

# QueryLog: A Collaborative Web Platform for Knowledge Sharing

Siddartha Reddy Boreddy,

SUNY Binghamton,

(607) – 296 - 8544

## ABSTRACT

In the era of rapid technological advancements and knowledge sharing, online platforms play a pivotal role in connecting individuals, fostering collaboration, and enabling the exchange of ideas. This project, titled “QueryLog: A Collaborative Knowledge Sharing and Discussion Platform,” aims to provide a feature-rich, user-friendly platform for users to ask questions, share knowledge, and engage in meaningful discussions.

QueryLog is a versatile platform combining question-and-answer modules, real-time chat in public and private brainstorming rooms, user notifications, live sessions, and personalized profiles to foster structured communication and knowledge sharing across categorized topics. Built with Flask, SQLAlchemy, and Socket.IO, it delivers a robust, interactive user experience. Notable features include private rooms, allowing creators to invite specific users for secure collaboration, alongside voting, commenting, and notification mechanisms to boost engagement. Designed as a prototype for academic submission, QueryLog showcases modern web development practices and holds potential for applications in education, professional forums, and community-driven projects.

## 1. INTRODUCTION

QueryLog is an innovative web-based platform crafted to facilitate knowledge sharing and enhance collaboration across diverse topics. By integrating features such as question-and-answer modules, real-time chat in both public and private brainstorming rooms, user notifications, live sessions, and personalized profiles, the platform provides a comprehensive space for structured communication and effective knowledge dissemination. Built using Flask as the backend framework, SQLAlchemy for robust database management, and Socket.IO for real-time communication, QueryLog ensures a seamless and interactive user experience.


The platform stands out with its private room

functionality, enabling creators to curate exclusive discussions by inviting specific users. This feature ensures secure and focused collaboration, ideal for tackling specialized topics or fostering group creativity. To further encourage participation, QueryLog incorporates voting and commenting features, as well as real-time notifications to keep users engaged and informed. The thoughtful categorization of topics enhances navigation, allowing users to explore and contribute to areas of their interest effortlessly.

Initially designed as an academic prototype, QueryLog embodies the potential for broader applications in education, professional forums, and community-driven projects. The project showcases the integration of modern web development tools and practices, highlighting scalability, efficiency, and user-centric design. By providing a platform that bridges the gap between structured knowledge sharing and interactive collaboration, QueryLog sets the stage for impactful engagement in a variety of domains. The next sections will elaborate each feature in detail:

### 1.1 QUESTION-AND-ANSWER MODULE

QueryLog provides a seamless Ask Question feature, empowering users to post queries on a wide range of topics. The intuitive interface allows users to input their question title, description, and select relevant categories and subcategories, ensuring their queries reach the appropriate audience. This structured input system facilitates effective categorization, making it easier for others to discover and engage with posted questions. The feature is designed to encourage curiosity and knowledge sharing within the community, offering a user-friendly experience for contributing to the platform. Below is the screenshot of the Ask Question page.



The screenshot shows a web form titled "Ask a Question". It contains the following elements: a text input field for "Question Title", a larger text area for "Type your question here...", a dropdown menu for "Select Category", another dropdown menu for "Select Subcategory", and a blue "Submit" button at the bottom.

Fig 1: Screenshot of Ask a question page

The Feed feature acts as the hub of engagement, showcasing questions posted by users and allowing for answers and comments. Users can filter the feed by newest, answered, or unanswered questions, ensuring they can focus on the content that matters most to them. Questions are displayed with user information, associated tags, and interactive options for commenting or providing answers. The system also incorporates voting mechanisms to highlight the most helpful responses, fostering quality interactions and enabling the community to collaboratively build a repository of valuable insights. Below is the screenshot of the Feed page.

