



北京航空航天大学
BEIHANG UNIVERSITY

数据库系统原理课程设计

深度学习文献检索系统

系统实现报告

北京航空航天大学

计算机学院

陈麒先，牛雅哲

16061160，16131059

二〇一八年十二月

郑重声明

关于诚实守信公约：

本实验报告中参考了文献或互联网资料的部分均有引用标注，另有与老师或同学研究后的成果引用将在特别致谢中说明，其余部分均为本人通过查阅课件、课堂笔记和教材后独立思考的结果。另，实验报告撰写仓促，如有错误，在所难免，欢迎批评指正！特此声明。

陈麒先，牛雅哲

原创性声明

本实验报告未注释部分内容由作者原创。抄袭行为在任何情况下都是不能容忍的 (COPY is strictly prohibited under any circumstances)！转载或引用须征得作者本人同意，并注明出处！**勿谓言之不预！**

陈麒先，牛雅哲



目录

一、实验概述	1
1、开发环境与技术说明	1
2、介绍所用的 WEB 服务器	1
3、开发语言	2
4、数据库管理系统	3
二、系统功能分析	4
1、注册与登陆	4
2、用户信息管理	5
3、管理员后台管理功能	5
4、论文检索查阅评论新增功能	5
5、主题讨论帖功能	5
三、数据库表的定义	5
1、管理员信息表 (ADMIN_CHECK_TAB)	5
2、用户信息表 (USER_LIST)	6
3、类别目录表 (CATEGORY)	6
4、论文类别表 (ARTICLE_CATEGORY)	6
5、论文信息表 (ARTICLE)	6
6、讨论帖表 (POSTS)	7
7、讨论帖回复信息表 (POSTS_INFO)	7
8、论文评论信息表 (COMMENTS)	7
四、存储过程与触发器	8
1、存储过程	8
(1) 用户注册信息登录至数据库存储过程 (create_user)	8
(2) 用户贡献资源登录至数据库存储过程 (create_user)	8
2、触发器	9

五、系统演示	9
1、初始登录界面	9
2、用户登录界面	10
3、用户注册界面	10
4、管理员登录	11
5、管理员操作	11
6、管理员访问数据库查询结果显示	11
7、主界面	12
8、论坛页面	12
9、发布讨论帖	13
10、查看讨论帖内容与参与讨论	13
11、查看主题文章列表详细信息	14
12、查看文章详细内容与评论	15
13、文献资源贡献	16
14、小额赞助界面	16
六、系统实现技术与程序结构解读	17
1. 程序结构解读	17
2. 系统实现技术	17
(1) 前端技术	17
(2) 后端技术	18
(3) 数据通信流图	18
七、体会与感想	18

一、实验概述

1、开发环境与技术说明

【操作系统】

Windows

【集成开发环境】

WampServer 2.5,

Apache : 2.4.9,

MySQL : 5.6.17 ,

PHP : 5.5.12 ,

PHPMyAdmin : 4.1.14 ,

SqlBuddy : 1.3.3,

XDebug : 2.2.5

【系统】

处理器: Intel(R) Core™ i5-3239M CPU @ 2.60GHz

RAM : 16.00 GB

系统类型: 64 位操作系统

2、介绍所用的 web 服务器

【Wamp Server】

WampServer 是一款由法国人开发的 Apache Web 服务器、PHP 解释器以及 MySQL 数据库的整合软件包。免去了开发人员将时间花费在繁琐的配置环境过程,从而腾出更多精力去做开发。

WampServer 就是 Windows Apache Mysql PHP 集成安装环境,即在 window 下的 apache、php 和 mysql 的服务器软件。

【特点】

- (1) 支持中文语言,一键安装,省时省力;任何人都可以轻松搭建;
- (2) 集成 Apache/MySQL/PHP/PhpMyadmin;支持 PHP 扩展、Apache 的 mod_rewrite;
- (3) 一键启动、重启、停止所有服务,一键切换到离线状态等等。

3、开发语言

【PHP】

PHP（外文名:PHP: Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。语法吸收了 C 语言、Java 和 Perl 的特点，利于学习，使用广泛，主要适用于 Web 开发领域。PHP 独特的语法混合了 C、Java、Perl 以及 PHP 自创的语法。它可以比 CGI 或者 Perl 更快速地执行动态网页。用 PHP 做出的动态页面与其他的编程语言相比，PHP 是将程序嵌入到 HTML（标准通用标记语言下的一个应用）文档中去执行，执行效率比完全生成 HTML 标记的 CGI 要高许多；PHP 还可以执行编译后代码，编译可以达到加密和优化代码运行，使代码运行更快。

PHP 原始为 Personal Home Page 的缩写，已经正式更名为 "PHP: Hypertext Preprocessor"。注意不是“Hypertext Preprocessor”的缩写，这种将名称放到定义中的写法被称作递归缩写。PHP 于 1994 年由 Rasmus Lerdorf 创建，刚刚开始是 Rasmus Lerdorf 为了要维护个人网页而制作的一个简单的用 Perl 语言编写的程序。这些工具程序用来显示 Rasmus Lerdorf 的个人履历，以及统计网页流量。后来又用 C 语言重新编写，包括可以访问数据库。他将这些程序和一些表单直译器整合起来，称为 PHP/FI。PHP/FI 可以和数据库连接，产生简单的动态网页程序。

在 1995 年以 Personal Home Page Tools (PHP Tools) 开始对外发表第一个版本，Lerdorf 写了一些介绍此程序的文档。并且发布了 PHP1.0！在这的版本中，提供了访客留言本、访客计数器等简单的功能。以后越来越多的网站使用了 PHP，并且强烈要求增加一些特性。比如循环语句和数组变量等等；在新的成员加入开发行列之后，Rasmus Lerdorf 在 1995 年 6 月 8 日将 PHP/FI 公开发布，希望可以透过社群来加速程序开发与寻找错误。这个发布的版本命名为 PHP 2，已经有 PHP 的一些雏型，像是类似 Perl 的变量命名方式、表单处理功能、以及嵌入到 HTML 中执行的能力。程序语法上也类似 Perl，有较多的限制，不过更简单、更有弹性。PHP/FI 加入了对 MySQL 的支持，从此建立了 PHP 在动态网页开发上的地位。到了 1996 年底，有 15000 个网站使用 PHP/FI。

在 1997 年，任职于 Technion IIT 公司的两个以色列程序设计师：Zeev Suraski 和 Andi Gutmans，重写了 PHP 的剖析器，成为 PHP 3 的基础。而 PHP 也在这个时候改称为 PHP: Hypertext Preprocessor。经过几个月测试，开发团队在 1997 年 11 月发布了 PHP/FI 2。随后就开始 PHP 3 的开放测试，最后在 1998 年 6 月正式发布 PHP 3。Zeev Suraski 和 Andi Gutmans 在 PHP 3 发布后开始改写 PHP 的核心，这个在 1999 年发布

的剖析器称为 Zend Engine，他们也在以色列的 Ramat Gan 成立了 Zend Technologies 来管理 PHP 的开发。

在 2000 年 5 月 22 日，以 Zend Engine 1.0 为基础的 PHP 4 正式发布，2004 年 7 月 13 日则发布了 PHP 5，PHP 5 则使用了第二代的 Zend Engine。PHP 包含了许多新特色，像是强化的面向对象功能、引入 PDO（PHP Data Objects，一个存取数据库的延伸函数库）、以及许多效能上的增强。PHP 4 已经不会继续更新，以鼓励用户转移到 PHP 5。

2008 年 PHP 5 成为了 PHP 唯一的有在开发的 PHP 版本。将来的 PHP 5.3 将会加入 Late static binding 和一些其他的功能强化。PHP 6 的开发也正在进行中，主要的改进有移除 register_globals、magic quotes 和 Safe mode 的功能。

PHP 最新稳定版本：5.4.30(2013.6.26)

PHP 最新发布的正式版本：5.5.14(2014.6.24)

