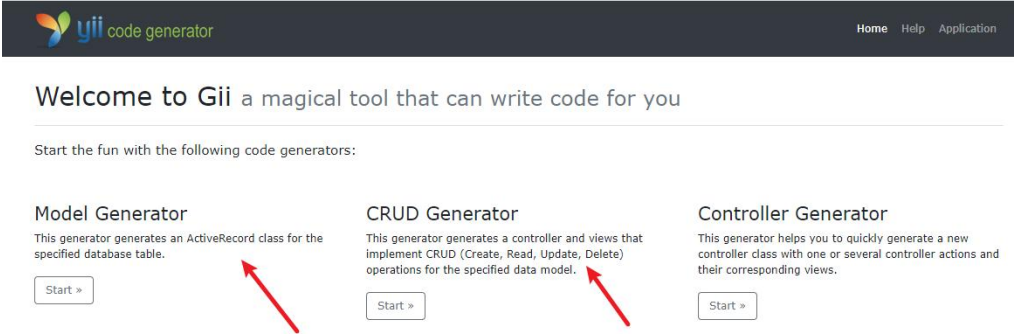
timeline_crawler.py 文件中，并在部署时，后台开始执行。

具体来说，crawl 函数用于解析新闻的 HTML 格式，输出需要的结果。updateDB 函数用于连接数据库并执行 SQL 语句进行更新。loop_func 函数每隔十分钟执行上述函数并获取最新的新闻。

2. 使用数据库中的数据表 news，字段包括：新闻名 name、发布时间 date、新闻链接 url、序号 num_id。

	name 新闻标题	date 发布时间	url 链接	num_id 序号
<input type="checkbox"/>	俄防长：美西方向乌提供重型武器将导致局势朝着不可预测方向升级	2023-02-07 22:18:58	http://www.chinanews.com.cn/gj/2023/02-07/9949306....	1
<input type="checkbox"/>	俄乌局势进展：俄称普京与格罗西本周末安排会面 泽连斯基召开最高统帅部会议	2023-02-07 21:11:28	http://www.chinanews.com.cn/gj/2023/02-07/9949259....	2

3. 借助 Yii 框架的 gii 工具在前台生成 MVC 结构。其中模型(News)映射到表 news；视图 (views/news) 展示数据表的相关内容；控制器 (NewsController) 管理模型和视图中的通信。



4. 在控制器中，获取数据表的所有行，并以字段 date 倒序排序。

```
$timelines = $query-> orderBy('date DESC')
```

5. 在控制器中创建动作，index 操作调用了活动记录 Timeline 的 find() 方法，去生成查询语句并从 news 表中取回所有数据。为了限定每个请求所返回的数据行数，查询在 yii\data\Pagination 对象的帮助下进行分页。


```

public function actionIndex()
{
    $query = Timeline::find();
    $pagination = new Pagination([
        'totalCount' => $query-> count(),
        'pageSize' => 200
    ]);
    $timelines = $query-> orderBy('date DESC')
        -> offset($pagination-> offset)
        -> limit($pagination-> limit)
        -> all();
    $searchModel = new TimelineSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

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

```

6. 将/frontend/views/timeline/index.php 的主体部分替换我们的模板主体代码。然后使用 php 循环，取出数据并展示其各个字段值。

```

<?php foreach ($timelines as $timeline) : ?>
    <?php
        if($cnt4image > 14) $cnt4image = 1;
    ?>
    <div class="item">
        <div class="top-wrap c-color-gray">
            <span class="dot"></span>
            <span class="time c-gap-left-small"><?= $timeline->date ?></span>
        </div>
        <div class="c-gap-left-large c-gap-inner-top-middle content pic-content">
            <div class="img-content">
                <a class="post-img" href="<?= $timeline->url ?>"></a>
            </div>
            <div class="c-text c-font-medium">
                <a class="content-link c-line-clamp3" target="_blank" href="<?= $timeline->url ?>">
                    <!--s-slot--> <?= $timeline->name ?> <!--/s-slot--> </a>
                </div>
            </div>
        </div>
    </div>
<?php endforeach; ?>