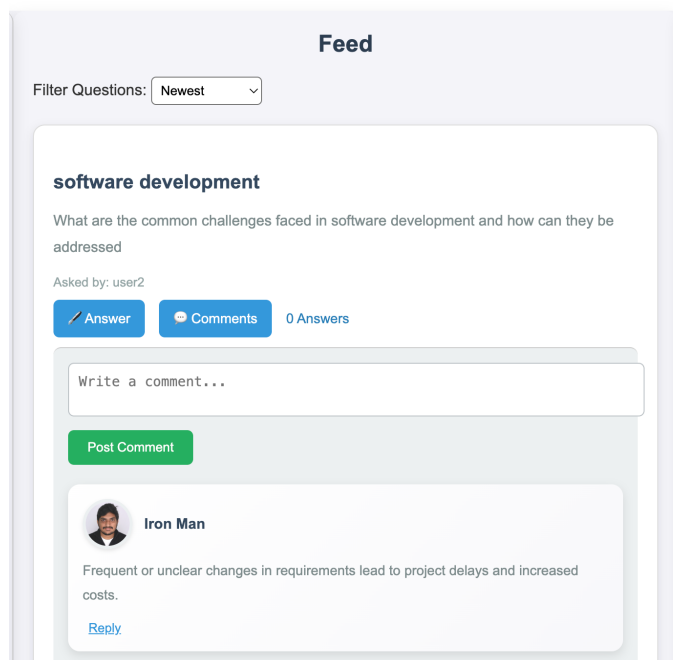


Fig 2: screenshot of feed page

## 1.2 BRAINSTROMING ROOMS

QueryLog introduces Brainstorming Rooms, a versatile module designed to foster collaboration and real-time discussions among users. This feature offers two distinct room types: public and private, catering to diverse interaction needs. Public rooms allow open conversations accessible to all users, encouraging discussions on shared interests or broad topics. In contrast, private rooms provide a focused and secure space for exclusive discussions, where access is managed through an invite-only mechanism. This functionality is particularly valuable for professionals or academic teams, offering a dedicated environment to brainstorm ideas, plan projects, or resolve queries with confidentiality.

Powered by Socket.IO, Brainstorming Rooms enable real-time chat functionality, ensuring messages are instantly

displayed for a seamless conversation flow. Private rooms ensure tailored participation by allowing creators to invite specific users or share unique access links. Additionally, all messages are stored in a secure database, enabling users to revisit past discussions when needed. With a user-friendly interface featuring an intuitive chat box and a scrollable display area, this feature combines flexibility, security, and efficiency. Query Log's Brainstorming Rooms embody its commitment to fostering community-driven collaboration and unlocking the potential of collective intelligence.

## Brainstorming Rooms

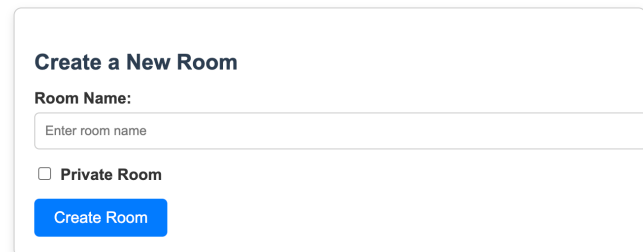


Fig 3: screenshot of brainstorming room creation

Fig 3 describes the room creation feature, where users can choose the private room option to ensure that only individuals with the access link can join, preventing unauthorized access

## 1.3 LIVE SESSIONS

The Live Sessions feature in QueryLog allows users to schedule and attend live discussions or presentations on various topics. This feature enables users to create a session by specifying the topic, date, time, and providing a Zoom link for the session. The sessions are listed on the platform, showing the upcoming events along with their respective details like the topic, date, and time.

