



南开大学
Nankai University

南 开 大 学

互联网数据库开发

实现文档

蒋薇

年级：2021 级

专业：计算机科学与技术

2023 年 12 月 24 日

简要介绍

课程大作业我们以环保-核污染为主题，以 yii2 为框架，使用 php 语言设计有关核污染的网页，包括核污染的相关事宜，核污染的分类、核事故、核污染新闻、法规政策、预防控制等。

yii2,php, 核污染

目录

一、 流程设计	1
二、 模块设计实现	1
(一) 后端	2
(二) 前端	2
(三) 首页、注册登录页面实现	2
(四) 留言评论页面实现	4
(五) 新闻页面实现	7
(六) 重大核事故记录	8
(七) 日本三次集中排放核废水统计	8
(八) 死亡及致病人数页面实现	10
(九) 团队人员介绍页面实现	10

NKF 小组包括蒋薇、周恂、王梦龙三位同学，在本次网页设计的课程大作业上，我们选择的是设计一个有关核污染的网页，向网友介绍相关事宜。

一、 流程设计

项目开发之初团队三人对用户需求进行分析，接着查阅资料确认项目主题，对其可行性进行研究，在确认其具有可行性后进行数据库表的设计，通过对数据的获取难易、用户的满意程度、制作的难易程度等一些列问题进行可行性分析后，进行实现功能的分配。本次项目中分为后端开发和前端网页设计。其中前端网页设计又根据具体的功能分为不同的模块，例如新闻页面、交流页面等等。最后对网页进行美化，添加细节元素。完成功能后进行测试，撰写文档。

如图12所示

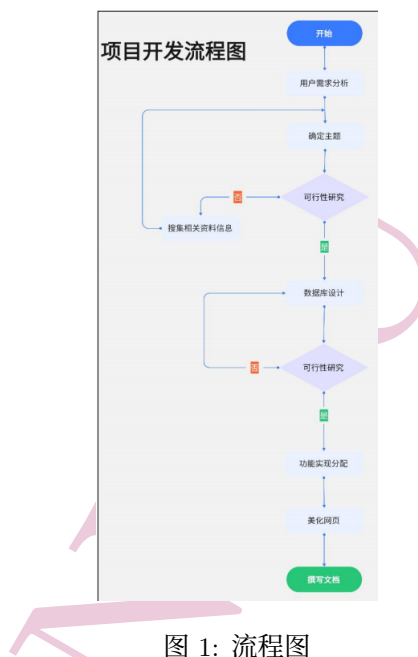


图 1: 流程图

二、 模块设计实现

项目整体开发中采用 MVC 的设计模式，在模块中定义相关类，将数据表中字段作为成员变量存取。在控制器中从数据库里取出数据。最后在视图中定义相关文件，实现对应页面的 php 设计。

如图12所示

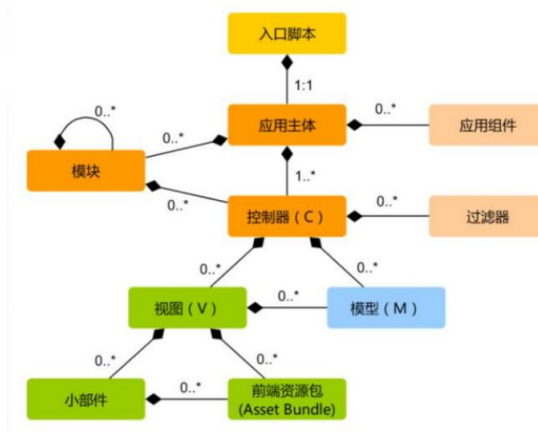


图 2: 关系控制图

（一） 后端

后端页面分两部分：CRUD、作业下载

CRUD 使用 gii 工具生成，并嵌入到原后端模板中，在页面左侧添加了下拉框供不同数据表的增删改查。

作业部分，为每一个作业的 div 设置点击事件，传入 value 参数以标识不同作业；再用 switch 语句定向到不同参数的 url，在 controller 层将读取参数、实现不同个人作业页面的区分，同样可以在左侧下拉框中选择团队、个人总体及个人作业。

