

MARE

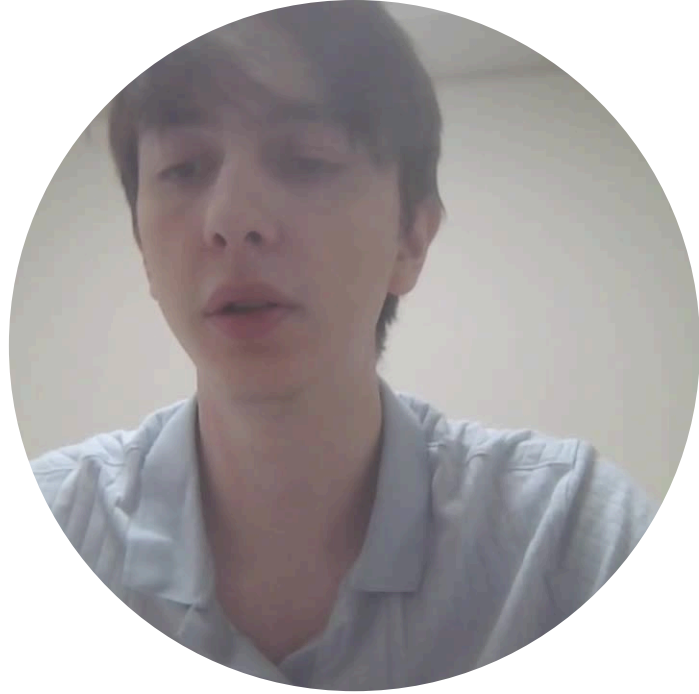
MINDFUL AGENT FOR
RESOLVING EXCEPTIONS



C3.ai

Team 33





TEAM MEMBERS



Andrea Núñez García

AI Model Fine Tuning



Emilio Berber Maldonado

Compiler Designer



Alfonso Ramirez Alvarado

Compiler Designers



Juan Zambrano Barajas

AI Model Fine Tuning



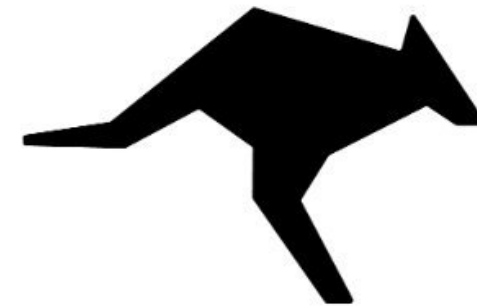
Fabian Lioner Rocha

LLM Integration Specialist



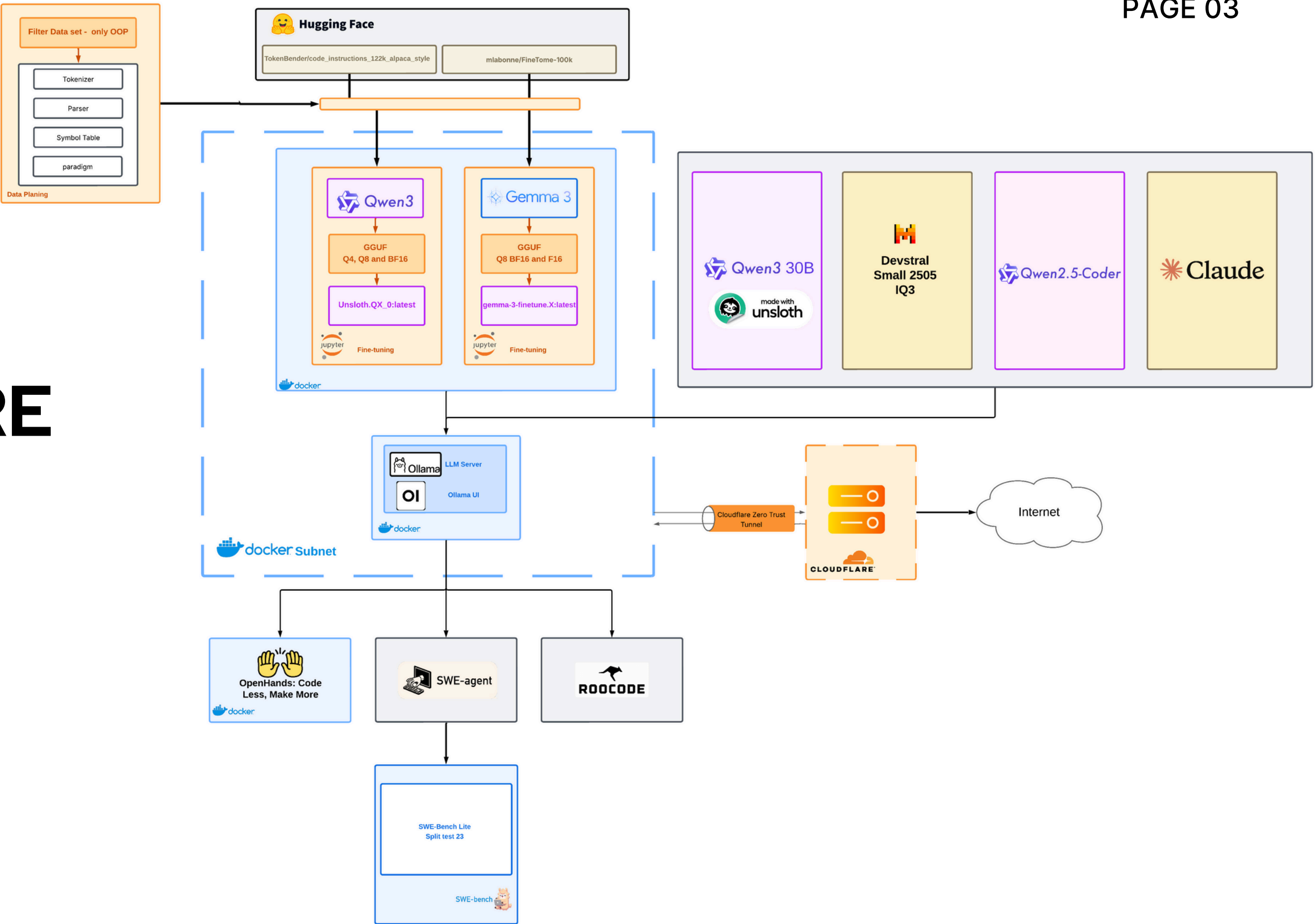
DESCRIPTION

This project introduces a multi-agent artificial intelligence system that enhances code quality and maintainability by autonomously addressing complex Object-Oriented Programming issues in GitHub repositories, featuring the specialized SWE-agent operating via terminal, RooCode integrated directly into Visual Studio for coding assistance, and an intuitive "Open Hands" web UI enabled agent for broader interaction, all powered by a diverse array of the language models for optimal performance and efficiency.