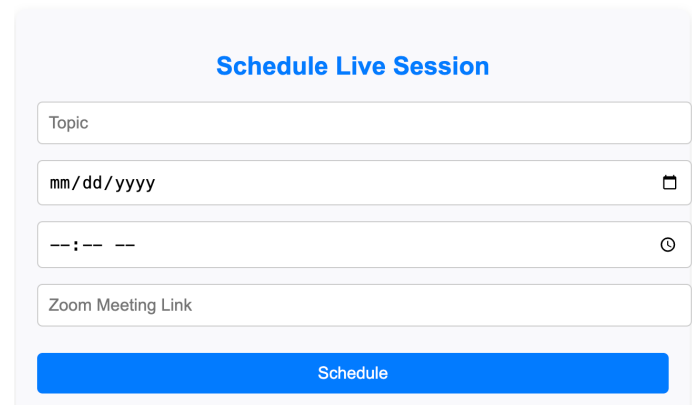


Fig 4: screenshot of Live Session Schedule feature

Fig 4 shows the live session schedule page where user can enter the topic details, date and time.

Users can view the list of live sessions, access the session details, and join via the provided Zoom link. The system ensures that all sessions are organized and displayed in chronological order,

making it easy for users to stay updated on upcoming events. This functionality provides an interactive way for users to engage in live discussions and enhances the collaborative experience within the platform.

### 1.4 VOTING AND COMMENTING

The **Voting and Commenting** feature in QueryLog enhance user interaction and engagement by allowing users to actively participate in discussions. This feature enables users to vote on answers, either upvoting or downvoting them based on their relevance, quality, or helpfulness. Each answer displays the current vote count, providing visibility into community feedback and helping users identify the most valuable contributions.

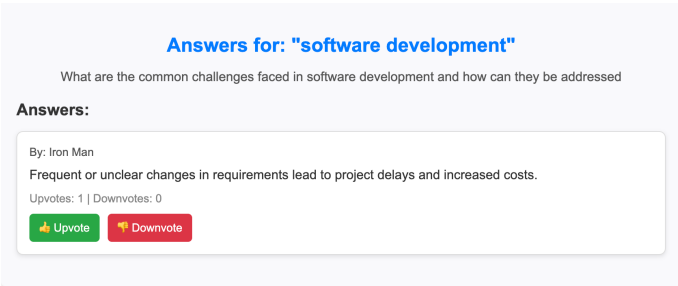


Fig 5: upvoting and downvoting feature

In addition to voting, users can comment on questions and answers to share insights, ask follow-up questions, or provide additional context. The commenting system supports hierarchical discussions by allowing replies to comments, enabling detailed and threaded conversations.

### 1.5 USER NOTIFICATIONS

User Notifications in QueryLog ensure that users remain informed about activities relevant to their interactions and contributions. This feature notifies users of key events such as new comments, replies, upvotes, or downvotes on their questions or answers. Additionally, users receive alerts for invitations to private Brainstorming Rooms or updates about scheduled Live Sessions they are part of. Notifications are displayed in a centralized dashboard or as a dropdown in the interface, ensuring visibility without disrupting the user experience.



Fig 5: Screenshot of Notification feature

Technically, the notification system is powered by event-driven architecture using WebSockets for real-time updates. The backend triggers notifications when specific actions occur, such as creating a comment or upvoting an answer, which are then stored in a database with timestamps and user identifiers. The frontend dynamically fetches and displays notifications using REST API calls or WebSocket events, ensuring low latency and seamless updates. Users can mark notifications as read, and the system maintains a history of notifications for easy reference. This robust design ensures that users stay engaged and informed while promoting efficient collaboration within QueryLog.

### 1.6 CATEGORIZATION AND FEED

Categorization and Feed in QueryLog is designed to streamline content organization and user accessibility. The platform categorizes questions into broad topics, each containing subcategories to provide hierarchical structure and better context. This categorization enables users to navigate the platform effectively, ensuring that they find questions and discussions relevant to their interests or expertise. The feed dynamically updates based on user preferences, such as category selection or applied filters like “Newest,” “Unanswered,” or “With Answers,” allowing a personalized browsing experience as shown in Fig 6.