```

7. 页面最终显示效果如下，新闻按时间降序给出，卡片上的标题使用的是数据表中的 name 字段，时间使用的是 date 字段，链接使用的是 url 字段。鼠标点击图片/标题/标签后即可跳转到新闻内容页面。

俄乌战争新闻时间线

更新至2023-02-07 22:18:58

○ 2023-02-07 22:18:58 (新)

俄防长：美西方方向乌提供重型武器将导致局势朝着不可预测方向升级

○ 2023-02-07 22:18:58



俄防长：美西方方向乌提供重型武器将导致局势朝着不可预测方向升级

○ 2023-02-07 21:11:28

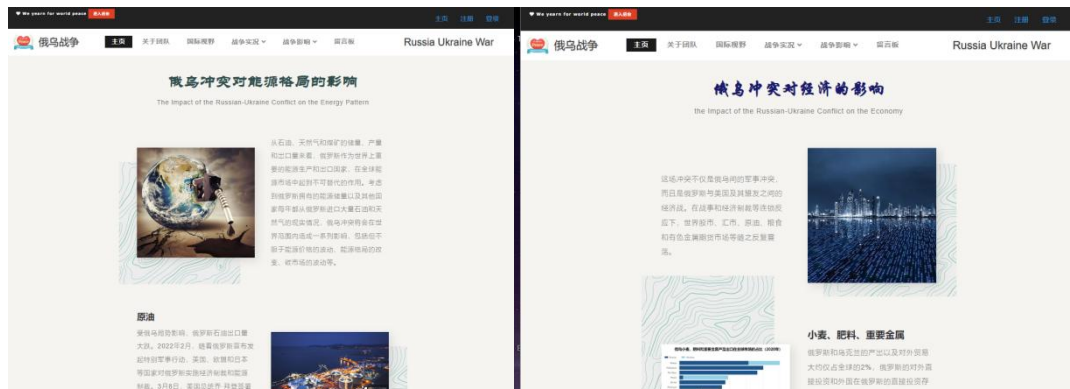


俄乌局势进展：俄称普京与格罗西本周末安排会面 泽连斯基召开最高统帅部会议

○ 2023-02-07 16:40:21

(四) “战争影响”界面

我们在战争影响部分创建了三个界面，分别为“国际关系”、“能源”和“经济”。和“国际视野”界面的实现步骤相同，创建数据表并生成 MVC，更改控制器中的动作以及视图代码的布局。此处不做赘述。



（五）“关于团队”界面

1、首先是团队介绍正文部分，添加一个图片和一段文字。



团队介绍

本网站是南开大学互联网数据库开发课程大作业，网站建站依靠yii框架进行搭建，网站页面分为前台部分和后台部分，同时网站也连接了数据库进行使用。

本网站由四名计算机学院和网络空间安全学院的大三学生创建，我们团队成员具有较强的个人能力和不错的团队协作能力，成员分工明确，各司其职，互帮互助，团队内有良好的学习氛围，遇到问题能及时沟通解决。如果有任何问题，可以在下面点击团队成员github链接访问主页，并联系我们。

2、接下来这部分介绍团队 git 提交数、团队成员、开发用的天数、文件总数的导引部分。

3、四个人的个人介绍部分，并且附上每个人的 github 主页跳转链接。



韩佳迅

后台模板导入，前台主页、国际视野（经济、能源）页面实现，后台用户管理、数据管理、作业下载功能实现。

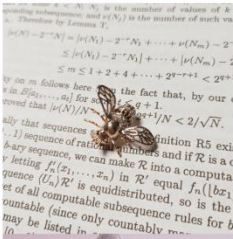
• [看我的github主页](#)



