

## 目录

一、	团队合作 .....	3
	使用 github 进行团队合作 .....	3
	使用 dbv 工具同步数据库结构 .....	4
	使用 navicat 同步数据库数据 .....	4
	分工情况 .....	4
二、	数据获取 .....	5
	使用在线工具获取数据 .....	5
	使用 c#编写爬虫获取数据 .....	6
三、	前后端模版使用 .....	16
	前端模版的使用 .....	16
	后端模版的使用 .....	16
四、	前端侧边按钮 .....	17
	“back-to-top”侧栏按钮 .....	17
	“收藏”侧栏按钮 .....	18
五、	前端部分 .....	19
	小组件的创建与使用 .....	19
	配合 bootstrap 搭建框架 .....	21
	yii 框架自带分页的使用 .....	22
	多表连接获取数据 .....	23
	检索工具的实现 .....	24
	controller 检查用户登录状态 .....	24

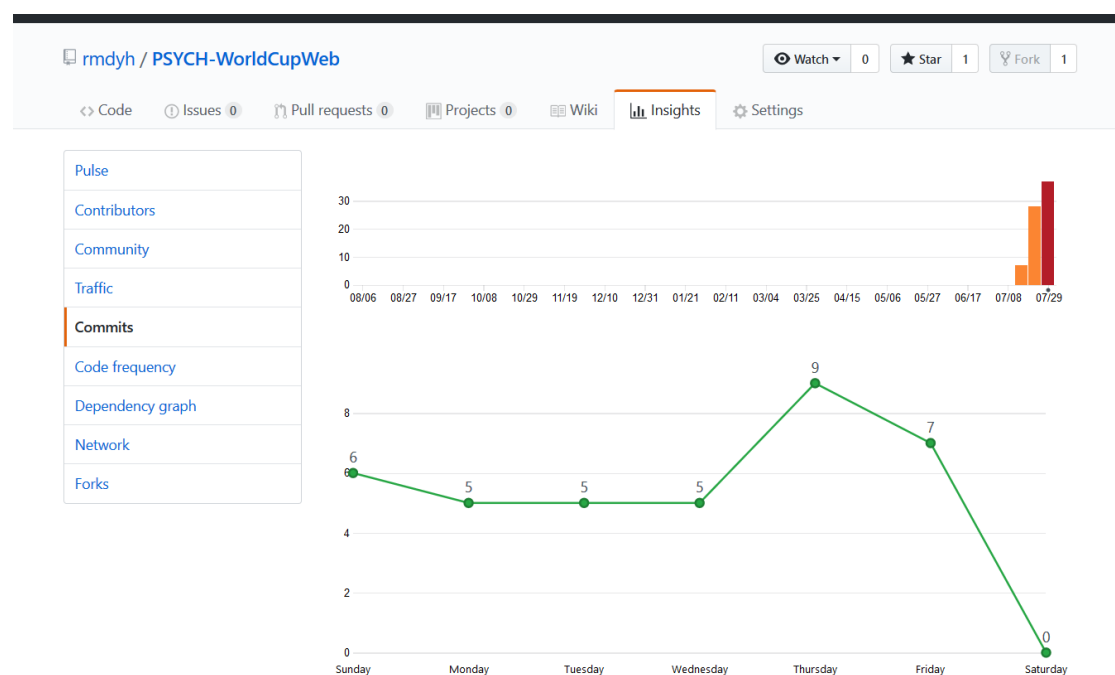
回复评论的实现 .....	25
文章编辑器的使用 .....	26
六、 后端部分 .....	27
七、 在服务器上搭建.....	28

# 一、团队合作

## 使用 github 进行团队合作

本项目采用的版本控制方式为 github，小组成员在 github 上 clone 下工程项目后在本地进行编写，将写完的改动 commit 后 push 到 github 远程端，即可方便地实现工程的版本控制和代码共享。

本项目的 github 地址为：<https://github.com/rmdyh/PSYCH-WorldCupWeb>



项目中 contributions 分别为：

rmdyh: 卢政先

xzy789: 徐子越

occulticplus: 文周之

lyj789: 刘一静

Collaborators	
	xzy789
	occulticplus
	lyj789

## 使用 dbv 工具同步数据库结构

本项目使用 dbv-1.1 对数据库结构进行版本控制。dbv-1.1 在项目根目录下的 dbv-1.1 中,使用仅需访问网址:<http://localhost/PSYCH-WorldCupWeb/dbv-1.1>, 输入用户名密码 (均为 dbv), 即可进行数据库结构同步, 最后 github 会将其一并并进行版本控制。

## 使用 navicat 同步数据库数据

为了对数据库中的数据进行同步, 我们使用 navicat 的数据同步功能。

## 分工情况

卢政先(组长): 主要负责前端(frontend)各个界面的 view 表现优化(index.css、knockoutCell.css 和前端各个小组件), 对 yii 框架各个功能的探索以及对工程进度的审核和修改。此外还负责用 c#进行数据爬取, 实现文档、用户手册文档以及设计文档的完善。

刘一静: 主要负责构建数据库、编写 controller 中的数据库查询修改操作以及后端的队伍管理、队员管理和比赛管理三个界面。此外还参与前端部分 view 界面

的编写，以及负责数据库大部分数据的爬取、录入操作、需求文档、设计文档、用户手册的编写。

文周之：主要负责研究文件上传功能和在线文章编辑器功能，以及后端的文章管理模块和用户权限操作模块的编写。此外还负责前端的用户界面的 view、controller 的编写和部署文档的编写。

