February 2025

“Eli5 tutors”

Tutors AI - Generative AI augmentation to www.tutors.dev

INTERIM REPORT

STUDENT NAME: Ruslan Zhabskyi
STUDENT NUMBER: 20104105
COURSE: Higher Diploma in Computer Science

Commercial Title:

Eli5 tutors

Academic Title:

Tutors AI - Generative AI augmentation to www.tutors.dev

Preface

This document should be read in conjunction with the following:
Project Road Map: https://bit.ly/3WTcexL
Project Web Page: http://bit.ly/4hxqnJb
Forked Tutors Repository: https://bit.ly/417eZhk
Chat Tutors AI branch: https://bit.ly/4aSre4M

Abstract

Augmenting Generative Artificial Intelligence into Tutors will enhance students' learning experience and will give power to content creators to adjust content based on demand. Tutors AI will include Chat with AI, quick explain (eli5), AI powered web search and AI for content creators features.

Keywords

Tutors, LLM, Large Language Model, GenAI, Generative AI, Artificial Intelligence, Granite, IBM, Web Search, Svelte

Table of Contents

Table of Figures

Table of Tables

List of Abbreviations

AI	Artificial Intelligence
BBH	Big Bench Hard
DFD	Data Flow Diagram
Eli5	Explain like I am five
GenAI	Generative Artificial Intelligence
GPQA	Graduate-Level Google-Proof Question & Answer
IFEval	Instruction-Following Evaluation
LLM	Large Language Model
MATH	Mathematics Aptitude Test of Heuristics
MMLU-Pro	Massive Multitask Language Understanding - Professional
MUSR	Multistep Soft Reasoning
RLHF	Reinforcement Learning from Human Feedback
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege
TB	Trust Boundary
UCD	Use Case Diagram

Introduction

Background

In my own studies, I found certain concepts challenging to understand, especially those that were abstract or conceptual. I started to look outside of tutors to find videos that could provide an alternative explanation.

At the same time I was taking the “Learn how to learn” Coursera course to improve learning efficiency and increase content memorization. One of the highlights was to use analogies to improve understanding of complex systems and memorise content.

Eventually I started to use GenAI to explain content by asking to provide analogies. I found that I could learn faster and understand better. However, accessing these explanations required extra steps outside the tutors web app, also external LLM products would not have context from tutors to see the full picture.

Generative AI augmentation to www.tutors.dev solves these issues and enhances students’ learning experience. Also, it will minimize the time lecturers spend explaining basic concepts, allowing them to dedicate more attention to addressing advanced questions. It will give details on what concepts students find more challenging to understand which will lead to further content improvements.

Scope

AI augmentation to Tutors involves 4 parts:

- Chat AI feature
- AI Powered Web Search
- Eli5 - quick explain feature
- AI for Content Creators

Objectives

The objectives of the project are:

- Augment GenAI into tutors.dev for students to improve learning experience
- Empower Content Creators with additional insights and AI tools to embed changes

Mockup Design

Link to 2 minutes brief demo of the mockup design

Eli5 quick explain and chat features

Students will be able highlight a part of the text and quickly generate explanations based on proposed prompts (ex. Eli5 - explain like I am five). In case there is a need to open a chat with Tutors AI - students can click on the floating button.

Students can like, dislike or copy LLM responses.