PHP 最新测试版本：5.6.0 RC2(2014.6.03)

2013 年 6 月 20 日，PHP 开发团队自豪地宣布推出 PHP 5.5.0。此版本包含了大量的新功能和 bug 修复。需要开发者特别注意的一点是不再支持 Windows XP 和 2003 系统。

2014 年 10 月 16 日，PHP 开发团队宣布 PHP 5.6.2 可用。四安全相关的错误是固定在这个版本，包括修复 cve-2014-3668，cve-2014-3669 和 cve-2014-3670。所有的 PHP 5.6 鼓励用户升级到这个版本。

4、数据库管理系统

【navicat】

Navicat 是一套快速、可靠并价格相宜的数据库管理工具，

专为简化数据库的管理及降低系统管理成本而设。它的设计符合数据库管理员、开发人员及中小企业的需要。Navicat 是以直觉化的图形用户界面而建的，让你可以以安全并且简单的方式创建、组织、访问并共用信息。

Navicat 是闻名世界、广受全球各大企业、政府机构、教育机构所信赖，更是从业员每天必备的工作伙伴。自 2001 年以来，Navicat 已在全球被下载超过 2,000,000 次，并且已有超过 70,000 个用户的客户群。《财富》世界 500 强中有超过 100 家公司也都正在使用 Navicat。

Navicat 提供多达 7 种语言供客户选择，被公认为全球最受欢迎的数据库前端用户界面工具。

它可以用来对本机或远程的 MySQL、SQL Server、SQLite、Oracle 及 PostgreSQL 数据库进行管理及开发。

Navicat 的功能足以符合专业开发人员的所有需求，而且对数据库服务器的新手来说又相当容易学习。有了极完备的图形用户界面 (GUI)，Navicat 让你可以以安全且简单的方法创建、组织、访问和共享信息。

Navicat 适用于三种平台 - Microsoft Windows、Mac OS X 及 Linux。它可以让用户连接到任何本机或远程服务器、提供一些实用的数据库工具如数据模型、数据传输、数据同步、结构同步、导入、导出、备份、还原、报表创建工具及计划以协助管理数据。

二、系统功能分析

深度学习论文交流平台的设计目标是：现在是一个信息爆炸的时代，通过搜索引擎等方式可以非常容易地获取大量信息，但是随之而来的问题是如何去分辨梳理出最有价值的信息。深度学习作为一个近些年来热度很高的新领域，有各种各样的教程和资料，但是对于入门者来说如何去挑选资料是一个比较大的难题。因此，我们决定设计这样一个深度学习论文交流平台，将最经典最有价值的论文按照一定的类别整理起来，为学习者和开发者提供一个可以根据研究方向和所需的具体知识内容来查阅论文的平台，同时还可以在平台上对论文的内容进行讨论，或者是针对一个主题进行发帖讨论，同时，用户还可以向平台的根据论文类别添加新的论文。所以，本系统应该具有以下几个功能：

1、注册与登陆

用户在使用本网站之前需要首先进行注册，否则将无法查阅论文并参与讨论。在进行注册时，用户需要输入注册所用的用户名、密码，联系电话及邮箱等信息。在注册完成后，用户便可以通过登录界面进行登陆或是自动跳转进入平台，这时就可以使用平台的全部功能，分门别类查阅论文并进行相关讨论。除此之外，本系统设置后台管理员，管理员可以对论文和讨论帖进行管理和编辑，为平台新增内容并规范监督用户的行为。本系统具有用户、后台管理员两种角色。两种角色分别可以使用不同的方式进行登陆，并且分属不同的角色组，具有不同的操作权限。因而本系统具有权限管理的功能，可以在一定程度上保证系统的安全性。

2、用户信息管理

用户可查看自己的个人资料，由于用户所发表的评论会显示为真实姓名，故用户可通过修改个人信息确保个人资料的安全性。

3、管理员后台管理功能

管理员通过管理员口令认证，即可登录至后台完成管理员操作。我们默认管理员是具有数据库领域知识的专业人士，管理员对系统数据库的操作必须通过 SQL 语句，查询结果显示为 PHP 默认的查询结果。

4、论文检索查阅评论新增功能

本平台将深度学习的论文划分为如下几个小的分类：监督学习，非监督学习，强化学习，数据处理，损失函数，卷积神经网络。用户可以通过类别进行检索，也可以通过论文名称进行检索。检索到论文之后可以查看论文的名称，作者，摘要，链接等基本信息，如需进一步了解可以打开链接继续深入学习。同时，用户还可以在本平台上某篇论文提出自己的问题或者是评论，所有用户可以根据某篇论文在平台上进行讨论。此外，用户还可以根据对应的类别为平台新增新的论文，平台在审查信息后会将用户添加的论文也加入论文列表中。

5、主题讨论帖功能

用户可以根据自己感兴趣的话题在讨论区中新建帖子进行讨论，帖子需要包含一个概括性的主题和具体的详细描述，用户可以对自己的帖子进行编辑，删除等操作。同时，用户也可以查看他人的讨论帖，讨论区首页将根据帖子的主题，简单描述，发帖时间，最新回复时间进行显示，用户可进入某个具体帖子进行查看和回帖，参与到讨论当中。

三、数据库表的定义

根据数据库系统设计的规范化原则和中期设计文档，本数据系统一共使用了 8 个基本表，分别是：管理员信息表(admin_check_tab)，用户信息表(user_list)，类别目录表(category)，论文类别表(article_category)，论文信息表(article)，讨论帖表(posts)，讨论帖信息表(posts_info)，评论信息表(comments)。所有基本表中的关系均已规范化到 3NF 意思以上。各个基本表的具体定义如下：

1、管理员信息表(admin_check_tab)

序号	字段名称	数据类型	字段属性				
			字段大小	必填属性	主键	索引	功能

1	admin_id	int	11	是	是	主键	管理员 id
2	admin_name	char	50	否	否	否	管理员名
3	admin_pwd	char	50	否	否	否	密码

2、用户信息表(user_list)

序号	字段名称	数据类型	字段属性				
			字段大小	必填属性	主键	索引	功能
1	user_id	int	11	是	是	主键	用户 id
2	tel	decimal	11	否	否	否	电话
3	user_name	char	30	是	否	否	用户名
4	password	char	30	是	否	否	密码
5	real_name	char	30	是	否	否	真实名
6	email	char	30	是	否	否	邮箱

3、类别目录表(category)

序号	字段名称	数据类型	字段属性				
			字段大小	必填属性	主键	索引	功能
1	category_id	int	11	是	是	主键	类别 id
2	category_name	char	255	否	否	否	类别名
3	category_info	varchar	600	否	否	否	类别信息

4、论文类别表(article_category)

序号	字段名称	数据类型	字段属性				
			字段大小	必填属性	主键	索引	功能
1	article_id	int	11	是	否	外键	论文 id
2	category_id	int	11	是	否	外键	类别 id

5、论文信息表(article)

序号	字段名称	数据类型	字段属性
----	------	------	------

			字段大小	必填属性	主键	索引	功能
1	article_id	int	11	是	是	主键	论文 id
2	article_title	char	200	是	否	否	论文题目
3	article_author	char	30	是	否	否	论文作者
4	article_abstract	varchar	600	是	否	否	论文摘要
5	article_date	date	0	是	否	否	上传日期
6	article_url	char	200	否	否	否	论文链接

6、讨论帖表(posts)

序号	字段名称	数据类型	字段属性				
			字段大小	必填属性	主键	索引	功能
1	post_id	int	11	是	是	主键	讨论帖 id
2	msg	char	200	否	否	否	讨论内容
3	user_id	char	30	否	否	外键	用户 id
4	time	int	11	否	否	否	发帖时间
5	topic	char	50	否	否	否	讨论话题

7、讨论帖回复信息表(posts_info)