徐子越：主要负责网站的美化设计，寻找图片及数据资源。

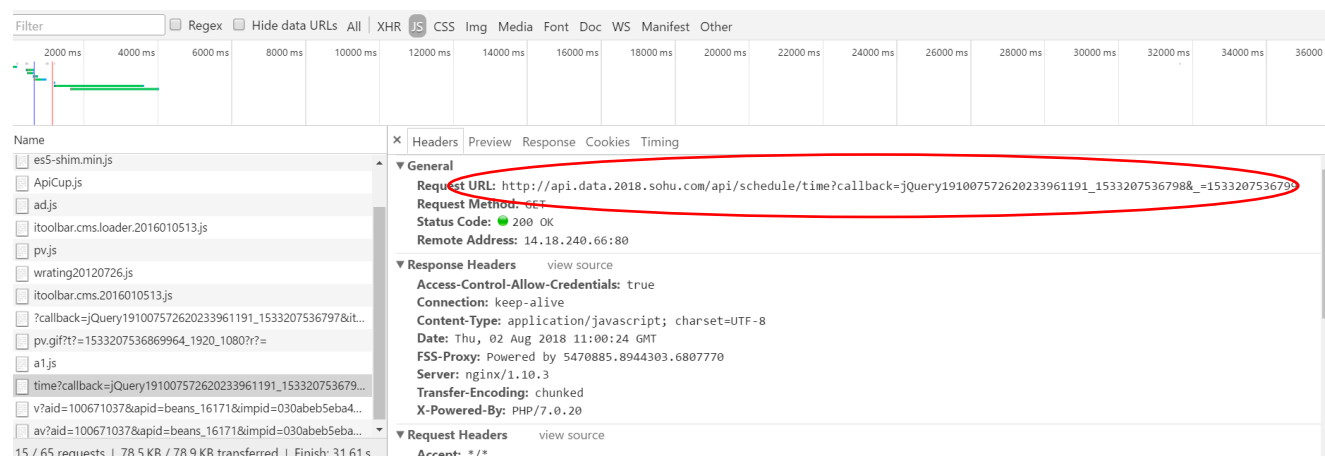
黄祎：主要负责数据库构架。

## 二、数据获取

### 使用在线工具获取数据

为了搭建 2018 年世界杯数据网站，我们必须建立一个数据全面准确的数据库作为前提，其中获取数据信息尤为关键，通过浏览多个世界杯数据网站，最终选择主要对搜狐体育的世界杯数据进行爬取（<http://data.2018.sohu.com/game-schedule.html?index=3>）下面以获取赛事信息为例详细说明爬取数据的实现过程：

首先通过右键查看源代码获取网页源代码，发现其中并没有包含页面中的所有元素。F12 查看 network，观察到网页中的关键数据信息都是通过 api 接口把包含数据的脚本传到界面上的。



访问该网址



据。具体 c#源代码如下：

获取队伍比赛信息：

```
using HtmlAgilityPack;
using System;
using System.IO;
using System.Net;
using System.Text;
using System.Drawing;

namespace ConsoleApp1
{
    class Program
    {
        public const String webur1 = "http://data.7m.com.cn/";
        public static String toFile = @"C:\Users\rmdyh\Desktop\datas\";
        public static String inFile = @"C:\Users\rmdyh\Desktop\";
        public static string[] countryA = new string[65];
        public static string[] countryB = new string[65];

        static void Main(string[] args)
        {
            string insert="";
            Program pg = new Program();
            if (!File.Exists("path.txt"))
            {
                Console.WriteLine("未能找到路径文档path.txt");
                return;
            }
            toFile=File.ReadAllText("path.txt");
            toFile=toFile+"\\ ";
            Console.WriteLine("路径为" + toFile+'\\n');

            string[] matchs = File.ReadAllLines("match.txt");
            for (int i = 1; i <= 64; i++)
            {
                countryA[i] = matchs[i - 1].Split(' ')[0];
                countryB[i] = matchs[i - 1].Split(' ')[1];
            }

            String s=pg.SpiderHTML(weburl+ "matches_data/149/gb/index.shtml",
            "data.7m.com.cn");
            HtmlAgilityPack.HtmlDocument doc = new HtmlAgilityPack.HtmlDocument();
            doc.LoadHtml(s);
```

```

var allCountry = doc.DocumentNode.SelectNodes("//div/a");
for(int np = 0; np < allCountry.Count; np++)
{
    HtmlNode country = allCountry[np];
    string title =
"http://team.7m.com.cn/v2/encrypt/fun/gethistorymatch.php?id=" +
country.Attributes["href"].Value.Split('/')[2] + "&lang=gb" ;

    String s1 = pg.SpiderHTML(title, "team.7m.com.cn");
    int number = 0;
    string[] aaa=s1.Split(' ');
    string[] aid=new string[10], bid=new string[10];
    for(int i = 0; i < aaa.Length; i++)
    {
        if (aaa[i] == "aid")
        {
            aid = aaa[i + 1].Substring(2, aaa[i + 1].Length -
4).Split(',');
        }
        if (aaa[i] == "bid")
        {
            bid = aaa[i + 1].Substring(2, aaa[i + 1].Length -
4).Split(',');
        }
        if (aaa[i] == "mid")
        {
            for (int j = aaa[i+1].IndexOf("149"); j != -1; j =
aaa[i+1].IndexOf("149", j + 1))
            {
                number++;
            }
        }
    }
    int cnt = 0, number2 = number;
    while (number > 0)
    {
        if (aaa[cnt].IndexOf("div>") != -1)
        {
            int id = 0;bool isfirst = true;
            if(bid[number2 - number] ==
country.Attributes["href"].Value.Split('/')[2])
            {
                isfirst = false;

