

互联网数据库实现文档

2011911 果丁 2012026 穆禹宸 2012885 陈静怡 2013743 林巧莹

Tower及Git提交记录

Tower项目管理

准备工作

准备工作 +	开始时间	截止时间	负责人	概述
<input type="checkbox"/> 板块内容讨论&数据收集分工	2023-01-17	2023-01-17	全员	进行第一次会议，对内容划分类型进行讨论，分配数据收集任务。了解yii2框架
<input type="checkbox"/> 数据收集	2023-01-17	2023-01-27	全员	分为经济、外交、军事与新闻四个板块，每个人根据自己负责的板块进行数据收集
<input type="checkbox"/> 数据整理讨论&建立完整数据库	2023-01-27	2023-01-27	全员	进行第二次会议，就个人收集数据进行整理，最终形成一个完整的数据库
<input type="checkbox"/> 配置环境&进一步了解框架	2023-01-27	2023-01-31	全员	进行搭建yii2框架(advanced)环境的配置，对yii2框架的MVC结构进行深入学习
<input type="checkbox"/> 开发工作分工	2023-01-31	2023-01-31	全员	进行第三次会议，进行前端/后端详细分工，就个人负责板块设计大致界面

开发工作

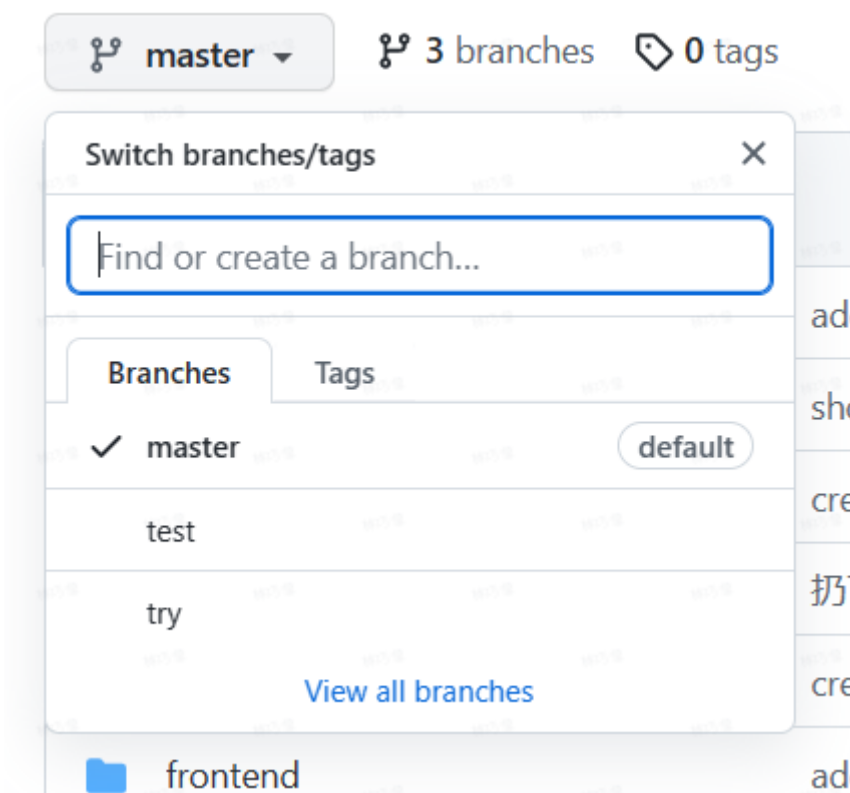
开发工作 +	开始时间	截止时间	负责人
<input type="checkbox"/> 导航栏制作	2023-01-31	2023-02-13	陈静怡、果丁
<input type="checkbox"/> 新闻时间线爬取/制作	2023-01-31	2023-02-03	陈静怡、穆禹宸
<input type="checkbox"/> 可互动世界地图制作	2023-01-31	2023-02-08	陈静怡
<input type="checkbox"/> 外交/Echarts图表制作	2023-01-31	2023-02-08	陈静怡、穆禹宸
<input type="checkbox"/> 经济/Echarts图表制作	2023-01-31	2023-02-08	果丁、林巧莹
<input type="checkbox"/> 军事/Echarts图表制作	2023-01-31	2023-02-08	果丁、穆禹宸
<input type="checkbox"/> 登录/注册功能实现	2023-01-31	2023-02-08	陈静怡、穆禹宸
<input type="checkbox"/> 用户权限限制（登录前后不同）	2023-01-31	2023-02-08	林巧莹
<input type="checkbox"/> 管理员权限实现（增删改查）	2023-01-31	2023-02-08	林巧莹
<input type="checkbox"/> 下载功能实现	2023-02-06	2023-02-12	果丁、林巧莹
<input type="checkbox"/> 上传文件功能实现	2023-02-06	2023-02-12	果丁、林巧莹
<input type="checkbox"/> 留言板实现	2023-02-06	2023-02-12	陈静怡、穆禹宸

文档编写

▼ 文档编写 +		
<input type="checkbox"/> 需求文档	2023-02-06	2023-02-12
<input type="checkbox"/> 设计文档	2023-02-06	2023-02-12
<input type="checkbox"/> 实现文档	2023-02-06	2023-02-12
<input type="checkbox"/> 用户手册	2023-02-06	2023-02-12
<input type="checkbox"/> 部署文档	2023-02-06	2023-02-12
<input type="checkbox"/> 小组PPT制作	2023-02-06	2023-02-12
...		