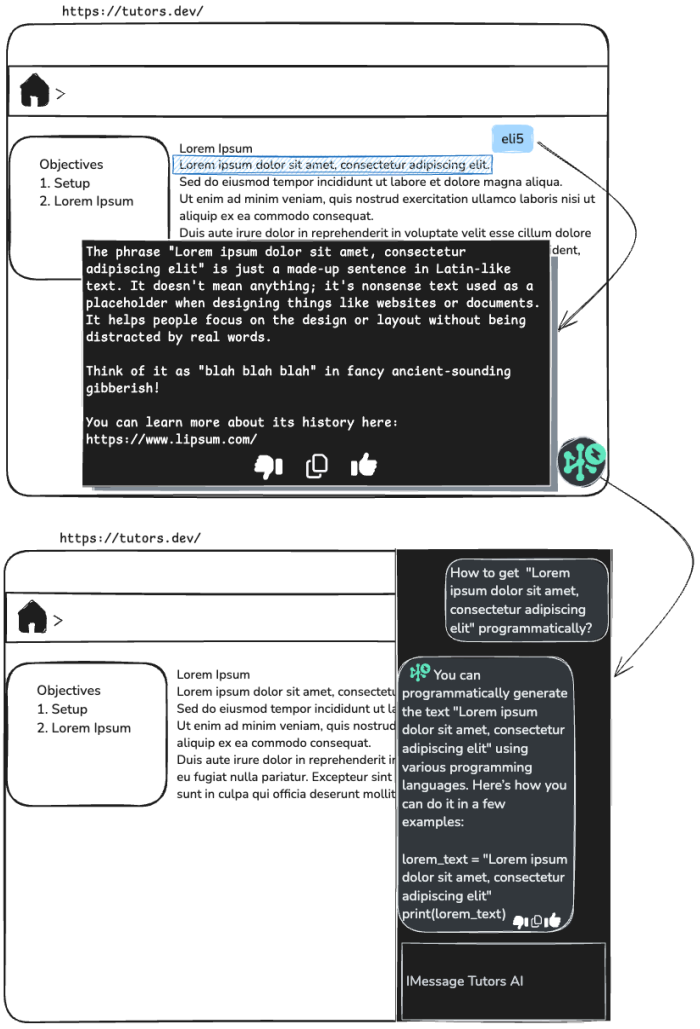


Figure 1: eli5 and chat Tutors AI features mockup

AI powered web search

In case a student needs to find specific information, they can use AI powered websearch which will list relevant resources with links and short explanations.

Figure 2: Tutors AI powered web search mockup

AI for content creators

Content creators will be able to see which topics require content improvements based on the amount of LLM calls on a specific article or lab.

It will be possible to double click and see what students found as the most helpful responses. Then content creators can in one click generate a content suggestion based on what students found as a perfect explanation for them. The suggested material modification can be used to update content if content creators see value.