序号	字段名称	数据类型	字段属性				
			字段大小	必填属性	主键	索引	功能
1	respond_id	int	11	是	是	主键	回复 id
2	post_id	int	11	否	否	外键	帖子 id
3	user_id	int	11	是	否	外键	用户 id
4	respond_msg	char	200	是	否	否	回复信息
5	respond_time	int	11	是	否	否	回复时间

8、论文评论信息表(comments)

序号	字段名称	数据类型	字段属性				
			字段大小	必填属性	主键	索引	功能

1	comment_id	int	11	是	是	主键	评论 id
2	article_id	int	11	是	否	外键	文章 id
3	user_id	int	11	是	否	外键	用户 id
4	comment_text	varchar	600	是	否	否	评论内容
5	comment_time	int	11	是	否	否	评论时间

四、存储过程与触发器

1、存储过程

(1) 用户注册信息登录至数据库存储过程 (create_user)

【作用对象】用户信息表(user_list)

【功能】为登录信息较多的存储建立存储过程

【实现代码】

```

1 CREATE PROCEDURE create_user
2 (IN u_name CHAR(11), r_name CHAR(20),
3  password CHAR(20) , mailadd CHAR(30) ,
4  tel_num decimal(11,0))
5 BEGIN
6 INSERT INTO user_list (tel, password , real_name , email , user_name)
7 VALUES (tel_num, password, r_name, mailadd , u_name);
8 END

```

(2) 用户贡献资源登录至数据库存储过程 (create_user)

【作用对象】论文信息表(article_list)

【功能】为登录信息较多的存储建立存储过程

【实现代码】

```

1 CREATE PROCEDURE article_store
2 (IN title CHAR(11), author CHAR(20),
3  dat DATE , abstract VARCHAR(600) ,
4  link VARCHAR(300))
5 BEGIN
6 INSERT INTO article(article_title , article_author ,
7  article_date , article_abstract ,
8  article_url )
9 VALUES (title , author , dat , abstract , link);
10 END

```

【作用对象】论文信息表(article_list)

【功能】保证插入信息的正确性，即在插入信息前对类别存在性的验证。

【实现代码】

```
$sql0 = "SELECT * FROM category WHERE category_name LIKE '$category'";
$result0 = mysqli_query($connect , $sql0);
if(!$result0){
    echo "<script>alert('cannot find a certain category!'); history.go(-1);</script>";
}
```

2、触发器

讨论帖删除更新触发器(post_delete_update)

【作用对象】讨论帖回复信息表(posts_info)

【触发条件】讨论帖表(posts)执行对应的删除操作后

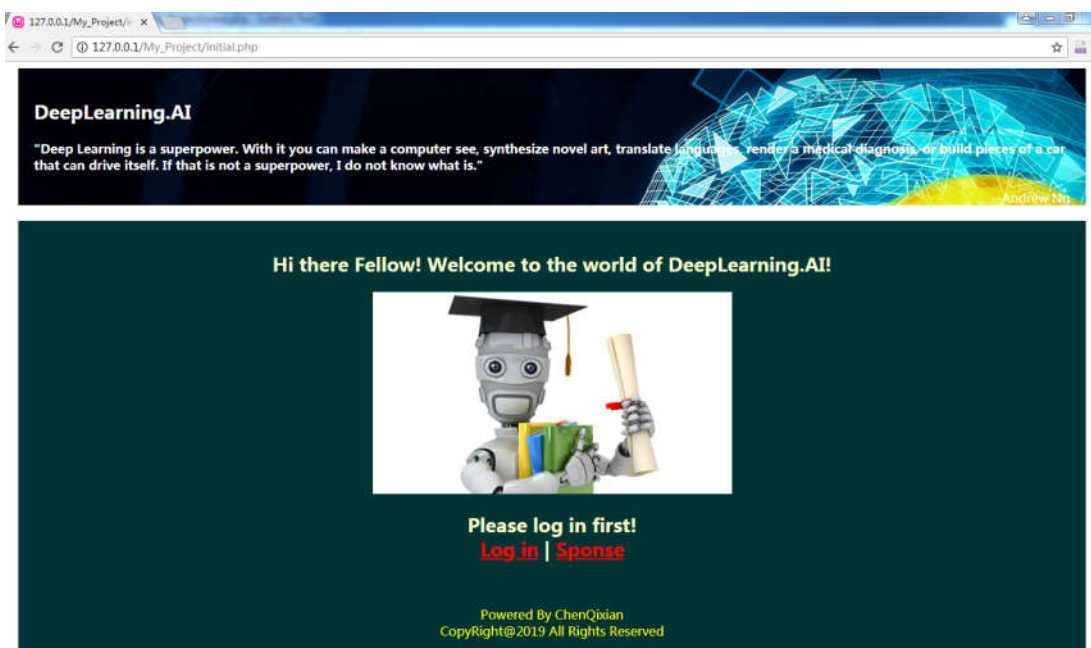
【功能】在讨论帖表删除一条帖子之后，会将之前帖子中所有的讨论回复内容对应删除，即删除讨论帖对应的所有讨论帖回复信息表。

【实现代码】

```
1 CREATE TRIGGER post_delete_update
2 AFTER DELETE ON posts
3     FOR EACH ROW BEGIN
4         DELETE FROM posts_info WHERE posts_info.post_id = old.post_id
5     END
```

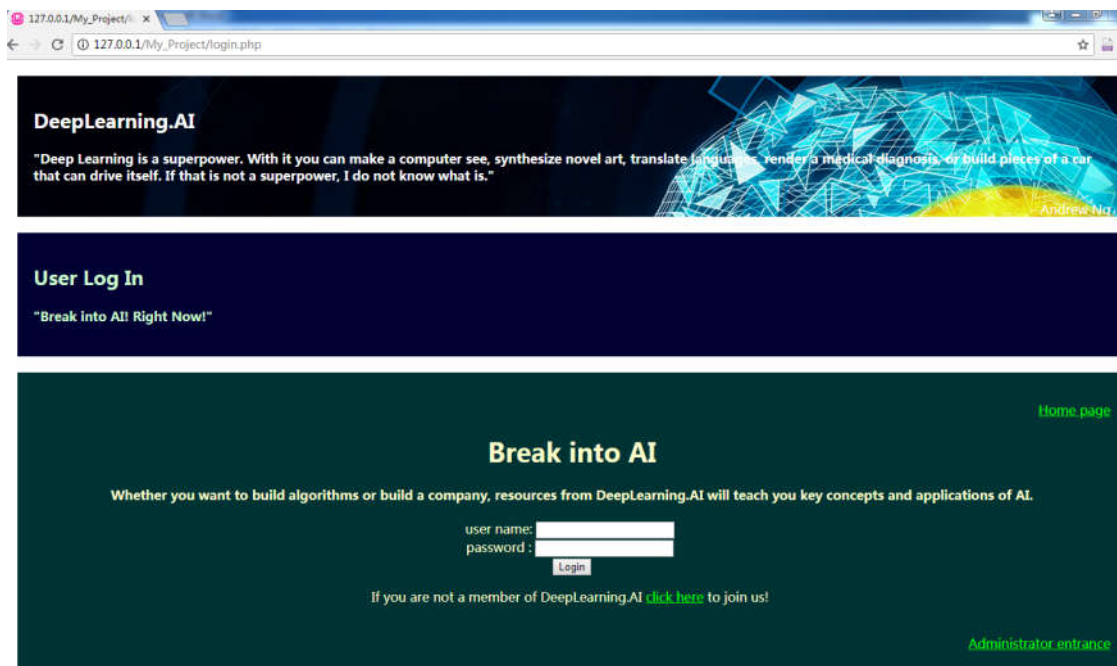
五、系统演示

1、初始登录界面



用户点击 Log In 跳转到登录页

2、用户登录界面



DeepLearning.AI

"Deep Learning is a superpower. With it you can make a computer see, synthesize novel art, translate languages, render a medical diagnosis, or build pieces of a car that can drive itself. If that is not a superpower, I do not know what is."

Andrew Ng

User Log In

"Break into AI! Right Now!"