Git提交记录/版本控制

共建立了三个分支，用于分工合作



提交次数非常多

12fd0bd 20 minutes ago ⌚ 114 commits

3 hours ago

具体的一些提交情况，由于提交太多，这里仅展示部分提交情况














<div>Commits on Feb 4, 2023</div> <div><div>fix: erroneous legend color</div><div>ephatica committed last week</div></div> <div><div>added aid, sanction and cost world map</div><div>ephatica committed last week</div></div> <div><div>added aid, sanction and cost world map</div><div>ephatica committed last week</div></div> <div><div>added refugee world map</div><div>ephatica committed last week</div></div> <div><div>added world map</div><div>ephatica committed last week</div></div> <div><div>add table files to store info of uploaded files</div><div>Roslin-v committed last week</div></div> <div><div>add access control and upload files</div><div>Roslin-v committed last week</div></div> <div><div>后台的体力活结束</div><div>TephrocactusHC committed last week</div></div> <div><div>sanction-type\sanction-date\aid add bar. pie/radar bugs</div><div>alittle-f committed last week</div></div>	<div>Commits on Feb 5, 2023</div> <div><div>new ddl script</div><div>ephatica committed last week</div></div> <div><div>modified button style</div><div>ephatica committed last week</div></div> <div><div>optimized timeline styles</div><div>ephatica committed last week</div></div> <div><div>new index page</div><div>ephatica committed last week</div></div> <div><div>designed about page</div><div>ephatica committed last week</div></div> <div><div>add comment</div><div>ephatica committed yesterday</div></div> <div><div>Minor optimization</div><div>alittle-f committed yesterday</div></div> <div><div>navigation bar&button</div><div>alittle-f committed yesterday</div></div> <div><div>Merge branch 'test' of github.com:ephatica/yii-application into test</div><div>Roslin-v committed yesterday</div></div> <div><div>finished upload and download</div><div>Roslin-v committed yesterday</div></div> <div><div>很好，留言板的提交修复了</div><div>TephrocactusHC committed yesterday</div></div> <div><div>添加了穆禹宸与果丁的file，修复留言功能</div><div>TephrocactusHC committed yesterday</div></div> <div><div>上传个人作业</div><div>TephrocactusHC committed yesterday</div></div> <div><div>add comment in backend(view and delete)</div><div>Roslin-v committed 8 hours ago</div></div> <div><div>fixed bugs in comment</div><div>Roslin-v committed 9 hours ago</div></div> <div><div>update comment (cannot get discuss)</div><div>Roslin-v committed 10 hours ago</div></div> <div><div>something</div><div></div></div> <div><div>对应主页修改更新了展示的PPT</div><div>TephrocactusHC committed 3 hours ago</div></div> <div><div>add slides</div><div>ephatica committed 5 hours ago</div></div> <div><div>ppt上传</div><div>TephrocactusHC committed 6 hours ago</div></div>
<div>Commits on Feb 7, 2023</div>	
<div>Commits on Feb 9, 2023</div> <div><div>fix some bugs and add download</div><div>Roslin-v committed 5 days ago</div></div> <div><div>ukraine-round-trip database bug/bar chart done</div><div>alittle-f committed 5 days ago</div></div> <div><div>ukraine-russia-military-exp-linechart</div><div>alittle-f committed 5 days ago</div></div> <div><div>ukraine-civilian-casualties-bar</div><div>alittle-f committed 5 days ago</div></div> <div><div>delete line chart</div><div>alittle-f committed 5 days ago</div></div> <div><div>增加了留言在主页的展示</div><div>TephrocactusHC committed yesterday</div></div> <div><div>backend/views</div><div>alittle-f committed yesterday</div></div> <div><div>fixed some bugs</div><div>Roslin-v committed yesterday</div></div> <div><div>嘶，还在尝试修改</div><div>TephrocactusHC committed yesterday</div></div> <div><div>更改了about和加了简陋版的留言</div><div>TephrocactusHC committed yesterday</div></div>	

代码框架及目录展示

项目主要架构

```
1  .
2  |— backend
3  |   |— assets
4  |   |— config
5  |   |— controllers
6  |   |— models
7  |   |— runtime
8  |   |— tests
9  |   |— views
10 |   |— web
11 |   |— codeception.yml
12 |   └─ Dockerfile
13 |— common
14 |— console
15 |— data
16 |   |— personal
17 |   |   |— 2011911-果丁
18 |   |   |— 2012026-穆禹宸
19 |   |   |— 2012885-陈静怡
20 |   |   └─ 2013743-林巧莹
21 |   |— team
22 |   |   |— Untitled_设计文档
23 |   |   |— Untitled_部署文档
24 |   |   |— Untitled_需求文档
25 |   |   |— Untitled_实现文档
26 |   |   └─ Untitled_用户手册
27 |   └─ install.sql
28 |— environments
29 |— frontend
30 |   |— assets
31 |   |— config
32 |   |— controllers
33 |   |— models
34 |   |— runtime
35 |   |— tests
36 |   |— views
37 |   |— web
38 |   |— codeception.yml
39 |   └─ Dockerfile
40 |— uploads
41 └─ vagrant
```

还包含以下文件：

-  .bowerrc
-  .gitignore
-  codeception.yml
-  composer.json
-  composer.lock
-  docker-compose.yml
-  init
-  init.bat
-  LICENSE.md
-  README.md
-  requirements.php
-  Vagrantfile
-  yii.bat

