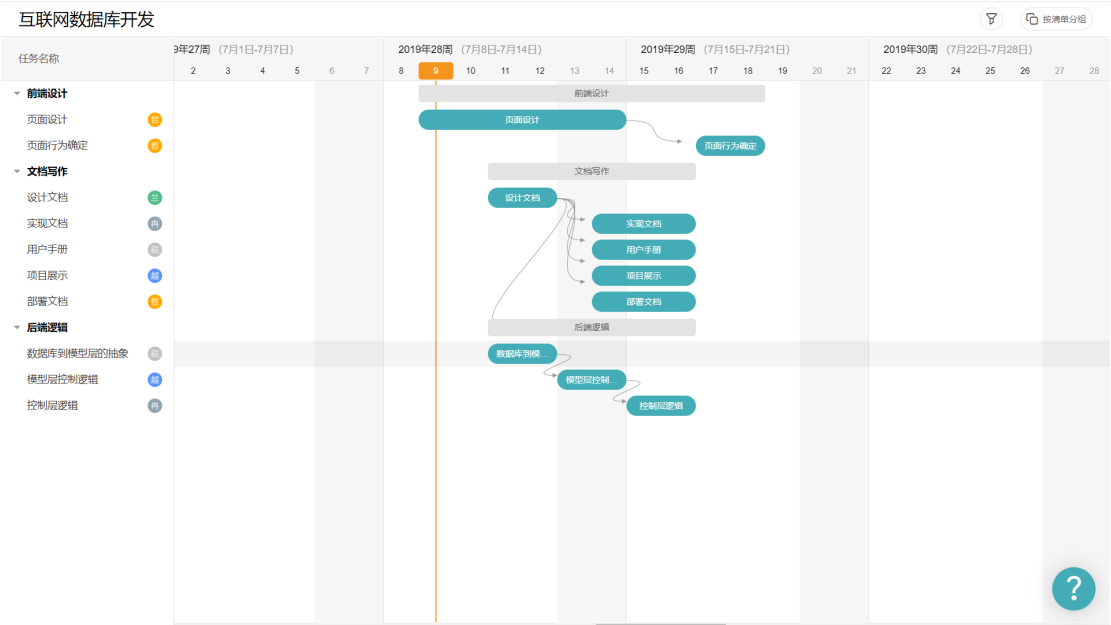


百年南开百科实现文档

1. 任务分工过程

1. 具体分工：
- 前台设计、部署文档、需求文档写作：杨侯哲；
- 后台逻辑之控制层逻辑、实现文档写作：孙一冉；
- 后台逻辑之模型层控制逻辑、项目展示 PPT 写作：杨越；
- 后台逻辑之数据库到模型层的抽象、用户手册写作：王心荻；
- 后台逻辑补充、设计文档写作：解亚兰；
- 数据库及相关触发器的设计，数据的填入：杨侯哲，孙一冉，杨越，王心荻，解亚兰；
2. Tower 截图：
- (其中的代表词：哲：杨侯哲；兰：解亚兰；冉：孙一冉；荻：王心荻；越：杨越。)











2. Git 提交记录

1. 项目 GitHub 完成后任务完成数目分布整体截图：

















2. 其中个人任务完成后提交的部分截图：

Commits on Jul 13, 2019

layui gilsaia committed 8 days ago	 d4bfe4d	
Merge branch 'master' of https://github.com/Although1998/nkpediaof in... gilsaia committed 9 days ago	 f7cd6f1	
7.13 lansehun committed 9 days ago	 e6f6461	

