



香港大學

THE UNIVERSITY OF HONG KONG

# From SQL to a Real-World Application

## Group Project

HKU COMP 3278'26

**Prof.** Ping Luo

**TA:** Tianxing Chen, Mengkang Hu

# Requirements



香港大學  
THE UNIVERSITY OF HONG KONG

**Deadline - Final Presentation (each group  $\leq 5$  members, pre ~8mins) - Apr 23, 28, 30**

**Everyone should present otherwise the score for those who didn't present will be zero**

**You need to SUBMIT:**

**(1) PPT ( $\leq 10$  pages): Requirement Description, ER Modeling, ER Design, UI design, how your group collaborates and the distribution of workload, a Demo Video. (2) Source Code (a zip file).**

- 1. Design and implement relational database tables**
- 2. Query the database using a Text-to-SQL or SQL system**
- 3. Design a user interface for data visualization (UI)**
- 4. Deploy and demonstrate the project on a website (local deployment is acceptable)**

# Scenario 1: Chat Group Application

*Easy but low-weighted*



香港大學  
THE UNIVERSITY OF HONG KONG

- **Core Features**

- User accounts with usernames
- Chat messages with timestamps
- Chat history that can be queried and analyzed

- **Optional / Bonus Features**

- Support Multi-Device & Multi-User Online Chatting
- Support for sending and visualizing images via image URLs
- A chat input box for sending text and image messages (URL)
- A button that allows users to select predefined emoji or stickers
- Support multiple chat groups
- Propose your own SQL-related extensions or creative features

## Example

### Chat Group - COMP 3278'26

#### Ping Luo – 2026/01/02

Provide students with two scenarios. Each group chooses one scenario and designs several tables. Students then use Text-to-SQL to query the data. The project also includes UI design.

#### Mengkang Hu – 2026/01/12

Copy that !!!

#### Tianxing Chen – 2026/01/18



Type Here

Send

# Scenario 2: Social Media Application



香港大學  
THE UNIVERSITY OF HONG KONG

*Difficult but high-weighted*

- **Core Features**

- User accounts with unique usernames
- Support for image posts via image URLs
- Visualization of images directly in the feed
- Each post displays: Username, Text description, Image, Timestamp
- Like functionality for posts

- **Optional / Bonus Features**

- Support multi-device and multi-user concurrent access
- Users can create and publish posts with text content
- Automatic maintenance and update of like counts for each post
- Commenting on posts
- Viewing posts sorted by time or popularity
- Displaying user-specific post history
- A button to like or unlike a post
- Basic analytics (e.g., most-liked posts, most active users)
- Propose your own SQL-related extensions or creative features

## Example

HKUgram

User 1: Tianxing

2026/01/18 – 19:05

Photo Here

Likes 16

Text Here

Like

2026/01/18 – 19:00

Photo Here

Likes 99+

Text Here

Like

Refresh

New Post

# Demo Tutorial

We use Vanna (vanna.ai) to enable Text-to-SQL, making it easier to work with our SQL database. We also encourage vibe coding. In fact, many of our tutorial codes are generated by Vibe Coding (Human verified 🧑🏻💻).

We use GitHub: <https://github.com/TianxingChen/HKU-COMP-3278-26> to organize the course code.

Github is an online platform for Open-sourcing, Developer Collaboration. We strongly recommend you learn to use Github for daily code development.



香港大學  
THE UNIVERSITY OF HONG KONG

HKU-COMP-3278-26 Public Pin Watch 0

main 1 Branch 0 Tags Go to file Add file Code

TianxingChen init repo 8287b46 · now 1 Commit

README.md	init repo	now
demo_chat_app.py	init repo	now
demo_index.html	init repo	now
requirements.txt	init repo	now

README

## HKU Comp 3278'26

### Clone this repo

Please make sure you have registered Github account.

Run the following code in your terminal.

```
git clone git@github.com:TianxingChen/HKU-COMP-3278-26.git
```