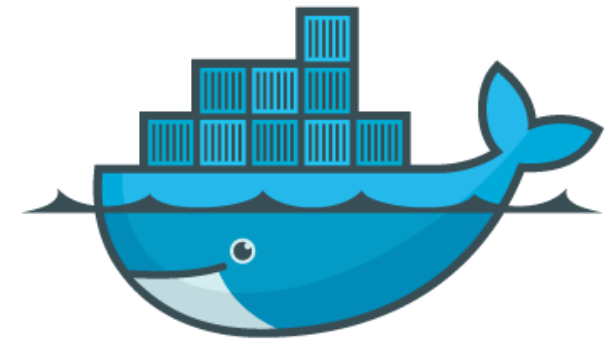




ARCHITECTURE DIAGRAM



FULL TECH STACK

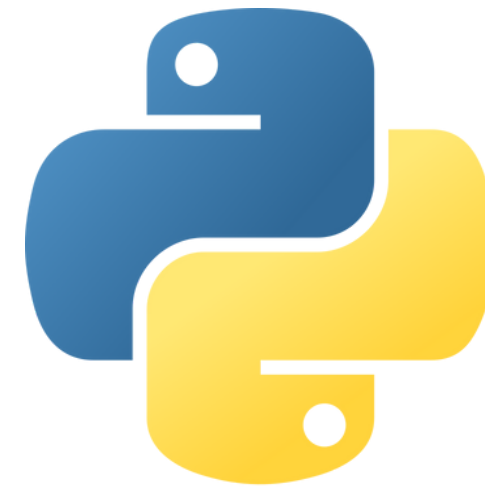


docker