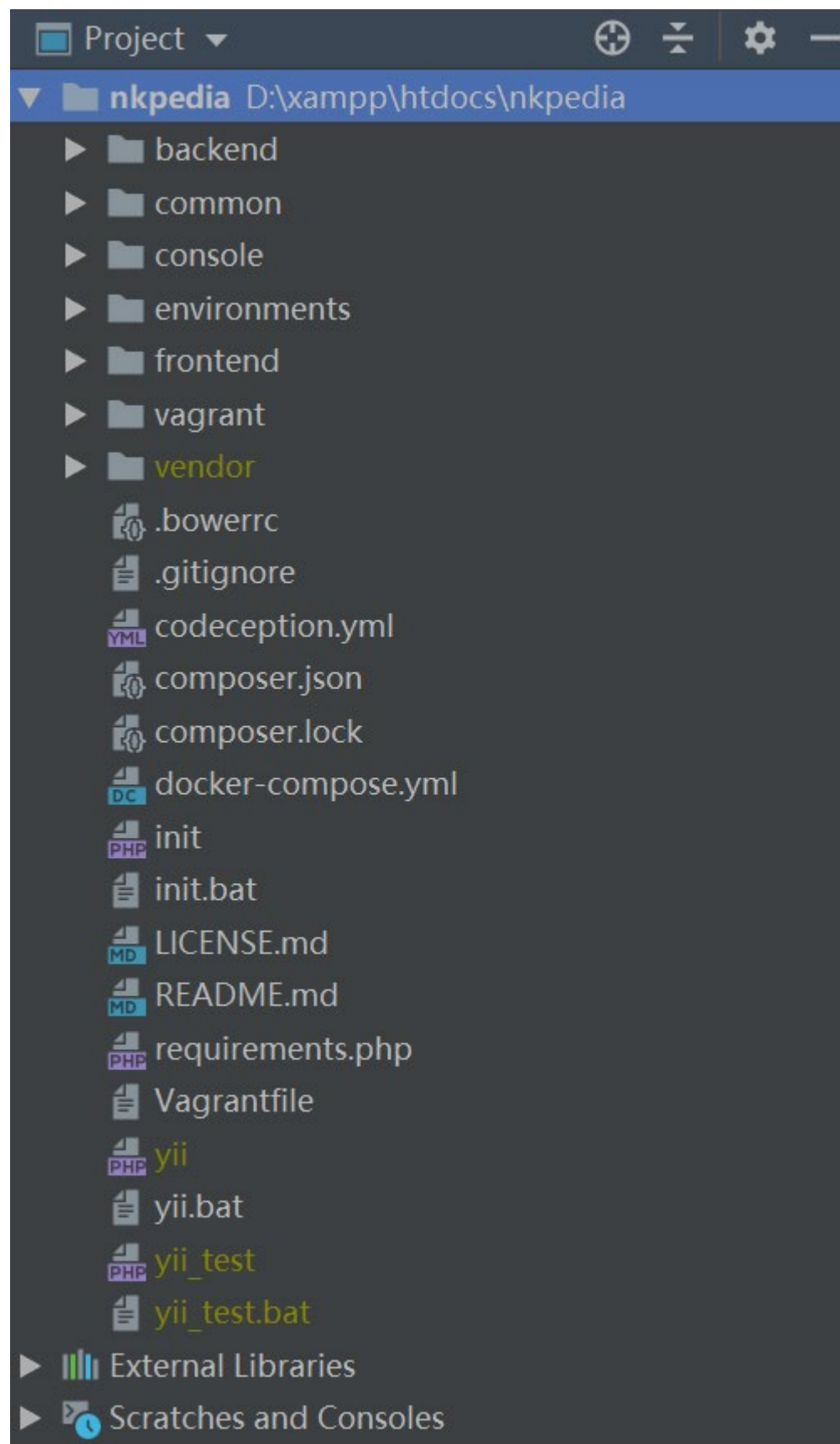
Commits on Jul 12, 2019

Merge branch 'master' of https://github.com/Mr-Xindi/nkpd lh777777 committed 9 days ago	 b7e3eef	
最终版 Although1998 committed 9 days ago	 77bc305	
Merge branch 'master' of https://github.com/Mr-Xindi/nkpd Although1998 committed 9 days ago	 cc229bd	
yy222 lh777777 committed 9 days ago	 a3e2754	
Merge branch 'master' of https://github.com/lh777777/nkpedia Although1998 committed 9 days ago	 a0b6816	
xd backend Mr-Xindi committed 9 days ago	 e16d0c0	
xd backend Mr-Xindi committed 9 days ago	 dab9e81	

3. 主要代码及展示

3.1 前端-网页显示

3.1.1 总体目录:



分析解释:

backend:后台模板

common: 前台后台共用的模板

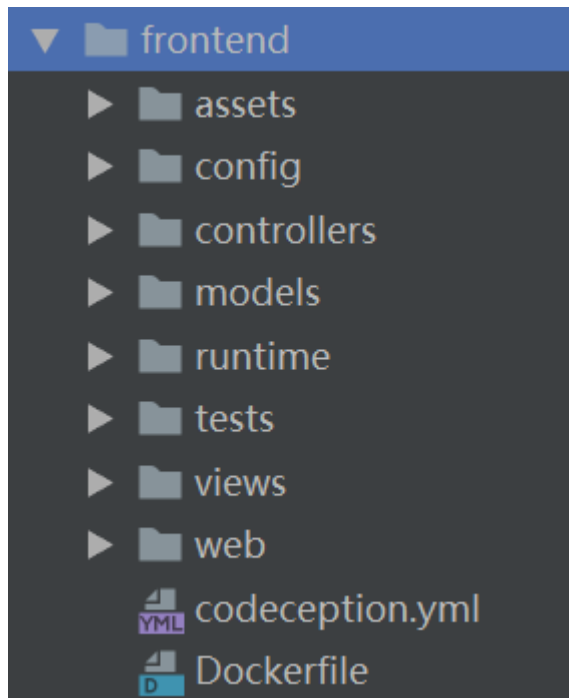
console: 控制台

environments: 环境

frontend: 前台模板

vendor: 高级模板

3.1.2 前台 frontend



1. 调用 ui 库

我们决定在前台使用 layui 作为 ui 库来搭建起我们自己所需要的网页，所以在 asset 处加入 layui 注册 php

```
LayuiAsset.php
7  * Team: 没有蛀牙, NKU
8  * Coding by 杨俊哲 1711396, 20190713
9  * This is ui Asset
10
11  */
12  * Main backend application asset bundle.
13  */
14  class LayuiAsset extends AssetBundle
15  {
16      public $basePath = '@webroot';
17      public $baseUrl = '@web';
18      public $css = [
19          'layui/css/layui.css',
20          'css/my.css',
21          'resources/css/background.css',
22          'font-awesome/css/font-awesome.css'
23      ];
24      public $js = [
25          'layui/layui.js',
26      ];
27      public $depends = [
28      ];
29      //定义按需加载JS方法, 注意加载顺序在最后
30      public static function addScript($view, $jsfile) {
31          $view->registerJsFile($jsfile, [AppAsset::className(), 'depends' => 'frontend/assets\LayuiAsset']);
32      }
33
34      //定义按需加载css方法, 注意加载顺序在最后
35      public static function addCss($view, $cssfile) {
36          $view->registerCssFile($cssfile, [AppAsset::className(), 'depends' => 'frontend/assets\LayuiAsset']);
37      }
38  }
39
40  \frontend\assets > LayuiAsset > addScript()
```

