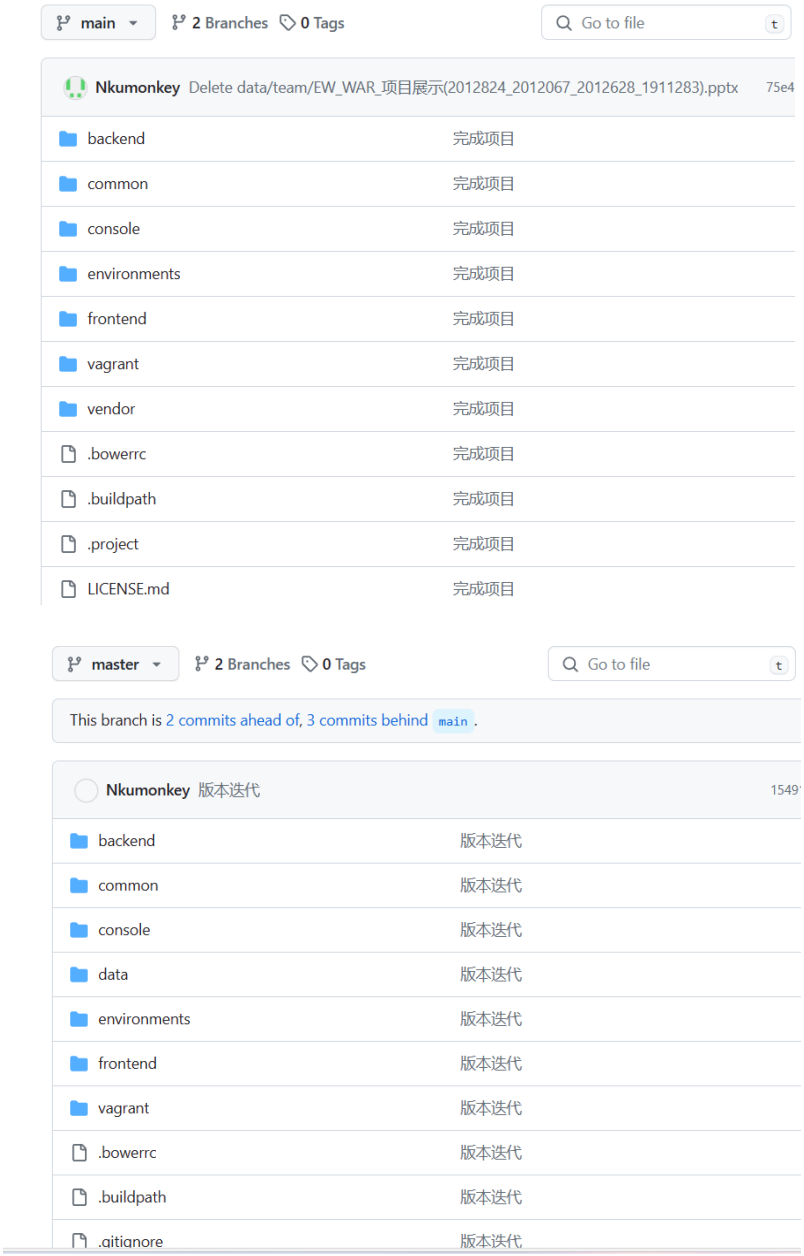
实现文档

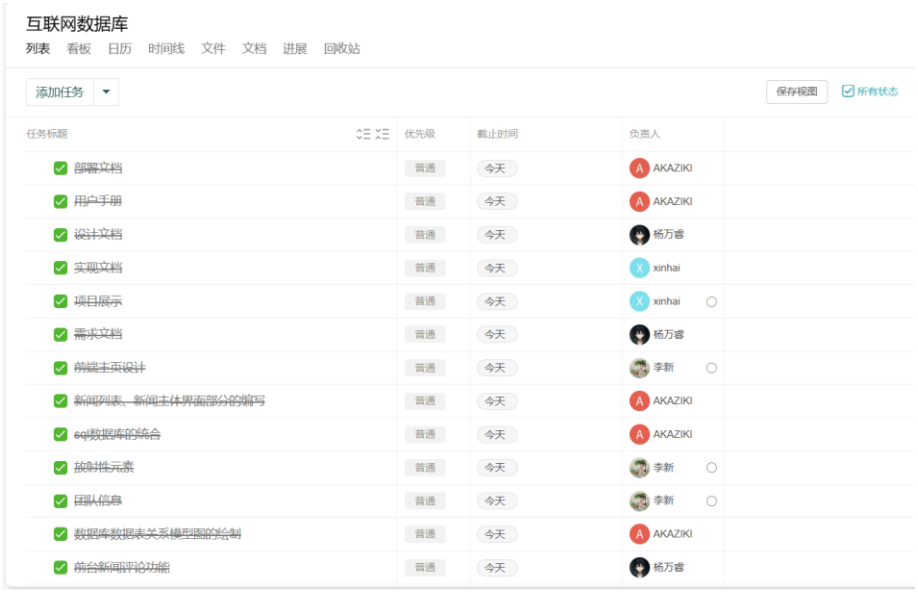
队伍名称：互联网团队

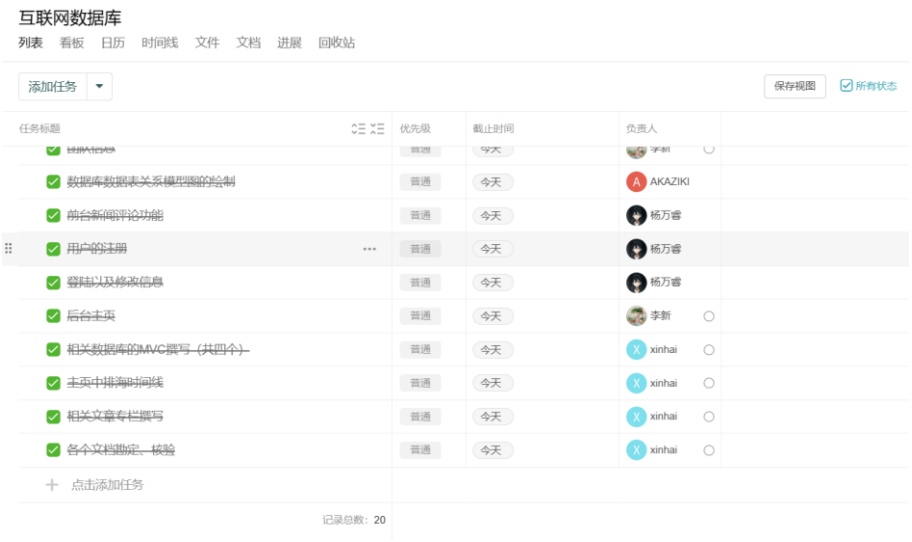
一、团队项目实施

我们选择了 Tower 作为项目管理工具，github 作为代码提交工具。



上图是版本的迭代示意图。





二、前台用户页面实现

2.1 网页模板导入

从 https://www.free-css.com/网站中下载对应的网页样式模板，本次使用的模板链接如下：https://www.free-css.com/free-css-templates/page282/leadmark，该部分将下载的压缩包解压后放入 frontend\web\assets\frontend 目录下，修改 frontend\view\layouts\main.php 文件，配置 css 和 js 文件，具体需要配置的文件可以在 css 模板的 index.html 中查看，需要注意的是，css 模板中 index.php 的资源配置与本次 Yii 框架网页中需要的配置路径不同，因为我们修改了 css 和 js 文件的位置。

为了 PHP 代码编写，我们需要在模板首页源代码外层加上<?php

$this->beginPage() ?><?php $this->endPage() ?>，在 html 的<body>标签外加上<?php $this->beginBody() ?><?php $this->endBody() ?>。

```
<?php $this->beginPage() ?>
    <!DOCTYPE html>
    <html lang="<?= Yii::$app->language ?>" class="h-100">
        <head>