Docker



Ollama



Python



Jupyter Notebook



Qwen

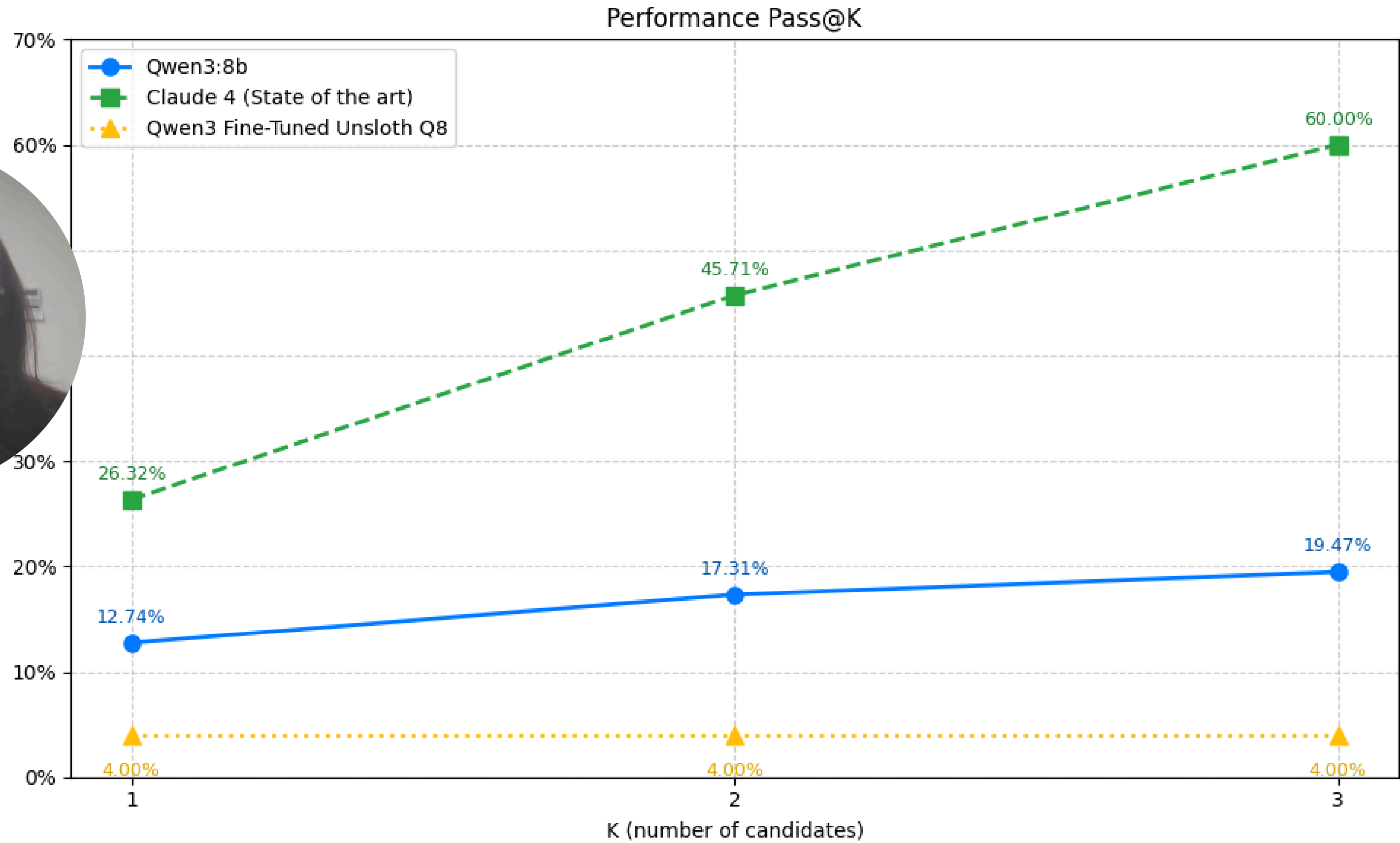
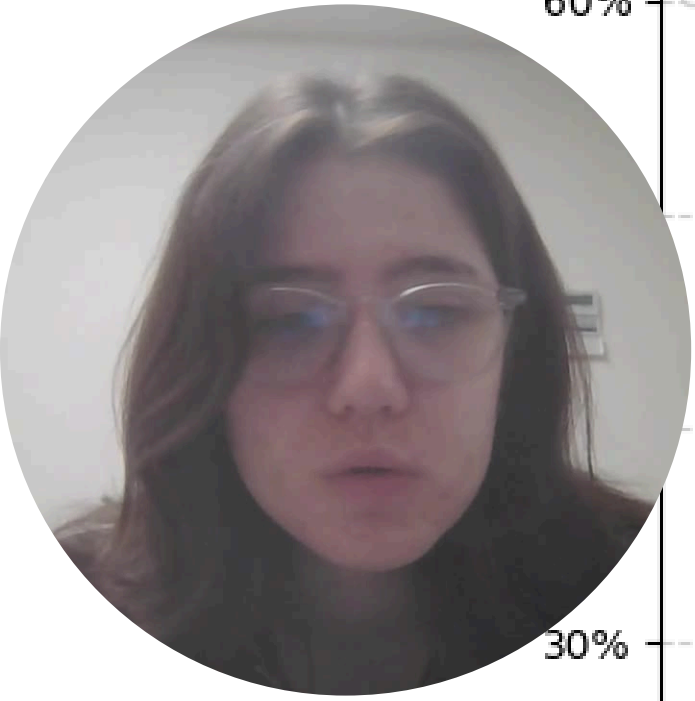