这样在网页中可以随时调用 ui 库相关样式，对于需要使用的 js 和 css 文件也可以调用相应函数进行注册

2. 查询词条

```
{
    $this->layout='laymain';
    if($report!=null)
    {
        $reportcol=new PediaEntryReport();
        $reportcol->vid=$report;
        $reportcol->description=' 举报词条';
        $reportcol->save();
    }
    if($wordse!=null)
    {
        // 做些有意义的事 ...
        $wordsi=PediaEntryBasicinfo::find()->where(['title'=>$wordse]);
        if($wordsi->count() !=0)
        {
            $wordsi=$wordsi->one();
            if($wordsi->isshow==0)
            {
                return $this->render(['view' => 'error', ['message' => 'There is No This Word', 'name' => 'Can\'t Search']]);
            }
            $wordsi->clicktimes+=1;
            $wordsi->update();
            return $this->render(['view' => 'search', ['word' => $wordsi, 'report' => $report]]);
        }
        else
        {
            return $this->render(['view' => 'error', ['message' => 'There is No This Word', 'name' => 'Can\'t Search']]);
        }
    }
}

$model = new SearchWordForm();
if ($model->load(Yii::$app->request->post()) && $model->validate()) {
```

```

        return $this->render( view: 'error', ['message' => 'There is No This Word', 'name' => 'Can\'t Search']);
    }
}

$model = new SearchWordForm();
if ($model->load(Yii::$app->request->post()) && $model->validate()) {
    // 验证 $model 收到的数据

    // 做些有意义的事 ...
    $word=PediaEntryBasicinfo::find()->where(['like','title',$model->word]);
    if($word->count() !=0)
    {
        $word=$word->one();
        if($word->isshow==0)
        {
            return $this->render( view: 'error', ['message' => 'There is No This Word', 'name' => 'Can\'t Search']);
        }
        $word->clicktimes+=1;
        $word->update();
        return $this->render( view: 'search', ['word'=>$word,'report'=>$report]);
    }
    else
    {
        return $this->render( view: 'error', ['message' => 'There is No This Word', 'name' => 'Can\'t Search']);
    }
} else {
    // 无论是初始化显示还是数据验证错误
    return $this->render( view: 'index', ['model' => $model]);
}

/**
 * Displays ranking page
 */

```

对于查询词条可能是带参的 get 请求或是无参的 get 请求，所以首先判断是否是举报词条，然后判断是否是精确搜索词条，如果都不是那么就是模糊搜索词条，根据条件的不同进行不同的判断，如果是模糊查询说明需要一个搜索表单提交 post 请求，同时对于搜索到的词，它的点击次数立刻增加存回数据库

3. 类别搜索和热词排行

两功能实现方式相近，在此对热词排行功能做一个说明，热词排行功能重要的部分是下图的部分

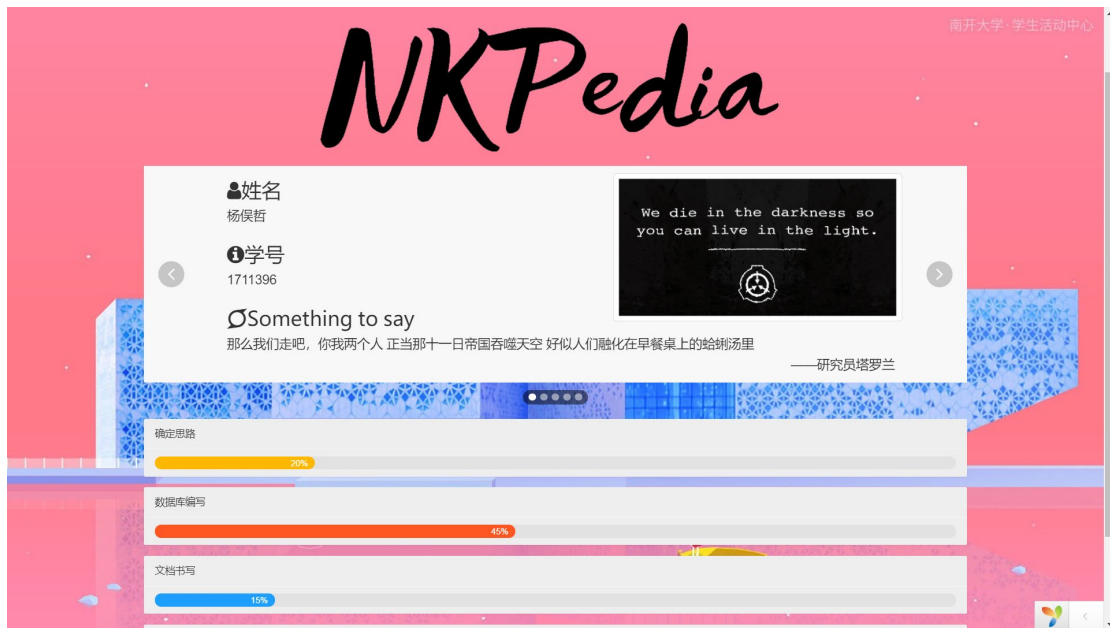

```
30 <th>热点新闻</th>
31 <th>词条类别</th>
32 <th>点击次数</th>
33 </tr>
34 </thead>
35 <tbody>
36 <?php $words=PediaEntryBasicinfo::findBySql('select eid,title,clicktimes,isshow FROM pedia_entry_basicinfo ORDER BY clicktimes DESC')->all();
37 $colors=array('layui-btn-normal','layui-btn-warm','layui-btn-danger');
38 foreach ($words as $word)
39 {
40     if($word->isshow==0)
41     {
42         continue;
43     }
44     echo '<tr>';
45     '<td><a target="_self" href="/pkpedia/frontend/web/index.php?r=site%2Findex&wordde='.$word->title.'">'.$word->title.'</a></td>';
46     '<td><div class="layui-btn-container">';
47     $scates=PediaEntryClassification::find()->where(['eid'=>$word->eid])->all();
48     foreach ($scates as $scate)
49     {
50         $cname=PediaEntryCategory::find()->where(['cid'=>$scate->cid])->one();
51         echo '<a target="_self" href="/pkpedia/frontend/web/index.php?r=site%2Fcategory&cid='.$scate->cid.'"><button type="button" class="layui-btn';
52         echo '</div></td>';
53         '<td>'.$word->clicktimes.'</td>';
54     }
55 }
56 ?>
57 </tbody>
58 </table>
59 </div>
60 <?php LayuiAsset::addscript($this, 'jsfile' => '@web/resources/js/rank.js')?>
61
```

