



02 项目设计文档

《互联网数据库开发》课程团队作业

小组成员：姜宇，唐明昊，王禹衡，徐海濛

学号：2210705、2113927、2213040、2212180



1 项目设计目标:

使用 yii2 框架, 完成一个关于人工智能展示主题的网页开发。

2 数据表设计:

(1) 用户表:

表名	列名	数据类型	约束及说明
Users	UserID	int	主键, 自动递增 (AUTO_INCREMENT), 唯一标识每个用户
	Username	varchar(255)	用户名, 不能为空
	Password	varchar(255)	密码, 不能为空
	Role	enum('admin','user')	用户角色, 默认为 'user', 可以是 'admin' 或 'user'
	CreatedAt	timestamp	用户创建时间, 默认为当前时间 (CURRENT_TIMESTAMP)
主键	UserID		主键约束, 确保每个用户唯一

```
DROP TABLE IF EXISTS `Users`;  
/*!40101 SET @saved_cs_client      = @@character_set_client */;  
/*!40101 SET character_set_client = utf8 */;  
CREATE TABLE `Users` (  
  `UserID` int NOT NULL AUTO_INCREMENT,  
  `Username` varchar(255) NOT NULL,  
  `Password` varchar(255) NOT NULL,  
  `Role` enum('admin','user') DEFAULT 'user',  
  `CreatedAt` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`UserID`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;  
/*!40101 SET character_set_client = @saved_cs_client */;
```

根据代码, 我们可以看出, 我们的用户表, 记录了用户的所有信息, 其中用户的密码进行了加密保护, 同时为每个用户规定了一个 Role 属性, 表示用户的权限, 有 'admin' 和 'user' 两种, 默认为 'user'。

(2) 网页访问信息表:

表名	列名	数据类型	约束及说明
WebsiteVisits	ID	int	主键, 自动递增 (AUTO_INCREMENT), 唯一标识每次访问记录
	VisitCount	int	网站访问计数, 默认为 0
主键	ID		主键约束, 确保每次访问记录唯一

```

DROP TABLE IF EXISTS `WebsiteVisits`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `WebsiteVisits` (
  `ID` int NOT NULL AUTO_INCREMENT,
  `VisitCount` int DEFAULT '0',
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

这张表中存储了访问网页的用户 id，以及对应的访问量。

(3) 文章信息表:

Articles 表中，存储了我们需展示到页面上的文章的基本信息，包括对应的点赞量，文章的浏览量，以及对应的发布时间和作者。

表名	列名	数据类型	约束及说明
Articles	ArticleID	int	主键，自动递增 (AUTO_INCREMENT)，唯一标识每篇文章
	Title	varchar(255)	文章标题，不能为空
	Content	text	文章内容，不能为空
	AuthorID	int	外键，指向 Users 表的 UserID 列
	PublishedAt	timestamp	发布时间，默认为当前时间 (CURRENT_TIMESTAMP)
	UpdatedAt	timestamp	最后更新时间，默认为当前时间，更新时会自动更新
	ViewCount	int	文章浏览量，默认为 0
	LikeCount	int	文章点赞数，默认为 0
主键	ArticleID		主键约束，确保每篇文章唯一
外键	AuthorID		参照 Users 表中的 UserID 列
索引	AuthorID		索引 AuthorID，加速与 Users 表的连接查询

```

DROP TABLE IF EXISTS `Articles`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Articles` (
  `ArticleID` int NOT NULL AUTO_INCREMENT,
  `Title` varchar(255) NOT NULL,
  `Content` text NOT NULL,
  `AuthorID` int NOT NULL,
  `PublishedAt` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `UpdatedAt` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `ViewCount` int DEFAULT '0',
  `LikeCount` int DEFAULT '0',
  PRIMARY KEY (`ArticleID`),
  KEY `AuthorID` (`AuthorID`),
  CONSTRAINT `Articles_ibfk_1` FOREIGN KEY (`AuthorID`) REFERENCES `Users` (`UserID`)
) ENGINE=InnoDB AUTO_INCREMENT=35 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