（二） 前端

（三） 首页、注册登录页面实现

所有页面均使用同一模板样式；用户可以在导航栏选择 Home、about、backend、signup、signin 等，分别表示跳转到首页、about 页、后端、注册、登录；其中，鼠标悬浮在 about 下，会出现下拉框，有 comment、damage、member、news、officer、victim 页面；



图 3: 页面

用户可以在模板的 footer 部分浏览查看最新的事件 (event)，点击后跳转到详情页面。

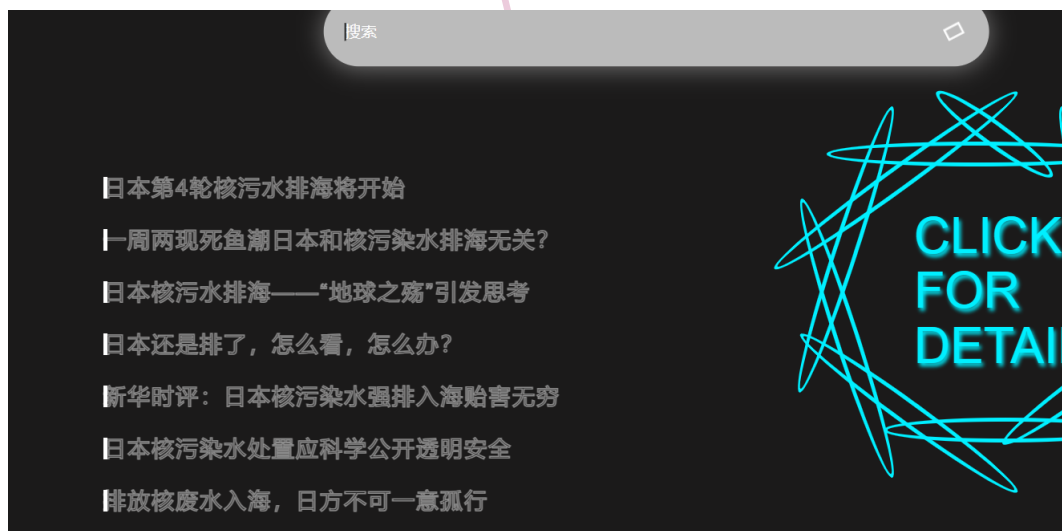


图 4: search



图 5: event



图 6: 详情

点击按钮返回首页，其中按钮具有流光动画；

首页实现了滚动视差动画，文字和图片滚动的速度不一致。

注册和登录页面具有良好的用户交互效果，会提示用户错误信息，按钮具有动画效果：当鼠标悬停时改变颜色

(四) 留言评论页面实现

标题具有动画效果：字体染光，自左到右、线性变化

Wj 2023-12-19 00:00

这是我们的互联网数据库开发的期末作业~

Nku 2023-12-19 00:00

请发表您的评论吧

Jc 2023-12-22 00:00

还有两天ddl

发表评论

昵称

2110957

内容

我的评论

发表

图 7: comment

用户可以输入留言，并发表在上方板块中 (同时添加到数据库)；在留言页面中，我们需要对数据库表进行查询从而显示出历史用户留言，因此需要设计一个数据表模型 `Comment`，该模型只需简单地继承 `ActiveRecord` 类，从而在对表进行查询时可以直接使用继承自父类的 `find` 函数。同时，我们还需要对 `comment` 表进行插入操作，因此这里我们在 `Comment` 类中实现一个静态函数 `add`，用于将用户输入插入到数据库中。

此外，留言页面还需要一个提交用户评论的表单，对于该表单的实现，我们同样需要建立一个模型 `CommentForm`，来接收用户的输入，表单需要两个参数，分别为存储用户昵称的 `user` 变量和存储用户留言内容的 `comment` 变量。