这里执行词条表对应 model 的查询函数，这里是直接按照 sql 语句执行选出相应信息并按点击次数(在这里认为是词条热度)排序并列出现相应信息，而在 layui 中表格的使用需要 js 函数来激活，所以在最下方注册了对应的 js 文件以启动 layui 的 table 模块

4. 关于我们

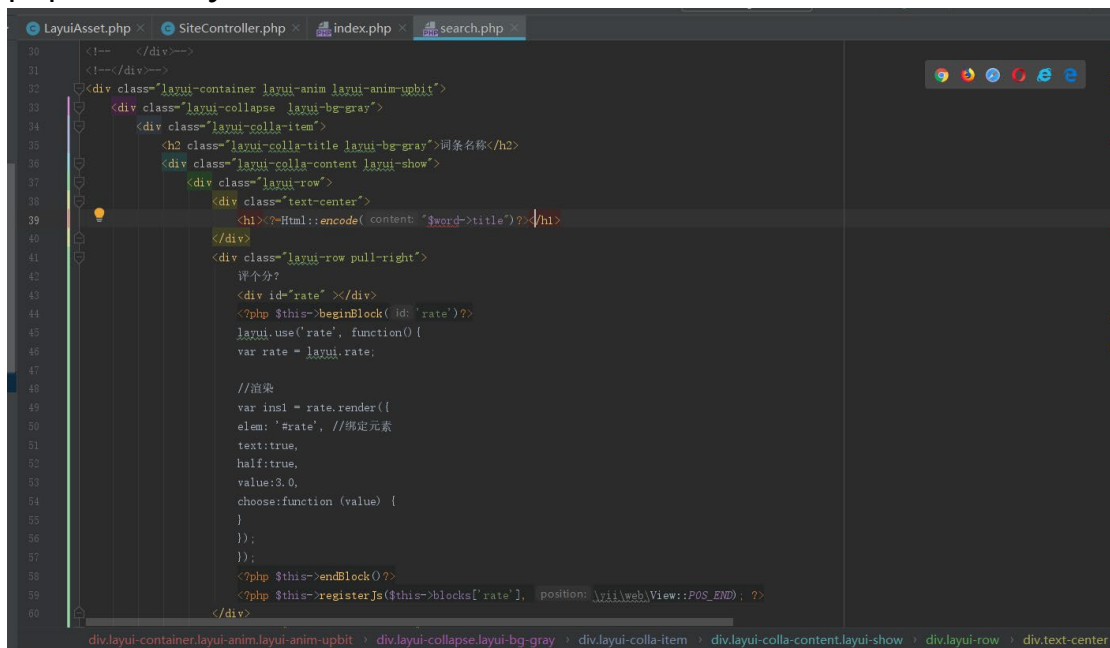
在这里用到的是 css 的样式技巧，对于轮播的界面，其中既想要放入相应文字又想空白处置入图片，那么文字位置是设为相对，通过边框调整设到合适位置，而图片位置设为绝对，从而不影响文字的栅格系统，不影响整体排版

```
LayuiAsset.php x SiteController.php x aboutus.php
</div>
<div class="layui-row">
  <div class="layui-col-md3" style="position: relative; left: 100px;">
    <p class="h4">杨侯哲</p>
  </div>
</div>
<br>
<div class="layui-row">
  <div class="layui-col-md3" style="position: relative; left: 100px;">
    <i class="fa fa-info-circle fa-2x">学号</i>
  </div>
</div>
<div class="layui-row">
  <div class="layui-col-md3" style="position: relative; left: 100px;">
    <p class="h4">1711396</p>
  </div>
</div>
<br>
<div class="layui-row">
  <div class="layui-col-md3" style="position: relative; left: 100px;">
    <i class="fa fa-superpowers fa-2x">Something to say</i>
  </div>
</div>
<div class="layui-row">
  <div class="layui-col-md10" style="position: relative; left: 100px;">
    <p class="h4">那么我们走吧，你我两个人 正当那十一日帝国吞噬天空 好似人们融化在早餐桌上的黏糊汤里</p>
    <p class="h4 text-right">——研究员塔罗兰</p>
  </div>
</div>
</div>
div.layui-container > div#uscarousel.layui-carousel > div > div.layui-container > div.layui-row > div.layui-col-md10
```