前台MVC架构

```
1 |— frontend
2 |   |— assets
3 |   |— config
4 |   |— controllers
5 |       |— AidController.php
6 |       |— CommentController.php
7 |       |— MigrationController.php
8 |       |— MilitaryCompareTotalController.php
9 |       |— RUMilitaryComparisonController.php
10 |      |— RussiaLossController.php
11 |      |— SanctionCountryController.php
12 |      |— SanctionDateController.php
13 |      |— SanctionTypeController.php
14 |      |— SeStatusController.php
15 |      |— SiteController.php
16 |      |— UkraineCivilianCasualtiesController.php
17 |      |— UkraineRoundTripController.php
18 |      |— UkraineRussiaMilitaryExpenditureController.php
19 |   |— models
20 |       |— Aid.php
21 |       |— Comment.php
22 |       |— CommentSearch.php
23 |       |— ContactForm.php
24 |       |— Migration.php
```

```

25 | | └─ MigrationSearch.php
26 | | └─ MilitaryCompareTotal.php
27 | | └─ PasswordResetRequestForm.php
28 | | └─ ResendVerificationEmail.php
29 | | └─ ResetPasswordForm.php
30 | | └─ RUMilitaryComparison.php
31 | | └─ RussiaLoss.php
32 | | └─ SanctionCountry.php
33 | | └─ SanctionDate.php
34 | | └─ SanctionType.php
35 | | └─ SeStatus.php
36 | | └─ SignUpForm.php
37 | | └─ UkraineCivilianCasualties.php
38 | | └─ UkraineRoundTrip.php
39 | | └─ UkraineRussiaMilitaryExpenditure.php
40 | | └─ VerifyEmailForm.php
41 | └─ runtime
42 | └─ tests
43 | └─ views
44 | | └─ aid
45 | | └─ comment
46 | | └─ layouts
47 | | └─ migration
48 | | └─ military-compare-total
49 | | └─ r-u-military-comparison
50 | | └─ russia-loss
51 | | └─ sanction-country
52 | | └─ sanction-date
53 | | └─ sanction-type
54 | | └─ se-status
55 | | └─ site
56 | | └─ ukraine-civilian-casualties
57 | | └─ ukraine-round-trip
58 | | └─ ukraine-russia-military-expenditure
59 | └─ web
60 | └─ codeception.yml
61 | └─ Dockerfile

```

后台MVC架构

```

1 | └─ backend
2 |   └─ assets
3 |   └─ config
4 |   └─ controllers
5 |   └─ AidController.php

```

6			— AllCountriesController.php
7			— CommentController.php
8			— MigrationController.php
9			— MilitaryCompareTotalController.php
10			— RUMilitaryComparisonController.php
11			— RussiaLossController.php
12			— SanctionCountryController.php
13			— SanctionDateController.php
14			— SanctionTypeController.php
15			— SeStatusController.php
16			— SiteController.php
17			— UkraineCivilianCasualtiesController.php
18			— UkraineRoundTripController.php
19			— UkraineRussiaMilitaryExpenditureController.php
20			— models
21			— Aid.php
22			— AidSearch.php
23			— AllCountries.php
24			— AllCountriesSearach.php
25			— Comment.php
26			— CommentSearch.php
27			— ContactForm.php
28			— Migration.php
29			— MigrationSearch.php
30			— MilitaryCompareTotal.php
31			— MilitaryCompareTotalSearch.php
32			— PasswordResetRequestForm.php
33			— ResendVerificationEmail.php
34			— ResetPasswordForm.php
35			— RUMilitaryComparison.php
36			— RUMilitaryComparisonSearch.php
37			— RussiaLoss.php
38			— RussiaLossSearch.php
39			— SanctionCountry.php
40			— SanctionCountrySearach.php
41			— SanctionDate.php
42			— SanctionDateSearach.php
43			— SanctionType.php
44			— SanctionTypeSearch.php
45			— SeStatus.php
46			— SeStatusSearch.php
47			— SignUpForm.php
48			— UkraineCivilianCasualties.php
49			— UkraineCivilianCasualtiesSearch.php
50			— UkraineRoundTrip.php
51			— UkraineRoundTripSearch.php
52			— UkraineRussiaMilitaryExpenditure.php

```
53 | └─ UkraineRussiaMilitaryExpenditureSearch.php
54 | └─ runtime
55 | └─ tests
56 | └─ views
57 |   └─ aid
58 |   └─ all-countries
59 |   └─ comment
60 |   └─ layouts
61 |   └─ migration
62 |   └─ military-compare-total
63 |   └─ r-u-military-comparison
64 |   └─ russia-loss
65 |   └─ sanction-country
66 |   └─ sanction-date
67 |   └─ sanction-type
68 |   └─ se-status
69 |   └─ site
70 |   └─ ukraine-civilian-casualties
71 |   └─ ukraine-round-trip
72 |   └─ ukraine-russia-military-expenditure
73 | └─ web
74 | └─ codeception.yml
75 | └─ Dockerfile
```

主要功能及实现思路

管理员注册/登录

管理员注册/登录的功能借鉴了yii advanced模板自带的用户注册/登录的功能。我们添加了admin表，以与普通用户区分开来。只有管理员才能够登录后台，普通用户不能登录后台。

权限控制

```
1 // SiteController中的权限控制
2 'access' => [
3     'class' => AccessControl::class,
4     'only' => ['logout', 'signup', 'upload'],
5     'rules' => [
6         [
7             'actions' => ['logout', 'signup', 'upload'],
8             'allow' => true,
9             'roles' => ['@'],
10        ],
11    ],
12 ],
```



```

13 // 其它Controller中的权限控制
14 'access' => [
15     'class' => AccessControl::className(),
16     'only' => ['create', 'update', 'delete'],
17     'rules' => [
18         [
19             'allow' => true,
20             'actions' => ['create', 'update', 'delete'],
21             'roles' => ['@'],
22         ],
23     ],
24 ],