CommentForm

```

1  class Comment extends ActiveRecord
2  {
3      static public function add($name, $comment){
4          Yii::$app->db->createCommand()->insert('comment', [
5              'cuser' => $name,
6              'ccomment' => $comment,
7              'cdate' => date('Y-m-d_H-i-s', time())
8          ])->execute();
9      }
10 }
11 class CommentForm extends Model
12 {
13     public $user;
14     public $comment;
15
16     public function rules()

```

```

17     {
18         return [
19             [['user', 'comment'], 'required'],
20         ];
21     }
22 }

```

我们创建 CommentController 控制器。控制器的页面响应函数处理逻辑为，首先新建一个 CommentForm 模型对象，然后判断页面是否接收到了 post 请求并对 post 请求数据的合法性进行判断，如果成功，则表示成功接收到了用户的留言信息，此时我们将该条留言信息插入到数据库中。之后我们调用 Comment 模型的 find 函数，从数据库中查询所有的历史评论记录，并将这些记录以及 CommentForm 模型对象传递给视图 index。

CommentController

```

1  class CommentController extends Controller
2  {
3      public $layout = "main_layout";
4      public function actionIndex()
5      {
6          $model = new CommentForm();
7          if ($model->load(Yii::$app->request->post()) && $model->validate()) {
8              Comment::add($model->user, $model->comment);
9              $model->comment = "";
10         }
11         $query = Comment::find();
12         $comments = $query->orderBy('cdate')->all();
13         return $this->render('index', ['model' => $model, 'comments' =>
14             $comments]);
15     }
16 }

```

视图中我们需要将传递自控制器的历史评论信息给逐条显示出来，并使用 ActiveForm 类来生成一个提交用户评论的表单，表单有一个输入框和一个文本框，分别与 CommentForm 模型的用户昵称属性和用户评论内容属性对应，最后创建一个提交按钮。

CommentView

```

1  <div id="comments">
2      <?php foreach ($comments as $comment): ?>
3          <hr class="comment-hr">
4          <div class="comment-usr-info">
5              <span class="comment-usr-name"><?= Html::encode("{
6                  $comment->cuser}") ?></span>
7              <span class="comment-usr-date"><?= Html::encode("{
8                  $comment->cdate}") ?></span>
9          </div>
10         <p class="comment-content"><?= Html::encode("{ $comment->
11             ccomment}") ?></p>
12     <?php endforeach; ?>

```



```

10 </div>
11 <div class="comment-form">
12     <div class="comment-header">发表评论</div>
13     <!-- form开始 -->
14     <?php $form = ActiveForm::begin([
15         'fieldConfig' => [
16             'template' => "{label}\n<div class=\"comment-inputdiv\">{
17                 input}</div>",
18             'labelOptions' => ['class' => 'col-lg-1 control-label'],
19             'inputOptions' => ['class' => 'comment-input'],
20         ]
21     ]); ?>
22     <?=$form->field($model, 'user')->label('昵称') ?>
23     <?=$form->field($model, 'comment')->textarea()->label('内容') ?>
24     <div class="form-group">
25         <?=$form->submitButton('发表', ['class' => 'comment-btn']) ?>
26     </div>
27 </div>
28 <?php ActiveForm::end(); ?>
    <!-- form结束 -->

```

(五) 新闻页面实现



图 8: news

新闻界面有三部分：轮播图、搜索引擎接口、新闻列表。与之前类似，通过 model 查询、controller 匹配 url、将数据送至 view 层；其中新闻按时间倒序排列。