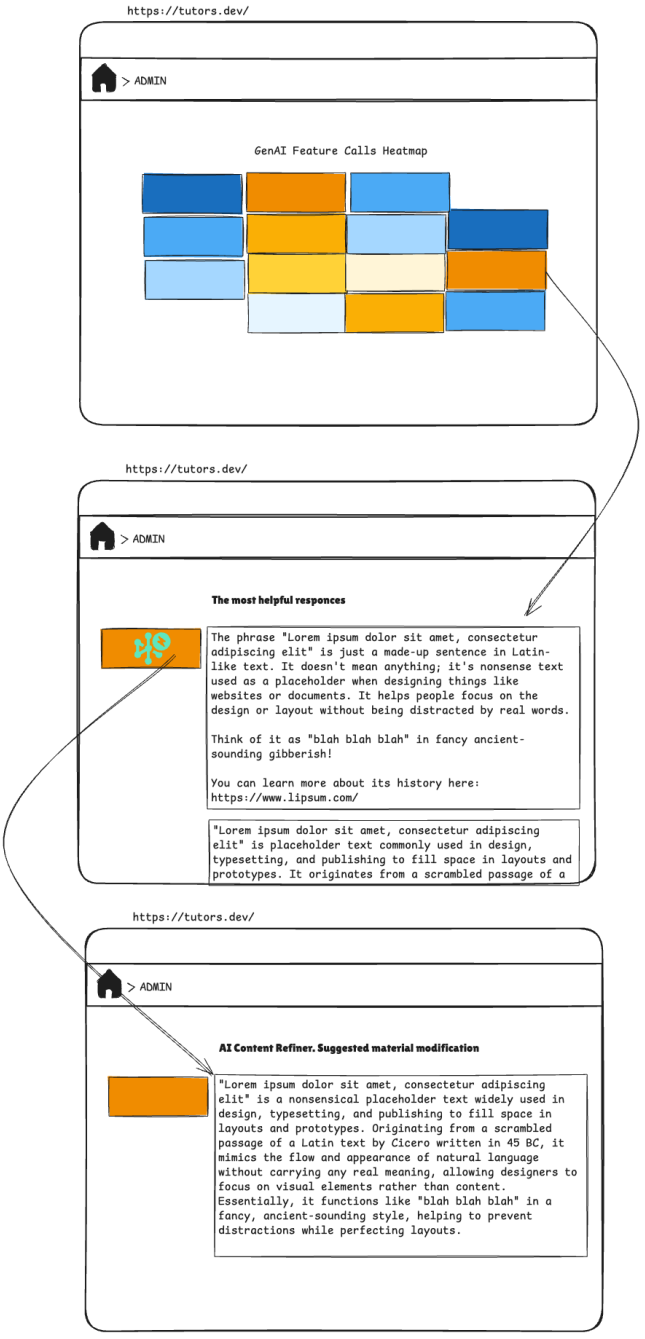


Figure 3: Tutors AI for content creators mockup

System Design

I am planning to run the system locally, however I decided to explore deployment options, their cost and provide high level diagrams. I am also adding Threat Modelling which is critical for security considerations of deployed systems.

Functional requirements

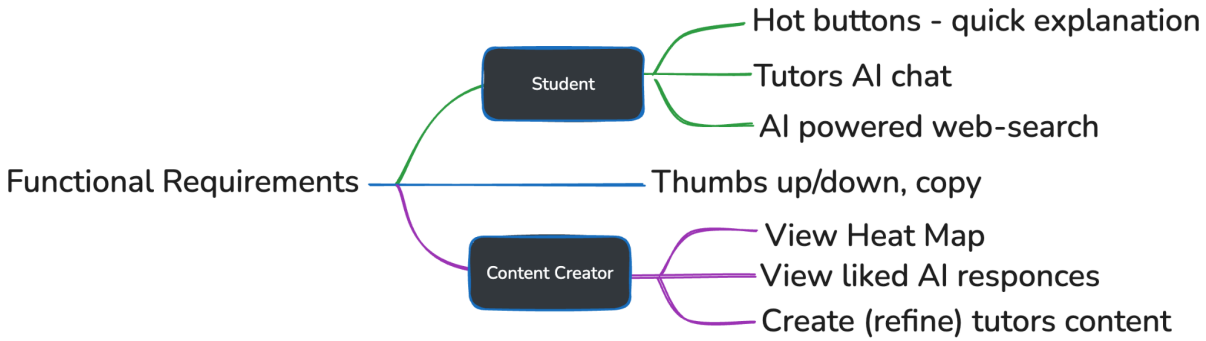


Figure 4: Functional Requirements

1. AI for students:
    a. Hot buttons - quick explanation of a selected text by predefined prompts (example: explain like I am five …)
    b. Chat with tutors AI - interactive AI assistant for students
    c. AI powered web-search - enhanced web search using AI powered summaries and result selection

2. AI for creators:
    a. AI requests heat map - visualize explanation demand per content
    b. View AI responses - analyse AI responses
    c. Create (refine) tutors content - AI powered tutors content creation based on students feedback

3. Thumbs up/down, copy

Operation	URL	Request	Response

Generate Quick Explanation	**POST** **http://localhost:11434/api/generate**	{
		"model": "granite3.1-dense:2b",
		"prompt": "System: You need to explain like I am five.\n\nUser: Why is the sky blue?",
		"stream": false,
		"options": {
		"temperature": 0.8
		},
		"format": "json"
		}
			responseId, status code, genAiResponse