```



```

    }
    for(int i = 1; i < 65; i++)
    {
        if(aaa[cnt].IndexOf(countryA[i])!=-1&&
aaa[cnt].IndexOf(countryB[i]) != -1)
        {
            if((isfirst &&
countryA[i]==country.InnerText )||(isfirst && countryB[i] == country.InnerText))
            {
                id = i;
                break;
            }
        }
    }
    int start = aaa[cnt].IndexOf("<td>");
    if (bid[number2-number] ==
country.Attributes["href"].Value.Split('/')[2])
    {
        for(int j = 1; j <= 10; j++)
        {
            start = aaa[cnt].IndexOf("<td>", start + 1);
        }
    }
    insert += "INSERT INTO `team_match` (`team`, `match_ID`,
`jiaoqiu`, `yuewei`, `shemen`, `huangpai`, `huanren`, `fangui`, `renyiqiu`,`jiuqiu`,
`kongqiulv`) VALUES (";
    insert += " '" + country.InnerText + "', '"+id+"', ";
    for(int i = 1; i < 10; i++)
    {
        start = aaa[cnt].IndexOf("<td>", start + 1);
        string temp = aaa[cnt].Substring(start + 4,
aaa[cnt].IndexOf("<", start + 1) - start - 4);
        insert += "'" + temp + "', ";
    }
    insert = insert.Substring(0, insert.Length - 1);
    insert += ");\n";
    number--;
}
cnt++;
}

}
File.WriteAllText("result.txt", insert);
}

```

```

public void SaveImg(String URL, String host, String address)
{
    HttpWebRequest Myrq = (HttpWebRequest)WebRequest.Create(URL);
    Myrq.KeepAlive = false;
    Myrq.Timeout = 30 * 1000;
    Myrq.Method = "GET";
    Myrq.Accept =
"text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8";
    Myrq.Host = host;
    Myrq.UserAgent = "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/49.0.2623.221 Safari/537.36 SE 2.X MetaSr 1.0";

    HttpWebResponse Myrp = (HttpWebResponse)Myrq.GetResponse();
    if (Myrp.StatusCode != HttpStatusCode.OK) return;

    //using (FileStream file = new FileStream("test.txt", FileMode.Create))
    //{
    //    Myrp.GetResponseStream().CopyTo(file);
    //}
    try
    {
        Bitmap bmp = new Bitmap(Myrp.GetResponseStream());
        bmp.Save(address);
    }
    catch (Exception)
    {
        SaveImg(URL, host, address);
    }
    return;
}

public string SpiderHTML(String URL, String host)
{
    HttpWebRequest Myrq = (HttpWebRequest)WebRequest.Create(URL);
    Myrq.KeepAlive = false;
    Myrq.Timeout = 30 * 1000;
    Myrq.Method = "GET";
    Myrq.Accept = "*/*";
    Myrq.Host = host;
    Myrq.UserAgent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0)
Gecko/20100101 Firefox/61.0";

```

```

        HttpResponseMessage Myrp = (HttpResponseMessage)Myrq.GetResponse();
        if (Myp.StatusCode != HttpStatusCode.OK) return null;

        //using (FileStream file = new FileStream("test.txt", FileMode.Create))
        //{
        //    Myrp.GetResponseStream().CopyTo(file);
        //}

        Stream ReceiveStream = Myrp.GetResponseStream();
        Encoding encode = System.Text.Encoding.GetEncoding("utf-8");

        StreamReader readStream = new StreamReader(ReceiveStream, encode);
        try
        {
            String result = readStream.ReadToEnd();
            return result;
        }
        catch (Exception)
        {
            return SpiderHTML(URL, host);
        }
    }
}

```

获取队员比赛表现信息：

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Newtonsoft.Json;
using System.IO;
using System.Net;

namespace ConsoleApp3
{
    public class Player
    {
        public int team_id { get; set; }
        public string team_name { get; set; }
        public int player_number { get; set; }
        public string player_id { get; set; }
        public string player_name_cn { get; set; }
        public string photo { get; set; }
    }
}

```

```

public class AllPlayers
{
    public int status { get; set; }
    public string reason { get; set; }
    public Player[] result { get; set; }
}

public class PlayerPer
{
    public int goalsconceded { get; set; }
    public int goalsscored { get; set; }
    public int assists { get; set; }
    public int offsides { get; set; }
    public int redcards { get; set; }
    public int yellowcards { get; set; }
    public int? starter { get; set; }
    public int duelstacklesuccessful { get; set; }
    public int passesshortsucessful { get; set; }
    public int shotsongol { get; set; }
    public int penaltiessaved { get; set; }
    public int passesmediumsuccessful { get; set; }
    public int interceptions { get; set; }
    public int shotsfacedsaved { get; set; }
    public int shotsblocked { get; set; }
    public int passeslongsuccessful { get; set; }
    public int shotsfacedtotal { get; set; }
    public int passesshorttotal { get; set; }
    public int passeslongtotal { get; set; }
    public int chancescreated { get; set; }
    public int shotsoffgoal { get; set; }
    public int duelsheadersuccessful { get; set; }
    public int passesmediumtotal { get; set; }
    public string player_name_cn { get; set; }
    public string home_team_outcome { get; set; }
    public string visiting_team_outcome { get; set; }
}

public class PlayerPers
{
    public int status { get; set; }
    public string reason { get; set; }
    public PlayerPer[] result { get; set; }
}

```