(4) 文章评论表:

对于每一篇文章来说, 这个表中存储了对应的评论和评论者的信息, 这样可以保证我们在访问信息的过程中, 通过外键, 主键约束, 实现正确的文章评论。

表名	列名	数据类型	约束及说明
ArticleComments	CommentID	int	主键, 自动递增 (AUTO_INCREMENT)
	UserID	int	外键, 指向 Users 表的 UserID 列
	ArticleID	int	外键, 指向 Articles 表的 ArticleID 列
	Content	text	评论内容, 不能为空
	CommentedAt	timestamp	评论时间, 默认为当前时间 (CURRENT_TIMESTAMP)
主键	CommentID		主键约束, 确保每个评论唯一
外键	UserID		参照 Users 表中的 UserID 列
	ArticleID		参照 Articles 表中的 ArticleID 列
索引	UserID		索引 UserID, 加速与 Users 表的连接查询
	ArticleID		索引 ArticleID, 加速与 Articles 表的连接查询

```
DROP TABLE IF EXISTS `ArticleComments`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `ArticleComments` (
  `CommentID` int NOT NULL AUTO_INCREMENT,
  `UserID` int NOT NULL,
  `ArticleID` int NOT NULL,
  `Content` text NOT NULL,
  `CommentedAt` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`CommentID`),
  KEY `UserID` (`UserID`),
  KEY `ArticleID` (`ArticleID`),
  CONSTRAINT `ArticleComments_ibfk_1` FOREIGN KEY (`UserID`) REFERENCES `Users` (`UserID`),
  CONSTRAINT `ArticleComments_ibfk_2` FOREIGN KEY (`ArticleID`) REFERENCES `Articles` (`ArticleID`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

(5) 文章点赞表:

表中存入了点赞的信息, 可以存储点赞的文章信息, 用户信息以及对应的点赞信息。

表名	列名	数据类型	约束及说明
ArticleLikes	LikeID	int	主键, 自动递增 (AUTO_INCREMENT)
	UserID	int	外键, 指向 Users 表的 UserID 列
	ArticleID	int	外键, 指向 Articles 表的 ArticleID 列
	LikedAt	timestamp	点赞时间, 默认为当前时间 (CURRENT_TIMESTAMP)
主键	LikeID		主键约束, 确保每个点赞唯一
外键	UserID		参照 Users 表中的 UserID 列
	ArticleID		参照 Articles 表中的 ArticleID 列
索引	UserID		索引 UserID, 加速与 Users 表的连接查询
	ArticleID		索引 ArticleID, 加速与 Articles 表的连接查询

```

DROP TABLE IF EXISTS `ArticleLikes`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `ArticleLikes` (
  `LikeID` int NOT NULL AUTO_INCREMENT,
  `UserID` int NOT NULL,
  `ArticleID` int NOT NULL,
  `LikedAt` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`LikeID`),
  KEY `UserID` (`UserID`),
  KEY `ArticleID` (`ArticleID`),
  CONSTRAINT `ArticleLikes_ibfk_1` FOREIGN KEY (`UserID`) REFERENCES `Users` (`UserID`),
  CONSTRAINT `ArticleLikes_ibfk_2` FOREIGN KEY (`ArticleID`) REFERENCES `Articles` (`ArticleID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

(6) 用户历史聊天记录表:

存储了用户和大语言模型对话的记录信息。

表名	列名	数据类型	约束及说明
Conversations	ConversationID	int	主键, 自动递增 (AUTO_INCREMENT), 唯一标识每个对话
	UserID	int	外键, 指向 Users 表的 UserID 列
	StartedAt	timestamp	对话开始时间, 默认为当前时间 (CURRENT_TIMESTAMP)
	EndedAt	timestamp	对话结束时间, 默认为 NULL
	Status	varchar(50)	对话状态, 默认为 'active'
主键	ConversationID		主键约束, 确保每个对话唯一
外键	UserID		参照 Users 表中的 UserID 列
索引	UserID		索引 UserID, 加速与 Users 表的连接查询


```

DROP TABLE IF EXISTS `Conversations`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Conversations` (
  `ConversationID` int NOT NULL AUTO_INCREMENT,
  `UserID` int NOT NULL,
  `StartedAt` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `EndedAt` timestamp NULL DEFAULT NULL,
  `Status` varchar(50) DEFAULT 'active',
  PRIMARY KEY (`ConversationID`),
  KEY `UserID` (`UserID`),
  CONSTRAINT `Conversations_ibfk_1` FOREIGN KEY (`UserID`) REFERENCES `Users` (`UserID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

(7) 大模型对话表:

记录了用户和大语言模型的对话交互的内容和过程。

表名	列名	数据类型	约束及说明
Messages	MessageID	int	主键, 自动递增 (AUTO_INCREMENT), 唯一标识每条消息
	ConversationID	int	外键, 指向 Conversations 表的 ConversationID 列
	Sender	enum('user', 'model')	消息发送者, 必须是 'user' 或 'model'
	Content	text	消息内容, 不能为空
	Timestamp	timestamp	消息发送时间, 默认为当前时间 (CURRENT_TIMESTAMP)
主键	MessageID		主键约束, 确保每条消息唯一
外键	ConversationID		参照 Conversations 表中的 ConversationID 列
索引	ConversationID		索引 ConversationID, 加速与 Conversations 表的连接查询

```

DROP TABLE IF EXISTS `Messages`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Messages` (
  `MessageID` int NOT NULL AUTO_INCREMENT,
  `ConversationID` int NOT NULL,
  `Sender` enum('user', 'model') NOT NULL,
  `Content` text NOT NULL,
  `Timestamp` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`MessageID`),
  KEY `ConversationID` (`ConversationID`),
  CONSTRAINT `Messages_ibfk_1` FOREIGN KEY (`ConversationID`) REFERENCES `Conversations` (`ConversationID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

(8) 视频评论表:

表名	列名	数据类型	约束及说明
VideoComments	CommentID	int	主键, 自动递增 (AUTO_INCREMENT), 唯一标识每条评论
	UserID	int	外键, 指向 Users 表的 UserID 列
	VideoID	int	外键, 指向 Videos 表的 VideoID 列
	Content	text	评论内容, 不能为空
	CommentedAt	timestamp	评论时间, 默认为当前时间 (CURRENT_TIMESTAMP)
主键	CommentID		主键约束, 确保每条评论唯一
外键	UserID		参照 Users 表中的 UserID 列
	VideoID		参照 Videos 表中的 VideoID 列
索引	UserID		索引 UserID, 加速与 Users 表的连接查询
	VideoID		索引 VideoID, 加速与 Videos 表的连接查询

```

DROP TABLE IF EXISTS `VideoComments`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `VideoComments` (
  `CommentID` int NOT NULL AUTO_INCREMENT,
  `UserID` int NOT NULL,
  `VideoID` int NOT NULL,
  `Content` text NOT NULL,
  `CommentedAt` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`CommentID`),
  KEY `UserID` (`UserID`),
  KEY `VideoID` (`VideoID`),
  CONSTRAINT `VideoComments_ibfk_1` FOREIGN KEY (`UserID`) REFERENCES `Users` (`UserID`),
  CONSTRAINT `VideoComments_ibfk_2` FOREIGN KEY (`VideoID`) REFERENCES `Videos` (`VideoID`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

(9) 视频点赞表:

表名	列名	数据类型	约束及说明
VideoLikes	LikeID	int	主键, 自动递增 (AUTO_INCREMENT), 唯一标识每个点赞
	UserID	int	外键, 指向 Users 表的 UserID 列
	VideoID	int	外键, 指向 Videos 表的 VideoID 列
	LikedAt	timestamp	点赞时间, 默认为当前时间 (CURRENT_TIMESTAMP)
主键	LikeID		主键约束, 确保每个点赞唯一
外键	UserID		参照 Users 表中的 UserID 列
	VideoID		参照 Videos 表中的 VideoID 列
索引	UserID		索引 UserID, 加速与 Users 表的连接查询
	VideoID		索引 VideoID, 加速与 Videos 表的连接查询