```

```
<body data-spy="scroll" data-target=".navbar" data-offset="40" id="home">
<?php $this->beginBody() ?>

<div class="site-wrap">
    <!-- Page Header -->
    <header class="header">
```

css 配置主要在<body>标签前的<head>标签中实现，举例如下：

```
<!-- font icons -->
<link rel="stylesheet" href="assets/frontend/leadmark/public_html/assets/vendors/themify-icons/css/themify-icons.css">
<!-- Bootstrap + LeadMark main styles -->
<link rel="stylesheet" href="assets/frontend/leadmark/public_html/assets/css/leadmark.css">
```

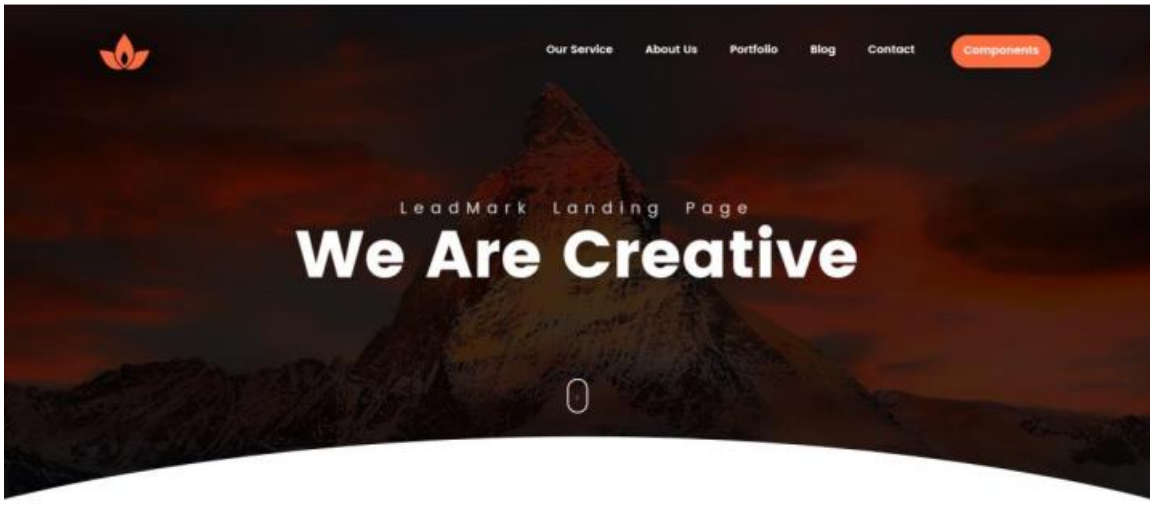
js 配置主要在<body>标签的末尾实现，举例如下：

```
<!-- Isotope -->
<script src="assets/frontend/leadmark/public_html/assets/vendors/isotope/isotope.pkgd.js"></script>

<!-- LeadMark js -->
<script src="assets/frontend/leadmark/public_html/assets/js/leadmark.js"></script>

<?php $this->endBody() ?>
```