```
}
```

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
    {
```

```
        Program pg = new Program();
```

```
        string[] reflect = new string[750];
```

```
        /*
```

```
        int cnt = 0;
```

```
        string[] playerre = File.ReadAllText("player_db.txt").Split('\n');
```

```
        for(int i = 0; i < playerre.Length; i++)
```

```
        {
```

```
            reflect[i] = playerre[i].Split(' ')[1];
```

```
            reflect[i] = reflect[i].Substring(0, reflect[i].Length - 1);
```

```
            cnt++;
```

```
        }
```

```
        */
```

```
        string[] matchs = File.ReadAllLines("match.txt");
```

```
        string[] countryA = new string[65];
```

```
        string[] countryB = new string[65];
```

```
        for (int i = 1; i <= 64; i++)
```

```
        {
```

```
            countryA[i] = matchs[i - 1].Split(' ')[0];
```

```
            countryB[i] = matchs[i - 1].Split(' ')[1];
```

```
        }
```

```
        string jsonText = File.ReadAllText("allplayer.json");
```

```
        AllPlayers obj2 = JsonConvert.DeserializeObject<AllPlayers>(jsonText);
```

```
        for(int i = 0; i < obj2.result.Length; i++)
```

```
        {
```

```
            reflect[i] = obj2.result[i].player_id;
```

```
            /*
```

```
            int flag = 0;
```

```
            for(int j = 0; j < cnt; j++)
```

```
            {
```

```
                if (reflect[j]==obj2.result[i].player_name_cn)
```

```
                {
```

```
                    //reflect[j] = obj2.result[i].player_id;
```

```
                    flag++;
```

```
                }
```

```
            }
```

```
            if (flag > 1)
```

```
            {
```

```
                throw new Exception();
```

```

    }
    if (flag == 0)
    {
        reflect[cnt++] = obj2.result[i].player_name_cn;
    }
    */
}

string insert = "";
for(int i = 0; i < reflect.Length; i++)
{
    string url =
"http://api.data.2018.sohu.com/api/stats/player?player_id=" + reflect[i];
    jsonText = pg.SpiderHTML(url, "api.data.2018.sohu.com");
    PlayerPers obj = JsonConvert.DeserializeObject<PlayerPers>(jsonText);
    for (int j = 0; j < obj.result.Length; j++)
    {
        PlayerPer temp = obj.result[j];
        int matchid = 0;
        for(int z = 1; z < countryA.Length; z++)
        {
            if(temp.home_team_outcome==countryA[z] &&
temp.visiting_team_outcome == countryB[z])
            {
                matchid = z;
                break;
            }
        }
        if (matchid == 0)
        {
            throw new Exception();
        }
        insert += "INSERT INTO `player_match` (`player_ID`, `match_ID`,
`shoufa`, " +
            "`jinqiu`, `zhugong`, `shezheng`, `huangpai`, `hongpai`,
`qiangduan`, `fengdu`, " +
            "`lanjie`, `touqiu`, `shepian`, `chuanqiu`, `chengchuan`,
`yuewei`, `shiqiu`, `pujiucheng`, " +
            "`pujiuzong`, `dianqiucheng`, `changchuan`, `jihui`) VALUES
(' "+
            (i+1)+"" , "" +
            matchid+"" , "" +
            temp.starter+"" , "" +
            temp.goalsscored + "" , "" +

```

```

        temp.assists+"", ' " +
        temp.shotsongol + "", ' " +
        temp.yellowcards+"", ' " +
        temp.redcards+"", ' " +
        temp.duelstacklesuccessful + "", ' " +
        temp.shotsblocked + "", ' " +
        temp.interceptions + "", ' " +
        temp.duelsheadersuccessful + "", ' " +
        temp.shotsoffgoal + "", ' " +
        ((int)temp.passesshorttotal +(int)temp.passesmediumtotal
+(int)temp.passeslongtotal) + "", ' " +
        (temp.passesshortsuccessful +temp.passesmediumsuccessful
+temp.passeslongsuccessful) + "", ' " +
        temp.offsides + "", ' " +
        temp.goalsconceded + "", ' " +
        temp.shotsfacedsaved + "", ' " +
        temp.shotsfacedtotal+"", ' " +
        temp.penaltiessaved + "", ' " +
        temp.passeslongsuccessful + "", ' " +
        temp.chancescreated + "" );\n";

    }
}

File.WriteAllText("result.txt", insert);

}

```

```

public string SpiderHTML(String URL, String host)
{
    HttpWebRequest Myrq = (HttpWebRequest)WebRequest.Create(URL);
    Myrq.KeepAlive = false;
    Myrq.Timeout = 30 * 1000;
    Myrq.Method = "GET";
    Myrq.Accept = "*/*";
    Myrq.Host = host;
    Myrq.UserAgent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0)
Gecko/20100101 Firefox/61.0";

    HttpWebResponse Myrp = (HttpWebResponse)Myrq.GetResponse();
    if (Myrp.StatusCode != HttpStatusCode.OK) return null;
}

```

```

        //using (FileStream file = new FileStream("test.txt", FileMode.Create))
        //{
        //    Myrp.GetResponseStream().CopyTo(file);
        //}

        Stream ReceiveStream = Myrp.GetResponseStream();
        Encoding encode = System.Text.Encoding.GetEncoding("utf-8");

        StreamReader readStream = new StreamReader(ReceiveStream, encode);
        try
        {
            String result = readStream.ReadToEnd();
            return result;
        }
        catch (Exception)
        {
            return SpiderHTML(URL, host);
        }
    }
}
}

