

# Client Query Management System

**RAHUL RAJ**

Batch Code: DS-C-WE-E-B89

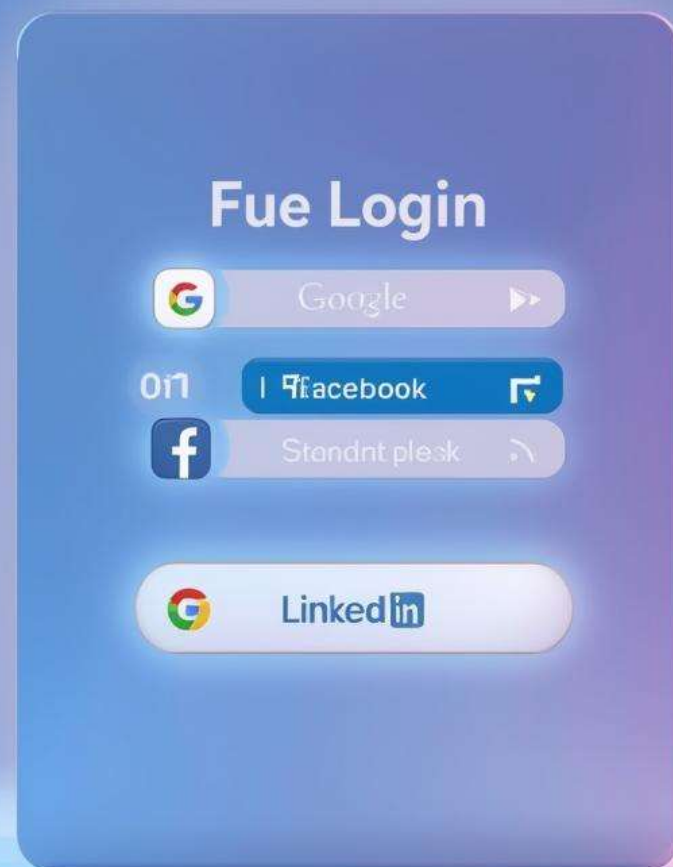




# Introduction

- 1) The *Client Query Management System (CQMS)* is a web-based responsive application designed to manage and track client queries efficiently.
- 2) It provides a seamless interface for clients to register, log in, and submit their issues or requests.
- 3) Built using **Python, Streamlit, and MySQL**, the system ensures smooth data handling and real-time updates.
- 4) This project simplifies communication between clients and admin , enhancing overall service quality.





# FEATURES OF CQMS

- 1) **User Management** – Allows clients and admins to register, log in, and manage their accounts securely.
- 2) **Query Submission & Tracking** – Clients can easily submit queries and track their status (Open/Closed).
- 3) **Admin Dashboard** – Admins can view, manage, and update all client queries efficiently.
- 4) **Database Integration** – All client and query data are securely stored and managed using SQLite/MySQL.

# SYSTEM WORKFLOW

## CLEINT REGISTRATION AND LOGIN

- 1) Clients create an account by entering name, email, and password.
- 2) After validation ensure, registered clients gain access to the query submission form.
- 3) Clients log in securely using registered email and password.
- 4) Invalid login attempts show proper error messages for user clarity.

## ADMIN REGISTRATION AND LOGIN

- 1) Admin registration is managed securely through backend setup.
- 2) Only authorized personnel can register as admins.
- 3) Admins log in to view, manage, and update client queries and ensure validation.
- 4) Admin dashboard provides real-time data visibility.

## QUERY SUBMISSION

- 1) Clients submit queries with name, email, mobile, and description.
- 2) Each query gets a unique ID and stored in the database.
- 3) Clients can track the query status (Open or Closed).

## ADMIN PANEL

- 1) Displays all submitted client queries in an organized table.
- 2) Admins can update query status and view detailed information.
- 3) It helps monitor overall client support progress efficiently.

