CS307 Project2 Database Application

> Main Contributors:
>
> ZHU Yueming
>
> WANG Lishuang, ZHANG Chaozu
>
> Review: MA Yuxin

1. Basic Requirements: (70%)

1.1 API Specification：

To provide basic functionality of accessing a database system, you are required to build a backend library which exposes a set of application programming interfaces (APIs). The general descriptions for each API are listed below.

1. Register a new User.
2. User can favorites, shares and likes posts.
3. User can view the posts that he/she has been favorited, shared and liked.
4. User can follow or unfollow other users, and he/she can also view the user list he/she has been followed.
5. User can create a post.
6. User can reply a post or reply a reply.
7. User can view his/her own posts and replies.

1. 注册账户
2. 用户点赞、收藏、转发帖子
3. 用户查看的点赞、收藏、转发帖子的列表
4. 用户关注其他作者或取消对其他作者的关注，并查看用户自己关注作者的列表。
5. 用户发布帖子。
6. 用户回复帖子或回复回复。
7. 用户查看自己发布的帖子以及自己已回复的帖子

1.2 Functional Requirements:

- It is required to use a general-purpose **programming language** which can interact with the database to fulfill all the requirements mentioned in Section 1.1.
- To test all the APIs with necessary input data and display the result set, you should provide a type of **interface to interact with your program**. The interface can be:
  - Command-line-based application for input and output.
  - HTTP/RESTful interface services.
  - GUI-based desktop application.
  - Webpage-based.

- **Prepare testing data**: The two json files in project 1 as the testing data and store them in the database for the project 2.

2. Advanced Requirements: (30%)

> If you would like to get the full mark for any advanced requirement, please try to demonstrate yourself by providing a high-quality solution.

- Complete the project using open gauss database. (up to 5%)
    - https://edu.hicomputing.huawei.com/teaching
    - https://www.modb.pro/db/611481
- Based on the basic requirements in Section 1, further enhance the usability of the APIs to accept more flexible types of requests. You may think about more requirements than those proposed in Section 2 and implement the new requirements. Such as: (up to 15%)
    - Anonymous reply
    - Support pictures and video play. (Requires a page to display)
    - Shield or Block function
    - Hot search list function
    - Multi-parameter search function. Any parameter may be null or not. For example: search post by time period, keyword and type.
    - Others
- Encapsulate the features and implement a real back-end server instead of several independent scripts or programs. The server should provide socket-based communication or HTTP/RESTful web services. (up to 10%)
- Apply database connection pools; (up to 2%)
- Big data management. (Up to 4%)
- Page display design (up to 4%)
    - A usable and beautiful GUI design or webpage design for data presentation. (up to 4%)
    - Giving a wonderful input and output format display based on the command line. (up to 2%)
- Proper use of user privileges, procedures, indexing, and views in a reasonable manner; (up to 5%)
- Support high-concurrent with proper pressure tests. (up to 10%)

1. 使用到了open gauss数据库完成项目。
2. 在保证Section2需求的基础上，进一步完善API设计，设计出更多的系统功能性需求。
    - 匿名发言
    - 支持图片、视频播放。 要求用页面展示出来
    - 屏蔽或拉黑功能
    - 热搜榜功能
    - 多参数搜索功能。例如： 关键词及类型、时间段搜索帖子功能
    - 其他功能
3. 封装并实现一个真正的后端服务器，而不是几个独立的脚本程序。服务器应提供基于套接字的通信或HTTP/RESTful Web 服务。

4. 使用数据库连接池。
5. 大数据管理。
6. 页面显示设计。
    - 展示数据时，提供实用性强有好看的GUI页面或网页。
    - 展示数据时，提供一个设计好看的基于命令行输入输出交互的页面
7. 合理使用用户权限、过程、索引、视图等功能
8. 支持高并发，并要通过适当的压力测试。

3 What to Submit

Please submit your report and attachments before **23:55 on June 2nd 2022**, including:

- All scripts you have written.
- A report no longer than 12 pages. The following content should be include:
    - Basic information of your group: Please follow the same requirement as described in Project 1.
    - API specification of your code: Please describe the purpose and use of interfaces (you may only use 1-2 sentences for each API in case the report becomes too long). Also, you need to illustrate the types and meanings of the parameters and return values. You can take any API documents of mature open source projects as references of how to organize the specification of your interfaces.
    - If you have finished any advanced requirements, describe what you have done and how you did it.