Chat with Tutors AI	**POST http://localhost:11434/api/chat**	{
		"model": "granite3.1-dense:2b",
		"messages": [
		{
		"role": "system",
		"content": "You need to explain like I am five."
		},
		{
		"role": "user",
		"content": "Why is the sky blue?"
		},
		{
		"role": "assistant",
		"content": "{\n\"answer\": \"Imagine you're holding up your hand and looking at the sunlight …\"\n}"
		},
		{
		"role": "user",
		"content": "magnifying?"
		}
		],
		"stream": false,
		"options": {
		"temperature": 0.8
		},
		"format": "json"
		}
			responseId, status code, genAiResponse

~~Thumbs up~~	~~**POST http://localhost:XXXXX/api/mongo{/responseId}/helpful**~~	~~{"helpful": true}~~	~~status code~~
~~Thumbs down~~	~~**POST http://localhost:XXXXX/api/mongo/responseId/helpful**~~	~~{"helpful": false}~~	~~status code~~
AI Powered Web Search	**POST http://localhost:11434/api/chat**	{ "model": "granite3.1-dense:2b", "messages": [{ "role": "system", "content": "Your role is to provide a concise and accurate answer to the user's question using the provided search results. Ensure your answer is derived from the most relevant search snippets and	status code, answer query

		include links to sources with a brief description for additional context."
		},
		{
		"role": "search results",
		"content": "{\"results\": [{\"title\": \"Dublin - Wikipedia\", \"url\": \"https://en.wikipedia.org/wiki/Dublin\", \"snippet\": \"Dublin is the capital and largest city of Ireland. It is located on the east coast of the island, at the mouth of the River Liffey.\", \"displayed_url\": \"en.wikipedia.org/wiki/Dublin\"}, {\"title\": \"What is the Capital of Ireland? - WorldAtlas\", \"url\": \"https://www.worldatlas.com/cities/dublin.html\", \"snippet\": \"The capital of Ireland is Dublin, a city known for its historical landmarks, vibrant culture, and bustling economy.\", \"displayed_url\": \"www.worldatlas.com/cities/dublin.html\"}, {\"title\": \"Capital of Ireland | Britannica\", \"url\": \"https://www.britannica.com/place/Dublin-Ireland\", \"snippet\": \"Dublin, Irish Baile Átha Cliath, is the city, capital, and chief port of Ireland. It has been the center of Ireland's cultural and political life for centuries.\", \"displayed_url\": \"www.britannica.com/place/Dublin-Ireland\"}, {\"title\": \"Visit Dublin: Official Tourism Site\", \"url\": \"https://www.visitdublin.com/\", \"snippet\": \"Explore Dublin, the capital of Ireland. Discover its historic attractions, exciting activities, and local hospitality.\", \"displayed_url\": \"www.visitdublin.com\"}, {\"title\": \"Is Dublin the capital of Ireland? - Quora\", \"url\": \"https://www.quora.com/Is-Dublin-the-capital-of-Ireland\", \"snippet\": \"Yes, Dublin is the capital of Ireland. It is the largest city in the country and serves as its political, cultural, and economic hub.\", \"displayed_url\": \"www.quora.com/Is-Dublin-the-capital-of-Ireland\"}]}"
		},
		{
		"role": "user",
		"content": "What is the capital of Ireland?"
		}
		],
		"stream": false,
		"options": {
		"temperature": 0.7
		},
		"format": "json"
		}
		
		OpenAI Chat GPT 4. 2025. Response to Ruslan Zhabskyi, 11 Jan. 2025.	