Home page

Break into AI

Whether you want to build algorithms or build a company, resources from DeepLearning.AI will teach you key concepts and applications of AI.

user name:

password:

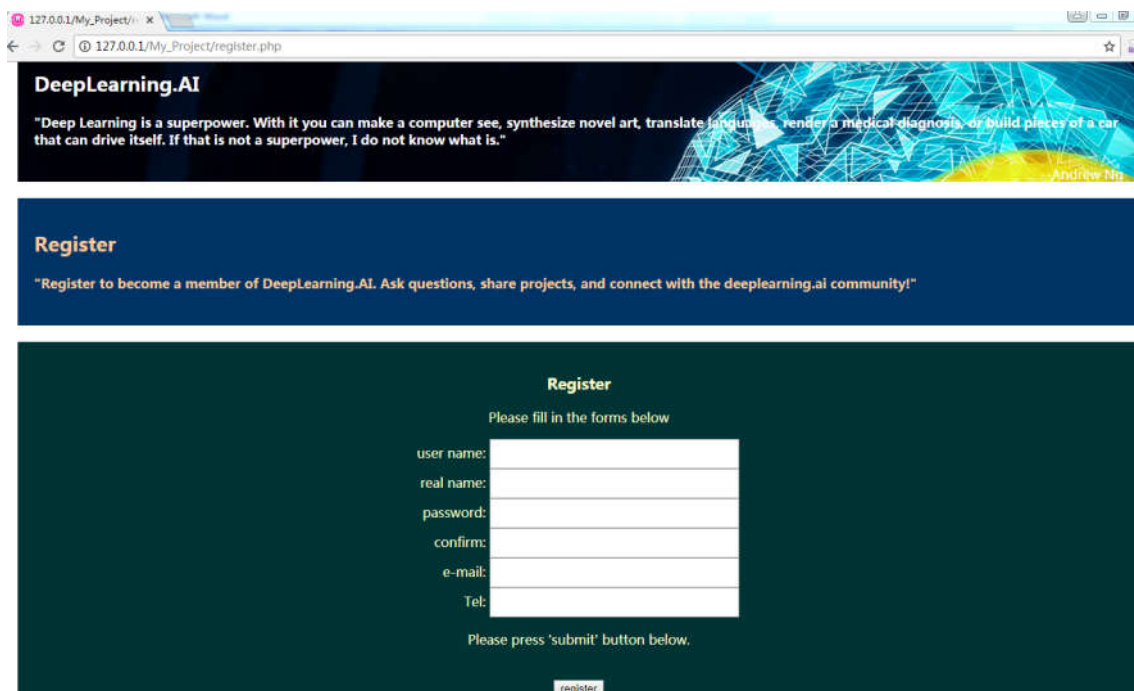
Login

If you are not a member of DeepLearning.AI [click here](#) to join us!

Administrator entrance

用户根据提示进行登录或注册，管理员进入管理员登录入口。

3、用户注册界面



DeepLearning.AI

"Deep Learning is a superpower. With it you can make a computer see, synthesize novel art, translate languages, render a medical diagnosis, or build pieces of a car that can drive itself. If that is not a superpower, I do not know what is."

Andrew Ng

Register

"Register to become a member of DeepLearning.AI. Ask questions, share projects, and connect with the deeplearning.ai community!"

Register

Please fill in the forms below

user name:

real name:

password:

confirm:

e-mail:

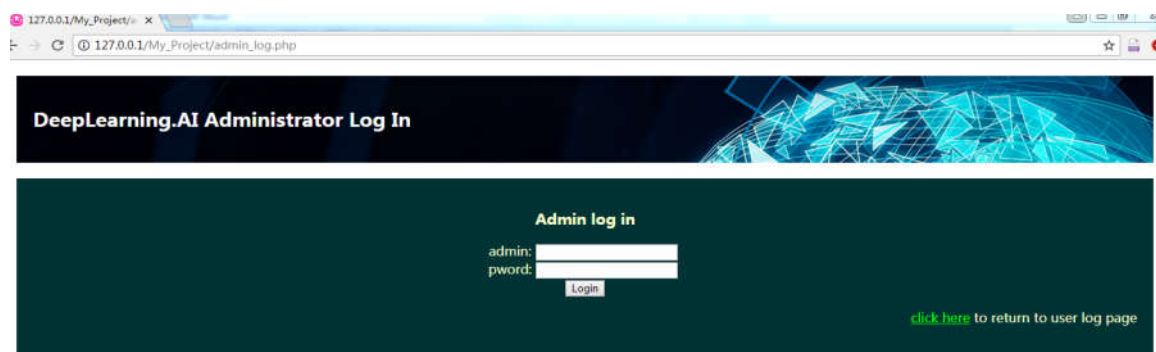
Tel:

Please press 'submit' button below.