```

DROP TABLE IF EXISTS `VideoLikes`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `VideoLikes` (
  `LikeID` int NOT NULL AUTO_INCREMENT,
  `UserID` int NOT NULL,
  `VideoID` int NOT NULL,
  `LikedAt` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`LikeID`),
  KEY `UserID` (`UserID`),
  KEY `VideoID` (`VideoID`),
  CONSTRAINT `VideoLikes_ibfk_1` FOREIGN KEY (`UserID`) REFERENCES `Users` (`UserID`),
  CONSTRAINT `VideoLikes_ibfk_2` FOREIGN KEY (`VideoID`) REFERENCES `Videos` (`VideoID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

(10) 视频表:

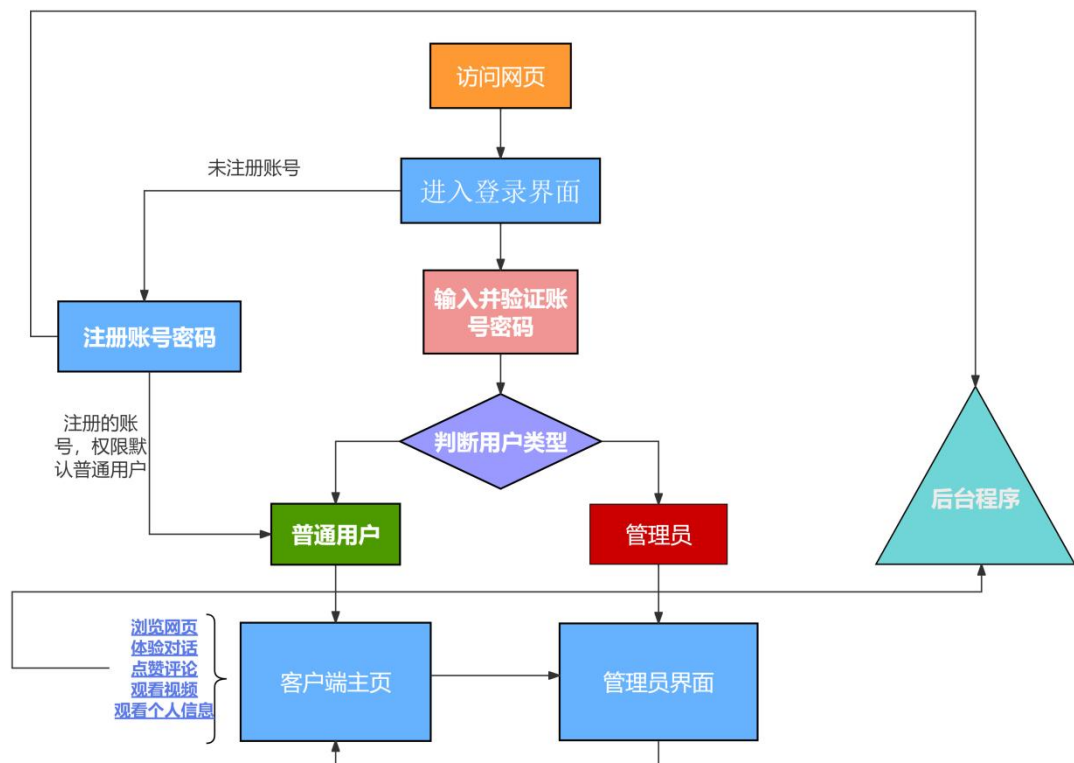
表名	列名	数据类型	约束及说明
Videos	VideoID	int	主键, 自动递增 (AUTO_INCREMENT), 唯一标识每个视频
	Title	varchar(255)	视频标题, 不能为空
	URL	varchar(255)	视频文件的 URL 地址, 不能为空
	UserID	int	外键, 指向 Users 表的 UserID 列
	PictureURL	varchar(255)	视频封面图的 URL 地址, 不能为空
	UploadedAt	timestamp	上传时间, 默认为当前时间 (CURRENT_TIMESTAMP)
	UpdatedAt	timestamp	更新时间, 默认为当前时间, 当更新时自动修改 (ON UPDATE CURRENT_TIMESTAMP)
	ViewCount	int	视频观看次数, 默认为 0
	LikeCount	int	视频点赞次数, 默认为 0
主键	VideoID		主键约束, 确保每个视频唯一
外键	UserID		参照 Users 表中的 UserID 列
索引	UserID		索引 UserID, 加速与 Users 表的连接查询


```

DROP TABLE IF EXISTS `Videos`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Videos` (
