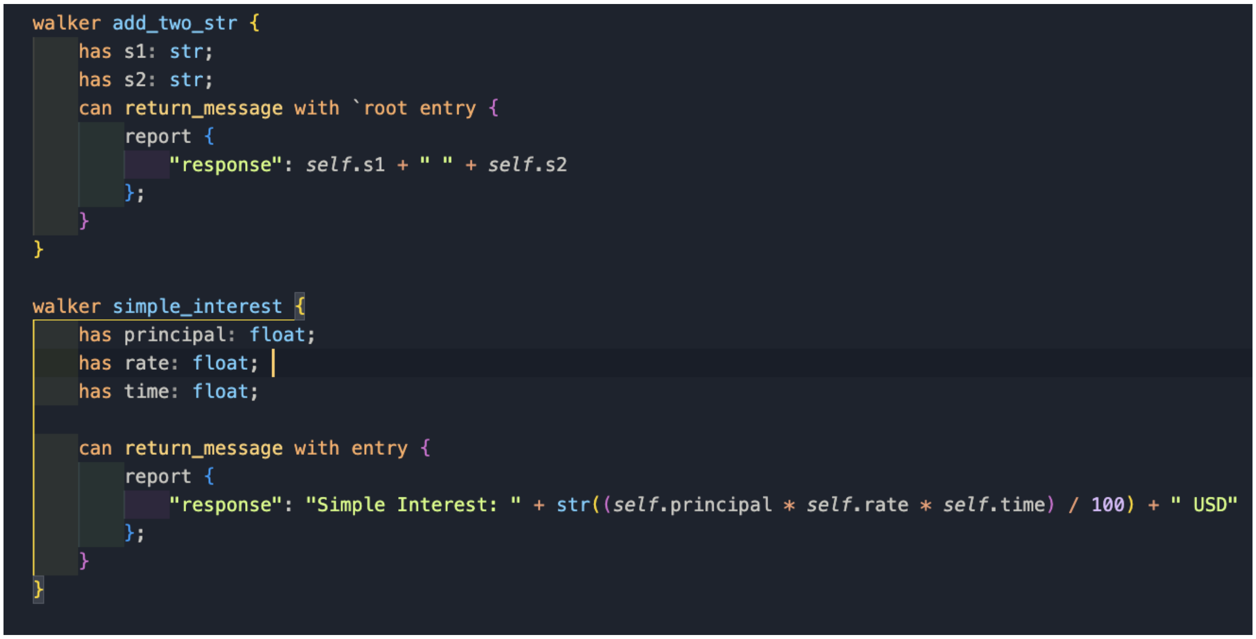
# EECS 449 Warm-up Assignment

The code to all this is present in this GitHub link:
https://github.com/ikalwani/EECS-449-Warm-Up-Assignment.git

## Part 1

I added two walkers in server.jac: add_two_str and simple_interest.
Add_two_str just concatenates two strings together with two parameters - two strings.
simple_interest just uses the simple interest formula to calculate simple interest with three float parameters: principal, rate and time.

Below is a screenshot of the two walkers.