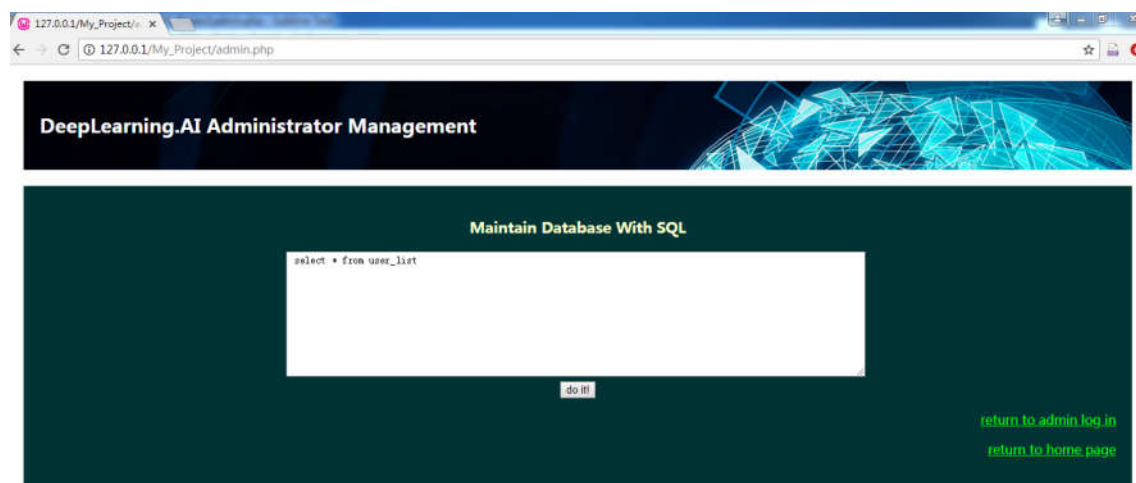
register

用户根据提示完成注册

4、管理员登录

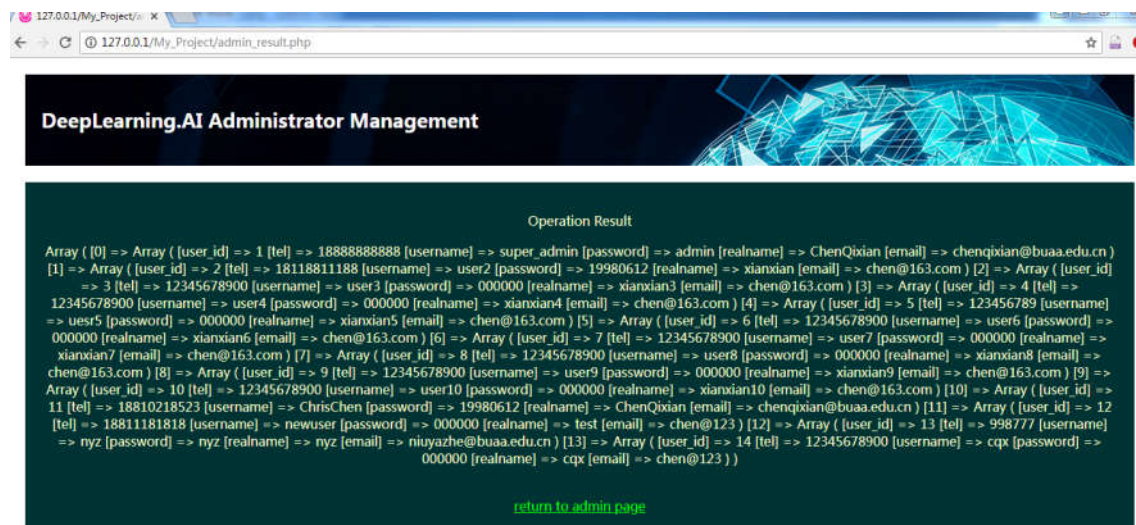


5、管理员操作



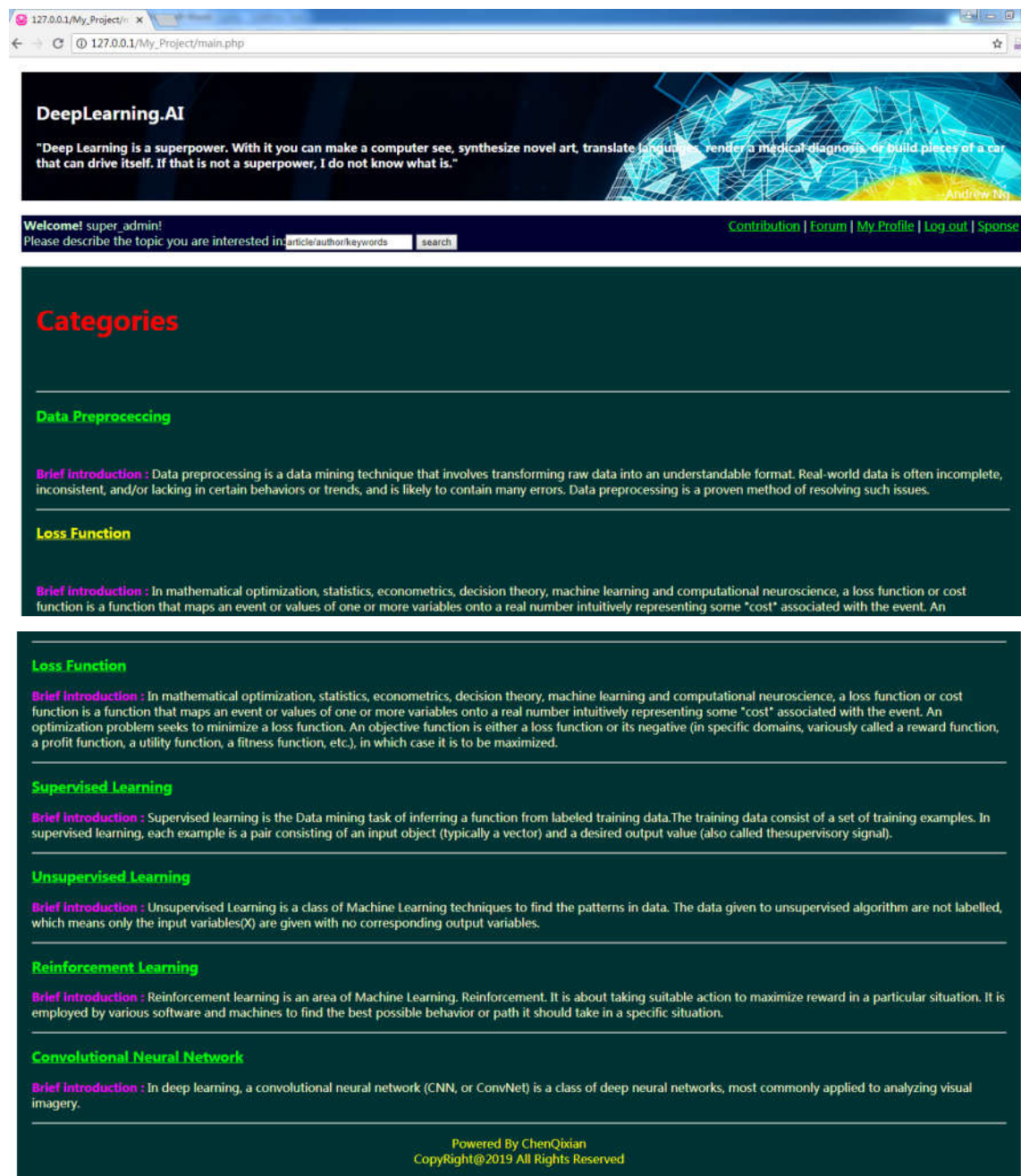
管理员通过 SQL 语句对数据库统一维护管理

6、管理员访问数据库查询结果显示