# DATABASE SCHEMA

USER TABLE

Field Name	Data Type	Description
id	INT (Primary Key, Auto Increment)	Unique user ID
name	VARCHAR(100)	Full name of the user
email	VARCHAR(100)	User email (used for login)
password_hash	VARCHAR(255)	Hashed password for security
role	VARCHAR(20)	Defines user type: 'client' or 'admin'
created_at	TIMESTAMP	Time of registration

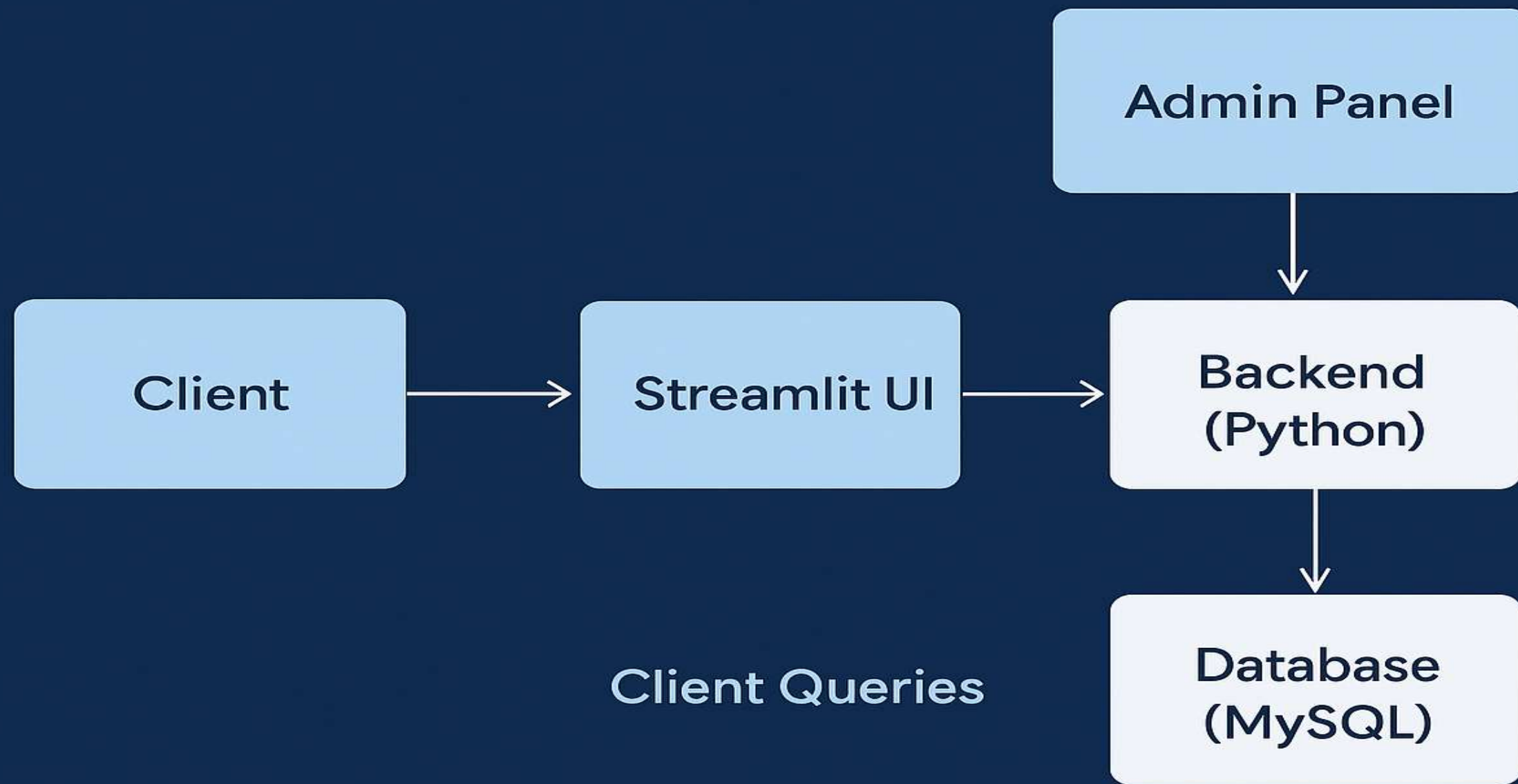
QUERY TABLE

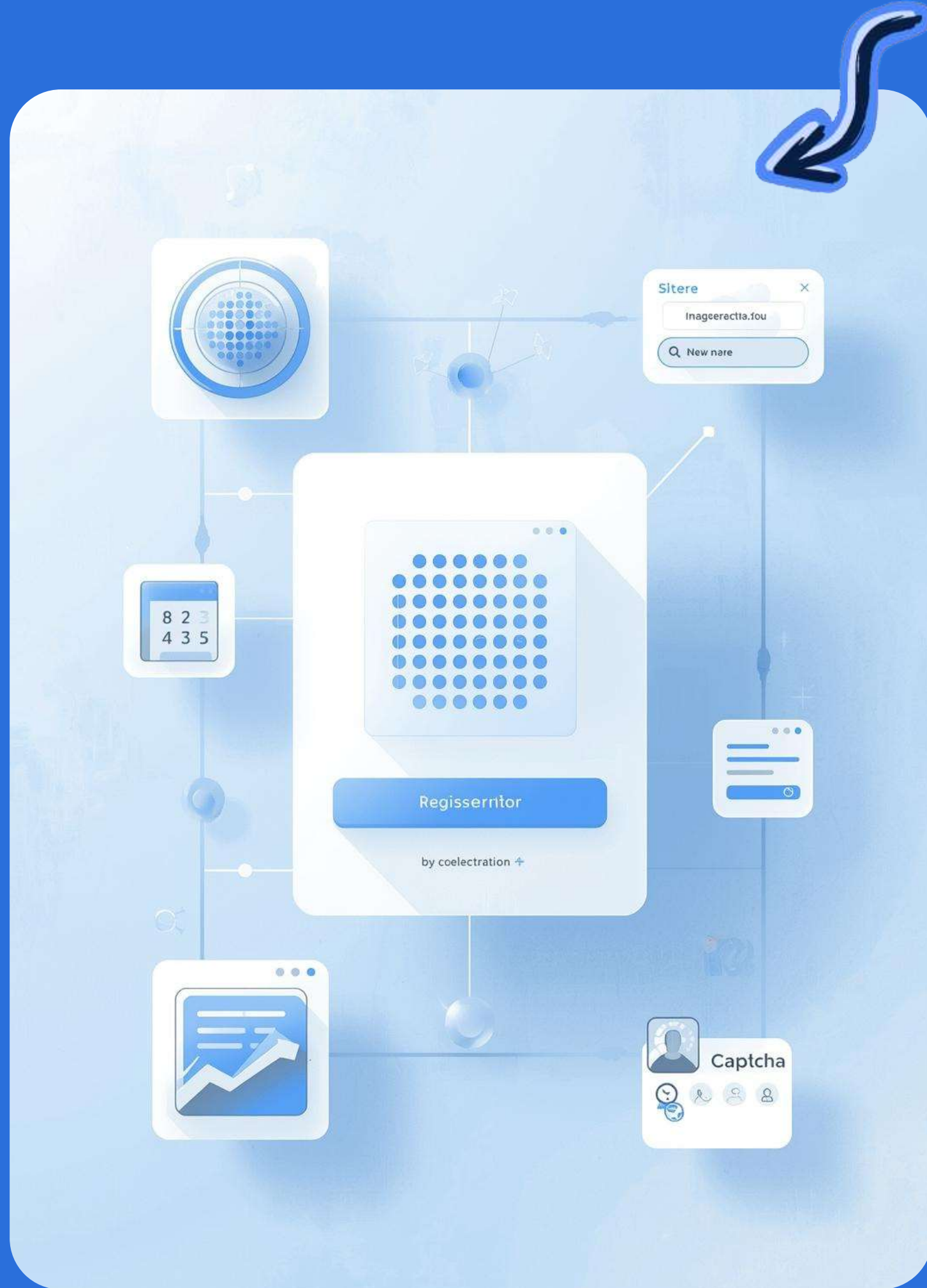
Field Name	Data Type	Description
id	INT (Primary Key, Auto Increment)	Unique query ID
user_id	INT (Foreign Key → users.id)	ID of the user who submitted the query
heading	VARCHAR(255)	Short title of the query
description	TEXT	Detailed description of the issue
status	VARCHAR(20)	Query status ('Open' or 'Closed')
created_at	TIMESTAMP	Time when the query was submitted

RESPONSE TABLE

Field Name	Data Type	Description
id	INT (Primary Key, Auto Increment)	Unique response ID
Query_id	INT (Foreign Key → queries.id)	The query being responded to
admin_id	INT (Foreign Key → users.id)	Admin who responded
message	TEXT	Admin's reply message
created_at	TIMESTAMP	Time of response

# Diagram





# FUTURE SCOPE

- 1) Future versions can include AI-powered assistants to automatically reply to basic client queries, reducing manual effort and improving response time.
- 2) The system can be enhanced with dashboards and visual reports to analyze client trends, response efficiency, and team performance.
- 3) Hosting the CQMS on cloud platforms like AWS or Streamlit Cloud will make it accessible from anywhere, improving scalability and reliability.
- 4) Future upgrades can allow multiple admins or support agents to handle queries simultaneously, ensuring faster query resolution and better teamwork.



# THANK YOU

We welcome your feedback and questions

**Email**

Kshanu.2003@gmail.com

**Phone**

7667891605

