

Google Summer of Code 2025 Proposal

Personal Information

Name: Shubhanshu kumar

Email: subhanshukumar290@gmail.com

GitHub Profile:[github](#)

LinkedIn Profile:[LinkedIn](#)

University:Dr. A.P.J Abdul kalam technical university

Degree Program: B.Tech in Information technology

Year of Study: Final Year (8th Semester)

Title: FAQ Detection Assistant

Synopsis

Rocket.Chat, being a powerful open-source communication platform, is widely used by communities and organizations. One recurring challenge is handling frequently asked questions in a way that reduces manual effort and improves response quality. This project proposes the implementation of an **FAQ Detection Assistant** — an intelligent module that can detect if a user query has already been answered in the form of an FAQ.

The assistant will:

- Monitor specific channels (e.g., support) or GitHub Discussions
- Detect whether an incoming message matches existing FAQs using similarity algorithms and prompt-chained LLM responses
- Automatically respond, or notify a human agent for approval based on configurations
- Support various configurable actions (e.g., DM user, respond with approval, link to FAQ)

This project will make support faster, reduce load on moderators, and improve overall user satisfaction.

Benefits to the Community

- **Scalability:** Reduced burden on human moderators and community support members
 - **User Satisfaction:** Users receive faster and more accurate answers to common questions
 - **Consistency:** Ensures that responses are uniform across the platform
 - **Integration Friendly:** Can be easily adapted for Rocket.Chat, GitHub Discussions, and more
 - **Configurability:** Users can set thresholds, preferred actions, and customize how the assistant behaves
-

Deliverables

Time Period	Deliverable
Community Bonding	Deep understanding of Rocket.Chat APIs, set up dev environment, study prior FAQ structures
Phase 1 (First 6 weeks)	Webhook listener for messages LLM prompt chaining setup FAQ dataset schema and integration Matching algorithm for similarity detection
Evaluation 1	Working prototype that detects similar questions and logs suggested FAQ matches
Phase 2 (Next 6 weeks)	Configurable actions (DM, reply, notify) Feedback loop for user corrections UI panel for FAQ management (if applicable) Testing and improvements

Final Evaluation on	Complete FAQ Detection Assistant with deployment instructions, docs, and testing
---------------------------	---

Technical Approach

1. Message Listening via Webhook

Rocket.Chat apps support event-driven architecture. We will use Rocket.Chat's webhook system to listen to messages in configured rooms.

```
export async function webhookHandler(req, res) {  
  const { message } = req.body;  
  const userQuery = message.text;  
  const response = await detectFAQMatch(userQuery);  
  // Take action (respond/notify/etc.)  
  res.sendStatus(200);  
}
```

2. Prompt Chaining for LLM

To evaluate whether a message matches an FAQ, we can chain prompts like:

```
const systemPrompt = `You are an FAQ assistant. Given a user query and a list of  
FAQs, tell whether the query is similar to any FAQ.`;
```

```
const prompt = `User Query: ${query}`
```

```
FAQs:
```

```
1. ${faqList.join("\n")}
```

```
Output: Most similar FAQ or "No Match";
```

```
const response = await callLLM(systemPrompt + prompt);
```

3. Similarity Matching with Vector Embeddings

To improve accuracy, we'll also implement vector-based semantic matching using cosine similarity.

```
function cosineSimilarity(a, b) {  
  const dotProduct = a.reduce((sum, val, i) => sum + val * b[i], 0);  
  const magA = Math.sqrt(a.reduce((sum, val) => sum + val ** 2, 0));  
  const magB = Math.sqrt(b.reduce((sum, val) => sum + val ** 2, 0));  
  return dotProduct / (magA * magB);  
}
```

4. Configurable Actions

Users can configure what happens after an FAQ match:

- Direct message the user
- Post reply in channel
- Notify admin or approver before responding

This will be based on a simple JSON config:

```
{  
  "actions": {  
    "onMatch": "notify",  
    "notifyUsers": ["admin"]  
  }  
}
```

Timeline

Date Range	Milestone
May 20 - June 16	Community bonding, architecture discussions, first PR (setup)
June 17 - July 28	Build core detection logic, webhook listener, LLM prompt chaining
July 29 - August 26	Integrate feedback, refine matching, configurable actions, final testing
August 27 - September 9	Final documentation, blog post, and submission

About Me

I am **Shubhanshu**, a final-year B.Tech student specializing in Information Technology. My interests lie in building intelligent systems using AI, full-stack development, and open-source collaboration.

Skills:

- TypeScript, JavaScript, Node.js
- Prompt engineering and LLM apps
- REST APIs, Webhooks
- MongoDB, Express, React

Projects:

- **Tradexa:** A Zerodha-like stock trading app (MERN stack)
- **NEXAMEET:** Zoom-like video conferencing with WebSocket
- **Wandernest:** Travel community platform
- **Gemini Clone:** Chatbot interface with Gemini AI API

Contributions:

Currently exploring Rocket.Chat repositories and working on understanding apps-engine. I plan to start contributing through beginner-friendly issues during the community bonding period.

Final Thoughts

This project aligns perfectly with my skill set and goals. I'm confident I can deliver a useful, configurable, and intelligent FAQ Detection Assistant that will improve Rocket.Chat's support capabilities. I am excited about the opportunity to contribute to the Rocket.Chat community through GSoC 2025.

Thank you for your time and consideration.

End of Proposal