```

权限控制借用AccessControl类实现，后台管理员必须登录后才能修改数据库、上传文件。出于安全考虑，我们设置了一个默认管理员，管理员名字为admin，密码为myadminadmin。已有的管理员可以为其他新的管理员注册账号，而不像前台那样所有访问网页的人都可以注册账号。

用户注册/登录

权限控制

```

1 // SiteController中的权限控制
2 'access' => [
3     'class' => AccessControl::class,
4     'only' => ['logout', 'signup', 'download'],
5     'rules' => [
6         [
7             'actions' => ['signup'],
8             'allow' => true,
9             'roles' => ['?'],
10        ],
11        [
12            'actions' => ['logout', 'download'],
13            'allow' => true,
14            'roles' => ['@'],
15        ],
16    ],
17 ],
18 // CommentController中的权限控制
19 'access' => [
20     'class' => AccessControl::className(),
21     'only' => ['create'],
22     'rules' => [
23         [
24             'allow' => true,

```

```

25         'actions' => ['create'],
26         'roles' => ['@'],
27     ],
28 ],
29 ],

```

前台的用户都可以进行注册、查看可视化数据、查看网友评论，但只有登录的用户才可以发表评论、下载文件。

导航栏

前台与后台导航栏原理相同，此处仅举一个例子（外交数据→援助数据/制裁数据→6个界面）

```

1 use yii\bootstrap5\Nav;
2 use yii\bootstrap5\NavBar;
3 use kartik\bs5dropdown\Dropdown;
4
5 <header>
6     <?php
7         NavBar::begin([
8             'brandLabel' => Yii::$app->name,
9             'brandUrl' => Yii::$app->homeUrl,
10            'options' => [
11                'class' => 'navbar navbar-expand-md navbar-dark fixed-top shadow-lg
12            ],
13        ]);
14        $menuItems = [
15            'label' => '外交数据',
16            'items' => [
17                [
18                    'label' => '援助数据',
19                    'items' => [
20                        ['label' => '各国对乌克兰的援助(国家)', 'url' => ['/aid/cour
21                        ['label' => '各国对乌克兰的援助(时间)', 'url' => ['/aid/date
22                        ['label' => '各国对乌克兰的援助(类型)', 'url' => ['/aid/type
23                    ]
24                ],
25            ],
26            [
27                'label' => '制裁数据',
28                'items' => [
29                    ['label' => '各国对俄罗斯的制裁(国家)', 'url' => ['/sanctior
30                    ['label' => '各国对俄罗斯的制裁(时间)', 'url' => ['/sanctior
31                    ['label' => '各国对俄罗斯的制裁(类型)', 'url' => ['/sanctior
32            ]
33        ]

```

```

33     ]
34 ];
35 echo Nav::widget([
36     'options' => ['class' => 'navbar-nav me-auto mb-2 mb-md-0'],
37     'dropdownClass' => Dropdown::classname(),
38     'items' => $menuItems,
39 ]);
40 if (Yii::$app->user->isGuest) {
41     echo Html::tag('div',Html::a('登录',['/site/login'],['class' => ['btn btn-
42 } else {
43     echo Html::beginForm(['/site/logout'], 'post', ['class' => 'd-flex'])
44         . Html::submitButton(
45             '登出 (' . Yii::$app->user->identity->adminname . ')',
46             ['class' => 'btn btn-link logout text-decoration-none']
47         )
48         . Html::endForm();
49 }
50 NavBar::end();
51 ?>
52 </header>

```

本项目中导航栏主要运用下拉列表进行展示与跳转。其与附带的Bootstrap下拉JavaScript插件交互，是通过点击来切换的（并非悬停）。

运用HTML中 `Menuitems`元素 / `items` 快速创建，并进行分类的嵌套。

注：不同版本提供的功能不同。

可互动世界地图

基于d3.js模板实现，首先定义四个按钮用于选择查看四张地图。

```

1 <div class="d-flex align-items-center flex-column">
2     <div class="card m-3">
3         <ul class="nav nav-pills">
4             <li class="nav-item">
5                 <a class="nav-link" id="map-refugee" href="#">难民地图</a>
6             </li>
7             <li class="nav-item">
8                 <a class="nav-link" id="map-sanction" href="#">制裁地图</a>
9             </li>
10            <li class="nav-item">
11                <a class="nav-link" id="map-aid" href="#">援助地图</a>
12            </li>
13            <li class="nav-item">
14                <a class="nav-link" id="map-cost" href="#">花销地图</a>

```

```

15     </li>
16   </ul>
17 </div>
18   <svg id="my_dataviz" width="800" height="450"></svg>
19 </div>

```

然后将地图必要的数据传到view层，由上至下分别为：国家代码、难民数量、花销、是否援助、是否制裁

```

1 worldData = [
2   <?php foreach ($countries as $country): ?>
3     {'code': "<?=" . Html::encode("{ $country->code}") . ">",
4      'number_of_refugees': <?=" . Html::encode("{ $country->number_of_refugees}") . ">",
5      'cost': <?=" . Html::encode("{ $country->total_cost_in_billion_euros}") . ">",
6      'aid': <?=" . Html::encode("{ $country->aid}") . ">",
7      'sanction': <?=" . Html::encode("{ $country->sanction}") . ">",
8     },
9   <?php endforeach; ?>
10 ];