  `VideoID` int NOT NULL AUTO_INCREMENT,
  `Title` varchar(255) NOT NULL,
  `URL` varchar(255) NOT NULL,
  `UserID` int NOT NULL,
  `PictureURL` varchar(255) NOT NULL,
  `UploadedAt` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `UpdatedAt` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `ViewCount` int DEFAULT '0',
  `LikeCount` int DEFAULT '0',
  PRIMARY KEY (`VideoID`),
  KEY `UserID` (`UserID`),
  CONSTRAINT `Videos_ibfk_1` FOREIGN KEY (`UserID`) REFERENCES `Users` (`UserID`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```

3 流程图:



如流程图所示，当我们访问网页后会导入登录界面，此时，

(1) 如果没有注册账号，那么会跳转到账号注册页面，在创建好对应的账

号密码之后，我们就创建了一个用户的账号，默认通过注册界面注册的账号都是普通用户。

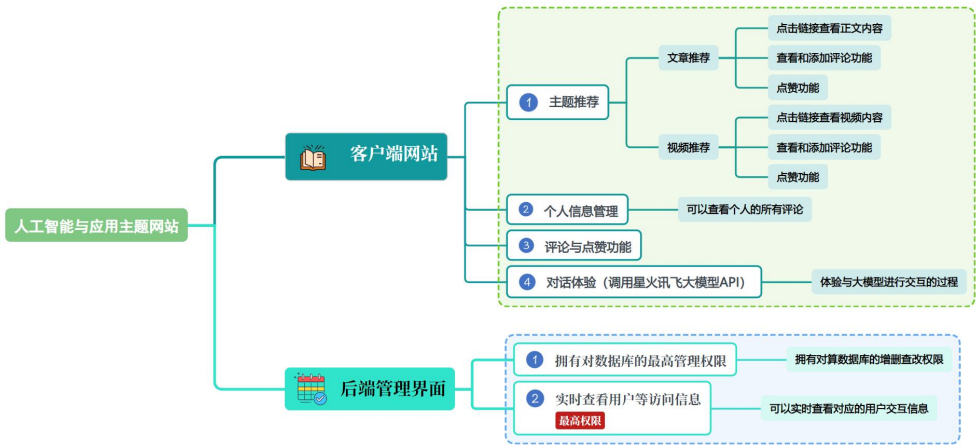
(2) 如果我们已经拥有账号了，那么我们可以输入对应的账号并进行验证，此时，根据用户的类型，我们可以判断一个用户是普通用户还是管理员用户。分别转入不同的页面。

(3) 普通用户会进入客户端页面，可以完成浏览页面，体验大模型，观看视频等操作。

(4) 在前端的页面中，我们实现了点赞和评论的功能。因此，我们每次修改对应的数据，都可以将对应的信息反馈给后台程序。

(5) 在后台的程序中，我们可以完成对于前台数据的记录，例如每注册一个账号，我们都可以后台的程序中进行记录。每条评论和记录都可以实现在后台中的记录添加。

4 模块图：



(1) 在本次的项目中，我们实现了客户端（前台）和后端的两个界面，在前台界面中，我们实现了主题推荐，个人信息管理，评论点赞功能，以及引入了星火讯飞大语言模型的 API 来让用户真实体验大语言模型的工作方式。

(2) 在后端的管理页面中，我们拥有对于程序修改的最高权限，可以查看数据库中的所有信息，同时可以对所有表实现增删查改的功能。