### Install Python Environment

```
pip install -r requirements.txt
```

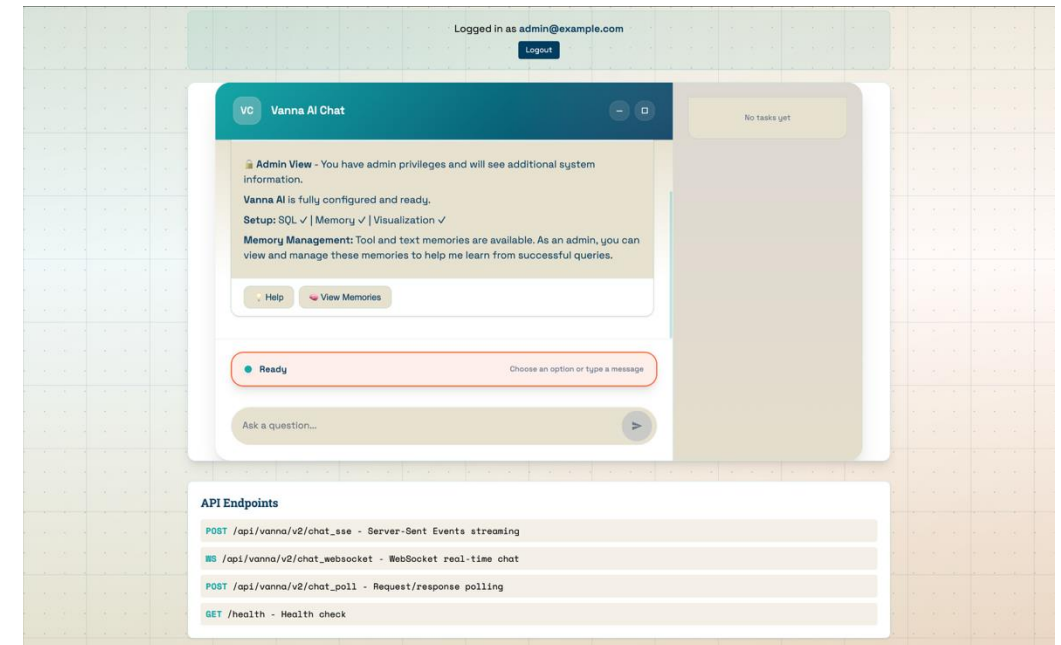
# Demo Tutorial



香港大學  
THE UNIVERSITY OF HONG KONG

- Clone the repository from GitHub.
- Follow the instructions in **README.md** to set up the Python environment.
- Initialize the database (python demo\_chat\_app.py).
- Review the provided example and make sure you understand the workflow.
- Visualize the database by opening the demo\_index.html with your web browser.

```
tianxingchen@Mac ~/D/c/project (main)> python demo_chat_app.py
INFO: Started server process [84358]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO: 127.0.0.1:58219 - "GET / HTTP/1.1" 200 OK
INFO: 127.0.0.1:58241 - "POST /api/vanna/v2/chat_sse HTTP/1.1" 200 OK
INFO: 127.0.0.1:58244 - "GET /favicon.ico HTTP/1.1" 404 Not Found
```



# Demo Tutorial



香港大學  
THE UNIVERSITY OF HONG KONG

- Clone the repository from GitHub.
- Follow the instructions in **README.md** to set up the Python environment.
- Initialize the database (python demo\_chat\_app.py).
- Review the provided example and make sure you understand the workflow.
- Visualize the database by opening the demo\_index.html with your web browser.

Add users "Ping Luo", "Tianxing Chen" and "Meng Kang Hu", and add them to Group "Chat Group - COMP 3278'26"

group_id	group_name	user_id	username	role	joined_at
1	Chat Group - COMP 3278'26	1	Ping Luo	member	2026-01-18 12:52:57
1	Chat Group - COMP 3278'26	2	Tianxing Chen	member	2026-01-18 12:52:57
1	Chat Group - COMP 3278'26	3	Mengkang Hu	member	2026-01-18 12:52:57

Show me all messages sent in group "Chat Group - COMP 3278'26"

No data to display

Tianxing Chen sent a message "Hello I am your TA @ COMP 3278'26 !!!", and then Ping Luo sent "Welcome."

Show me all messages sent in group "Chat Group - COMP 3278'26"

message_id	content	sender_id	sender_name	group_id	group_name	created_at
1	Hello I am your TA @ COMP 3278'26 !!!	2	Tianxing Chen	1	Chat Group - COMP 3278'26	2026-01-18 13:09:22
2	Welcome Tianxing.	1	Ping Luo	1	Chat Group - COMP 3278'26	2026-01-18 13:10:28

# Demo Tutorial



香港大學  
THE UNIVERSITY OF HONG KONG

- Clone the repository from GitHub.
- Follow the instructions in **README.md** to set up the Python environment.
- Initialize the database.
- Review the provided example and make sure you understand the workflow.
- Visualize the database by opening the demo\_index.html with your web browser.

**Chat Group Viewer (Fetch GET /groups first, then load messages, to avoid name mismatch)**

Backend Base URL:  Load groups:  Limit (1-500):  Auto Refresh:

Your FastAPI address:  Click this first so the page can get the "real group names" from the DB. Max number of messages to fetch each time:  Polling refresh (snapshot style):

Group Picker (from /groups):  Or type to match (fuzzy):  Type part of the name; I will pick the closest group (still) Sort:  View:

Pick here to match the database exactly (including ' / ' / spaces, etc.):  Example: type COMP 3278 and it will select the closest match.

After (optional):  Before (optional):

Search (client-side):

Messages loaded:  Loaded:  Filtered:

Request URL:

Matched group:  
- id: 1  
- name: Chat Group - COMP 3278'26

Messages loaded: 2

Key point: this page always uses the real name returned by /groups to build the URL, and applies encodeURIComponent to the path parameter to guarantee an exact match.

Tips: If an image doesn't show, the source may block CORS or the link may be invalid.

**Ping Luo** 2026-01-18 13:10:28 - #2  
Welcome Tianxing.

**Tianxing Chen** 2026-01-18 13:09:22 - #1  
Hello I am your TA @ COMP 3278'26 !!!