```

之后定义四个WordMap函数，对于前两种浮点型数据，还要手动设置好数据大小的分界线。在这里仅展示花销的写法。

```

1 function costWorldMap() {
2   svg.selectAll("*").remove();
3   data = d3.map();
4   legendText = "花销 （百万欧元） ";
5   mode = 0;
6   for (const d of worldData) {
7     data.set(d['code'], d['cost']);
8   }
9   colorScale = d3.scaleThreshold()
10     .domain([0.1, 0.2, 0.3, 0.5, 0.6, 0.7, 1, 5, 8])
11     .range(d3.schemeGreens[9]);
12   // Load external data and boot
13   d3.queue()
14     .defer(d3.json, "/world.geojson")
15     .await(ready);
16
17   // Add clickable background
18   svg.append("rect")
19     .attr("class", "background")
20     .attr("width", width)

```

```
21     .attr("height", height)
22     .on("click", click);
23 }
```

默认显示的是难民地图，当四个按钮中的某一个被点击，则执行绘制相应地图的函数，清空地图导航栏样式，然后为当前按钮设置阴影。

```
1  refugeeWorldMap();
2  document.getElementById("map-refugee").className = "nav-link active";
3
4  function resetMapNavStyles() {
5      document.getElementById("map-refugee").className = "nav-link";
6      document.getElementById("map-cost").className = "nav-link";
7      document.getElementById("map-aid").className = "nav-link";
8      document.getElementById("map-sanction").className = "nav-link";
9  }
10
11 document.getElementById("map-refugee").addEventListener("click", () => {
12     refugeeWorldMap();
13     resetMapNavStyles();
14     document.getElementById("map-refugee").className = "nav-link active";
15 });
16
17 document.getElementById("map-cost").addEventListener("click", () => {
18     costWorldMap();
19     resetMapNavStyles();
20     document.getElementById("map-cost").className = "nav-link active";
21 });
22
23 document.getElementById("map-aid").addEventListener("click", () => {
24     aidWorldMap();
25     resetMapNavStyles();
26     document.getElementById("map-aid").className = "nav-link active";
27 });
28
29 document.getElementById("map-sanction").addEventListener("click", () => {
30     sanctionWorldMap();
31     resetMapNavStyles();
32     document.getElementById("map-sanction").className = "nav-link active";
33 });
```

最后修改图例和鼠标悬停显示的内容，框架默认显示的是浮点类型数据，如果是后两种布尔类型数据，则要单独判断。

```

1 .text(function(d, i) {
2     if (mode == 0) {
3         if (i === 0) return "< " + d[1];
4         if (d[1] < d[0]) return d[0];
5         return d[0] + " - " + d[1];
6     } else {
7         if (i === 0) return "否";
8         return "是";
9     }
10 });

```

新闻线展示

按时间顺序查询数据库中的新闻数据表

```

1 /**
2  * Displays homepage.
3  *
4  * @return mixed
5  */
6 public function actionIndex()
7 {
8     $query = Article::find()->orderBy('date');
9     $articles = array_reverse($query->all(), true);
10    return $this->render('index', [
11        'articles' => $articles
12    ]);
13 }

```

在主页遍历每条数据，以时间+新闻标题作为时间线的标题，同时在时间线上展示简介。时间线的样式在timeline.css中定义。

```

1 <div class="row d-flex">
2     <div class="mt-5 card bg-light border-light p-5" style="border-radius: 50px;
3         <h2>新闻动态</h2>
4         <div class="timeline w-100">
5             <div class="timeline-outer">
6                 <?php foreach ($articles as $article): ?>
7                     <div class="timeline-card">
8                         <div class="timeline-info">
9                             <h3 class="timeline-title">
10                                 <?= Html::a("{ $article->date } { $article->title }", ["/si
11                                 </h3>

```

```
12             <?= $article->summary ?>
13         </div>
14     </div>
15     <?php endforeach; ?>
16 </div>
17 </div>
18 </div>
19 </div>
```

前台可视化数据展示

为了方便前台用户查看，我们将数据库的内容进行了处理，并通过可视化的方式展示。

aid表中存储了各个国家对乌克兰的援助，属性有国家、日期、类型、价值等。我们希望分别按照国家、日期和类型统计对乌克兰的援助价值。接下来我们以aid表的country页面为例，介绍如何实现前台可视化数据展示。

Model 模型支持

我们需要为aid表创建一个模型。Active Record 提供了一个面向对象的接口，用以访问和操作数据库中的数据。Active Record 类与数据库表关联，Active Record 实例对应于该表的一行，Active Record 实例的属性表示该行中特定列的值。因此，我们只需要创建Aid模型，继承ActiveRecord类即可。

Controller 数据处理

第一，我们要根据展示的需求对数据库的数据进行处理，如筛选、统计等。

```
1 $model = Aid::find()
2     ->select('country, sum(value) as sum')
3     ->groupBy('country')
4     ->having('sum IS NOT NULL')
5     ->orderBy('country ASC')
6     ->asArray()->all();
```

在本例中，我们通过find函数获取aid表中的所有记录，然后再进行筛选和统计。

第二，我们要将获取的数据由行变为列。这是因为我们实现可视化借助的是echart，而echart需要按列读取数据。

```
1 $country = ArrayHelper::getColumn($model, 'country');
2 $value = ArrayHelper::getColumn($model, 'sum');
```

这个功能我们通过ArrayHelper类的getColumn函数实现。

第三，我们要将获取的按列读取的顺序传递到view层，交给对应的界面进行渲染。

```
1 return $this->render('country', [  
2     'country' => $country,  
3     'value' => $value,  
4 ]);
```