彭钰婷

前台新闻页面、时间线页面设计和实现，实现爬虫，对news数据表进行数据爬取。

• [看我的github主页](#)



林语盈

前台国际关系页面实现，实现了爬虫，对news数据表进行数据爬取，实现了一键部署功能。

• [看我的github主页](#)



姚鑫

前台模板导入，导航栏部分页面设计，团队介绍、网友评论页面设计和实现。

• [看我的github主页](#)

4、每个人的个人作业、团队报告部分 github 跳转链接展示。

组员作业展示

下面是我们团队成员在课程中个人作业的github链接，欢迎浏览。（下载功能提供在后台，请登录后台下载）



Web前端初探

- > 韩佳迅第一次作业
- > 林语盈第一次作业
- > 彭钰婷第一次作业
- > 姚鑫第一次作业



Web前端设计

- > 韩佳迅第二次作业
- > 林语盈第二次作业
- > 彭钰婷第二次作业
- > 姚鑫第二次作业



开源建站工具初试

- > 韩佳迅第三次作业
- > 林语盈第三次作业
- > 彭钰婷第三次作业
- > 姚鑫第三次作业



团队作业开发文档

- > 需求文档
- > 设计文档
- > 实现文档
- > 用户手册
- > 部署文档
- > 展示PPT

5、最后加上网站统一的 footer 部分。



（六）“网友评论”界面

创建过程和之前相似：首先在数据库创建 `comment` 数据表，之后在网站使用 `gii` 工具生成 MVC 模型，并修改 `model` 部分和 `controller` 部分。在 `view` 的 `comment/index.php` 中修改页面布局，并加上统一的 footer 部分。最后呈现效果如下。