Unsloth

KEY FINDINGS

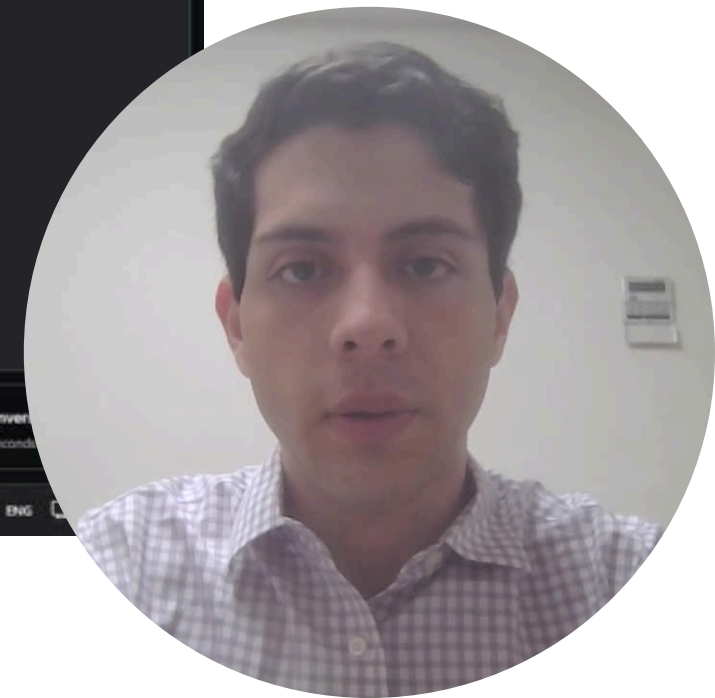
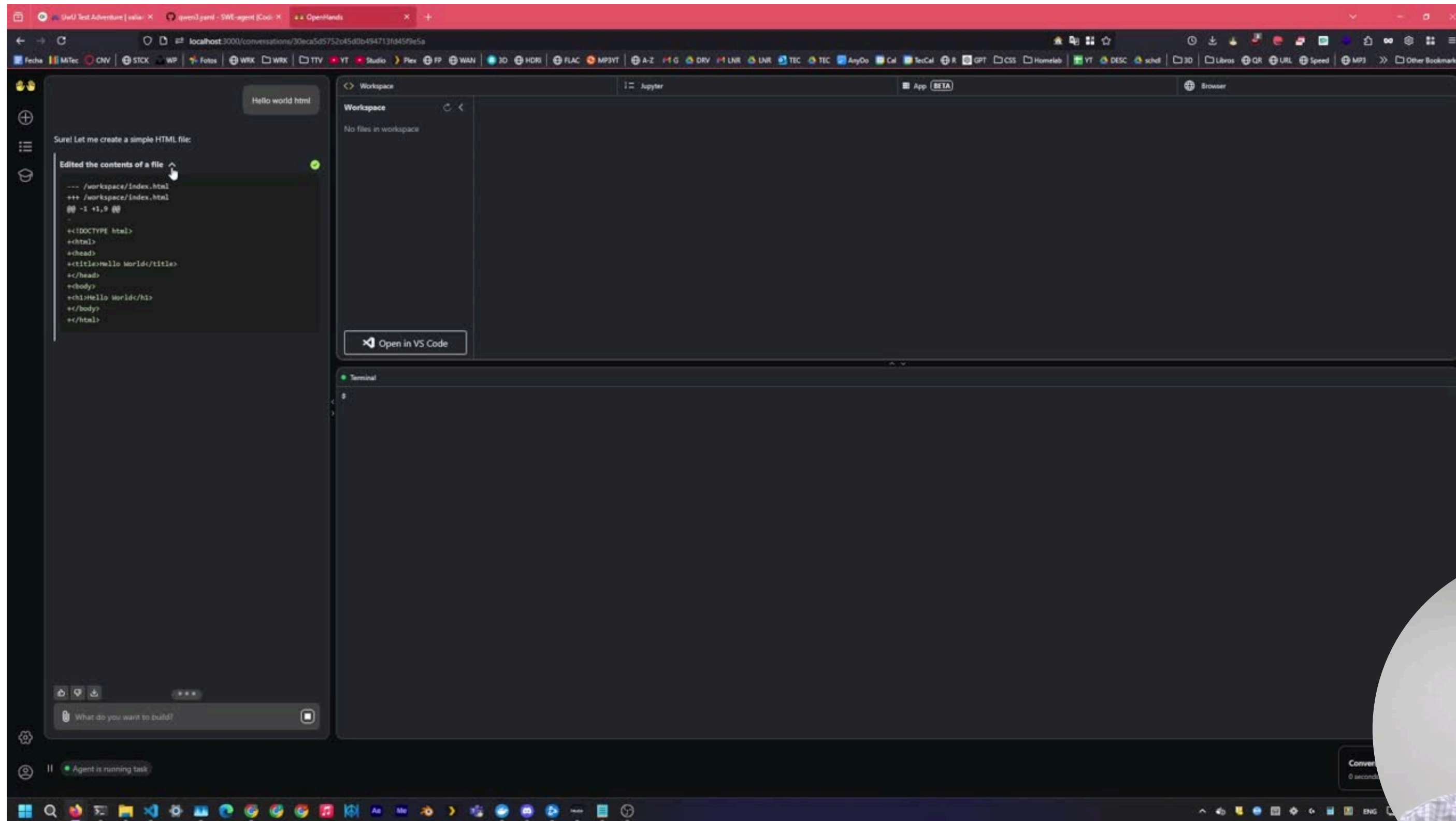