```

### 三、前后端模版使用

#### 前端模版的使用

前端本次并没有直接使用模版，而是直接通过写 index.css 的形式构建了网站。

在构成网站的时候对

<http://sc.chinaz.com/mobandemo.aspx?downloadid=201723056261&qq-pf->

[to=pcqq.group](http://sc.chinaz.com/mobandemo.aspx?downloadid=201723056261&qq-pf-to=pcqq.group)

中的模版进行了一定程度的参考。

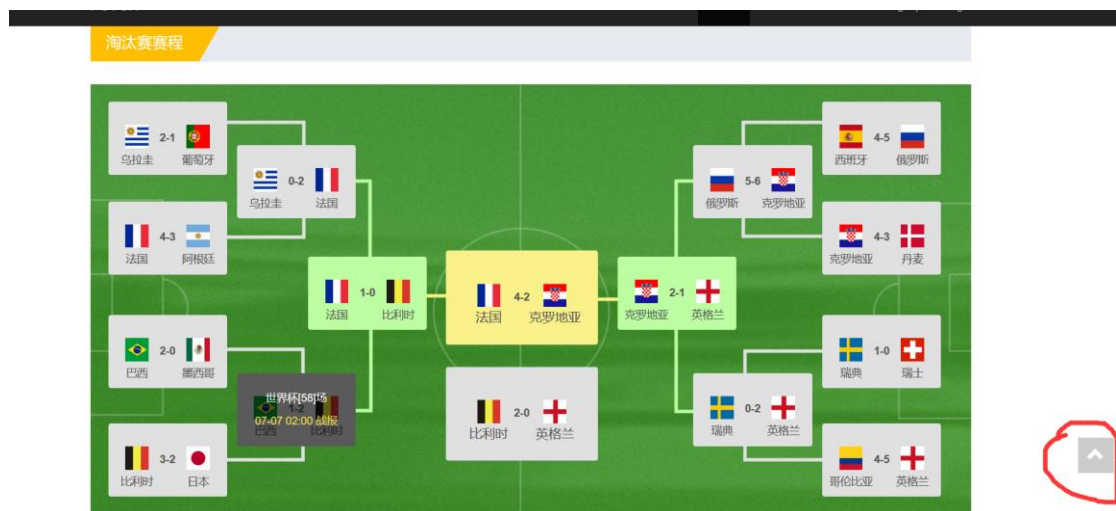
#### 后端模版的使用

后端模版使用了老师在课上提供的 absolute，在此基础上进行了一定的修改和删除。



## 四、前端侧边按钮

“back-to-top” 侧栏按钮



back-to-top 按钮出现在网页的右下角，当页面向下滚动到一定程度的时候会出现，点击后会自动滚动到页面顶部。具体实现为：

在 main\_layout.php 中写入如下代码

```
<a class="back-to-top invalidate" title="回到顶部">
|   <span class="glyphicon glyphicon-chevron-up"></span>
</a>
```

在 index.js 中写入如下代码

```
$(document).scroll(function () {
    let backTop = $(".back-to-top");
    if ($(document).scrollTop() > 200) {
        if (backTop.hasClass("invalidate")) {
            backTop.removeClass("invalidate");
        }
    }
    else {
        if (!backTop.hasClass("invalidate")) {
            backTop.addClass("invalidate");
        }
    }
})
```

并在 index.css 中写入相应的样式即可实现。

## “收藏”侧栏按钮



收藏侧栏按钮会在查看文章详细信息的时候显示，具体实现如下：

在 view/passage/view.php 中写入如下代码

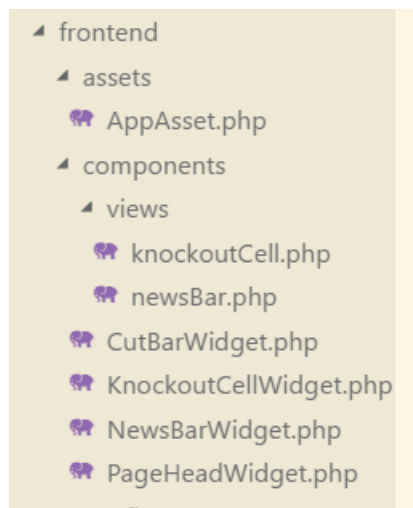
```
<? = Html::beginTag('a', ['class' => "fav-icon", 'title' => "喜欢! 收藏这篇文章", 'href' => "?r=passage/favorite&Favorite[passage_ID]=" . $model->ID])
| <span class="glyphicon glyphicon-heart"></span>
</a>
```

并添加相应的 controller 即可实现收藏侧栏按钮。

## 五、前端部分

### 小组件的创建与使用

在前端中用到了一些自己创建的小组件，其目录为 frontend/components



共创建了四个组件，其中 KnockoutCellWidget 用于主页中显示淘汰赛的对阵信息；NewsBarWidget 用于显示文章列表；PageHeadWidget 用于显示一个小页面的头部。

KnockoutCellWidget:



NewsBarWidget:

宽屏时 (lg):



## 2018年世界杯:足坛风云变幻 中国足球需奋勇直追

2018年世界杯:足坛风云变幻 中国足球需奋勇直追 15日,一场瓢泼大雨中,法国队再次高高举起大力神杯,2018年俄罗斯世界杯随之大幕落下。法国、比利时、英格兰等劲旅刮起一场青春风暴,冲击着旧有的世界足坛格局。面对风云变幻的世界足