搜索引擎接口使用 js 跳转到百度，并传入用户参数。

页面具有动画效果：轮播图、搜索框当鼠标悬浮时会展开、新闻标题当鼠标悬浮时会高亮、右侧有提示动画装饰

(六) 重大核事故记录

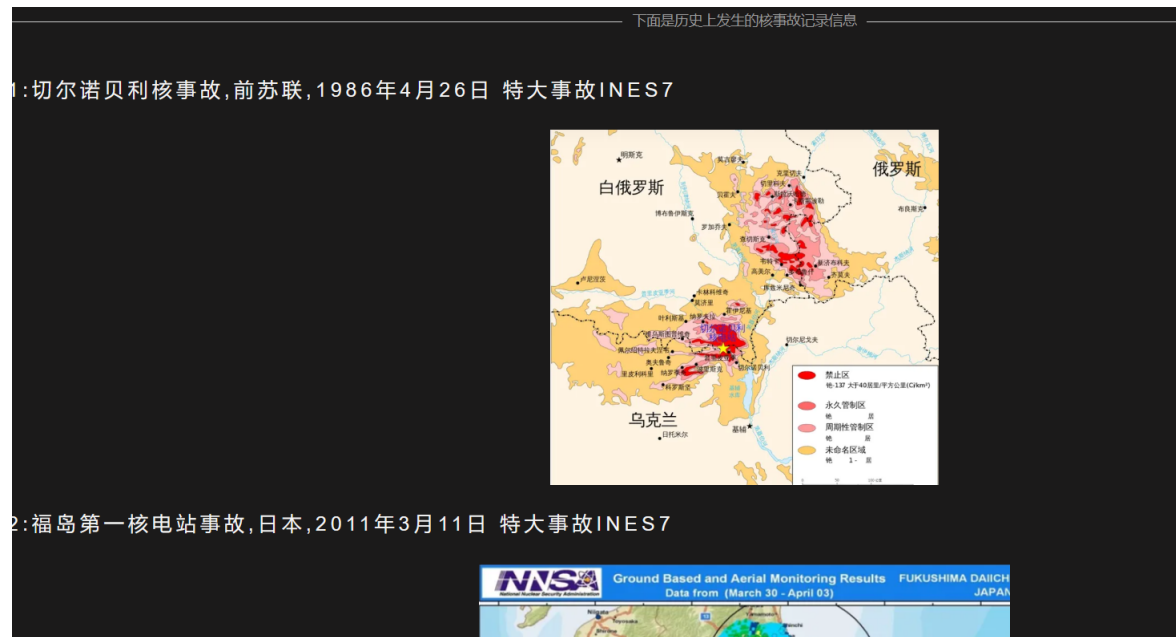


图 9: event record

整体思路和上面介绍的 MVC 一致, 在 VIEW 中, 当取得数据后, 循环打印数据表 OFFICER 中的事故信息, 同时打印显示事故图片。页面动画效果是: 事故部分的滚动视差效果。