模板示例截图如下：



2.2 前台主页

2.2.1 页眉部分

该部分主要在 frontend\view\layouts\main.php 文件中进行编辑，其中，“放射

性元素”和“最新文章”为页面内跳转，该部分通过 href="#id"和<section>标签中的 id 来实现页内跳转；

```
<li class="nav-item">
    <a class="nav-link" href="#portfolio">放射性元素</a>
</li>
<li class="nav-item">
    <a class="nav-link" href="#blog">最新文章</a>
</li>
<!-- <li class="nav-item">
    <a class="nav-link" href="#contact">Contact</a>
```

```
<!-- Portfolio Section -->
<section id="portfolio" class="section portfolio-section">
    <div class="container">
        <h6 class="section-title text-center">日本非法排放核污水大数据分析</h6>
        <h6 class="section-subtitle mb-5 text-center">The big data analysis of Jap
```

```
<!-- Blog Section -->
<section class="section" id="blog">
    <div class="container">
        <h6 class="section-title mb-0 text-center">最新文章</h6>
        <h6 class="section-subtitle mb-5 text-center">注：本网站为南开大学学
```

其余部分为页面跳转，该部分主要通过$menuItems[]变量实现，lable 属性用来显示文字，url 属性用来显示相对跳转地址，如"/site/about"表示相对跳转地址为 site 目录下的 about.php 文件。

```
$menuItems = [
    ['label' => '主页', 'url' => ['/site/index']],
    ['label' => '新闻', 'url' => ['/site/news']],
    ['label' => '关于', 'url' => ['/site/about']],
    //这里做文章，最后改为想打开的php文件名，然后controller里增添对应action某某
    ['label' => '排海时间线', 'url' => ['/usupport/index']],
];

if (Yii::$app->user->isGuest) {
    $menuItems[] = ['label' => '注册', 'url' => ['/site/signup']];
    $menuItems[] = ['label' => '登录', 'url' => ['/site/login']];
} else {
    $menuItems[] = ['label' => '修改信息', 'url' => ['/site/modify']];


    $menuItems[] = "          <li><span>"
        . Html::beginForm(['/site/logout'], 'post')
        . "<button class='btn btn-outline-secondary'>登出("
        . Yii::$app->user->identity->username . ')'
        . "</button>"
        . Html::endForm()
        . '</span></li>';

}
```

2.2.2 放射性元素部分

采用模板中的的案例，修改对应的图片及文字即可。其中，数据在数据库中存储，通过如下方式获取其中的数据：首先在 common/modules 中建立对应的 module 文件，

该部分主要文件为 Refugee.php 文件，代码如下：该部分类名与数据库中表名相同，通过 tableName 这一静态函数保证该 module 对应数据库中的数据表。

```php
class Refugee extends ActiveRecord
{
    public static function tableName()
    {
        return '{{%Refugee}}';
    }

    public function setNationality($nationality)
    {
        $this->nationality = $nationality;
    }

    public function setDestination($destination)
    {
        $this->destination = $destination;
    }

    public function setNum($num)
    {
        $this->num = $num;
    }
```

在使用时通过如下方式使用:使用 findOne 或者 findAll 函数查找对应记录使用。

① ECharts 导入南丁格尔图

该部分主要通过 ECharts 实现，通过 style 中的 height 和 width 调整大小，代码主体采用 ECharts 的案例代码修改获得，数据通过数据库获取，代码如下

```
<div class="col-md-6 col-lg-4 advertising">
<div id="container" style="height: 600%;width: 300%"></div>
    <script type="text/javascript" src="https://fastly.jsdelivr.net/npm/echarts@5.4.1/dist/echarts.min.js"></script>
    <script type="text/javascript">
        var dom = document.getElementById('container');
        var myChart = echarts.init(dom, null, {    "echarts": Unknown word.
        renderer: 'canvas',
        useDirtyRect: false
        });
        var app = {};

        var option;

        option = {
        legend: {
            top: 'bottom'
        },
        toolbox: {
            show: true,
            feature: {
            mark: { show: true },
            dataView: { show: true, readOnly: false },
            restore: { show: true },
            saveAsImage: { show: true }
            }
        },
```

```
series: [
    {
    name: 'Nightingale Chart',
    type: 'pie',
    radius: [50, 250],
    center: ['50%', '50%'],
    roseType: 'area',
    itemStyle: {
        borderRadius: 7
    },
    data: [
        { value: <?= $element_1->time ?>, name: '锶-90' },
        { value: <?= $element_2->time ?>, name: '锑-124' },
        { value: <?= $element_3->time ?>, name: ' 钴-58' },
        { value: <?= $element_4->time ?>, name: '铯-137' },
        { value: <?= $element_5->time ?>, name: '镍-63' },
        { value: <?= $element_6->time ?>, name: '氚' },
        { value: <?= $element_7->time ?>, name: '碳-14' },
        { value: <?= $element_8->time ?>, name: '铯-134' },
        { value: <?= $element_9->time ?>, name: '碲-129' }
    ]
    }
]
};

if (option && typeof option === 'object') {
myChart.setOption(option);
}

window.addEventListener('resize', myChart.resize);
```