[READ MORE](#)

作者: 123456

81

4

7

中屏时 (md):



## 2018年世界杯:足坛风云变幻 中国足球需奋勇直追

2018年世界杯:足坛风云变幻 中国足球需奋勇直追 15日,一场瓢泼大雨中,法国队再次高高举起大力神杯,2018年俄罗斯世界杯随之大幕落下。法国、比利时、英格兰等劲旅刮起一场

[READ MORE](#)

作者: 123456

82

4

7

平板时 (sm):



## 2018年世界杯:足坛风云变幻 中...

2018年世界杯:足坛风云变幻 中国足球需奋勇直追 15日,一场瓢泼大雨中,法国队再次高高举起大力神杯,2018年

[READ MORE](#)

作者: 123456

82

4

7

手机时 (xs):

## 2018年世界杯:足坛风云变幻 中国足球需...

2018年世界杯:足坛风云变幻 中国足球需奋勇直追 15日,一场瓢泼大雨中,法国队再次高高举起大力神杯,2018年俄罗斯世界杯随之大幕落下。法国、

[READ MORE](#)

作者: 123456

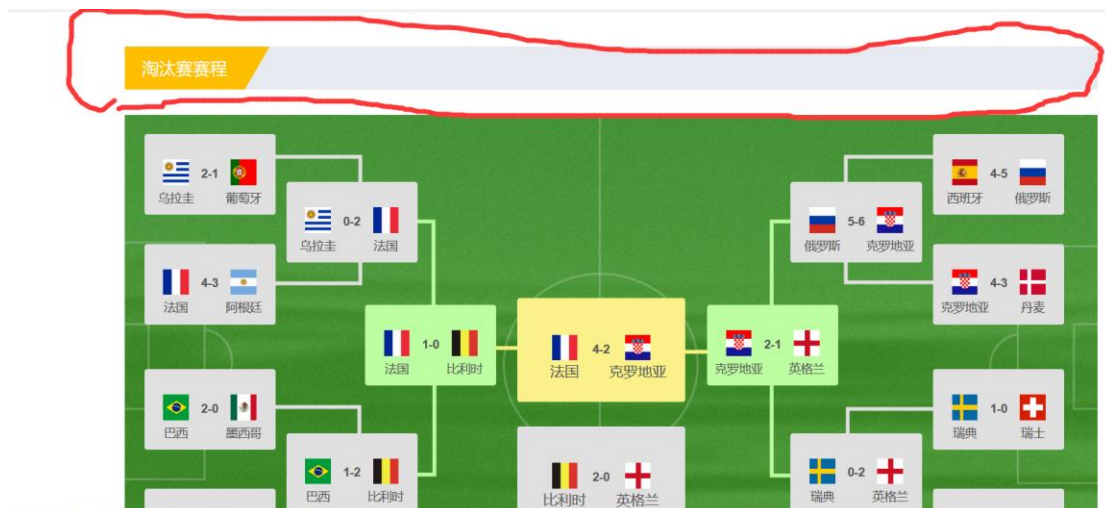
82

4

7

PageHeadWidget:

border 无时样式:



border 有时样式:

球队基本信息		
乌拉圭		
	斯巴雷斯	12次
	球队教练	参加世界杯次数
	1900年	2次
	球队成立时间	冠军次数
	1923年	14名
	加入世界杯时间	世界排名

配合 bootstrap 搭建框架

在前端中用到了很多 bootstrap 全局样式来搭建网站，例如主页下的

射手榜		
球员姓名	所属国家	进球数
哈里·凯恩	英格兰	6 个
丹尼斯-切里舍夫	俄罗斯	4 个
C罗	葡萄牙	4 个
格列兹曼	法国	4 个
姆巴佩	法国	4 个
卢卡库	比利时	4 个
曼朱基奇	克罗地亚	3 个
佩里西奇	克罗地亚	3 个
卡瓦尼	乌拉圭	3 个
米纳	哥伦比亚	3 个

助攻榜		
球员姓名	所属国家	助攻数
戈洛温	俄罗斯	2 个
卡洛斯-桑切斯	乌拉圭	2 个
J罗	哥伦比亚	2 个
金特罗	哥伦比亚	2 个
久巴	俄罗斯	2 个
库蒂尼奥	巴西	2 个
维克多-赫拉森	瑞典	2 个
格列兹曼	法国	2 个
阿扎尔	比利时	2 个
卢卡斯-埃尔南德斯	法国	2 个

采用了 bootstrap 的栅格系统和 table 表格样式等等，以构造一个有良好的窗口自适应的网页，其部分代码如下（view/site/index.php）

```

<div class="row">
  <div class="col-md-6">
    <?php PageHeadWidget::begin([
      'text' => "射手榜",
      'hasBorder' => true,
    ]) ?>
    <div class="page-wrap">
      <table class="table table-striped" style="margin-top:0">
        <thead>
          <tr>
            <th>
              球员姓名
            </th>
            <th>
              所属国家
            </th>
            <th>
              进球数
            </th>
          </tr>
        </thead>
        <tbody>

```

## yii 框架自带分页的使用

在文章列表中采用了 yii 框架自带的分页,当文章数量超过 5 个时会出现分页栏。

其实现为:

(controllers/passageController.php)

```

40 | */
41 | public function actionIndex()
42 | {
43 |     // $passage = Passage::find()->all();
44 |     // return $this->render('index', [
45 |     //     'passage' => $passage,
46 |     // ]);
47 |     $query = Passage::find()->where(['status'=>'success']);
48 |     $count= $query->count();
49 |     $pagination = new Pagination(['totalCount' => $count, 'pageSize'=>5]);
50 |     $passage=$query->offset($pagination->offset)
51 |         ->limit($pagination->limit)
52 |         ->all();
53 |     return $this->render('passage',[
54 |         'passage'=>$passage,
55 |         'pagination' => $pagination,
56 |     ]);
57 | }
58 |

```

(view/passage/passage.php)

```
<?= LinkPager::widget([
|   'pagination' => $pagination,
|]);
```

## 多表连接获取数据

在前端中大量用到了多表链接（尤其是在文章详细界面中），其多表链接建立如下：(models/passage.php)

```
66 | */
67 | public function getFavorites()
68 | {
69 |     return $this->hasMany(Favorite::className(), ['passage_ID' => 'ID']);
70 | }
71 |
72 | /**
73 |  * @return \yii\db\ActiveQuery
74 |  */
75 | public function getMatches()
76 | {
77 |     return $this->hasMany(Match::className(), ['url' => 'ID']);
78 | }
79 |
80 | public function getPassageKeys()
81 | {
82 |     return $this->hasMany(PassageKey::className(), ['passage_ID' => 'ID']);
83 | }
84 |
85 | public function getComments()
86 | {
87 |     return $this->hasMany(Comment::className(), ['passage_ID' => 'ID']);
88 | }
89 | }
90 |
```

使用：(view/passage/view.php)

```
110 | <p>
111 | | <?= $relation->seen?>
112 | | </p>
113 | </div>
114 | <div class="col-md-4 col-xs-2">
115 | | <span class="glyphicon glyphicon-heart" aria-hidden="true" style="display: inline-block"></span>
116 | | <p>
117 | | | <?= sizeof($relation->favorites)?>
118 | | | </p>
119 | | </div>
120 | <div class="col-md-4 col-xs-2">
121 | | <span class="glyphicon glyphicon-comment" aria-hidden="true" style="display: inline-block"></span>
122 | | <p>
123 | | | <?= sizeof($relation->comments)?>
124 | | | </p>
125 | | </div>
126 | </div>
127 | </div>
```



## 检索工具的实现

在队员一览中，用到了检索工具，其实现如下：(PlayerController.php)

```
Yii::$app->request->queryParams
```

获得 get 参数并传入 searchModel 中执行 search，返回 dataProvider，显示出来。

```
42 public function actionIndex()  
43 {  
44     // $searchModel = new PlayerSearch();  
45     // $dataProvider = $searchModel->search(Yii::$app->request->queryParams);  
46  
47     // return $this->render('index', [  
48     //     'searchModel' => $searchModel,  
49     //     'dataProvider' => $dataProvider,  
50     // ]);  
51     $searchModel = new PlayerSearch();  
52     $dataProvider = $searchModel->search(Yii::$app->request->queryParams);  
53  
54     $players=array();  
55     foreach ($dataProvider->getModels() as $player) {  
56         if(!isset($players[$player->country])){  
57             $players[$player->country]=array();  
58         }  
59         array_push($players[$player->country],$player);  
60     }  
61  
62     return $this->render('index',[  
63         'players'=>$players,  
64         'model' => $searchModel,  
65         'data'=>$dataProvider->getModels(),  
66     ]);  
67 }  
68
```

在 view 中 (view/player.index):

```
8 <?php beginForm(['index'], 'get', ['enctype' => 'multipart/form-data', 'class' => 'form-inline']) ?>  
9 <?php  
10     echo "球队检索: ";  
11     echo Html::activeDropDownList($model, 'country', array_merge([""] => "全部", ArrayHelper::map($data, 'country', 'country')));  
12 >  
13 <div class="form-group">  
14     <div class="col-lg-offset-1 col-lg-11">  
15         <?php submitButton('查询', ['class' => 'btn btn-primary']) ?>  
16     </div>  
17 </div>  
18  
19 <!--button class="btn btn-default" type="submit">确定</button-->
```

即可实现。

## controller 检查用户登录状态

在评论文章时，需要检查用户是否已经登录。方法为在 controller 中进行一次判

断，如果没有登录则跳转到登录界面。代码如下：(passageController.php)



```

114 // $model->save();
115 if (Yii::$app->user->isGuest)
116 {
117     Yii::$app->user->setReturnUrl(Yii::$app->request->getUrl()); // 设置返回的url, 登录后原路返回
118     Yii::$app->user->loginRequired();
119     Yii::$app->end();
120 }
121 else{
122     date_default_timezone_set("Asia/Shanghai");
123     if($model->load(Yii::$app->request->post()))
124     {
125         $model->user_ID = Yii::$app->user->identity->ID;
126         $model->date = date("Y-m-d h:i:sa");
127         $model->username= Yii::$app->user->identity->username;
128         $model->status = "normal";
129         if($model->save()){
130             return $this->redirect(['view', 'id' => $model->passage_ID]);
131         }else{
132             var_dump("Something Wrong!");
133             return;
134         }
135     } else {
136         // return $this->redirect(['view', 'id' => 2]);
137         //return $this->render(Yii::$app->request->hostInfo);
138     }
139 }
140 }