这里我们默认管理员为数据库相关技术人员。

7、主界面

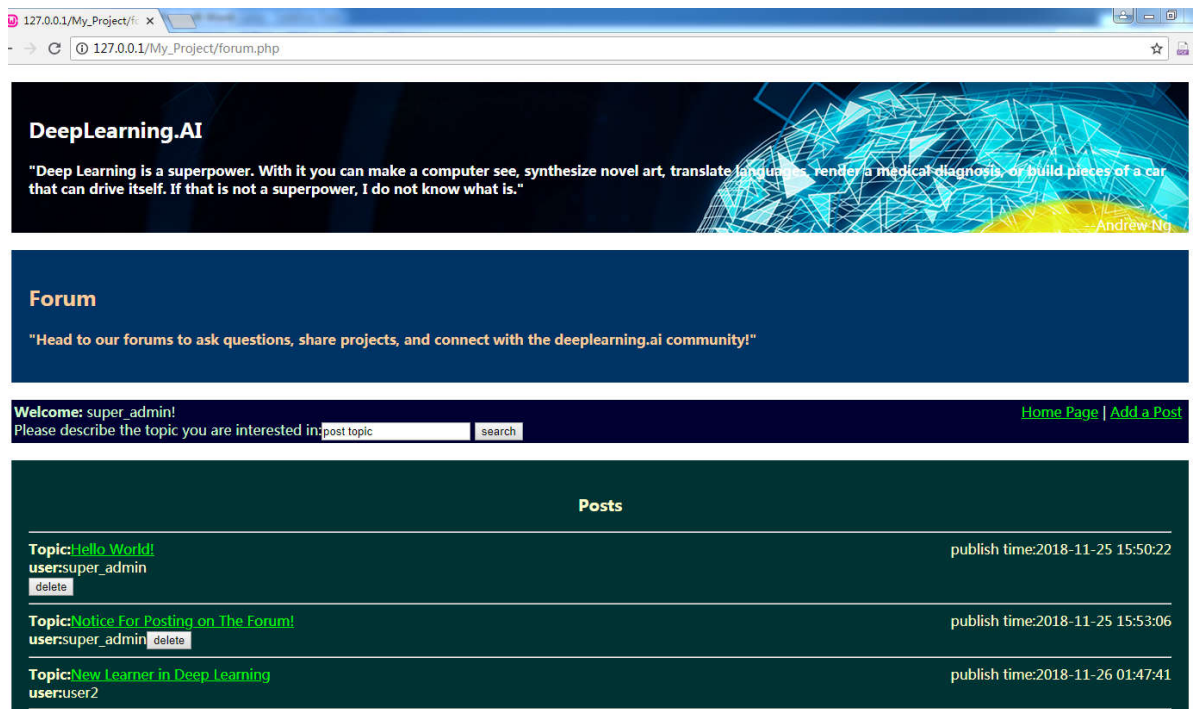


用户成果登录后自动跳转到主界面

主界面包括了本系统的全部功能导航。

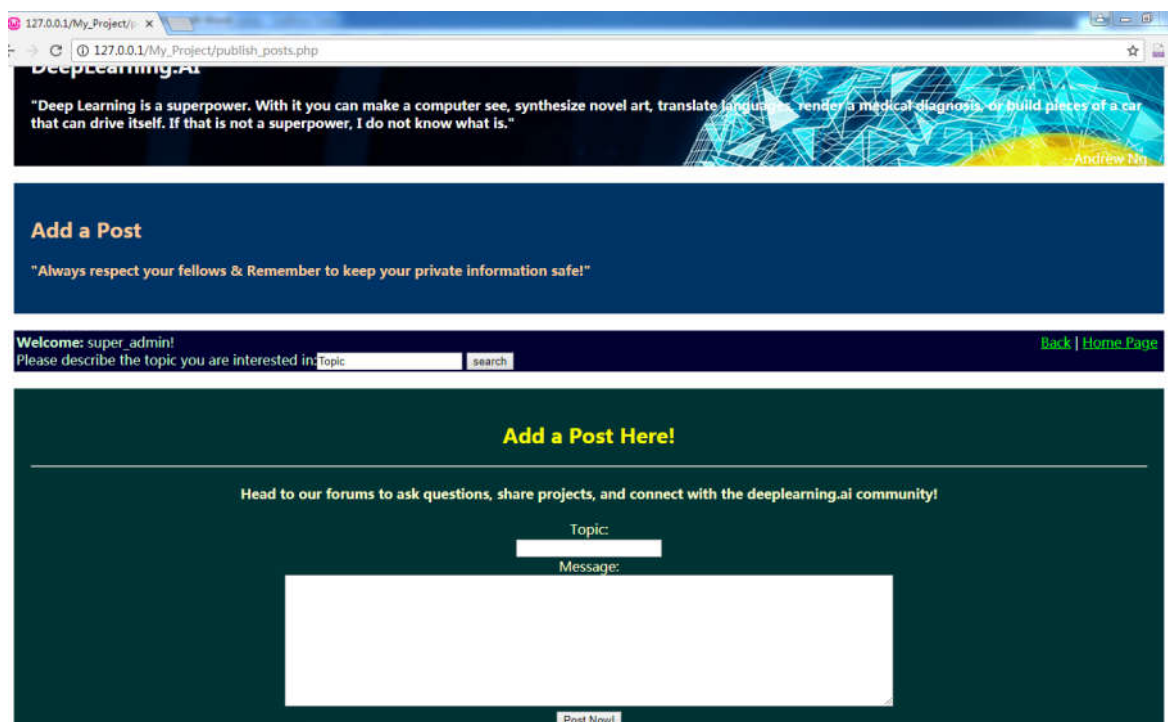
包含了网站目录索引、检索功能、用户论坛链接、用户资料修改、用户文献资源贡献区、用户注销、小额赞助的全部功能。