(七) 日本三次集中排放核废水统计

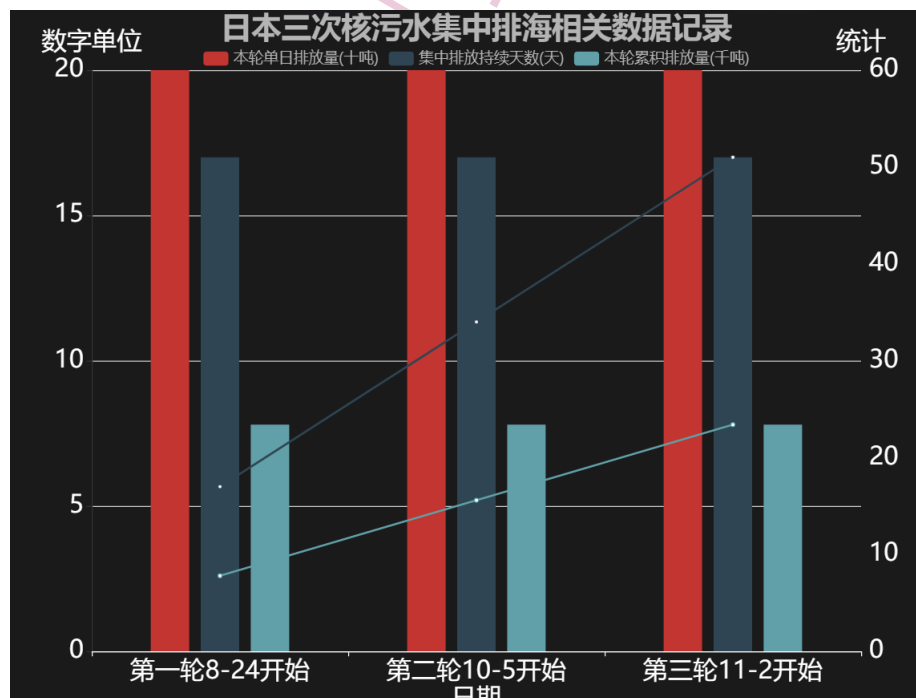


图 10: 日本三次集中排放核废水统计

在数据库中设计好对应的数据表，来在 damage 下的 echart.php

文件里连接数据库，将数据表 damage 中的数据取出，并存到 Damage 类对象中，接着将数据存放在 data 变量中以便后面使用。

在 index.php 中实现柱形图和折线图。首先取出我们刚刚存取的数据

echart

```
1 //调用ajax来实现异步的加载数据
2 function getUsers() {
3     var pathName = window.location.href
4     var DirName = pathName == '?' ? '' : pathName.substring(0, pathName.
        lastIndexOf('/'));
5     $.ajax({
6         type: "GET", //使用POST会出现多余的数据，导致无法解析
7         async: false,
8         url: DirName + '/echart',
9         data: {},
10        dataType: "json",
11        success: function(result){
12            if(result){
13                for(var i = 0 ; i < result.length; i++){
14                    dates.push(result[i].date);
15                    dmgTank.push(result[i].tank);
16                    dmgUAV.push(result[i].UAV);
17                    dmgCar.push(result[i].car);
18                    dmgTankAll.push(result[i].tankAll);
19                    dmgUAVAll.push(result[i].UAVAll);
20                    dmgCarAll.push(result[i].carAll);
21                }
22            }
23        },
24        error: function(errmsg) {
25            console.log("Ajax获取服务器数据出错了！" + errmsg);
26        }
27    });
```

接着进行柱状图相关项的设置，类如 x、y 轴坐标、名称、颜色和字体大小等等：

这里实现了分组和柱状图折线图同图。对于分组，在 legend 中写入组别名称，接着在 series 中将 name 项写入具体的组别

对于折线图和柱状图同图，则要注意 y 轴坐标轴的实现，要分为左右两个坐标轴，并在使用时标明使用的是哪一个，即 yAxisIndex 为 0（这里默认 0 表示第一个）还是 1。