- Fine tuning certain models, like Qwen3 can be problematic, since its reasoning capabilities and Mixture of Experts are concentrated in the top (final) layers.
- Large foundation models (Qwen3, gemma3) and expert models (Qwen2.5-coder, Devstral) often leads to performance degradation.
- Ollama presents certain limitations on the models and adapters it can use, as well as unusual, but reported docker performance drawbacks.
- Dataset size and quality is essential for fine tuning, LoRA adapters benefit from 1000 or more examples, while a full fine tune can require tens of thousands of rows.




DEMO OPEN HANDS (GEMMA 3)

PAGE 07



DEMO SWE-AGENT (QWEN 3 FINE-TUNED)



```
flioner@DESKTOP-HUJSSR0: ~$ sweagent run --config=config/ollama.yaml --env.repo.github_url=https://github.com/SWE-agent/test-repo --problem_statement.github_url=https://github.com/SWE-agent/test-repo/issues/1
[flioner@DESKTOP-HUJSSR0 sweagent]$ sweagent run --config=config/ollama.yaml --env.repo.github_url=https://github.com/SWE-agent/test-repo --problem_statement.github_url=https://github.com/SWE-agent/test-repo/issues/1
This is SWE-agent version 1.1.0 (hash='ec0f7cb811ddb7625b4bd893e1c130acc0af721') with SWE-ReX version 1.2.2
(rex_hash='unavailable').
Setting problem statement based on github issue url
Loaded environment variables from /home/flioner/Projects/Tec/sweagent/.env
Starting environment
Building image python:3.11 to install a standalone python to /swe: This might take a while (but you only have to do it once). To skip this step, set 'python_standalone_dir' to None.
Found free port 39617
Starting container python3.11-828da794-490b-4b61-b243-58a7c3ddfcfb with image python:3.11 serving on port 39617
Command: "docker run --rm -p 39617:8000 --name python3.11-828da794-490b-4b61-b243-58a7c3ddfcfb sha256:d11a882c169b297620c3216f812a83218bf2efae562f59deb4cbc76c88788b57 /bin/sh -c '/root/python3.11/bin/swerex-remote --auth-token dbfe718f-79af-4d4f-97a9-8ed706ec352b'"
Starting runtime at 39617
Runtime started in 1.03s
Environment Initialized
Resetting repository SWE-agent__test-repo to commit HEAD
Running agent
Setting up agent for instance SWE-agent__test-repo-11
Trajectory will be saved to
/home/flioner/Projects/Tec/sweagent/trajectories/flioner/ollama_ollama/gemini:8b__t-0.00__p-1.00__c-0.00__SWE-agent__test-repo-11/SWE-agent__test-repo-11/SWE-agent__test-repo-11.traj
