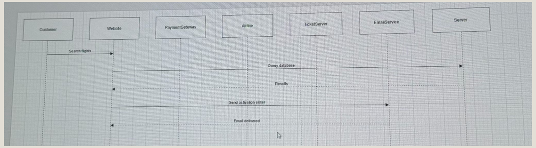
COMP 560

Kristine Cho David Bucheli
Alex Luna Tomas Sosa
Nicolas Sapriza

Generating Diagrams with an AI Model

Fine tuning an open source model to generate diagrams for draw.io

Project roadmap



IDEA

We first came up with a motivation to use an AI model capable of locally generating technical diagrams. These models would ultimately be promoted with natural language.

Research

We researched models and found the best one to use in our case would be Gemma 3 used in conjunction with draw.io for the diagram generation

Fine-Tuning

We worked on fine tuning the open sourced Gemma-3 model on the UNC library computers which had T1000 GPUs for more computing power. The specifically used the 1 billion parameter model

Problems

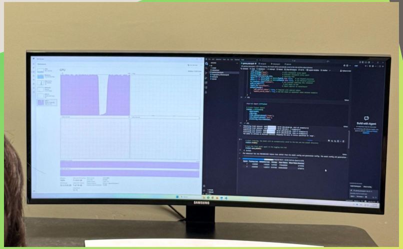
We ran into some problems with maxing out our GPU as well as finding datasets to be able to train our model.

Progress

AI models were unable to generate models in XML so we decided to pivot to MermaidJS.

Goal

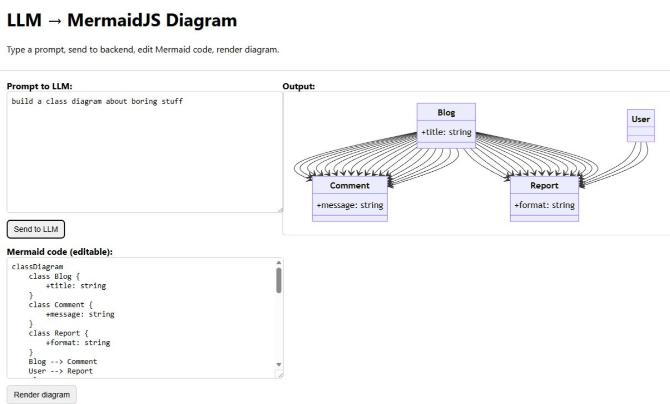
Ultimately, we ended with a small model capable of consistently generating valid diagrams.



COMP 560

RESULTS

1



LLM → MermaidJS Diagram

Type a prompt, send to backend, edit Mermaid code, render diagram.

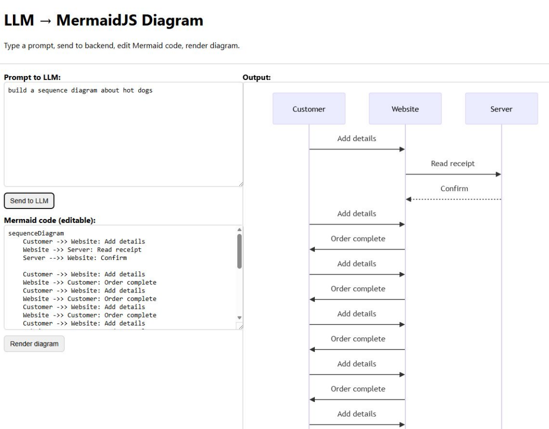
Prompt to LLM:

```
build a class diagram about boring stuff
```

Mermaid code (editable):

```
classDiagram
    class Blog {
        +title: string
    }
    class Comment {
        +message: string
    }
    class Report {
        +format: string
    }
    Blog --> Comment
    User --> Report
```

2



LLM → MermaidJS Diagram

Type a prompt, send to backend, edit Mermaid code, render diagram.

Prompt to LLM:

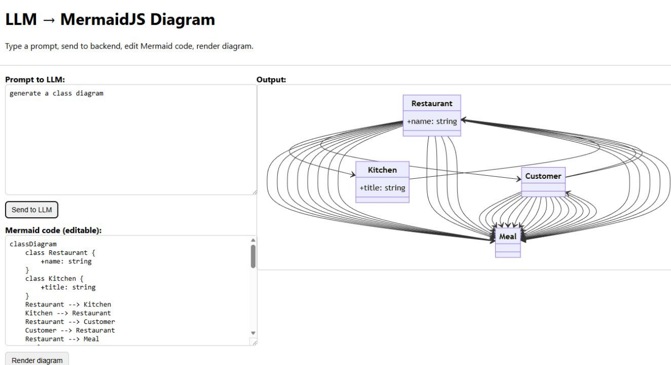
```
build a sequence diagram about hot dogs
```

Mermaid code (editable):

```
sequenceDiagram
    Customer ->> Website: Add details
    Website ->> Server: Read receipt
    Server -->> Website: Confirm

    Customer ->> Website: Add details
    Website ->> Customer: Order complete
    Customer ->> Website: Add details
    Website ->> Customer: Order complete
    Customer ->> Website: Add details
    Website ->> Customer: Order complete
    Customer ->> Website: Add details
```

3



LLM → MermaidJS Diagram

Type a prompt, send to backend, edit Mermaid code, render diagram.

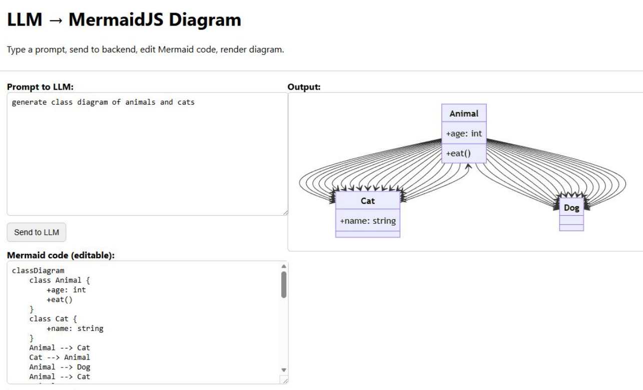
Prompt to LLM:

```
generate a class diagram
```

Mermaid code (editable):

```
classDiagram
    class Restaurant {
        +name: string
    }
    class Kitchen {
        +title: string
    }
    Restaurant --> Kitchen
    Kitchen --> Restaurant
    Restaurant --> Customer
    Customer --> Restaurant
    Restaurant --> Meal
```

4



LLM → MermaidJS Diagram

Type a prompt, send to backend, edit Mermaid code, render diagram.

Prompt to LLM:

```
generate class diagram of animals and cats
```

Mermaid code (editable):

```
classDiagram
    class Animal {
        +age: int
        +eat()
    }
    class Cat {
        +name: string
    }
    Animal --> Cat
    Cat --> Animal
    Animal --> Dog
    Animal --> Cat
```