

诗 享 人 生

数据库设计文档

# 一、数据库设计概述

本数据库旨在为“诗享人生”网页提供用户管理功能，支持用户注册、登录以及与古诗词机器人对话。同时在描述了数据库在安全性与扩展性方面的设计。

核心需求包括：

**用户信息管理：**存储用户的基本信息（用户名、邮箱等）和加密后的密码。

**安全性：**确保用户密码的安全性，避免直接存储明文密码。

**扩展性：**设计应便于未来功能的添加，如用户角色管理、活动记录等。

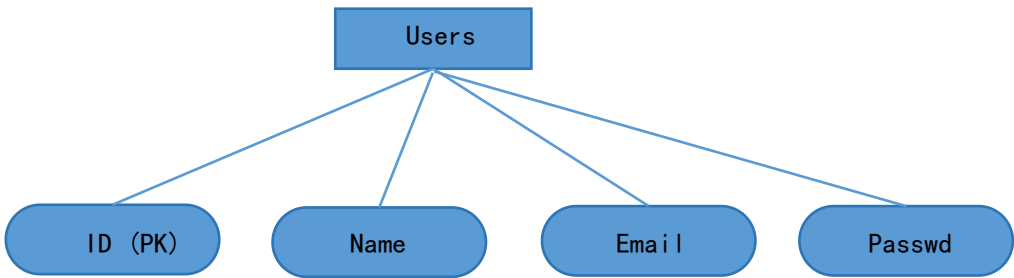
# 二、选用的数据库

考虑到项目的规模和需求，我们选用了 MySQL 作为后端数据库系统。MySQL 是一个广泛使用的关系型数据库管理系统，以其高性能、可靠性和易用性著称，适合中小型应用的开发。

# 三、E-R 图

在这个简单的用户管理系统中，主要实体为用户（User），其属性包括：

- ID（主键，自增）
- Username（用户名，唯一）
- Email（邮箱，唯一）
- Password（密码，加密存储）



# 四、实现细节与表结构

## 4.1 数据库初始化

首先，创建数据库文件：

```
CREATE DATABASE poetry_life;
USE poetry_life;
```

## 4.2 用户表结构

接着，创建 users 表来存储用户信息：

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL UNIQUE,
  email VARCHAR(100) NOT NULL UNIQUE,
  password VARCHAR(255) NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

列名	数据类型	属性	描述
id	INT	PRIMARY KEY, AUTO_INCREMENT	用户 ID, 主键
username	VARCHAR(50)	NOT NULL, UNIQUE	用户名, 唯一
email	VARCHAR(100)	NOT NULL, UNIQUE	邮箱, 唯一
password	VARCHAR(255)	NOT NULL	密码, 加密存储
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	创建时间戳

### 4.3 密码加密处理

为了安全起见，用户密码在存入数据库前应进行加密处理。可以使用 bcrypt 等算法对密码进行哈希处理后再存储。

```
// 注册接口
app.post('/register', async (req, res) => {
  const { username, email, password } = req.body;

  // 检查邮箱是否已存在
  pool.execute('SELECT * FROM users WHERE email = ?', [email], async (err, results) => {
    if (err) {
      return res.json({ status: 'error', message: 'Database error' });
    }

    if (results.length > 0) {
      return res.json({ status: 'error', message: 'Email already in use' });
    }

    // 哈希密码
    const hashedPassword = await bcrypt.hash(password, 10);

    // 插入新用户
    pool.execute('INSERT INTO users (username, email, password) VALUES (?, ?, ?)',
      [username, email, hashedPassword], (err, results) => {
        if (err) {
          return res.json({ status: 'error', message: 'Database error' });
        }
        res.json({ status: 'success', message: 'User registered successfully' });
      });
  });
});
```

### 4.4 安全性考虑

SQL 注入防护：使用参数化查询或 ORM 框架来防止 SQL 注入攻击。

密码加密：如上所述，使用强哈希函数对密码进行加密，确保即使数据库被泄露，密码也不会轻易被破解。

数据验证：在应用层对用户输入的数据进行严格验证，确保数据的合法性和安全性。

通过上述设计与实现，可以为“诗享人生”网页提供一个安全、高效且可扩展的用户管理系统。