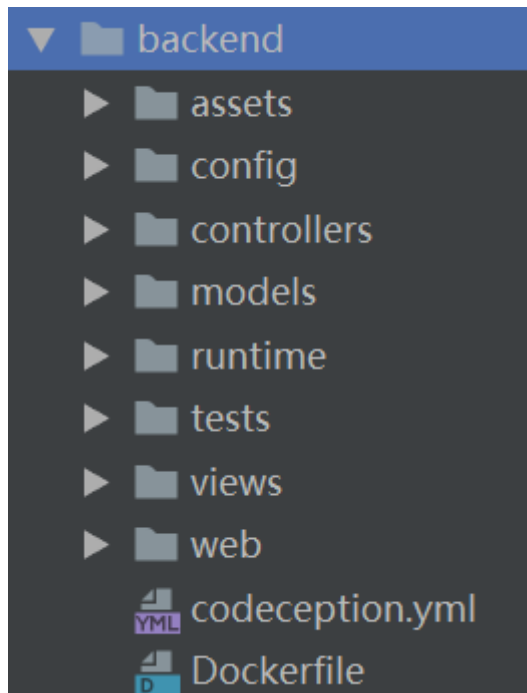


5. php 文件中 js 的使用

在 php 文件中直接写入 script 标签经常会造成种种奇怪的错误, 我们可以通过写 js 文件而后在 php 文件中注册的方式来解决, 这里提供另一种解决方式, 即 php 文件内的 js 代码块



3.1.3 后台 backend



a. 后台模板更换

使用 composer 安装所需要的包，在包中找到示例页面后拷贝到相应目录下后，将所选模板在布局中注册即可（见 /backend/views/layouts/backon.php 第 17 行）。使用时在 controller 文件中除 behaviors() 与 findModel() 的所有函数中写入 `$this->layout='backcon';` 即可应用新的模板。

```
backcon.php x main-local.php x PediaEntryBasicInfoController.php x index.php x form.php x _form.php x
7  /**
8  /**
9  * @var string $content
10 * @var \yii\web\View $this
11 */
12 use backend\assets\AppAsset;
13 use yii\helpers\Html;
14 use yiister\gentelella\assets\Asset;
15 use yii\helpers\Url;
16
17 $bundle = yiister\gentelella\assets\Asset::register($this);
18
19 <?>
20 <?php $this->beginPage(); ?>
21 <!DOCTYPE html>
22 <html lang="<? = Yii::$app->language ?>">
23
24
25 <head>
26 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
27 <meta charset="<? = Yii::$app->charset ?>" />
28 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
29 <meta name="viewport" content="width=device-width, initial-scale=1" />
30 <? = Html::csrfMetaTags() ?>
31 <title><? = Html::encode($this->title) ?></title>
32 <?php $this->head() ?>
33 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
34 <!--[if lt IE 9]>
35 <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
36 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
```

b.CRUD 页面的显示

使用 yii 自带工具 gii 创建 model 和 crud，创建 crud 时 model class 路径以 common\models\模块名的方式写出以便前后端共同使用，Search Model Class 路径以 backend\models\模块名+Search 给出，Controller Class 路径以 backend\models\模块名+Controller 给出。调用时在 /backend/views/layouts/backcon.php 中 label 的 URL 属性中填写相应路径即可。

```
backcon.php x main-local.php x PediaEntryBasicInfoController.php x index.php x form.php x _form.php x
88 v id="sidebar-menu" class="main_menu_side hidden-print main_menu">
89
90 <div class="menu_section">
91 <h3>General</h3>
92 <? =
93 \yiister\gentelella\widgets\Menu::widget(
94 [
95     "items" => [
96         ["label" => "Home", "url" => ['/site/index'], "icon" => "home"],
97         ["label" => "UserMember",
98             "icon" => "user-circle",
99             "url" => "#",
100             "items" => [
101                 ["label" => "user-member", "url" => ["/pedia-user-member/index"], "icon" => "user-circle-o"],
102                 ["label" => "user-medal", "url" => ["/pedia-user-medal/index"], "icon" => "graduation-cap"],
103                 ["label" => "user-adorn", "url" => ["/pedia-user-adorn/index"], "icon" => "lock"],
104             ],
105         ],
106         [
107             "label" => "UserGroup",
108             "icon" => "address-book",
109             "url" => "#",
110             "items" => [
111                 ["label" => "user-group", "url" => ["/pedia-user-group/index"], "icon" => "users"],
112                 ["label" => "user-perm", "url" => ["/pedia-user-perm/index"], "icon" => "key"],
113             ],
114         ],
115         [
116             "label" => "Entry",
117             "url" => "#",
```

c. 用户权限限制

数据库中的表项的创建与编辑对用户的身份要求十分复杂。例如：游客不允许对表项进行创建、编辑甚至查看，普通用户不允许创建表项，管理员可以创建、编辑甚至删除表项。有些词条需要专业领域的专家才可以编辑。所以我们对权限有以下处理：

1. 游客不允许对表项进行查看，即点击任意一个按钮都会显示登录界面：

```
public function actionIndex()
{
    if (Yii::$app->user->isGuest)
    {
        return $this->goHome();
    }
    $this->layout='backcon';
    $searchModel = new PediaEntryBasicinfoSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

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

在每个 model 的 controller 文件中，Index 操作中判断用户是否是游客，若是游客则返回登录界面。

2. 普通用户看不到 CRUD 界面中的“create”按钮：