② 视图部分

视图主要获取网页对应图片，通过 href 指定图片来源即可，这里不再赘述。

2.2.3 页脚部分

主页页脚在 index.php 文件中进行设置,分为“关于本站”、“相关网站”、“Send”和“返回顶部”几个部分，关于本站部分给出了一部分自己书写的 php 页面和框架中给定的 php 页面采用如下方式指定

```
<a href="<?= \yii\helpers\Url::to(['/site/index']); ?>">
```

相关网站则是给定的一些与日本排放核污水相关的网站，指定通过 href 指定；发送信息为本站原有代码，通过如下代码接收显示；最后是返回首页，也为页内跳转，与页眉部分类似，这里不再赘述。

```
<section class="banner_main">
    <div class="container-fluid">
        <div class="row d_flex">
            <div class="col-xl-6 col-lg-6 col-md-12">
                <div class="text-bg">
                    <span class="caption mb-3 d-block"></span>
                    <!--发送通知信息-->
                    <?php
                    if (isset($message))
                        echo "<h1 class=\"heading\">" . $message . "</h1>";
                    if(isset($_GET['message']))
                        echo "<h1 class=\"heading\">" . $_GET['message'] . "</h1>";
                    else echo "<h1 class=\"heading\"></h1>";
                    ?>
                </div>
            </div>
        </div>
    </div>
</section>
```

2.3 新闻页面

新闻页面主要由两部分页面组成：新闻列表页面与新闻内容界面。

2.3.1 新闻列表界面

新闻列表界面设计如下：

```
<style>
  #news-list {
    list-style-type: none;
    margin: 0 auto;
    padding: 0;
    width: 50%;
    background-color: #f1f1f1;
    border: 1px solid black;
    text-align: center;
  }
  #news-list li a {
    display: block;
    color: #000;
    padding: 8px 16px;
    text-decoration: none;
    font-size: 18px;
    font-weight: bold;
    text-align: center;
    border-bottom: 1px solid black;
  }
  #news-list li a:hover {
    background-color: #555;
    color: white;
  }
</style>
```

2.3.2 新闻主体界面

新闻主体界面设计如下：

请注意，为显示全页面图片对网页进行了缩放，请以实际网页大小为准。

新闻主体界面分为三个部分：标题部分、图片部分、正文部分。

标题部分包含新闻标题、新闻作者（即来源网站，下同）、新闻发布时间三部分。在页面的最上方为 36 号加粗白色居中字体，为新闻标题；新闻标题下起两行，分别为新闻作者、新闻发布时间，均为右对齐 20 号小字。具体实现代码如下：

```
.headerr {
  font-size: 36px;
  font-family: Songti;    "Songti": Unknown word.
  font-weight: bold;
  text-align: center;
  color: white;
}

.authorr {
  font-size: 20px;
  font-family: Songti;    "Songti": Unknown word.
  text-align: right;
  color: white;
}

.datee {    "datee": Unknown word.
  font-size: 20px;
  font-family: Songti;    "Songti": Unknown word.
  text-align: right;
  color: white;
}
```

图片部分包含一张与新闻有关的图片，大小为 35%界面宽度，锁定纵横比。具体实现代码如下：

```
.imagee {    "imagee": Unknown word.
  width: 35%;
  height: auto;
  object-fit: contain;
  margin: 20px auto;
  display: block;
}
```

正文部分包含新闻正文和新闻原 url 地址，字体为 15 号左对齐小字。具体实现代码如下：

```
.textt {    "textt": Unknown word.
  font-size: 15px;
  font-family: "WenQuanYi Zen Hei";    "Quan": Unknown word.
  line-height: 1.5;
  text-align: left;
  color: white;
  text-indent: 2em;
}
```

