# Client Query Management System

A web-based application for managing client queries, built with \*\*Streamlit\*\* and \*\*Python\*\*.

Supports user authentication, registration, query submission, query tracking, dashboard analytics, and admin/support features.

---

## Features

- \*\*User Authentication:\*\* Login/logout for Clients and Support/Admin roles.

- \*\*User Registration:\*\* New users can register with a role.

- \*\*Query Submission:\*\* Clients can submit queries with details and screenshots.

- \*\*Query Management:\*\* View, filter, search, and paginate queries.

- \*\*Query Update:\*\* Support/Admin can update query status.

- \*\*Dashboard:\*\* Visual analytics and summary of queries.

- \*\*Role-Based Access:\*\* Clients see only their queries; Support/Admin see all.

---

## Folder Structure

```

ClientQueryMS/

├── app.py # Main Streamlit entry point

├── dashboard.py # Dashboard analytics and visualizations

├── db.py # Database connection logic

├── login.py # Login/logout logic

├── registration.py # User registration logic

├── query.py # Query management (add/view/update)

├── utils.py # Helper functions

├── README.md # This file

```

---

## Setup Instructions

1. \*\*Clone the repository:\*\*

```

git clone <your-repo-url>

cd ClientQueryMS

```

2. \*\*Install dependencies:\*\*

```

pip install streamlit pandas

```

3. \*\*Configure your database:\*\*

- Update `db.py` with your database credentials.

- Ensure your database has tables `users` and `client\_query\_details` with appropriate columns.

4. \*\*Run the application:\*\*

```

streamlit run app.py

```

---

## Usage

- \*\*Login:\*\* Enter your username and password. Select your role.

- \*\*Register:\*\* Click "New User Registration" to create a new account.

- \*\*Submit Query:\*\* Clients can add queries with details and screenshots.

- \*\*View Queries:\*\* Use filters, search, and pagination to manage queries.

- \*\*Update Query:\*\* Support/Admin can update query status.

- \*\*Dashboard:\*\* View analytics and summary of queries.

---

## Database Schema

\*\*users\*\*

- `id` (Primary Key)

- `username` (Unique)

- `hashed\_password`

- `role` ("Client" or "Support")

\*\*client\_query\_details\*\*

- `query\_id` (Primary Key)

- `emailid`

- `mobilenumber`

- `query\_heading`

- `query\_description`

- `status` ("Open"/"Closed")

- `user\_id` (Foreign Key to users)

- `screenshot` (BLOB or bytea)

---

## Customization

- \*\*Styling:\*\* Custom CSS is injected for a modern, centered UI.

- \*\*Validation:\*\* Email and required fields are validated before submission.

- \*\*Error Handling:\*\* All database and form errors are shown in the UI.

- \*\*Dashboard:\*\* Customize analytics in `dashboard.py`.

---

## Troubleshooting

- \*\*Database Errors:\*\* Check your connection string and table schemas.

- \*\*File Upload Issues:\*\* Ensure your database supports BLOB/bytea for screenshots.

- \*\*Session Issues:\*\* Streamlit session state is used for navigation and role management.

---

## License

MIT License

---

## Authors

- Your Name

- Contributors

---

\*\*Enjoy managing your client queries efficiently!\*\*