AI requests heat map	**GET http://localhost:XXXXX/api/content/{contentUrl}/responses/count**		status code, responses count
View liked AI responses	**GET http://localhost:XXXXX/api/content/{contentUrl}/responses**	{“helpful”: true}	status code, list of responses
Create (refine) tutors content	**POST http://localhost:11434/api/chat**	{ "model": "granite3.1-dense:2b", "messages": [{ "role": "system", "content": "Your role is to refine tutors content. Your job is to combine original tutor content with AI-generated responses that students found helpful" }, { "role": "user", "content": "Here is the original content:\n'3*3=9'" }, { "role": "user", "content": "Here are some liked responses from the AI system:\n1. '3+3+3=9'" }, { "role": "user", "content": "Please generate a refined explanation based on the original content and the liked responses." }], "stream": false, "options": { "temperature": 0.8 }, "format": "json" }	status code, refinedContentId, generatedText

Table 1: REST API endpoints

AI Powered Web Search Diagram

According to Matt Williams (2024) and Ai Austin (2024) AI Powered websearch can be implemented by passing a metasearch engine or web search scraper code output to LLM via prompt with additional instructions. Below I graphed those two ideas.

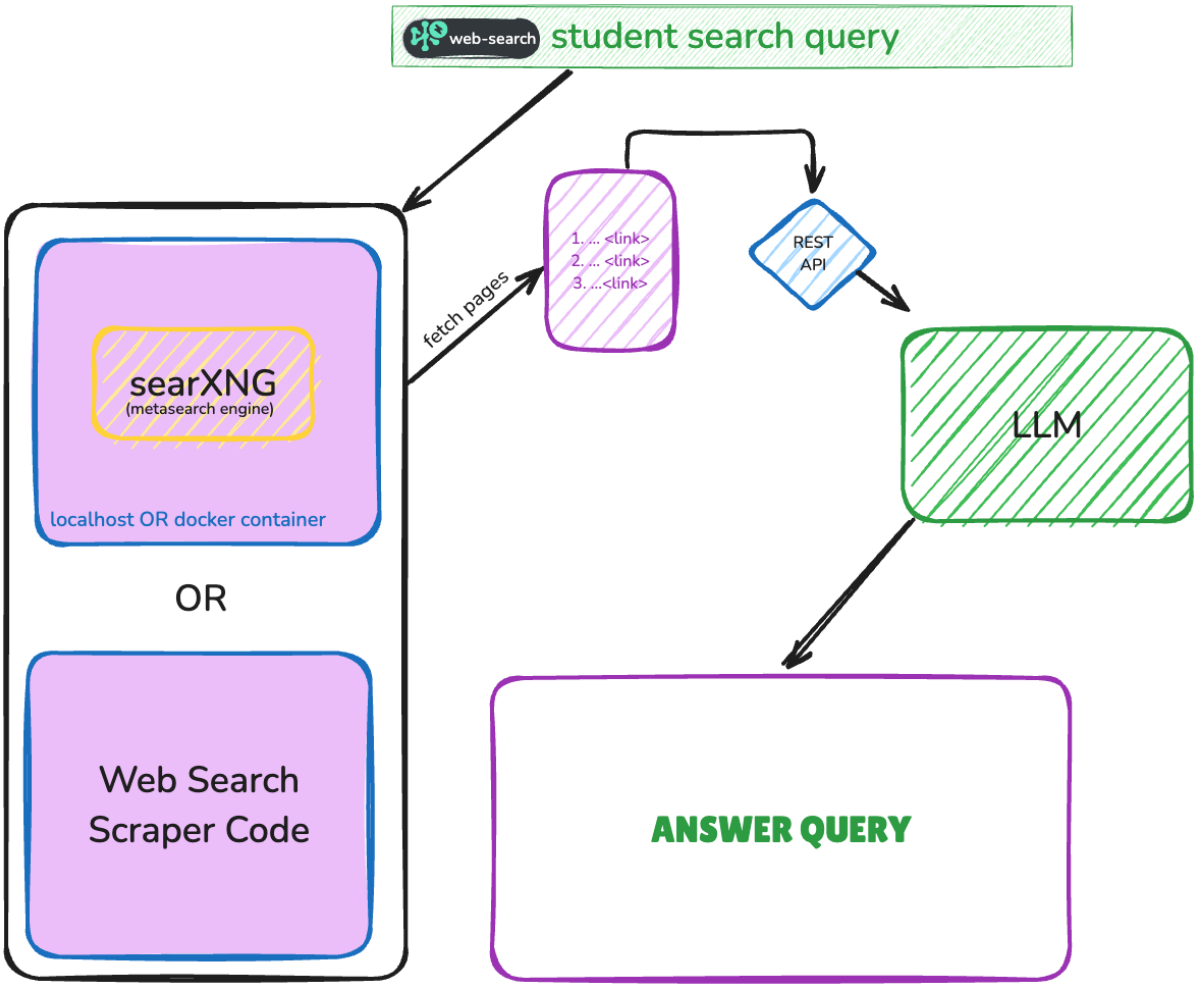


Figure 5: AI powered web search diagram

Non-functional requirements for deployed system

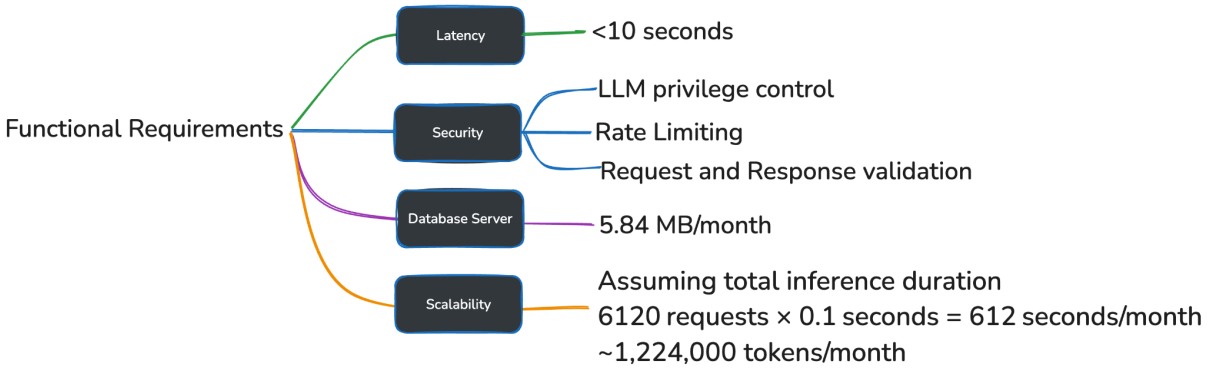


Figure 6: Non-functional requirements

1. Latency: <10 seconds