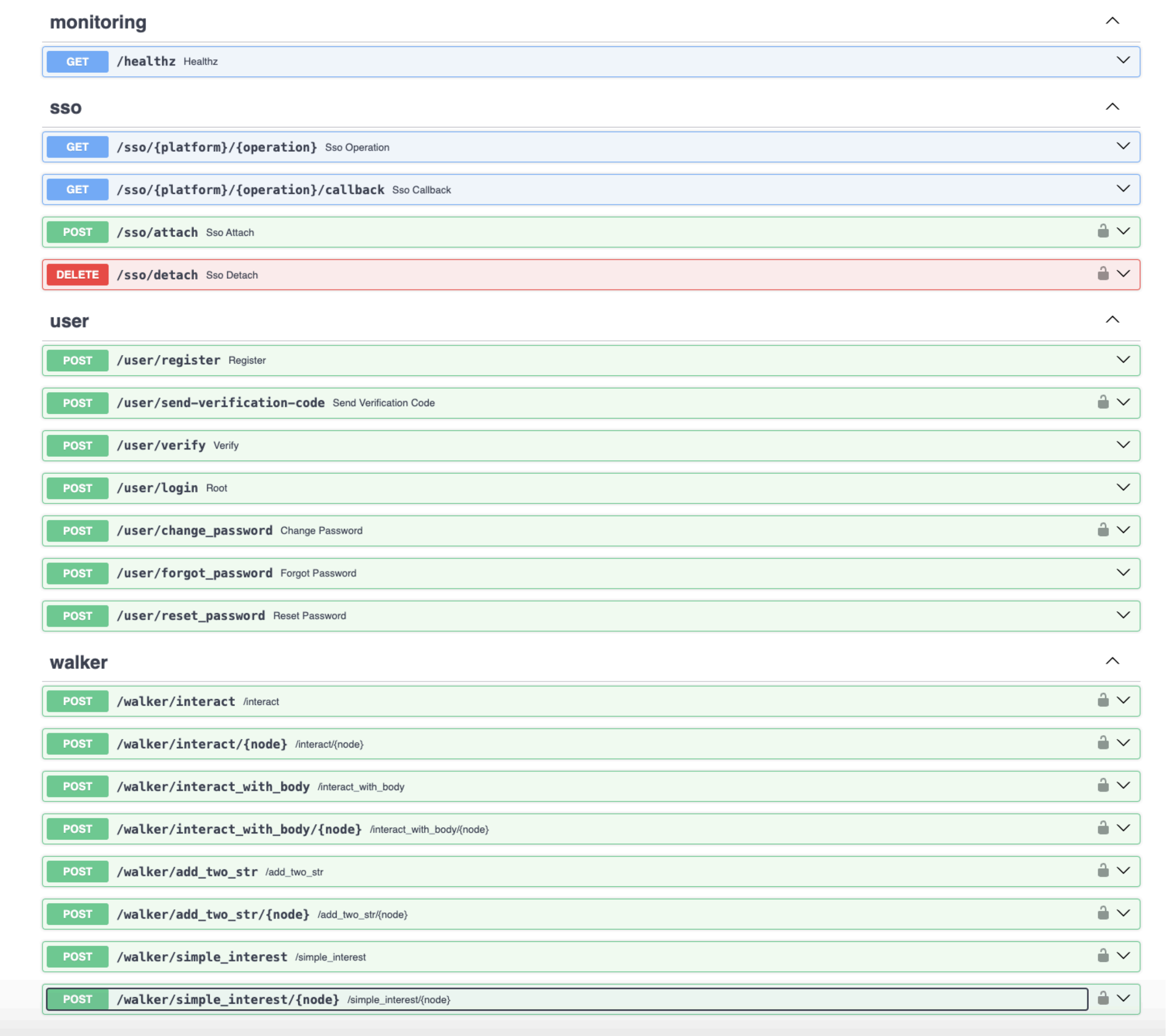


```
walker add_two_str {
    has s1: str;
    has s2: str;
    can return_message with `root entry {
        report {
            "response": self.s1 + " " + self.s2
        };
    }
}

walker simple_interest {
    has principal: float;
    has rate: float; 
    has time: float;

    can return_message with entry {
        report {
            "response": "Simple Interest: " + str((self.principal * self.rate * self.time) / 100) + " USD"
        };
    }
}
```

Below is a screenshot of the Swagger documentation showing the new walkers I implemented:



**monitoring**

GET /healthz Healthz

**sso**

GET /sso/{platform}/{operation} Sso Operation

GET /sso/{platform}/{operation}/callback Sso Callback

POST /sso/attach Sso Attach

DELETE /sso/detach Sso Detach

**user**

POST /user/register Register

POST /user/send-verification-code Send Verification Code

POST /user/verify Verify

POST /user/login Root

POST /user/change_password Change Password

POST /user/forgot_password Forgot Password

POST /user/reset_password Reset Password

**walker**

POST /walker/interact /interact

POST /walker/interact/{node} /interact/{node}

POST /walker/interact_with_body /interact_with_body

POST /walker/interact_with_body/{node} /interact_with_body/{node}

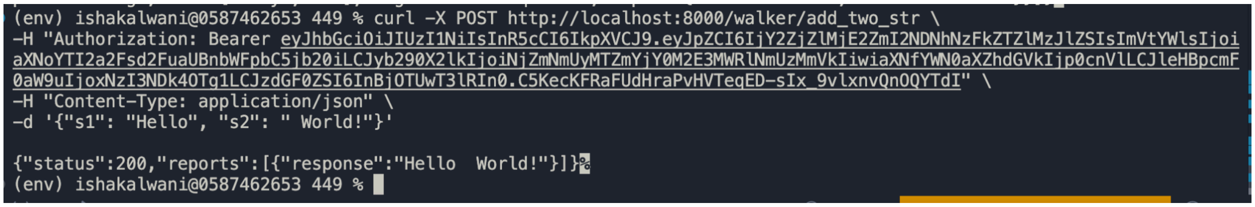
POST /walker/add_two_str /add_two_str

POST /walker/add_two_str/{node} /add_two_str/{node}

POST /walker/simple_interest /simple_interest

POST /walker/simple_interest/{node} /simple_interest/{node}

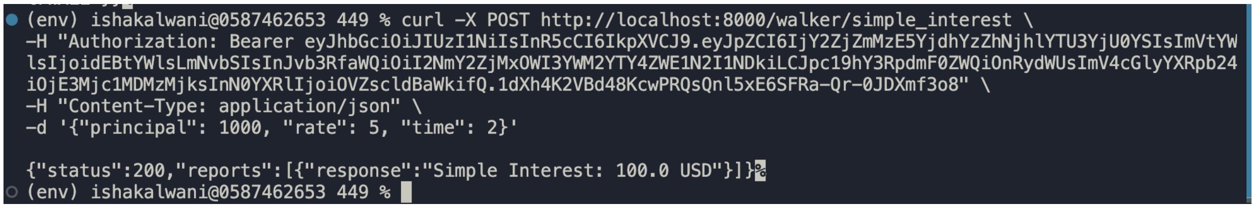
Below is a screenshot of the response from the API when I called my first walker - add_two_str:



```
(env) ishakalwani@0587462653 449 % curl -X POST http://localhost:8000/walker/add_two_str \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY2ZjZlMjE2ZmI2NDNhNzFkZTZlMzJlZSIsImVtYWlsIjoi
aXNoYTI2a2Fsd2FuaUBnbWFpbC5jb20iLCJyb290X2lkIjoiNjZmNmUyMTZmYjY0M2E3MWRlNmUzMmVkIiwiaXNfYWN0aXZhdGVkIjp0cnVlLCJleHBpcmF
0aW9uIjoxNzI3NDk4OTg1LCJzdGF0ZSI6InBjOTUwT3lRIn0.C5KecKFRaFUdHraPvHVTeqED-sIx_9vlxnvQnOQYTdI" \
-H "Content-Type: application/json" \
-d '{"s1": "Hello", "s2": " World!"}'

{"status":200,"reports":[{"response":"Hello  World!"}]}%
(env) ishakalwani@0587462653 449 %
```

Below is a screenshot of the response from the API when I called my second walker - simple_interest:



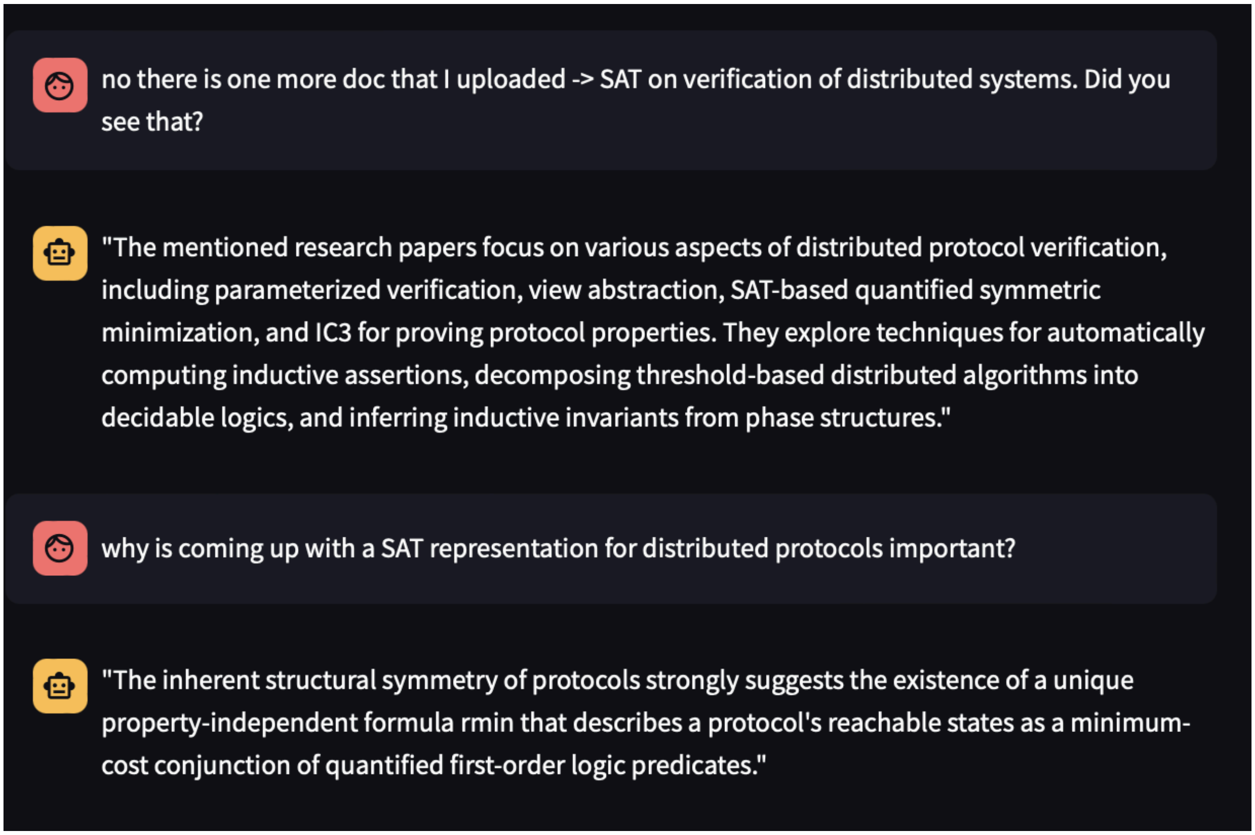
```
(env) ishakalwani@0587462653 449 % curl -X POST http://localhost:8000/walker/simple_interest \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY2ZjZmMzE5YjdhYzZhNjhlYTU3YjU0YSIsImVtYW
lsIjoidEBtYWlsLmNvbSIsInJvb3RfaWQiOiI2NmY2ZjMxOWI3YWM2YTY4ZWE1N2I1NDkiLCJpc19hY3RpdmF0ZWQiOnRydWUsImV4cGlyYXRpb24
iOjE3Mjc1MDMzMjksInN0YXRlIjoiOVZscldBaWkifQ.1dXh4K2VBd48KcwPRQsQnl5xE6SFRa-Qr-0JDXmf3o8" \
-H "Content-Type: application/json" \
-d '{"principal": 1000, "rate": 5, "time": 2}'

{"status":200,"reports":[{"response":"Simple Interest: 100.0 USD"}]}%
(env) ishakalwani@0587462653 449 %
```

## Part 2

I asked a question about a particular document about distributed systems.

Chatbot response:



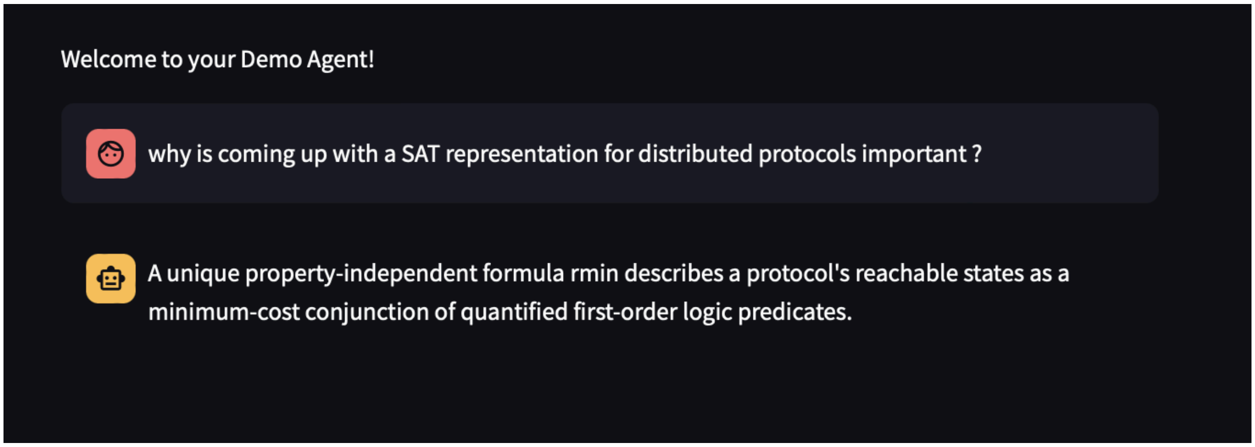
no there is one more doc that I uploaded -> SAT on verification of distributed systems. Did you see that?