```
<?php
$gid = PediaUserMember::find()->where(['loginname' => Yii::$app->user->identity->username])->asArray()->one()['gid'];
$pid = PediaUserGroup::find()->where(['gid' => $gid])->asArray()->one()['pid'];
$edit = PediaUserPerm::find()->where(['pid' => $pid])->asArray()->one()['allowedcreword'];
if ($edit != 0) {
    ?><p>
        <?= Html::a( text: 'Create Pedia Entry Basicinfo', ['create'], ['class' => 'btn btn-success']) ?>
    </p><?php
}
?>
```

在 view 文件夹中的各个 model 对应的 index 文件中，开头判断该登录用户是否有“allowcreword”权限，即能否创建表项的权限，若有，则显示 create 按钮。

3. 普通用户申请编辑或删除表项时会有弹窗警告，点击确定后返回上一界面：


```

public function actionUpdate($id)
{
    $gid = PediaUserMember::find()->where(['loginname' => Yii::$app->user->identity->username])->asArray()->one()['gid'];
    $pid = PediaUserGroup::find()->where(['gid' => $gid])->asArray()->one()['pid'];
    $edit = PediaUserPerm::find()->where(['pid' => $pid])->asArray()->one()['allowedcreword'];
    if ($edit == 0) {
        ?<script>alert("只有管理员可以编辑用户");history.back();</script><?php
        exit("0");
    }
    $this->layout='backcon';
    $model = $this->findModel($id);

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

    return $this->render( view: 'update', [
        'model' => $model,
    ]);
}

public function actionDelete($id)
{
    $gid = PediaUserMember::find()->where(['loginname' => Yii::$app->user->identity->username])->asArray()->one()['gid'];
    $pid = PediaUserGroup::find()->where(['gid' => $gid])->asArray()->one()['pid'];
    $edit = PediaUserPerm::find()->where(['pid' => $pid])->asArray()->one()['allowedcreword'];
    if ($edit == 0) {
        ?<script>alert("只有管理员可以删除用户");history.back();</script><?php
        exit("0");
    }
    $this->layout='backcon';
    $this->findModel($id)->delete();

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

```

在每个 model 的 controller 文件中，在 update 和 delete 操作中对相应权限进行判断，若不具备某种需求权限，则弹窗并返回上一界面。

d.首页用户身份、权限、勋章展示

1.用户身份

```

<p style="width: 100%;height: 15px;display: block;line-height: 30px;text-align: left;font-size: 20px"><?php
    $gid = PediaUserMember::find()->where(['loginname' => Yii::$app->user->identity->username])->asArray()->one()['gid'];
    $Gname = PediaUserGroup::find()->where(['gid' => $gid])->asArray()->one()['gname'];
    echo $Gname;
?>
</p>

```

找到此时登录的用户名所对应的 gid，再在 PediaUserGroup 中找到该 gid 对应的身份，打印在网页上即可。

2.权限

```

<p style="width: 100%;height: 15px;display: block;line-height: 30px;text-align: left;font-size: 20px"><?php
    $gid = PediaUserMember::find()->where(['loginname' => Yii::$app->user->identity->username])->asArray()->one()['gid'];
    $pid = PediaUserGroup::find()->where(['gid' => $gid])->asArray()->one()['pid'];
    $edit1 = PediaUserPerm::find()->where(['pid' => $pid])->asArray()->one()['alloweditword'];
    if ($edit1 != 0)
    {
        echo "Allow edit word:Yes";
    }
    else
    {
        echo "Allow edit word:No";
    }
?>
</p>

```

找到此时登录的用户名所对应的 gid，再在 PediaUserGroup 中找到该 gid 对应的 pid，在 PediaUserPerm 表中对于每一个权限进行检查，由于不同等级的用户组对应的权限数字从 1 到 10 不等，所以对权限与 0 进行比对，若不为

0，则说明该用户组有此权限，若为 0 则没有。后续几个权限以此类推。

3.勋章展示