另外，还实现了 container 和 content，用来控制文本显示位置和显示宽度。在内容获取方面，除新闻正文部分为直接文本，其余均与数据库相连，从数据库中获取内容。与数据库连接，实现内容部分具体实现代码如下：

```
<body>
  <div class="containerr">
    <div class="headerr"><?= $site1->title ?></div>
    <div class="authorr"><?= $site1->author ?></div>
    <div class="datee"><?= $site1->release_date ?></div>    "datee": Unknown word.
    <img class="imagee" src="news1.jpg" alt="图片">    "imagee": Unknown word.
    <div class="textt">    "textt": Unknown word.
      <div class="contentt">    "contentt": Unknown word.
        <p>当地时间8月22日，日本首相岸田文雄宣布如果气象和水文条件允许，将在24日开始向海洋放出用以处理福岛第一反应堆爆炸后的核污染水。据日本时事通讯社报道，22日，自民党干事长茂木
        <p>据日本共同社报道，东京电力（东电）公司计划，从24日下午1时开始对福岛第一核电站的处理水进行海洋释放，将在17天内排放第一批共7800吨核污染水。整个排放过程预计长达30年。2023
        <p>根据东电的信息，日本排放的核污染水经过ALPS技术处理，能够确保除了氚（tritium）之外的放射性物质浓度远远低于相关的安全规范值。氚是一种氢的放射性同位素。尽管日本经济产业省
        <p>此外，核污染水影响范围涉及全球，是最令人担忧的问题。德国海洋科学研究机构指出，福岛沿岸拥有世界上最强的洋流，从排放之日起57天内，放射性物质将扩散至太平洋大半区域，3年后
        <p>我国生态环境部相关负责人24日就日本启动福岛核污染水排海问题表示，“我部高度重视日本福岛核污染水排海问题。2021年、2022年先后组织开展了我国管辖海域海洋辐射环境监测，摸清了
        <p>韩国最大在野党共同民主党党首李在明23日谴责日本即将进行的核污染水排放，称这是一种“恐怖行为”，指责日本在过去以帝国主义战争威胁邻国后，又试图给韩国和太平洋国家带来不可挽回

中国外交部发言人汪文斌也表示，日本政府不顾国际社会严重关切和坚决反对，执意宣布将于8月24日启动福岛核污染水排海，公然向全世界转嫁核污染风险，将一己私利凌驾于全人类长远福祉之上，此举极
        <p>在日本国内最受舆论关注的问题则是渔业问题和福岛以及本地居民的污名化问题。据TBS News报道，岸田文雄在正式公布核污染水放出决定后，也宣布邀请反对核污染水释放的日本全国渔业协

此外，岸田文雄也宣布将设立业务持续基金，以确保渔民能够安心继续进行渔业活动。他强调说，“为了解决与ALPS（Advanced Liquid Processing System，多核素处理系统）处理的核污染水处置相关的声

然而，对于渔业不可避免的影响也使得日本全国渔业协会联合会表达了对于核污染水放出计划的不满。据NHK报道，围绕着核污染水排放计划，日本经济产业大臣西村康稔也与坂本雅信进行了会面。在会谈中
        <p>事实上，日本全国渔业协会联合会一直是反对日本政府向海洋倾倒核污染水的重要力量。日本全国渔业协会联合会是能够通过合作共同解决渔民无法独自处理的问题和挑战，从而汇集力量的组
        <p>据NHK报道，2011年3月11日发生的东日本大地震后，福岛县的第一产业为了恢复销售渠道和出货量，进行了各种努力。然而，福岛产的大米、牛肉、鱼等产品仍然面临被低价购买的趋势。日本

据日本共同社报道，为了避免污名化问题，日本水产厅8月11日宣布将在在核污染水放出开始之际，决定加强福岛第一核电站周边水产产品的检测。目前，在福岛县沿海，除了福岛第一核电站10公里范围内外
        <p>url:<?= $site1->url ?></p>
      </div>
    </div>
  </div>
```

2.4 关于页面

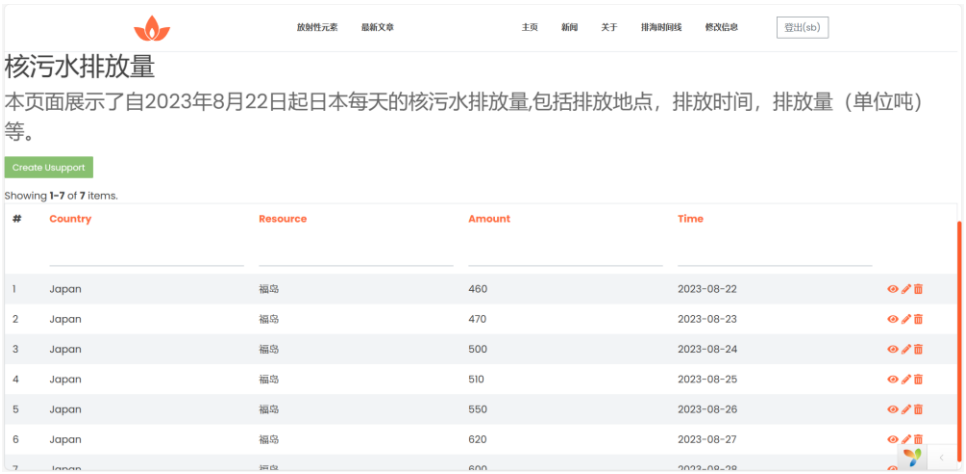
该部分主体为团队成员的信息显示，信息通过数据库获取，方式与主页中难民部分的数据获取和使用方式类似，这里不再赘述。为了使页面更美观，调整模板中的 class 参数来调整网页中团队成员所在框的大小，修改后代码如下所示：

```
<div class="col-md-6 my-3 my-md-0">…
</div>
```

2.5 援助页面

如“页眉部分节”，在$menuItems[]中添加了"/usupport/index.php"后，使用 gii 生成 vi 相应代码（具体步骤见下节）：

效果如下：



对 views/usupport 名/index.php 做出对应修改，可使跳转后页面显示不同文字、图片等。

```
<div class="usupport-index">

    <h1><?= Html::encode($this->title) ?></h1>
    <p>
        <font size="6">本页面展示了自2023年8月22日起日本每天的核污水排放量,包括排放地点，排放时间，排放量（单位吨）等。</font>
    </p>
    <p>
        <?= Html::a('Create Usupport', ['create'], ['class' => 'btn btn-success']) ?>
    </p>
```