```

## 回复评论的实现

回复评论功能，实质上是在评论表单中添加了一个隐藏的 input 元素，用来记录用户即将发送的评论的“parent”。如果是对文章进行评论，则“parent”为 0，否则“parent”的值是评论的 ID 号。在显示评论的时候会查看将要显示的评论的“parent”属性（数据库中有相应属性）来决定如何显示评论。

... 111111 2018-08-01 10:55:47pm  
黑化股份  
[回复](#)

回复 111111 在 2018-08-01 10:55:47pm 的评论:  
黑化股份

... 111111 2018-08-02 02:20:51pm  
查询出  
[回复](#)

... 111111 2018-08-02 02:21:24pm  
是的菜市场  
[回复](#)

评论内容

[提交](#)

样式如上。

点击回复评论后

您现在正在回复他人的评论。[取消回复](#)

评论内容

提交

点击回复和取消回复的 js 代码如下（view/passage/view.php）

```
35  */
36  function reply(recname, content) {
37      /*
38      $.get(window.Location.href + (!getUrlParam('repid')?('&repid=' + recid):') + '#comment',
39          function(data){
40              return data;
41          });
42      alert(window.Location.href + (getUrlParam('repid')?('&repid=' + recid):') + '#comment');
43      var something = getUrlParam('repid');
44      */
45      $('#comment-parent').val(recname);
46      $('.comment-tip').addClass("comment-tip-show");
47      //$('#commentarea').val("< Re: " + recname + " content: " + content + ">");
48  }
49  function cancelReply() {
50      $('#comment-parent').val('');
51      $('.comment-tip').removeClass("comment-tip-show");
52  }
```

## 文章编辑器的使用

文章编辑器使用的是适合 yii2 框架的 redactor 工具，由于 redactor 可以直接适用在 yii/activeform 下，所以使用的时候配合 post 可以达成信息交流的效果。

安装方法：利用 composer 进行安装。

```
composer require --prefer-dist yiidoc/yii2-redactor "*"

```

配置好配置文件

```
'modules' => [
    'redactor' => 'yii\redactor\RedactorModule',
    'user' => [
        // here is the config for user
    ],
],
```

使用实例：user/write&id=xxx;

```
echo $form->field($psg, attribute: 'title')->textInput();// textarea;
echo $form->field($psg, attribute: 'author_ID')->textInput()->hiddenInput(['value'=>$model->id])->label( label: false);
echo $form->field($psg, attribute: 'author')->textInput()->hiddenInput(['value'=>$model->username])->label( label: false);
echo $form->field($psg, attribute: 'seen')->textInput()->hiddenInput(['value'=>0])->label( label: false);
echo $form->field($psg, attribute: 'content')->widget(\yii\redactor\widgets\Redactor::className());
echo Html::submitButton( content: '提交', ['class'=>'btn btn-primary']);
ActiveForm::end();
```

可以看到使用了 yii\redactor

效果图

写文章  
您的文章审核后才不会被发表。

文章标题

文章内容

提交

## 六、 后端部分

考虑到数据库的可维护性，我们重定义删除操作：不再直接删除数据库当中的对应元组，而是把对应元组的状态值改变，之后在进行操作的时候仅仅筛选对应状态的元组。

一个实例如 backup\controllers\usercontroller 下的 actiondelete 方法

```
public function actionDelete($id)
{
    $target = $this->findModel($id);
    //var_dump(Yii::$app->request->post());
    //$target = user::findone($id);
    //var_dump($target);
    $target->status = 0; // 0 is think to be deleted condition.
    $target->save();
    //$this->findModel($id)->delete();

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

方法中对于删除操作，仅仅将 passage 下对应 id 的元素的 status 改为'deleted'，因为即使用户被删除了，用户的发言和文章也不应该直接被删除，这样会造成数据损失（因为对应的其他用户的信息会损失）

后端的界面实现，除使用 absolute 之外，表格的呈现有两种形式：1.利用 yii 自带的 girdview-widget 组件系，实现 crud。2.对于某些非资讯类的信息，即用户

相关的操作, 后端无权限进行完全的增删改查, 于是对这一部分代码进行了重构。  
一个实例是用户信息管理界面

User Admin				
userid	username	email	signuptime	operation
1	123		1970-01-01	<button>删除此用户</button>
5	321	123	1970-01-01	<button>删除此用户</button>
11	114514	acceed@asm.com	2018-08-03	<button>删除此用户</button>

此界面的实现使用了 bootstrap 自带的 hovetable 组件, 代码:

```
<table class="table table-hover">
  <thead>
    <tr>
      <th>userid</th>
      <th>username</th>
      <th>email</th>
      <th>signuptime</th>
      <th>operation</th>
    </tr>
  </thead>
  <tbody>
    <?php $users = user::find()->where(['status'=>'10'])->offset($pagination->offset)->limit($pagination->limit)->all
    <foreach($users as $user):?>
      <tr>
        <td>
          <?= $user->id?>
        </td>
        <td>
          <?= $user->username?>
        </td>
        <td>
          <?= $user->email?>
        </td>
        <td>
          <?= date( format: 'Y-m-d', $user->created_at)?>
        </td>
        <td>
          <button class="btn btn-success" onclick="deluser(<?=$user->id?>)">删除此用户</button>
        </td>
      </tr>
    </foreach>
  </tbody>
</table>
```

## 七、在服务器上搭建

我们选择了腾讯服务器, 在里面安装了 windows 的系统后, 按照部署文档对项目进行部署, 最终成功搭建。

网址为:

http://123.206.54.140:8080/PSYCH-WorldCupWeb/frontend/web/index.php?r=site%2Findex