```
103         }
104     } else
105     {
106         echo "Allow change perms:No";
107     }
108     ?></p>
109 </div>
110 </div>
111 <div id="w2" class="x_panel"><div class="x_title"><h1><i class="fa fa-grav"></i> Your Medal</h1><div class="clearfix">
112     <p style="..."><?php
113         $loginid = PediaUserMember::find()
114             ->where(['loginname' => Yii::$app->user->identity->username])->asArray()->one()['gid'];
115         $info = Yii::$app->db->createCommand( sql: 'SELECT mname
116 FROM pedia_user_medal,pedia_user_adorn
117 where pedia_user_medal.mid=pedia_user_adorn.mid
118 and pedia_user_adorn.uid=(:'.$loginid.')')->queryColumn();
119         reset( $array: $info);
120         foreach ( $info as $temp){
121             echo $temp;
122         }
123     ?></p>
124 </div>
125 </div>
126 </div>
127 </div>
128 </div>
129
130 div#w2.x_panel > div.x_content > p
```

找到此时登录的用户名所对应的 gid 后用 SQL 语句进行多表查询后将结果打印出来即可。

1. 下拉菜单

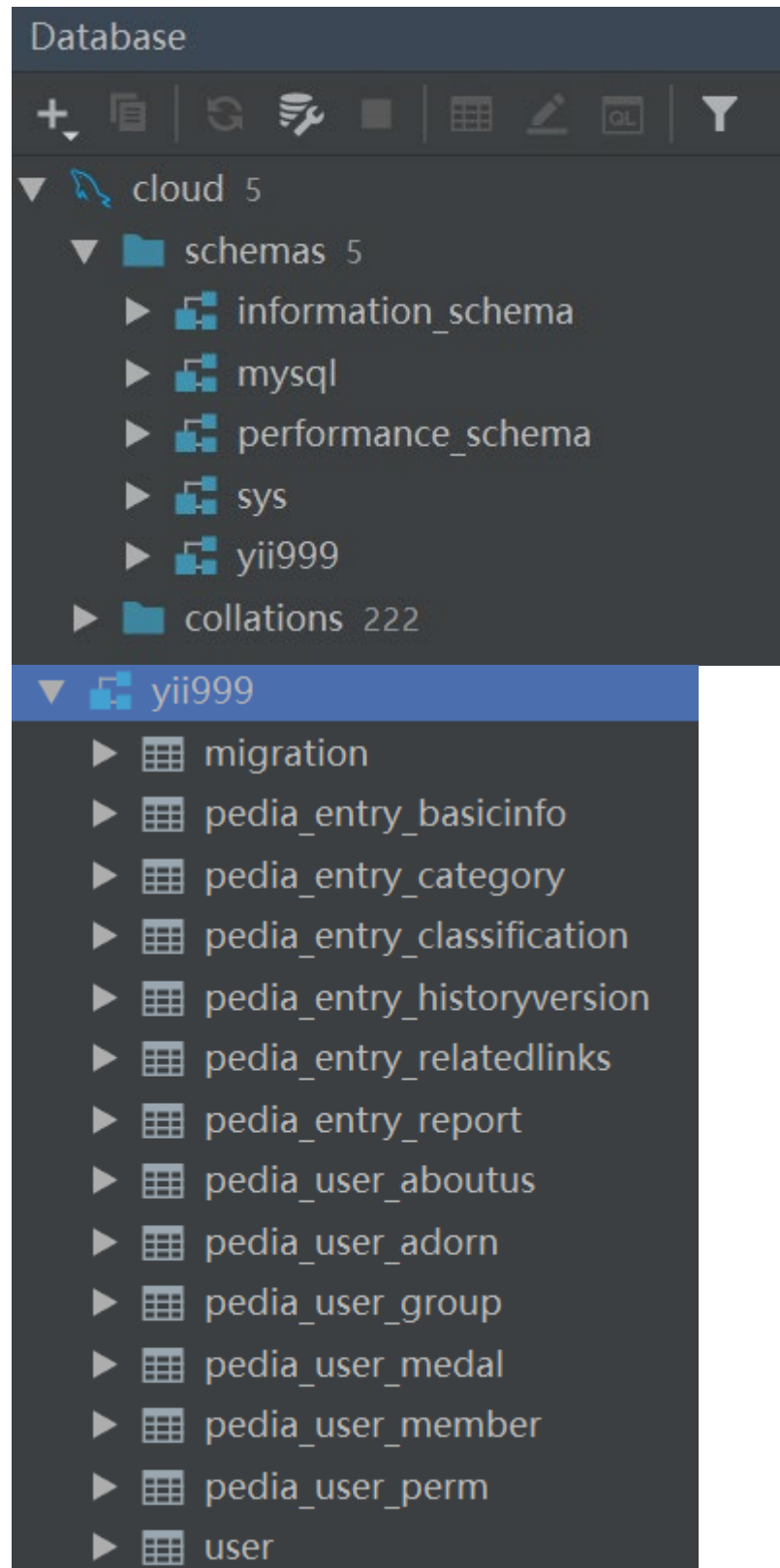
词条分类和用户佩戴勋章情况创建元组时，设计了下拉菜单。