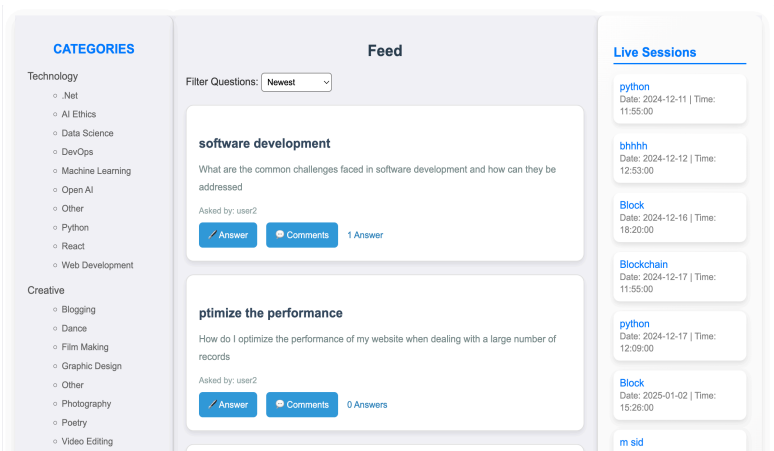
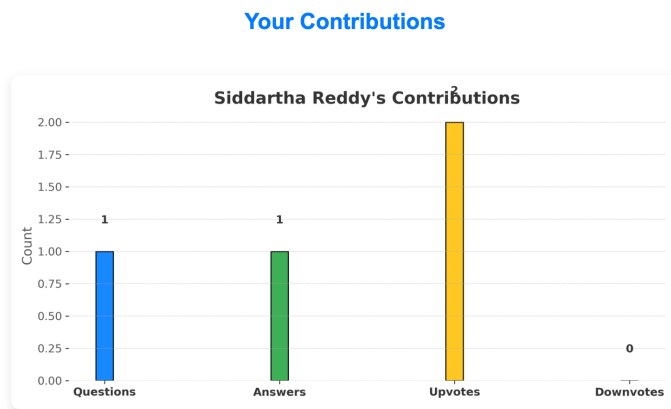


Fig 6: Feed page

From a technical standpoint, this feature relies on a relational database schema where questions are linked to categories and subcategories using foreign key relationships. When users interact with the feed, the backend dynamically queries the database based on selected filters and categories. The results are then rendered server-side into HTML templates using Flask, ensuring efficient and secure delivery of content. The feed’s filtering and sorting functionality leverages SQL queries to fetch and prioritize content based on timestamps, answer counts, or the presence of comments. To further optimize performance, pagination is employed to handle large datasets, ensuring the interface remains responsive and user-friendly.

## 1.7 PROFILE

The profile page in QueryLog serves as a personalized dashboard for users to view and manage their activities within the platform. It provides an overview of their contributions, such as the number of questions posted, answers given, comments added, and upvotes or downvotes received. Users can also edit their personal information, including their bio and profile picture, to keep their profile up to date. Additionally, the profile page features a visually appealing graph to represent user activity and engagement, offering insights into their contributions over time.



**Fig 7: Insights on Contributions over time**

From a technical perspective, the profile page leverages Flask as the backend framework to fetch and render user-specific data from a relational database. User data is stored in a dedicated User table, while activities such as questions, answers, and comments are linked via foreign key relationships. Upon accessing the profile page, Flask executes optimized database queries to gather user information and activity metrics, which are rendered using an HTML template. A standout feature is the dynamic graph generated with Matplotlib, which processes user activity data to create visualizations such as bar charts, saved as static images and embedded seamlessly in the page. Additionally, the page supports secure profile picture uploads through Flask's file-handling mechanisms, validating files for allowed formats (e.g., JPG, PNG) and storing them in a designated server directory. This structured approach ensures a comprehensive, interactive, and user-friendly profile experience.