页面具有的动画效果是，柱形图和折线图动态展示、当改变武器种类时的动态变化。

(八) 死亡及致病人数页面实现

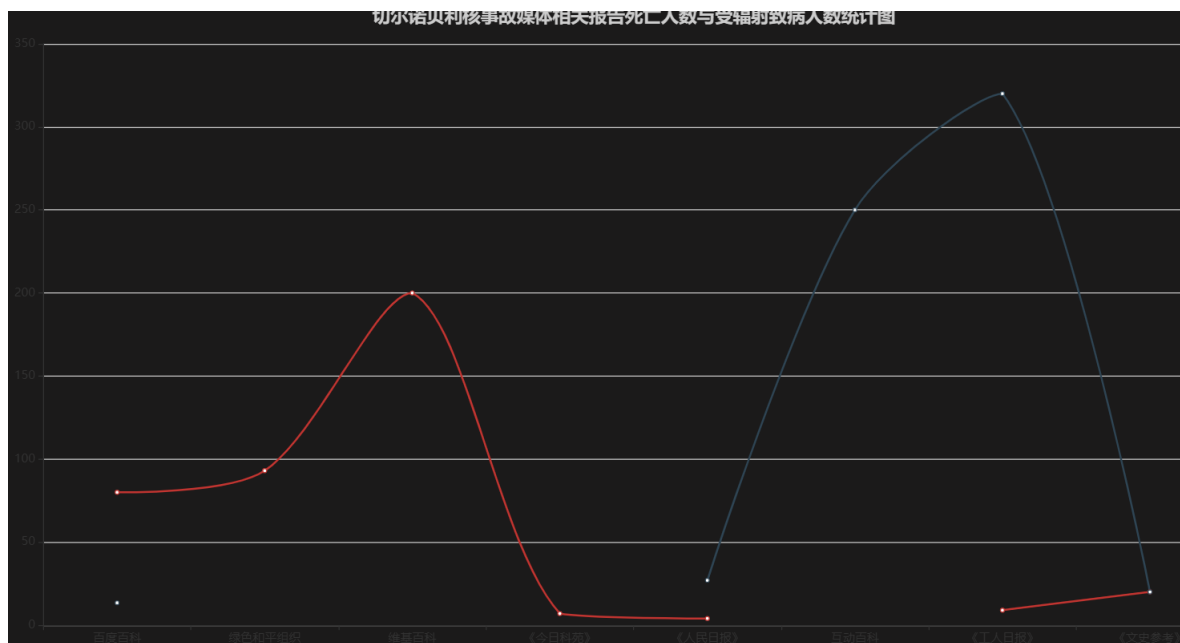


图 11: victim

采用 echarts 实现折线图的效果。页面的动画是刷新时折线图动态展开。

(九) 团队人员介绍页面实现

通过 MVC 模式从数据库中获取成员的姓名、自我介绍，并显示。

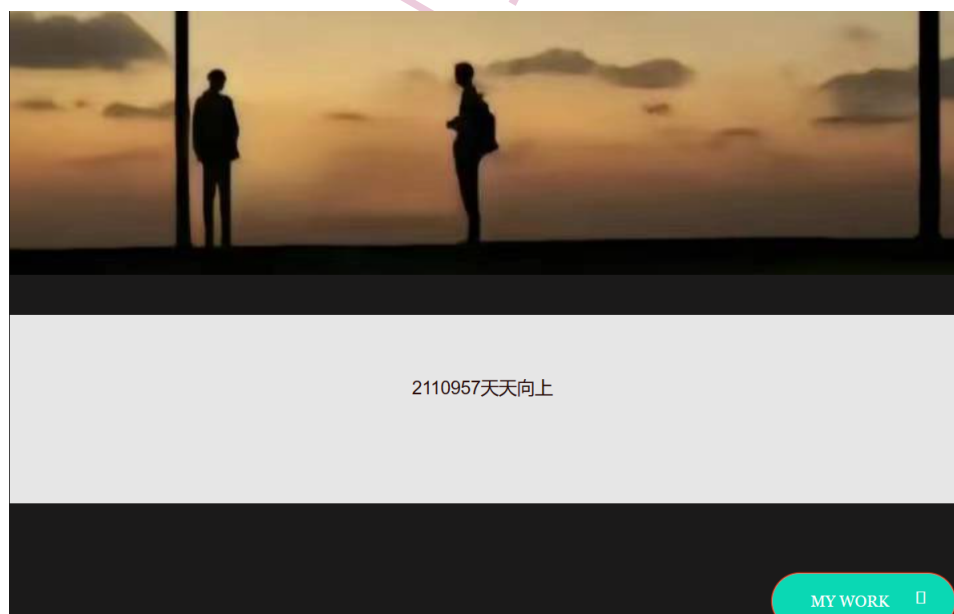


图 12: Members

动画效果是，当用户下滑时，新出现的 div 具有浮动效果。