```
4  /** Team:没有蛀牙,NKU ... */
10 use ...
12
13 /** @var $this yii\web\View */
14 /** @var $model common\models\PediaEntryClassification */
15 /** @var $form yii\widgets\ActiveForm */
16 ?>
17
18 <div class="pedia-entry-classification-form">
19
20     <?php $form = ActiveForm::begin(); ?>
21
22     <?= $form->field($model, attribute: 'eid')->textInput() ?>
23
24     <?php
25     $categories=\common\models\PediaEntryCategory::find()->all();
26     $listData=Yii\helpers\ArrayHelper::map($categories, from: 'cid', to: 'category');
27     echo $form->field($model, attribute: 'cid')->dropDownList($listData,['prompt'=>'请选择类别']);
28     ?>
29
30     <div class="form-group">
31         <?= Html::submitButton( content: 'Save', ['class' => 'btn btn-success']) ?>
32     </div>
33
34     <?php ActiveForm::end(); ?>
35
36 </div>
37
div.pedia-entry-classification-form
```

在/backend/views/模块名/_form.php 中写入如下代码，默认为“请选择类别”。

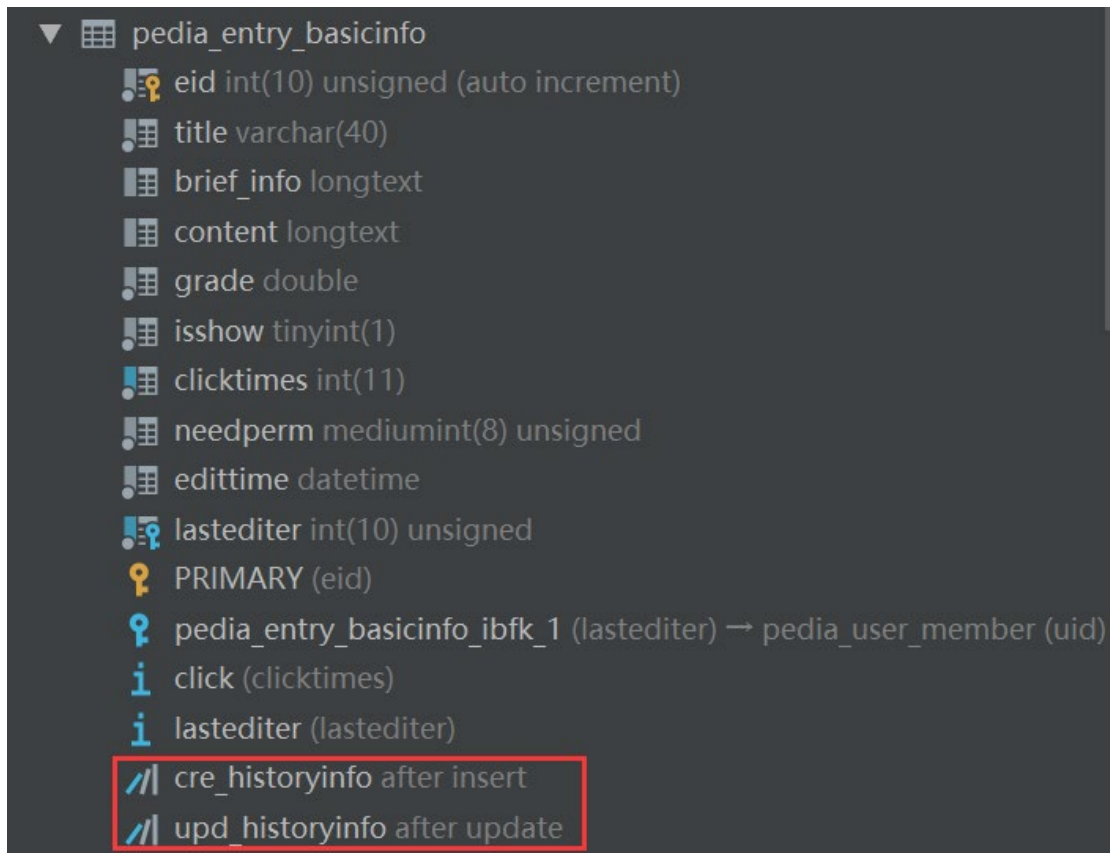
3.2 后端-数据库

3.2.1 数据库总体目录及表：



3.2.2 部分表及触发器举例：

1. 基本词条和词条历史版本两表的触发器：



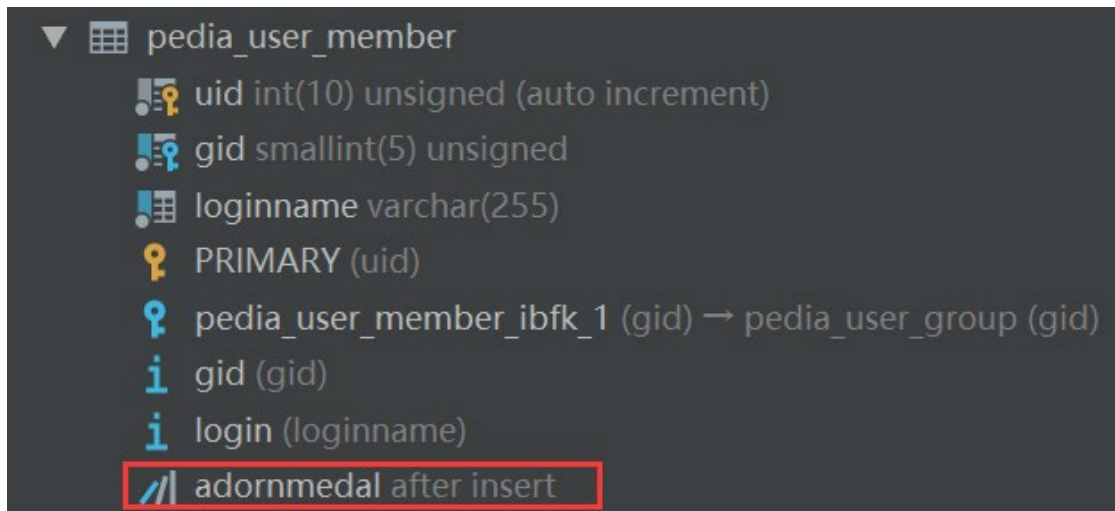
cre_historyinfo 触发器作用为在基本词条表中每插入一个新的词条，会在历史版本表中插入有"create word"以及其对应时间和管理人员等属性的新元组：

```
create definer = dbdev@`%` trigger yii999.cre_historyinfo
after INSERT
on yii999.pedia_entry_basicinfo
for each row
BEGIN
INSERT INTO pedia_entry_historyversion(edit_time,edit_cont,eid,uid) VALUES(NOW(),'create word',new.eid,new.lasteditor);
END;
```

同理，upd_historyinfo 触发器作用为在基本词条表中更新词条时在历史版本表中插入"update word"的相关元组：

```
create definer = dbdev@`%` trigger yii999.upd_historyinfo
after UPDATE
on yii999.pedia_entry_basicinfo
for each row
BEGIN
INSERT INTO pedia_entry_historyversion(edit_time,edit_cont,eid,uid) VALUES(NOW(),'update word',new.eid,new.lasteditor);
END;
```

2. 用户注册以及佩戴勋章两表触发器：



adornmedal 触发器的作用为用户每注册一个新号码，会自动给其佩戴上“初露锋芒”即初级勋章：