Uploading file from /home/flioner/Projects/Tec/sweagent/tools/registry to /root/tools/registry
Created zip file at /tmp/tmpvud6202/zipper_tzwwfec.zip
Uploading file from /home/flioner/Projects/Tec/sweagent/tools/search to /root/tools/search
Created zip file at /tmp/tmpiv777qr/zipper_tzwwfec.zip
Uploading file from /home/flioner/Projects/Tec/sweagent/tools/edit to /root/tools/edit
Created zip file at /tmp/tmp26630vt/zipper_tzwwfec.zip
Uploading file from /home/flioner/Projects/Tec/sweagent/tools/fixme to /root/tools/fixme
Created zip file at /tmp/tmpc8lrmal/zipper_tzwwfec.zip
Uploading file from /home/flioner/Projects/Tec/sweagent/tools/registry to /root/tools/registry
Created zip file at /tmp/tmp1hpa2b3/zipper_tzwwfec.zip
Uploading file from /home/flioner/Projects/Tec/sweagent/tools/diff_state to /root/tools/diff_state
Created zip file at /tmp/tmpx2ujcxh/zipper_tzwwfec.zip
Resetting tools
/home/flioner/Projects/Tec/sweagent/.venv/lib/python3.13/site-packages/ghapi/core.py:114: UserWarning: Neither GITHUB_TOKEN nor GITHUB_JWT_TOKEN found: running as unauthenticated
else: warn('Neither GITHUB_TOKEN nor GITHUB_JWT_TOKEN found: running as unauthenticated')
SYSTEM (main)
SETTING: You are an autonomous programmer, and you're working directly in the command line with a special interface.

IT IS ABSOLUTELY NECESSARY THAT YOU STICK TO THE TEMPLATE FORMATS, any deviation from these instructions will result in a complete disqualification

The special interface consists of a file editor that shows you 100 lines of a file at a time.
In addition to typical bash commands, you can also use the following commands to help you navigate and edit files.

You can also execute the given command directly in bash

[required]: The bash command to execute.

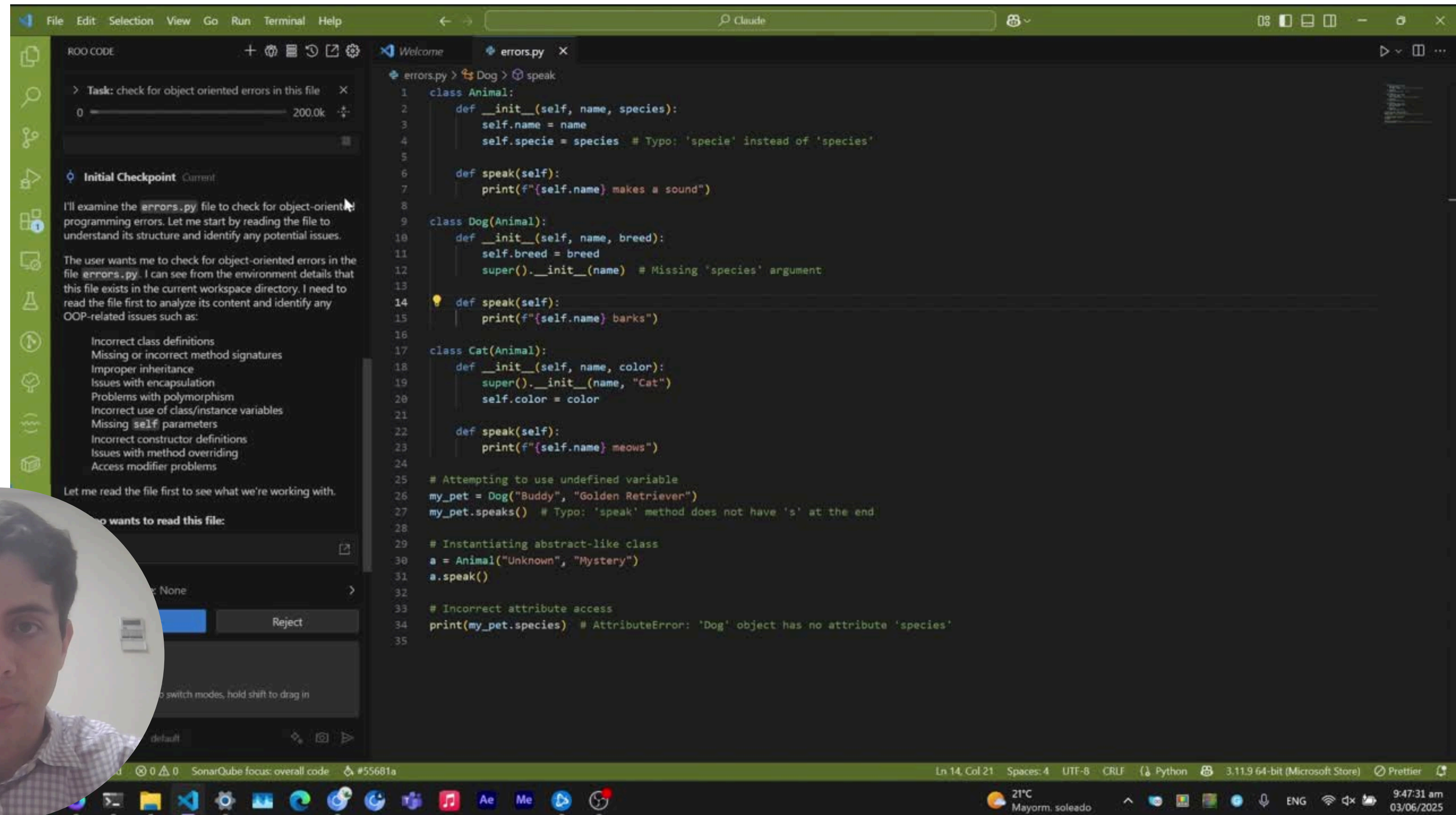
You can also search files with the given name or pattern in dir. If dir is not provided, searches in the current directory
[required]: the name of the file or pattern to search for. supports shell-style wildcards (e.g. *.py)
[optional]: the directory to search in (if not provided, searches in the current directory)

You can also search for search_term in all files in dir. If dir is not provided, searches in the current directory
[required]: the term to search for
[optional]: the directory to search in (if not provided, searches in the current directory)
```