View 可视化效果

本项目中使用了Echarts进行数据可视化。

```
1 <div class="aid-country-bar">  
2     <?= ECharts::widget([  
3         'responsive' => true,  
4         'options' => [  
5             'style' => 'height: 1300px;  
6         ],  
7         'pluginEvents' => [  
8             'click' => [  
9                 new JsExpression('function (params) {console.log(params)}'),  
10                new JsExpression('function (params) {console.log("ok")}'),  
11            ],  
12            'legendselectchanged' => new JsExpression('function (params) {console.lo  
13        ],  
14        'pluginOptions' => [  
15            'option' => [  
16                'title' => [  
17                    'text' => 'National assistance to Ukraine'  
18                ],  
19                'grid' => [  
20                    'left' => '3%',  
21                    'right' => '4%',  
22                    'bottom' => '3%',  
23                    'containLabel' => true  
24                ],  
25                'tooltip' => [  
26                    'trigger' => 'axis',  
27                    'axisPointer' => [  
28                        'type' => 'shadow'  
29                    ],  
30                ],  
31                'toolbox' => [  
32                    'feature' => [  

```



```

33         'dataView' => [
34             'show' => true,
35             'readOnly' => true
36         ],
37         //'magicType' => [
38             //'show' => true,
39             //'type' => ['line', 'bar', 'pie', 'tied', 'stack']
40         //],
41         'restore' => [
42             'show' => true
43         ],
44         'saveAsImage' => [
45             'show' => true
46         ],
47         'dataZoom' => [
48             'show' => true
49         ]
50     ],
51 ],
52 'yAxis' => [
53     'type' => 'category',
54     'data' => $country
55     //'axisLabel' => ['interval' => 0]
56 ],
57 'xAxis' => [
58     'scale' => true,
59     'type' => 'value',
60     'boundaryGap' => [0,0.01]
61 ],
62 'series' => [
63     [
64         //'data' => ['{{$searchModel[0]}}', '{{$searchModel[1]}}', '{{$sear
65         'data' => $value,
66         'type' => 'bar',
67         //'barGap' => '50%'
68     ]
69 ]
70 ]
71 ]
72 }); ?>
73 </div>

```

我们使用controller层传过来的数据（列传输），按照需求对横纵坐标进行赋值。例如本图表中使用了controller层传来的 `$country` 和 `$value` 对data进行赋值。

使用了Echarts进行图表制作。

- title: 设定标题
- grid: 设置布局
- tooltip: 光标悬停显示具体数据
- toolbox: 工具栏
 - dataView: 查看源数据
 - magicType: 图表类型转换
 - restore: 重新生成图表
 - saveAsImage: 以图片形式保存图表
 - dataZoom: 图表局部放大/缩小
- xAxis&yAxis: 设置横坐标/纵坐标
- series: 数据

后台数据库管理

后台管理员登录后，可以对数据库的内容进行增删改查。在后台网页中我们不需要再对数据进行可视化处理，只需要以表格的形式展示给管理员即可。我们同样以aid表为例，介绍如何实现后台数据库管理。

Model 模型支持

为了实现增删改查，我们不能像前台那样仅有一个简单的Aid模型，还需要创建一个AidSearch模型继承Aid，以便实现数据库的查找。

```
1 public function search($params)
2 {
3     $query = Aid::find();
4
5     // add conditions that should always apply here
6
7     $dataProvider = new ActiveDataProvider([
8         'query' => $query,
9     ]);
10
11     $this->load($params);
12
13     if (!$this->validate()) {
14         // uncomment the following line if you do not want to return any records
15         // $query->where('0=1');
16         return $dataProvider;
17     }
18 }
```

```

19 // grid filtering conditions
20 $query->andWhere([
21     'date' => $this->date,
22     'value' => $this->value,
23 ]);
24
25 $query->andWhere(['like', 'id', $this->id])
26     ->andWhere(['like', 'country', $this->country])
27     ->andWhere(['like', 'type', $this->type]);
28
29 return $dataProvider;
30 }

```

Controller 数据处理

对于主界面，我们只需要以表格形式展示数据库数据。因此，我们将通过刚刚创建的AidSearch类以及search方法，将模型和数据传递给view层。

```

1 public function actionIndex()
2 {
3     $searchModel = new AidSearch();
4     $dataProvider = $searchModel->search($this->request->queryParams);
5
6     return $this->render('index', [
7         'searchModel' => $searchModel,
8         'dataProvider' => $dataProvider,
9     ]);
10 }

```

“增”的实现需要先声明一个Aid的对象，相当于aid表的一条记录。管理员在view层通过POST方法提交了数据后，我们调用模型的load方法获取用户提交的数据，并通过save方法将新的记录插入到aid表中。若这些操作都成功，页面将重定向并展示管理员刚刚创建的那条记录。若这些操作失败，将返回create界面的默认状态。

```

1 public function actionCreate()
2 {
3     $model = new Aid();
4
5     if ($this->request->isPost) {
6         if ($model->load($this->request->post()) && $model->save()) {
7             return $this->redirect(['view', 'id' => $model->id]);
8         }
9     } else {

```

```

10         $model->loadDefaultValues();
11     }
12
13     return $this->render('create', [
14         'model' => $model,
15     ]);
16 }

```

“删”的实现较为简单，先调用findModel方法，通过主键id找到对应的记录，然后调用delete方法将其删除即可。

```

1 public function actionDelete($id)
2 {
3     $this->findModel($id)->delete();
4
5     return $this->redirect(['index']);
6 }