"The mentioned research papers focus on various aspects of distributed protocol verification, including parameterized verification, view abstraction, SAT-based quantified symmetric minimization, and IC3 for proving protocol properties. They explore techniques for automatically computing inductive assertions, decomposing threshold-based distributed algorithms into decidable logics, and inferring inductive invariants from phase structures."

why is coming up with a SAT representation for distributed protocols important?

"The inherent structural symmetry of protocols strongly suggests the existence of a unique property-independent formula rmin that describes a protocol's reachable states as a minimum-cost conjunction of quantified first-order logic predicates."

I then used another LLM -> another free model on Ollama called llama 3.2. This is the response that I got:

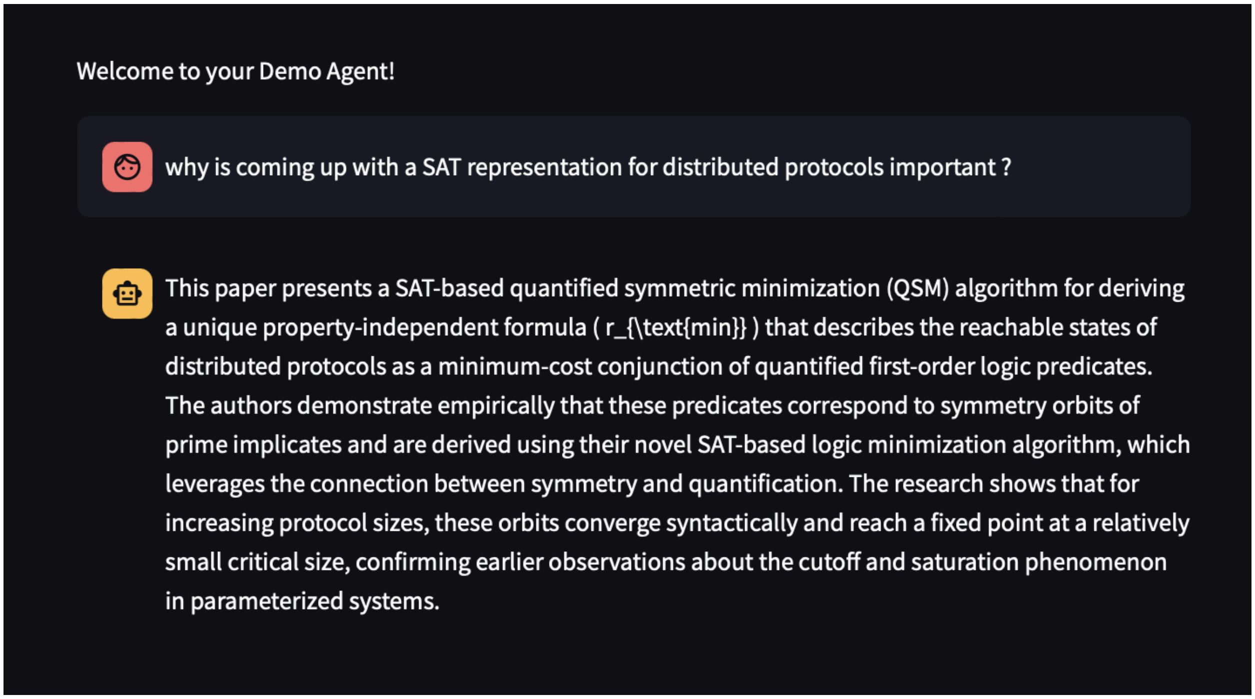


Welcome to your Demo Agent!

why is coming up with a SAT representation for distributed protocols important ?

A unique property-independent formula rmin describes a protocol's reachable states as a minimum-cost conjunction of quantified first-order logic predicates.

I then used another LLM -> another free model on Ollama called qwen2.5. This is the response that I got:



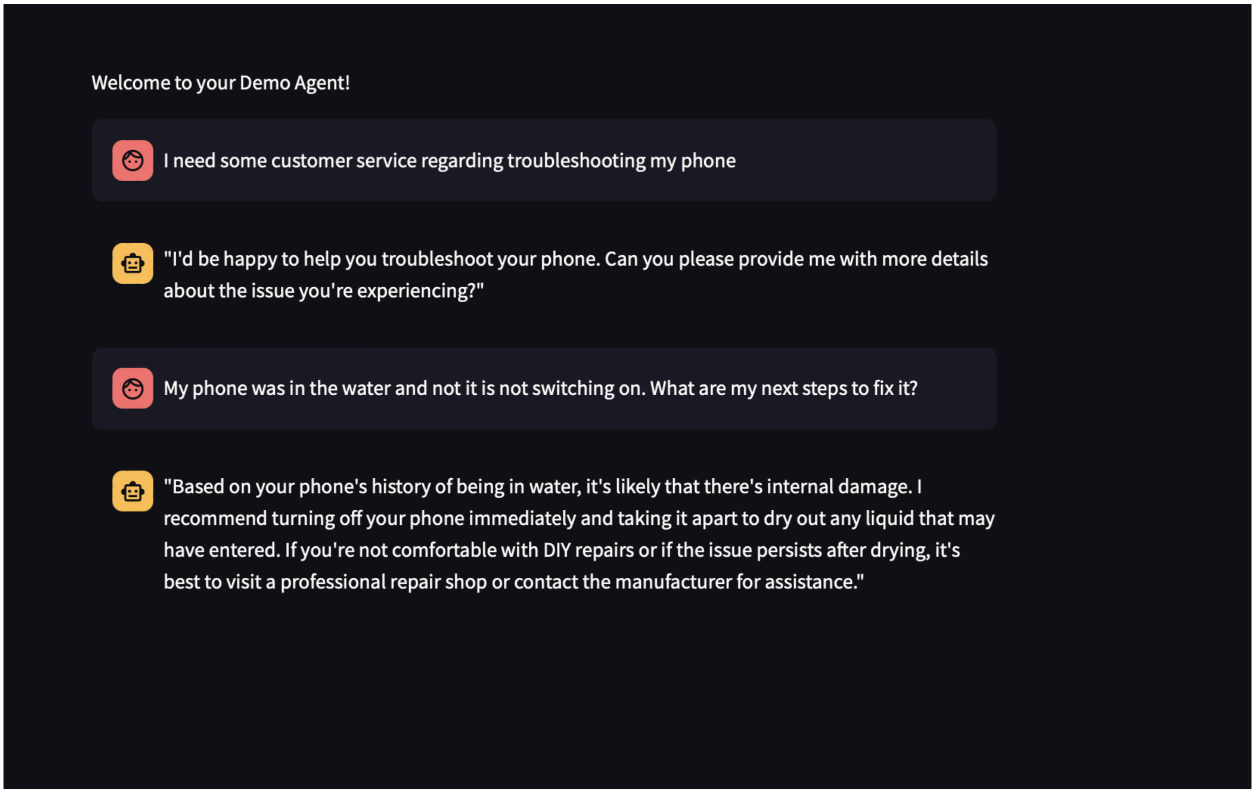
Welcome to your Demo Agent!

why is coming up with a SAT representation for distributed protocols important ?

This paper presents a SAT-based quantified symmetric minimization (QSM) algorithm for deriving a unique property-independent formula ( r_{\text{min}} ) that describes the reachable states of distributed protocols as a minimum-cost conjunction of quantified first-order logic predicates. The authors demonstrate empirically that these predicates correspond to symmetry orbits of prime implicates and are derived using their novel SAT-based logic minimization algorithm, which leverages the connection between symmetry and quantification. The research shows that for increasing protocol sizes, these orbits converge syntactically and reach a fixed point at a relatively small critical size, confirming earlier observations about the cutoff and saturation phenomenon in parameterized systems.

## **Part 3**

I added a SUPPORT state. It is designed to handle users who seek customer service, or troubleshooting, and general support queries within the chatbot. Users can interact with the chatbot to seek assistance regarding common issues. It is more of a customer support interaction. For example in the example below, I told the chatbot that my phone had water damage. It recommended next steps and provided assistance - acted as a customer service agent.



Welcome to your Demo Agent!

I need some customer service regarding troubleshooting my phone

"I'd be happy to help you troubleshoot your phone. Can you please provide me with more details about the issue you're experiencing?"

My phone was in the water and not it is not switching on. What are my next steps to fix it?

"Based on your phone's history of being in water, it's likely that there's internal damage. I recommend turning off your phone immediately and taking it apart to dry out any liquid that may have entered. If you're not comfortable with DIY repairs or if the issue persists after drying, it's best to visit a professional repair shop or contact the manufacturer for assistance."

## A Short Paragraph on How Things Went

Everything went smoothly for the most part. This was extremely useful - to actually code a working conversational AI to gain some experience and leverage these skills in the application that we make. It gave me a deeper understanding about RAG and the different models of Ollama. I liked how after each code snippet, there was an explanation given regarding what that piece of code does. One way to further improve the spec is to also attach some screenshots of what the chatbot is suppose to look like if our implementation is actually correct, so that we know that we are going on the right path.