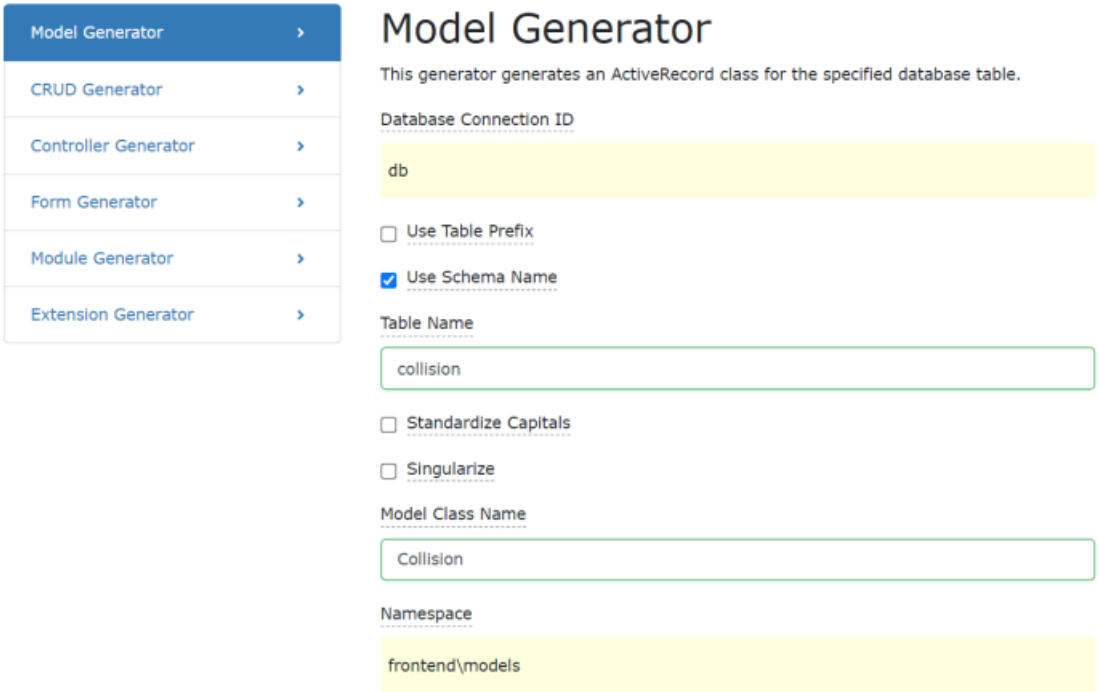
2.6 使用 gii 生成代码步骤

2.6.1 准备工作 Model Generator

①首先确保 MySQL 服务正常，且网页能连接数据库（通过 common/config/main-local.php 设置）。

②以...frontend/web 作为服务器根目录打开后，在网页输入?r=gii，点击 model generator 进入如下页面，在 table name 中选择数据库表名，使用该表名的首字母大写形式作为 Model Class Name。

③尤其注意下方的 namespace 选择，确定将 model 文件生成在 frontend 或 common 或 backend 对应的 model 文件夹下。（处于实现尽量简便的考量，将模型类生成在了 frontend 中，其实生成在 common 文件夹中可实现前后台共用）



2.6.2 Generate Model

其他选项保持默认不变即可，注意到所有的模型类都继承自\yii\db\ActiveRecord 类，ActiveRecord 类可以说是 yii 框架的精华核心所在。

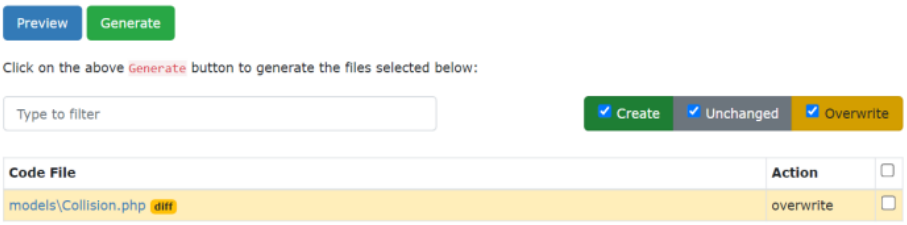
点击 preview，看到生成代码如下：

```php
class Collision extends \yii\db\ActiveRecord
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'collision';
    }

    /**
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['place', 'time'], 'required'],
            [['time'], 'safe'],
            [['rforce', 'uforce', 'rinjury', 'uinjury'], 'integer'],    "rforce": Unknown word.
            [['place'], 'string', 'max' => 255],
            [['place', 'time'], 'unique', 'targetAttribute' => ['place', 'time']],
        ];
    }
```

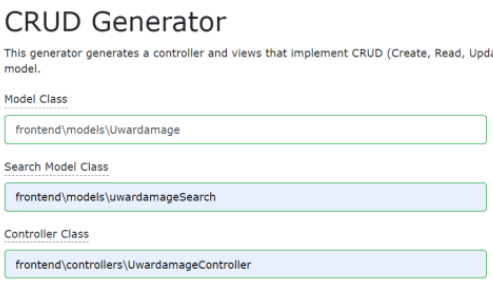
可以看到自动生成的代码较为简明规范，定义了表属性及相应结构，表具体的属性也在文件前以注释形式给出，为我们自己写代码提供了很好的参考。本文中战争相关的 model 均采用类似结构。

确认是否需要 overwrite 覆盖原有文件后，点击 generate



2.6.3 CRUD Generator

CRUD 即“增删改查”的首字母缩写，主要实现数据库的数据操作相关功能，并且也生成 view 类的代码，填写之前生成 model 类的位置，以及想要生成的 search model 类和 controller 类的位置（重要）。



2.6.4 generate controlller and view

同样确认无误后点击 preview->generate

生成的 controller 类大体代码如下：

```php
<?php

namespace frontend\controllers;

use frontend\models\collision;
use frontend\models\collisionSearch;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;

/**
 * CollisionController implements the CRUD actions for Collision model.
 */
class CollisionController extends Controller
{
    /**
     * @inheritDoc
     */
    public function behaviors()
    {
        return array_merge(
            parent::behaviors(),
            [
                'verbs' => [
                    'class' => VerbFilter::className(),
                    'actions' => [
                        'delete' => ['POST'],
                    ],
                ],
            ]
        );
    }
```