11:35:37 am
29/05/2025

DEMO ROO CODE (CLAUDE 4)

PAGE 07



The screenshot displays the Roo Code IDE interface. The main editor window shows a Python file named `errors.py` with the following code:

```
1 class Animal:
2     def __init__(self, name, species):
3         self.name = name
4         self.specie = species # Typo: 'specie' instead of 'species'
5
6     def speak(self):
7         print(f"{self.name} makes a sound")
8
9 class Dog(Animal):
10     def __init__(self, name, breed):
11         self.breed = breed
12         super().__init__(name) # Missing 'species' argument
13
14     def speak(self):
15         print(f"{self.name} barks")
16
17 class Cat(Animal):
18     def __init__(self, name, color):
19         super().__init__(name, "Cat")
20         self.color = color
21
22     def speak(self):
23         print(f"{self.name} meows")
24
25 # Attempting to use undefined variable
26 my_pet = Dog("Buddy", "Golden Retriever")
27 my_pet.speaks() # Typo: 'speaks' method does not have 's' at the end
28
29 # Instantiating abstract-like class
30 a = Animal("Unknown", "Mystery")
31 a.speak()
32
33 # Incorrect attribute access
34 print(my_pet.species) # AttributeError: 'Dog' object has no attribute 'species'
35
```

The left sidebar contains a task description and a list of OOP-related issues:

Task: check for object oriented errors in this file

Initial Checkpoint Current

I'll examine the `errors.py` file to check for object-oriented programming errors. Let me start by reading the file to understand its structure and identify any potential issues.

The user wants me to check for object-oriented errors in the file `errors.py`. I can see from the environment details that this file exists in the current workspace directory. I need to read the file first to analyze its content and identify any OOP-related issues such as:

- Incorrect class definitions
- Missing or incorrect method signatures
- Improper inheritance
- Issues with encapsulation
- Problems with polymorphism
- Incorrect use of class/instance variables
- Missing `self` parameters
- Incorrect constructor definitions
- Issues with method overriding
- Access modifier problems

Let me read the file first to see what we're working with.

The user wants to read this file:

None

Reject

to switch modes, hold shift to drag in

default

The bottom status bar shows the current line and column (Ln 14, Col 21) and the file encoding (UTF-8).

THANK YOU!



Team 33