## 1.8 SECURE USER AUTHENTICATION

The Secure User Authentication feature ensures that users can access their accounts safely and reliably. It provides a seamless login, sign-up, and logout experience while maintaining robust security protocols. Passwords are encrypted during storage, and authentication tokens

prevent unauthorized access. The system includes safeguards to prevent account duplication, ensuring each user has a unique identity. This feature is essential for protecting user data and providing a secure environment for interactions, such as posting questions, answering, voting, and commenting.

User authentication is implemented using Flask-Login for session management and Werkzeug's password hashing utility for secure password encryption. During registration, passwords are hashed and stored in the database, ensuring they cannot be retrieved in plaintext. Login functionality validates user credentials by comparing the provided password with the stored hash. Sessions are maintained using cookies, with a timeout mechanism to ensure inactive users are logged out automatically. Additionally, user input is sanitized to prevent injection attacks, and the application enforces HTTPS for secure communication. This comprehensive approach to authentication ensures the platform remains secure and reliable.

## 1.9 DATABASE ARCHITECTURE

Our platform's database is built on a robust relational schema designed to manage user data, content, and interactions efficiently. It serves as the backbone of the application, enabling seamless integration between features like user authentication, question and answer management, and content categorization. The database schema is designed with normalization principles to minimize redundancy and ensure data integrity.

Table	Description
answer	answers provided for questions
question	questions posted by users.
answer_votes	Tracks votes on answers.
category	Categorizes questions
comment	comments on questions and answers
message	Stores message
notification	Stores notifications
live_session	Maintains details about live sessions.
room	Store room details
room_invites	Invited users
subcategory	Stores subcategories
user	user-specific information

**Table 1: Database core tables**

Table 1 provides an overview of the key data tables used in the web application. It outlines their structure, relationships, and purpose within the system.

## 2. CHALLENGES AND SOLUTIONS

Real-time communication in the brainstorming rooms posed additional challenges, particularly in achieving instant updates with minimal latency. This was addressed through the integration of Socket.IO for WebSocket-based communication, complemented by database-backed message storage to ensure persistence. Similarly, implementing secure user authentication involved handling sensitive data securely, achieved by leveraging Flask-Login for session management and Werkzeug's password hashing for encrypting user credentials.

On the frontend, incorporating dynamic features such as live sessions, real-time messaging, and responsive interfaces required a focus on optimizing JavaScript and employing responsive frameworks like Bootstrap. Ensuring cross-device compatibility and rigorously testing real-time features were essential to providing a seamless user experience. These challenges, along with their solutions, highlight the technical depth and design considerations that contributed to the development of a robust, scalable, and user-friendly web application.

## 3. FUTURE ENHANCEMENTS

To further enhance the web application's capabilities and user experience, several future enhancements have been identified. One significant improvement is the development of a mobile application to complement the existing web platform, enabling users to access features like brainstorming rooms, notifications, and commenting on the go. Additionally, the integration of advanced analytics dashboards could provide users with valuable insights into their engagement and contribution patterns. This feature would utilize data visualization tools and machine learning to deliver personalized recommendations and trends.

Another key area for enhancement is the expansion of real-time collaboration features in brainstorming rooms. Future iterations could include functionalities such as collaborative document editing, screen sharing, and video conferencing, making the platform more appealing for academic and professional teams. Furthermore, gamification elements, such as user badges, leaderboards, and rewards for participation, could foster increased engagement and a sense of community among users.

Lastly, improving accessibility and integrating external platforms are priorities. Enhanced accessibility features, including screen reader support and multilingual options, would make the platform more inclusive. Integration with tools like Google Drive, Slack, or Zoom would streamline workflows and improve communication for collaborative

sessions. These planned enhancements aim to establish the web application as a versatile and inclusive platform for learning, collaboration, and professional development.

## 4. REFERENCES

- 1 – Flask documentation: <https://flask.palletsprojects.com/>
- 2 – Werkzeug Security Module: <https://werkzeug.palletsprojects.com/>
- 3 – PostgreSQL documentation: <https://www.postgresql.org/>