五、后台界面的实现

（一）主页

更改 `site/index.php`，通过更改字号、颜色、添加图片等方法，在主页实现了团队的介绍，如下图。

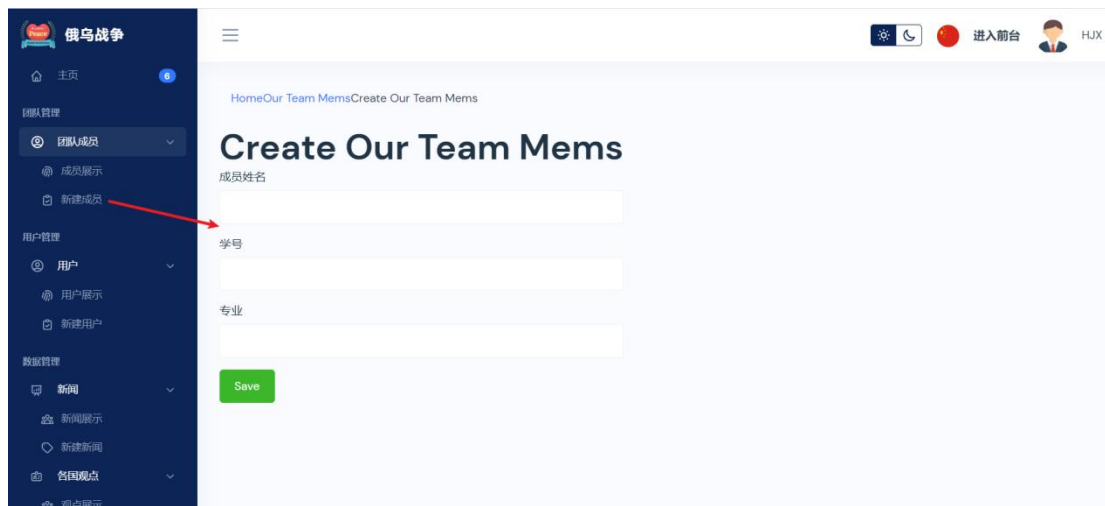




（二）团队管理和用户管理

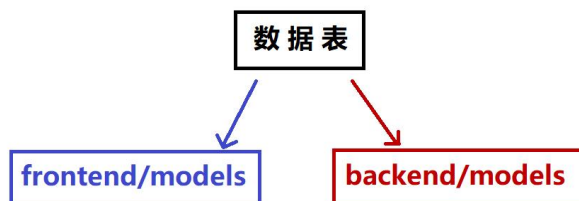
在数据库新建数据表 `Ourteammems`，并且使用 `gii` 工具为其创建 MVC 结构。更改 `main_layout` 中的导航栏名称和跳转地址，使其能够跳转到 `views/our-team-mems` 文件夹下，显示 `index.php`。并且将 `OurTeamMemsController` 的 `layout` 设置为我们的模板 `main_layout`。最终界面如下。（用户管理实现方法相同，不在此赘述）