```

“改”的实现与“增”类似，在此不再赘述。

View 可视化效果

首先，需要对数据表形成对应的表单。

```

1 <div class="aid-form">
2     <?php $form = ActiveForm::begin(); ?>
3     <?= $form->field($model, 'id')->textInput(['maxlength' => true]) ?>
4     <?= $form->field($model, 'country')->textInput(['maxlength' => true]) ?>
5     <?= $form->field($model, 'date')->textInput() ?>
6     <?= $form->field($model, 'type')->textInput(['maxlength' => true]) ?>
7     <?= $form->field($model, 'value')->textInput() ?>
8     <div class="form-group">
9         <?= Html::submitButton('Save', ['class' => 'btn btn-success']) ?>
10    </div>
11    <?php ActiveForm::end(); ?>
12 </div>

```

这里我们使用了ActiveForm部件去生成HTML表单，。其中的 `begin()` 和 `end()` 分别用来渲染表单的开始和关闭标签。在这两个方法之间使用了 `field()` 方法去创建输入框。之后使用 `yii\helpers\Html::submitButton()` 方法生成提交按钮。

然后，我们使用 GridView 控件，对所得到的表单进行格式化视图展示。GridView 控件一次可以从数据源中获取大量的数据，并且通过表格的形式将数据显示出来。

```
1 <?= GridView::widget([
2     'dataProvider' => $dataProvider,
3     'filterModel' => $searchModel,
4     'columns' => [
5         ['class' => 'yii\grid\SerialColumn'],
6
7         'id',
8         'country',
9         'date',
10        'type',
11        'value',
12        [
13            'class' => ActionColumn::className(),
14            'urlCreator' => function ($action, Aid $model, $key, $index, $co
15                return Url::toRoute([$action, 'id' => $model->id]);
16            }
17        ],
18    ],
19    ]); ?>
```

对于创建、更新、删除操作，大致类似，这里仅以创建说明：

直接通过 render 视图进行渲染，然后填入所需的内容。

```
1 <div class="aid-create">
2
3     <h1><?= Html::encode($this->title) ?></h1>
4
5     <?= $this->render('_form', [
6         'model' => $model,
7     ]) ?>
8
9 </div>
```

当然，我们还需要一个 view.php 进行更新和删除等属性，以实现对应功能的点击跳转。这里我们使用了 DetailsView 控件，DetailsView 控件的主要功能是以表格形式实现和处理来自数据源的单条数据记录。

```
1 <div class="aid-view">
```

```

2
3 <h1><?= Html::encode($this->title) ?></h1>
4
5 <p>
6     <?= Html::a('Update', ['update', 'id' => $model->id], ['class' => 'btn b
7     <?= Html::a('Delete', ['delete', 'id' => $model->id], [
8         'class' => 'btn btn-danger',
9         'data' => [
10             'confirm' => 'Are you sure you want to delete this item?',
11             'method' => 'post',
12         ],
13     ]) ?>
14 </p>
15
16 <?= DetailView::widget([
17     'model' => $model,
18     'attributes' => [
19         'id',
20         'country',
21         'date',
22         'type',
23         'value',
24     ],
25 ]) ?>
26
27 </div>

```

上传/下载文件

我们希望实现管理员在后台上传文件，例如源数据、个人作业、团队作业等，用户在前台可以下载这些文件。为此，我们需要在数据库中创建files表，保存这些文件的信息。在此基础上，我们介绍如何上传/下载文件，并和数据库的files表相关联。

Model 模型支持

为了与files表关联，与前文提到的类似，我们需要为files表创建一个Files类继承ActiveRecord类。

对于上传文件，我们需要声明一个UploadForm模型。

在rules方法中，我们要规定一些重要信息，如允许上传的文件类型、一次最多允许上传的文件个数等。

```

1 public function rules()
2 {
3     return [
4         [['dataFiles'], 'file', 'skipOnEmpty' => false, 'extensions' => 'csv, xl

```

```
5     ];  
6 }
```

在upload方法中，我们对上传的文件进行验证，并将文件保存在固定的路径下，最后返回上传文件是否成功。

```
1 public function upload()  
2 {  
3     if ($this->validate()) {  
4         foreach ($this->dataFiles as $file) {  
5             $file->saveAs('../..../uploads/' . $file->baseName . '.' . $file->extension,  
6                 $file->baseName . '.' . $file->extension);  
7             return true;  
8         } else {  
9             return false;  
10        }  
11    }
```

Controller 数据处理

在SiteController中，我们加入actionUpload动作。当提交表单的时候，yii\web\UploadedFile::getInstance()方法就被调用，上传的文件用一个UploadFile实例表示。然后，我们依靠模型的验证规则确保上传的文件是有效的，在数据库中增加一行记录，并将文件保存在服务器上。

```
1 public function actionUpload()  
2 {  
3     $model = new UploadForm();  
4  
5     if (Yii::$app->request->isPost) {  
6         $model->dataFiles = UploadedFile::getInstances($model, 'dataFiles');  
7         if ($model->upload()) {  
8             // 文件上传成功，在数据库中增加一行记录  
9             foreach ($model->dataFiles as $file) {  
10                $files = new Files();  
11                $files->name = $file->baseName . '.' . $file->extension;  
12                $files->save();  
13            }  
14            return $this->goHome();  
15        }  
16    }  
17  
18    return $this->render('upload', ['model' => $model]);
```