可以看到生成 controller 类需要先引入对应的 model 类和 modelSearch 类。

CRUD Generator 同时生成了 view 类，其中包含了每个操作模型的方法单独成文件（create、update 等），update 文件内容如下，负责在页面接受数据传递给模型：

```php
$this->title = 'Update Collision: ' . $model->place;
$this->params['breadcrumbs'][] = ['label' => 'Collisions', 'url' => ['index']];
$this->params['breadcrumbs'][] = ['label' => $model->place, 'url' => ['view', 'place' => $model->place, 'time' => $model->time]];
$this->params['breadcrumbs'][] = 'Update';
?>
<div class="collision-update">

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

    <?= $this->render('_form', [
        'model' => $model,
    ]) ?>

</div>
```

注意到视图 view 中每一个文件，在 controller 类里也有相应的 action 某某（actionCreate、actionUpdate 等），controller 类是连接 view 和 model 类的桥梁。

在 GridView 小部件是从数据提供者获取数据，并以一个表格的形式呈现数据。表中的每一行代表一个单独的数据项，列表示该项目的属性。可以使用 yii\grid\SerialColumn， yii\grid\CheckboxColumn 和 yii\grid\SerialColumn 进行定制样式。

2.7 最新文章部分

修改文章主页，基本都在\frontend\views\site\index.php

主要更改如下：

```
<div class="col-md-4">
    <div class="card border-0 mb-4">
        <img src="assets/frontend/leadmark/public_html/assets/imgs/blog-5.png" alt="" class="card-img-top w-100">
        <div class="card-body">
            <h6 class="card-title">blog2</h5>
            <p>我们的地球，我们的责任：反对日本非法核污水排放</p>
            <a href="index.php?r=rwardamage%2Findex" class="small text-muted">跳转到数据界面</a>    "rwardamage": Unkn
        </div>
    </div>
</div>
```

插入自己设定的图片，跳转的超链接为 gii 生成的 view 类下的 index.php 界面。之后跳转到的数据库页面实现思路与“援助“部分大体相同，不再赘述。

2.8 注册和登录

首先，需要在\common\models 中建立 User.php 来维护用户类，其代码如下(只展示核心代码)：

```
class User extends ActiveRecord implements IdentityInterface
{
    /*
     * @return 返回的是数据库中存储用户使用的表名
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return '{{%User}}';
    }

    /*
     * 通过身份证号寻找唯一的用户
     * 返回一个MyUser类
     */
    public static function findIdentity($account)
    {
        return static::findOne(['account' => $account]);
    }

    /*
     * 返回用户对象的主键
     */
    public function getId()
    {
        return $this->getPrimaryKey();
    }
```

2.8.1 注册

需要在\frontend\models 中建立 SignupForm 的模型，用来维护注册信息表单，SignupForm 类代码如下：

首先，声明注册表单需要的四个信息变量，即账号、密码、用户名以及电话号码

```php
class SignupForm extends Model
{
    public $account;
    public $password;
    public $username;
    public $tel;
```

接着，通过 rules 函数设置每个输入信息文本的规则，例如身份证号账号必须为 18 位且不能为空，用户名要在 1-10 位之间且非空等等，代码如下所示：

```php
public function signup()
{
    // if (!$this->validate()) {
    //     return null;
    // }

    $user = new User();
    $user->account = $this->account;
    $user->setPassword($this->password);
    $user->setUsername($this->username);
    $user->setTel($this->tel);
    $user->setType(0);
    $user->generateAuthKey();

    return $user->save();
}
```

```php
public function rules()
{
    return [
        [['account', 'name', 'tel'], 'trim'],
        ['account', 'required', 'message' => '账户不能为空'],
        ['account', 'unique', 'targetClass' => '\common\models\User', 'message' => '账户已存在'],
        ['account', 'string', 'min' => 18, 'max' => 18, 'tooShort' => '账户应为18位身份证号', 'tooLong' => '账户应为18位身份证号'],

        ['username', 'trim'],
        ['username', 'required', 'message' => '用户名不能为空'],
        ['username', 'string', 'min' => 1, 'max' => 10, 'tooShort' => '用户名的长度必须在1到10之间', 'tooLong' => '用户名的长度必须在1到10之间'],

        ['password', 'required', 'message' => '密码不能为空'],
        ['password', 'string', 'min' => 4, 'tooShort' => "密码长度必须大于等于4位"],

        ['tel', 'required', 'message' => '联系方式不能为空'],
        ['tel', 'number', 'min' => 10000000000, 'max' => 19999999999, 'message' => "请填写正确的联系方式", 'tooBig'=>"请填写正确的联系方式",'tooSmall'=>"请填写正确的联系
    ];
}
```

即，先新建一个注册表单类，如果注册成功，则返回首页；否则就渲染一个名为‘signup’的视图，并将该视图的 model 设置为当前的 model，因此，我们需要在视图模块中(\frontend\views)创建 signup.php 作为注册页面的视图，下面是主题部分的代码：

```php
<div class="row justify-content-center">
    <div class="col-lg-5">
        <?php $form = ActiveForm::begin(['id' => 'form-signup']); ?>

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



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

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

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




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

        <?php ActiveForm::end(); ?>
    </div>
</div>
</div>
```