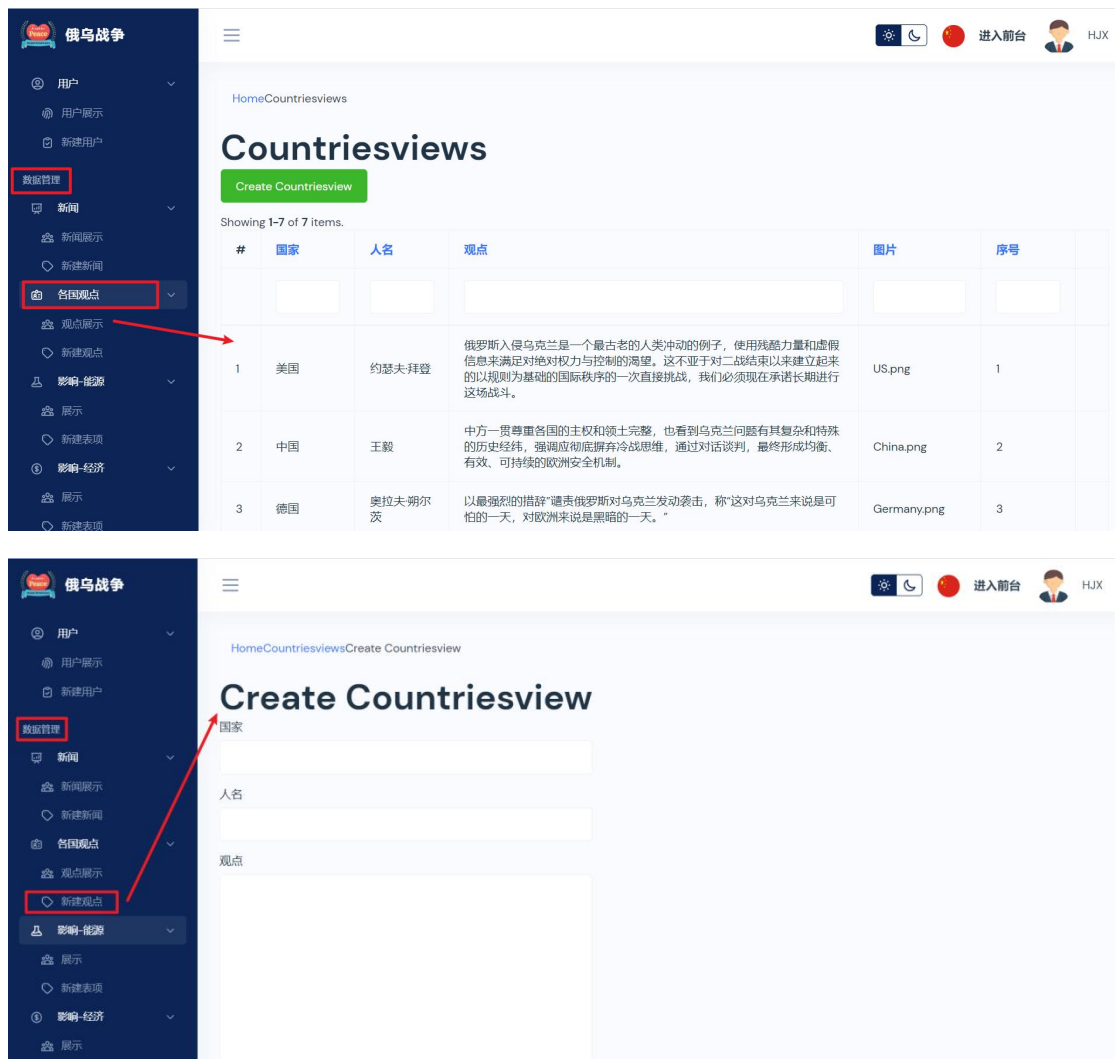


（三）数据管理

我们在前台建立的 MVC 结构的数据表，在后台也要进行相应的管理。我们采用前后台分别建立 MVC，且映射到同一个数据表的方法，如下图所示。



使得前后台对同一个数据库操作，前台负责展示，后台负责管理。生成 MVC 结构、更改视图的方法和“团队管理”界面相同，此处不做赘述。



（四）作业下载

建立数据表 **teamhomework**，字段包括作业名称和下载地址；同样地，建立数据表 **personhomework**，字段有人名、三个作业的地址、个人 GitHub 链接。分别为两张表生成 MVC，更改控制器中的动作以及视图代码的布局。视图代码以及最终的效果分别如下图。

```

<div class="site-index">

    <div class="jumbotron">
        <h1><p align="center">团队作业</p></h1>

        <p align="center" style="font-size:20px">团队成员：韩佳迅、林语盈、彭钰婷、姚鑫</p>
    </div>
    <body>
        <p style="text-align: center;">
            
        </p>
    </body>

    <div class="body-content">
        <div class="row">

            <?php foreach ($Teamhomework as $th) : ?>
                <div class="col-lg-4">
                    <p style="font-size:24px;color:navy"><b><?=$th->name ?></b></p>
                    <a href="../../data/team/<?=$th->position?>" ><span style="font-size:20px;">文件下载&nbsp;&nbsp;&nbsp;&nbsp;</span><br /></a></p>
                </div>
            <?php endforeach; ?>

        </div>
        <p align="center"><a class="btn btn-default btn-info" href="https://github.com/hanmaxmax/Yii_project">项目Github链接 >></a></p>
    </div>
</div>

```



```

<?php foreach ($Personhomework as $ph) : ?>
    <div class="col-lg-6">
        <h3><p style="color:navy"><?=$ph->name?></p></h3>
        <p>
            <p><span style="font-size:24px;">个人作业一：</span>
            <a href="../../data/personal/<?=$ph->hw1?>" ><span style="font-size:24px;">zip下载&nbsp;&nbsp;&nbsp;&nbsp;</span><br /></a></p>
            <p><span style="font-size:24px;">个人作业二：</span>
            <a href="../../data/personal/<?=$ph->hw2?>" ><span style="font-size:24px;">zip下载&nbsp;&nbsp;&nbsp;&nbsp;</span><br /></a></p>
            <p><span style="font-size:24px;">个人作业三：</span>
            <a href="../../data/personal/<?=$ph->hw3?>" ><span style="font-size:24px;">zip下载&nbsp;&nbsp;&nbsp;&nbsp;</span><br /></a></p>
        </p>
        <p><a class="btn btn-default" href="=<?=$ph->github?>">进入Github &raquo;</a></p>
    </div>
<?php endforeach; ?>

```



六、一键部署的实现

一键部署的实现主要在 `install` 文件夹中，其中分为三个文件。
`install.php` 是一键部署的主页面，它调用另外两个文件（管理员注册界面 `install-admin.php` 与数据库连接界面 `install-sql.php`）。其依次：
生成数据库，填写数据库的结构和数据，更改数据库的连接设置，并在最后调用 `Python` 在后台运行爬虫。