在前台，用户可以看到所有的文件。与“后台数据库管理-Controller 数据处理”类似地，我们将 FilesSearch 模型和获取的全部数据传给 view 层。用户在界面中点击下载时，将触发 actionDownload 动作，并将用户点击对应的文件名传递给 controller 层。我们通过文件流的方式下载对应的文件，并在数据库中将其“被下载次数”加一。

```

1 public function actionDownload($fileName)
2 {
3     $filePath = '../../uploads/';
4     $saveAsFileName = $fileName;
5     $file = fopen($filePath . $fileName, "rb");
6     Header("Content-type: application/octet-stream ");
7     Header("Accept-Ranges: bytes ");
8     Header("Content-Disposition: attachment; filename= {$saveAsFileName}");
9     $contents = "";
10    while (!feof($file)) {
11        $contents .= fread($file, 8192);
12    }
13    echo $contents;
14    fclose($file);
15    // 在数据库中将“被下载次数”+1
16    $files = Files::findOne($fileName);
17    $files->updateCounters(['download_times' => 1]);
18 }

```

View 可视化效果

在 view 层实现下载界面可视化的文件 `files.php` 中使用了 `GridView` 控件进行显示。

```

1 <?= GridView::widget([
2     'dataProvider' => $dataProvider,
3     'filterModel' => $searchModel,
4     'columns' => [
5         [
6             'class' => 'yii\grid\ActionColumn',
7             'template' => '{view}',
8             'buttons' => ['view' => function ($url, $model, $key) {
9                 $url = Url::to(['/site/download', 'fileName' => $model->name]);
10                return Html::a('下载 &raquo;', $url);
11            }],
12         ],
13 ]

```



```

14         'name',
15         'download_times:integer',
16         [
17             'class' => ActionColumn::className(),
18             'urlCreator' => function ($action, Files $model, $key, $index, $column) {
19                 return Url::toRoute([$action, 'name' => $model->name]);
20             }
21         ],
22     ],
23
24 ];

```

通过接收controller层传来的数据，我们可以在前台下载界面查看到文件名称、下载次数以及下载按钮。

光标点击下载按钮后，会反馈到controller层触发相应动作，进行下载。同时，下载次数会在数据库中+1，view层会读取相应数据，利用控件反馈到图表中。

在view层实现上传界面可视化的文件 `upload.php` 中使用了进行显示。

```

1 <div class="p-5 mb-4 bg-transparent rounded-3">
2     <div class="container-fluid py-5">
3         <h1 class="display-4">文件上传界面 📁 </h1>
4     </div>
5     <div>
6         <p class="fs-5 fw-light">操作步骤：“选择文件”→“Submit”</p>
7         <p class="fs-5 fw-light">支持上传csv、xls、xlsx、zip、rar格式的文件。</p>
8         <p class="fs-5 fw-light">注意：默认文件大小不得超过2MB，可在php.ini文件中
9     </div>
10 </div>
11
12 <?php $form = ActiveForm::begin(['options' => ['enctype' => 'multipart/form-data']]);
13
14 <?= $form->field($model, 'dataFiles[]')->fileInput(['multiple' => true, 'accept' => 'image/*,audio/*,video/*,text/*,application/*']);
15
16 <div class="form-group">
17     <?= Html::submitButton('Submit', ['class' => 'btn btn-success']) ?>
18 </div>
19
20 <?php ActiveForm::end() ?>

```

首先利用Html语言进行了简单的页面说明。然后利用ActiveForm方法实现两个交互按钮：上传文件、提交。

在视图里创建一个文件输入控件。需要注意的是要记得在表单选项里加入 `enctype` 属性以确保文件能被正常上传。 `fileInput()` 方法会渲染一个 `<input type="file">` 标签，让用户可以选择一个文件上传。

留言板

我们希望提供一个留言板，支持用户对我们的网页进行评论，提出反馈，以帮助我们更好地完善网页。对于后台而言，留言板的管理和“后台数据库管理”小节介绍的类似，在此不再赘述。我们将介绍前台如何实现留言板功能。

Model 模型支持

我们为支持留言板功能，增加了 `comment` 表。为了与 `comment` 表关联，与前文提到的类似，我们需要为其创建一个 `Comment` 类继承 `ActiveRecord` 类。用户还应该能够创建新的留言、查找留言等，因此我们还要创建 `CommentSearch` 类继承 `Comment` 类，以更好地实现这些功能。

Controller 数据控制

用户创建新的留言，将触发 `actionCreate` 动作。和“后台数据库管理”的“增”功能不同的是，用户只需要提供留言的内容，不需要填写用户名和留言的时间，这些应该是控制器可以自动获取的。因此，我们声明一个 `Comment` 类对象，用户提交表单后，我们通过获取 `POST` 请求中的留言内容，并通过系统获取其用户名和留言的时间，最后将该对象（一条新记录）插入数据库，并重定向界面，展示用户刚刚发表的留言。

```
1 public function actionCreate()
2 {
3     $model = new Comment();
4
5     if ($model->load(Yii::$app->request->post())) {
6         $model->discuss = $_POST['Comment']['discuss'];
7         $model->username = Yii::$app->user->identity->username;
8         $model->comment_time = date('Y-m-d H-i-s', time());
9         if ($model->save()) {
10             return $this->redirect(['view', 'comment_id' => $model->comment_id])
11         }
12     }
13
14     return $this->render('create', [
15         'model' => $model,
16     ]);
17 }
```

View 可视化效果

留言板的可视化渲染分为两个界面。一个是“网友评论”界面，以时间线的方式展示所有的留言，具体实现和“新闻线展示”小节相同。另一个是“提出反馈”界面，以 `ActiveForm` 的形式，获取用户输

入的留言内容，与“后台数据库管理”小节类似。在此不再赘述。