8、论坛页面

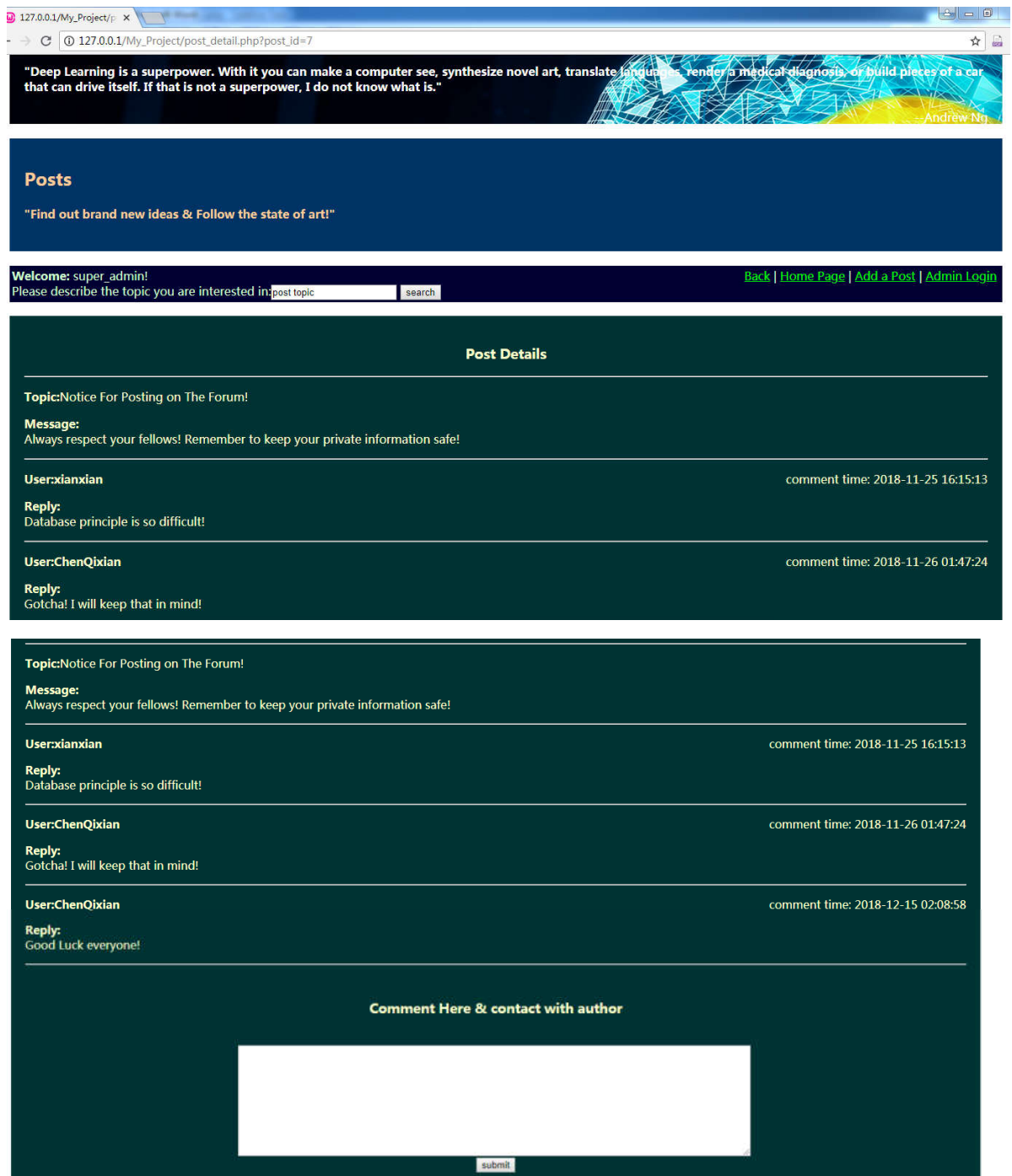


用户可以查看讨论帖、搜索话题、发布新帖以及删除自己曾发布的讨论帖

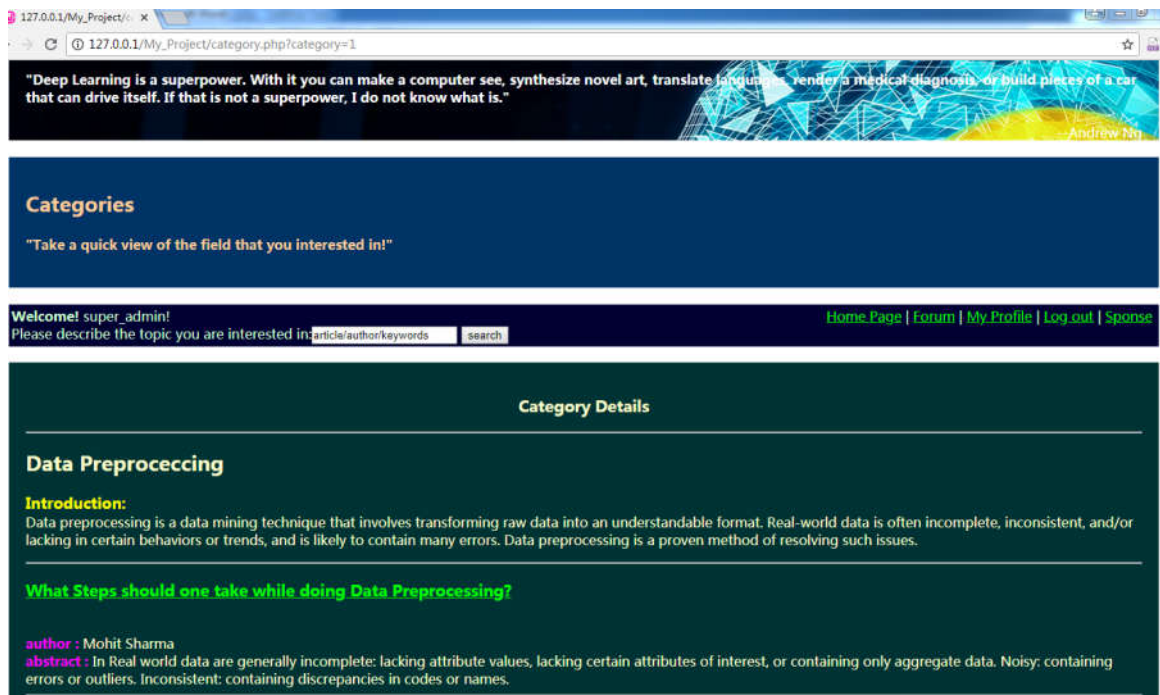
9、发布讨论帖



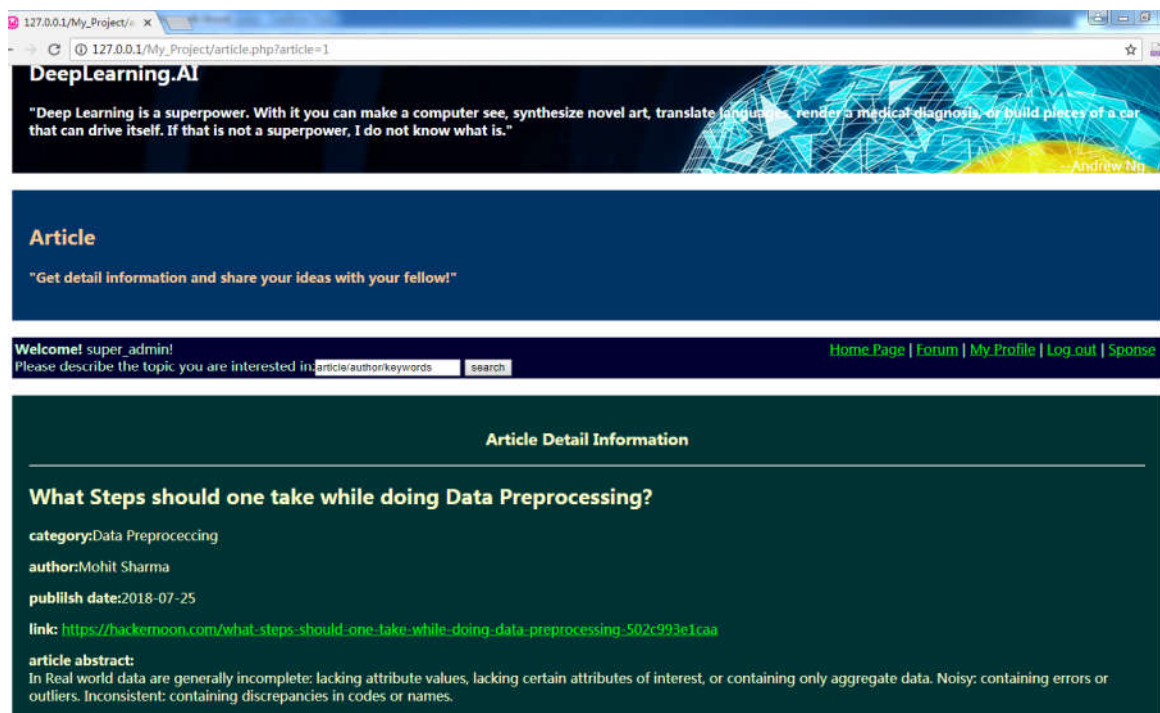
10、查看讨论帖内容与参与讨论



11、查看主题文章列表详细信息



12、查看文章详细内容与评论



user: user10

comments:
Hi chen, you can refet to this article to get some idea.

comment time:2018-12-08 10:49:10

user: newuser

comments:
Yes,I totally agree with user10.

comment time:2018-12-11 16:06:18

user: nyz

comments:
Can you release your code in github?

comment time:2018-12-12 12:41:38

Publish your comments HERE!

submit

13、文献资源贡献

127.0.0.1/My_Project/... X

127.0.0.1/My_Project/contribute.php

that can drive itself. If that is not a superpower, I do not know what is.

Andrew Ng

Contribution

"Share your resources and make contributions for DeepLearning.AI society!"

Home Page

Contribute Here!

Fill the information below & Make contributions !

*Notice : category must be exist already!

title :

author:

date :

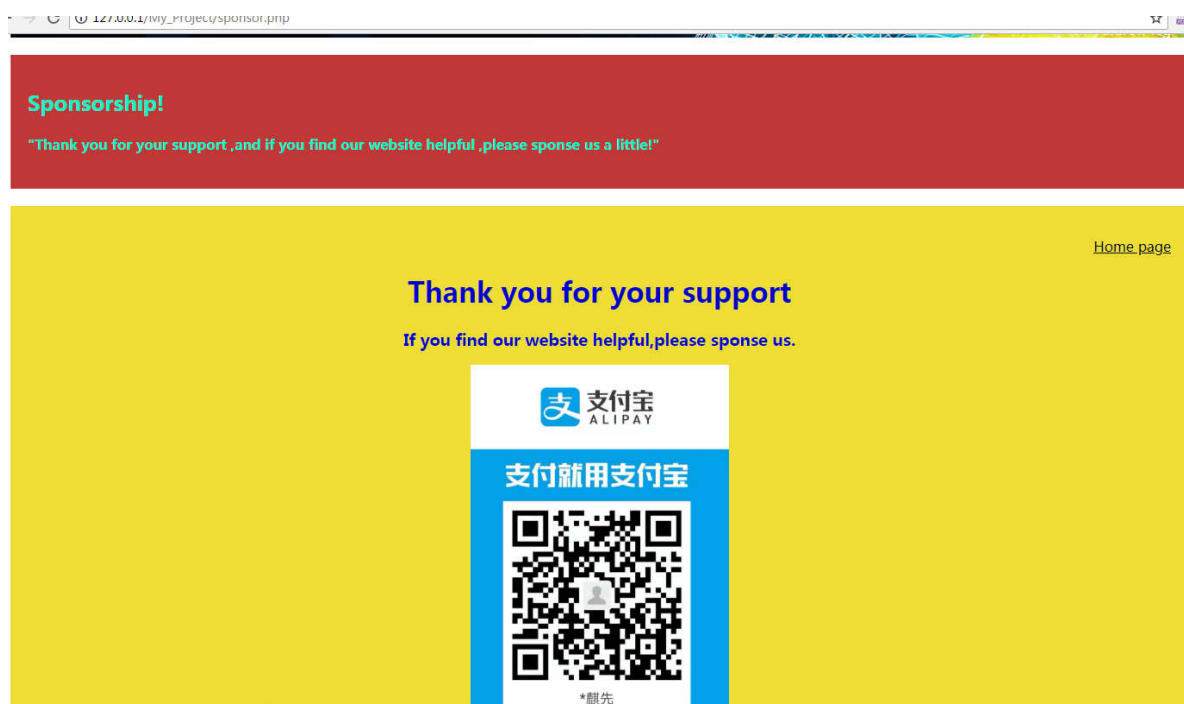
category:

abstract:

link:

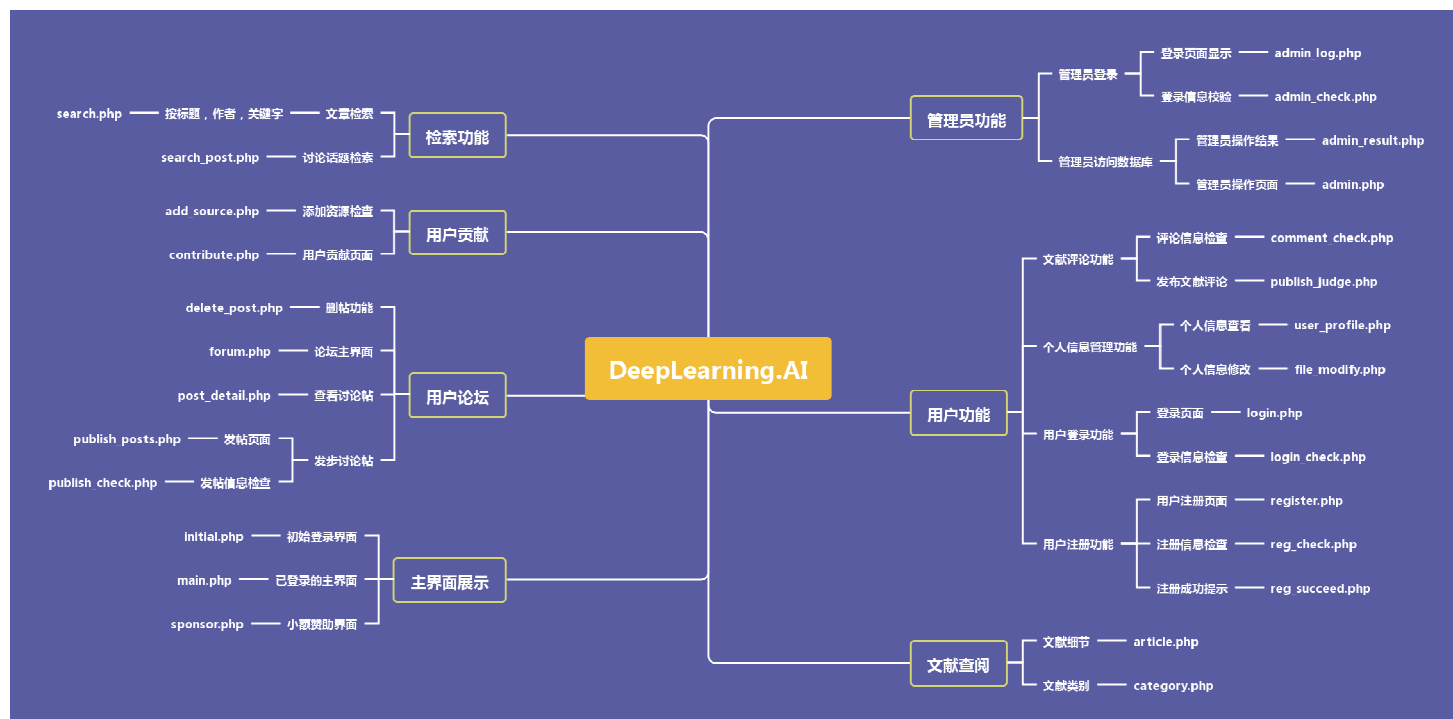
Contribute!

14、小额赞助界面



六、系统实现技术与程序结构解读

1. 程序结构解读



2. 系统实现技术

(1) 前端技术

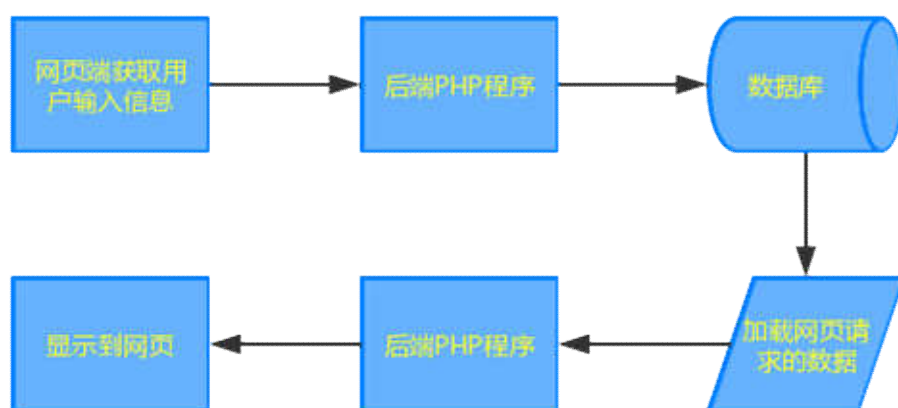
前端采用了少量 html5 和 css 提供的样式, 通过 html 与 php 语言结合的方式, 实现了前后端的数据通信。

(2) 后端技术

本项目的突出优点在于未使用成熟的后端框架，采用从 0 开始手工搭建网页的模式进行开发，对数据库的访问技术应用 PHP + MySQL 的方式，手工编写了大量的 SQL 语句，对 SQL 语言有了更进一步的深入理解。

同时在 PHP 编写过程中采用了最新的 PHP7 版本，代替了原有的 PHP5 版本，应用了 PHP 语言最新的特性，提升了系统的实现性能。

(3) 数据通信流图



七、体会与感想

本次数据库课程设计是一次综合性很高的作业，结合了数据库系统设计，web 前端和后端开发，团队协作代码管理多方面的知识。同时也是一次理论结合实践，设计模式指导代码实现的过程。通过这次数据库课程设计，我们真正将数据库的相关理论应用起来，不仅仅是理解概念和手写 SQL 语句或是暴力的代码实现，而是以理论知识为根基，先进行合理的数据库系统设计，确定实体，基本表，数据流图，建立合理完善的数据模型，然后进行数据收集和代码实现。这样一种设计导向的模式不但可以使我们更加深刻具体地理解数据库的理论知识，还让我们的开发过程更有条例，更高效率。计算机科学是非常重视实践的学科，任何脱离具体实践的理论知识都是没有意义的，与此相对，理论知识和设计理念，并不单单是概念定义，而是指导开发的核心骨干，暴力蛮干也许在一些小问题可以得到不错的结果，但是在系统级的设计上是完全行不通的，只有先规划好一个完善的设计方案，才能化繁为简，将复杂的系统分解成简单的局部去逐个击破。

在实际开发的过程中，由于编程语言没有二义性，一个简单的语法小错误就会导致全局系统的问题，这跟平时理论作业推导关系代数写 SQL 语句是完全不一样的，需要达到 100%的标准化规范化。而数据库编程中又有很多不起眼但很重要的小细节，需要一步步地在实际开发中去理解，在一次又一次的 debug 中去积累，这次课设的经历大大提升了我们的数据库编程经验。同时，我们在开发的过程中，也在思考为什么 SQL 语言要这样设计，为什么数据库理论要有那么多的规范和约束，真正体会到优秀设计在处理复杂多变的实际问题时的威力，将各式各样的具象实体进行抽象，从繁杂的事物关系中抽取出最核心的逻辑，这是我们以后开发中需要着重培养的能力。除此之外，由于本次作业是团队合作完成，且需要完成多块内容的开发，所以对我们的团队合作，版本控制能力也是一个考验，在此过程中我们使用 github 来帮助我们进行协作，进行项目的管理，我们在各个部分之间提前设定好合理的接口，采取“高内聚，低耦合”的设计模式，将整个大系统划分成一个个子系统，这样每个人开发的更新和改动都能限定在一定范围内，每个人的工作都可以看作一个模块，这样的开发模式条理清楚又十分高效。

总的来说，这次数据库课设使我们积累了宝贵的数据库系统设计，web 前后端开发的经验，通过实践强化理解了数据库理论知识，同时还提升了我们的团队开发能力。虽然在开发过程中遇到了一些令人焦头烂额的 bug，但克服困难的过程正是积累知识提升能力的过程。完成整个项目之后，我们有着非常大的成就感，切身感受到了自己的成长，是一次很有意义的实践经验。

陈麒先，牛雅哲

2018/12/16