可以看到，这里通过 ActiveForm 的类方法建立文本输入框和提交按钮，文本输入框通过调用 field 方法，对模型 model(也就是前面 SiteController 中新建立的 SignupForm)的相应属性进行填写。其中 textInput 是普通的可见的输入，passwordInput 是将输入内容设置为不可见的形式。提交按钮通过 Html::submitButton 进行创建即可。

2.8.2 登录

登陆的实现流程和注册基本一致，除了以下的几点不同：首先是登陆时只需要输入账号和密码，因此对应的 LoginForm 类只需要设置账号和密码作为其成员变量即可。其 SiteController 中的 action 函数代码如下：

```php
public function actionLogin()
{
    if (!Yii::$app->user->isGuest) {
        return $this->goHome();
    }

    $this->layout = 'blank';

    $model = new LoginForm();
    if ($model->load(Yii::$app->request->post()) && $model->login()) {
        return $this->goBack();
    }

    $model->password = '';

    return $this->render('login', [
        'model' => $model,
    ]);
}
```

即，如果已经登录，就返回主页。否则新建一个注册表单类，如果注册成功则回到主页，否则停留在所渲染的 login 视图页面，此时密码默认为空。

2.9 修改用户信息

与注册和登陆类似，也需要在 models 中建立 ModifyForm.php 用于维护修改信息的表单类，其中的成员变量包括新的密码、新的用户名等要修改的信息。在 SiteController 中实现如下 action 方法：

```php
/**
 *  重置信息的动作，用于修改当前的密码或者用户名
 */
public function actionModify()
{
    if (Yii::$app->user->isGuest) {
        return $this->goHome();
    }
    $model = new ModifyForm();
    if ($model->load(Yii::$app->request->post())) {
        if ($model->setMyUser()) {
            $model->setInfo();
            Yii::$app->user->logout();
            return $this->redirect(array('/site/index',
                'message' => "信息修改成功，请重新登录。"
            ));
        } else {
            return $this->render('modify', [
                'model' => $model,
                'message' => "用户名或密码错误"
            ]);
        }
    }
    return $this->render('modify', [
        'model' => $model,
    ]);
}
```

即，如果提交了表单且密码正确，则修改信息并登出当前和账号，否则留在当前页面。其中修改信息的视图与注册和登陆页面的布局类似，不加赘述。

2.10 评论页面

对于每一则新闻的评论均单独处理，以 id 为 1 的新闻为例，在 SiteController 中首先通过 findAll(['id] =>1)查找 id 为 1 的新闻，再通过 findAll(['New_id']) 查找所有 New_id 为 1 的评论，然后新建一个 CommentForm 类，该类在 models 中进行实现。随后，如果发布评论按钮提交成功，则仍重定向路由到当前页面：

```php
if ($model->submit())
{
    return $this->redirect(array('/site/comment1', 'message' => '发布成功！', 'id' => 1));
}
```

同时，在视图模块建立对应的评论界面，其中发布评论的文本框和提交按钮与注册登录基本一致。主要介绍展示评论的实现代码：

```php
<?php if ($comment1 != null)
    foreach ($comment1 as $_comment) : ?>
    <div class="comments-list-wrap">
        <div class="comment-list">
            <div class="single-comment-body">
                <div class="comment-text-body">
                    <h4> <?= $_comment->author ?></h4>
                    <p><?= $_comment->content ?></p>
                </div>
            </div>
        </div>
    </div>
    <br>
<?php endforeach; ?>
```

即，通过 php 的 foreach 对数据表进行循环遍历即可，这里注意需要在数据表非空的条件下进行。

三、后台管理页面实现

3.1 网页模板导入

该部分与前台的模板导入类似，这里只给出模板网址和样式图：https://www.free-css.com/free-css-templates/page282/edukate

3.2 用户信息管理

在控制器中加入 action，即点击相应按钮后渲染 userinfo.php 页面，为了防止非管理员用户访问，在 action 开始时通过 if 分支进行限制，代码如下：

```php
public function actionUserinfo()
{
    if (Yii::$app->user->identity->type==0)
    {
        return $this->goHome();
    }
    return $this->render('userinfo');
}
```

userinfo.php 页面主要获取所有非管理员用户的信息并统计数量，较为容易，这里不再赘述。

3.3 评论内容管理

首先，在后台控制器中实现以下的 action 方法：

```php
$news = news::findAll(['id' => 1]);
$comment = Comment::findAll(['New_id' => 1]);
$model = new CommentForm();
```

然后双重循环遍历评论表：

```php
<?php if ($news != null)
foreach ($news as $_news) : ?>
    <?php $comment_temp = \common\models\Comment::findAll(['New_id' => $_news->id]);?>


    <?php if ($comment_temp != null)
    foreach ($comment_temp as $_comment) : ?>
        <div class="d-flex justify-content-between border-bottom px-4">
            <h6 class="text-white my-3"><?= $_comment->New_id?></h6>
            <h6 class="text-white my-3"><?= $_comment->id?></h6>
            <h6 class="text-white my-3"><?= $_comment->content?></h6>
            <h6 class="text-white my-3"><?= $_comment->author?></h6>

        </div>

    <?php endforeach; ?>

<?php endforeach; ?>
```

最后设置删除评论的输入文本框和按钮：

```
<?php $form = ActiveForm::begin(); ?>
<div class="form-group">
    <h2><?= $form->field($model, 'id')->textInput(['autofocus' => true]) ?></h2>
</div>
<div class="form-group">
    <?= Html::submitButton('提交', ['class' => 'btn btn-primary']) ?>
</div>
<?php ActiveForm::end(); ?>
```