2. Security:
    a. LLM privilege control
    b. Rate Limiting
    c. Request and Response validation

    See Threat Modelling section for more details.
3. Database server
    - Assumptions:
        - Students usage breakdown:
          No Support Needed: 10%
          3 Prompts per Lab: 80%
          10 Prompts per Lab: 10%
    - Usage metrics
        - Requests: 6120 requests/month
        - Users: ~150 users/month
        - Activity: 3 labs/week, 3.4 AI requests per lab
    - Request Size:
      Assuming that one request with metadata and answer can be around 1000 characters (1 character = 1 byte). The size of one conversation is 1000 bytes which translates into 6 120 000 bytes/month (5.84 MB)
4. Scalability:
    - LLM web server - ensure scalability to handle growing requests
    - Model storage: 9 gb
    - Compute options:
        - **Amazon EC2 GPU** (g4dn.xlarge). **Cost: 0.584 $ per hour** (AWS, 2024)
        - **Amazon SageMaker** (On-Demand Serverless Inference).
            - Assuming total inference duration 6120 requests × 0.1 seconds = 612 seconds/month. Cost 612 seconds/month × 0.00004 $/second = 0.02448 $/month. Data cost: 0.00584 GB×0.016 $/GB=0.00009344 $/month.

(Amazon Web Services, Inc., 2024)

Total cost: $0.025/month

- **IBM watsonx-ai**
  - Pricing: $0.10 per 1 million tokens (Ibm.com, 2025).
  - Assuming that students will generate demand of ~1,224,000 tokens/month the **cost will be $0.12/month**

System Architecture Diagram for the system run locally

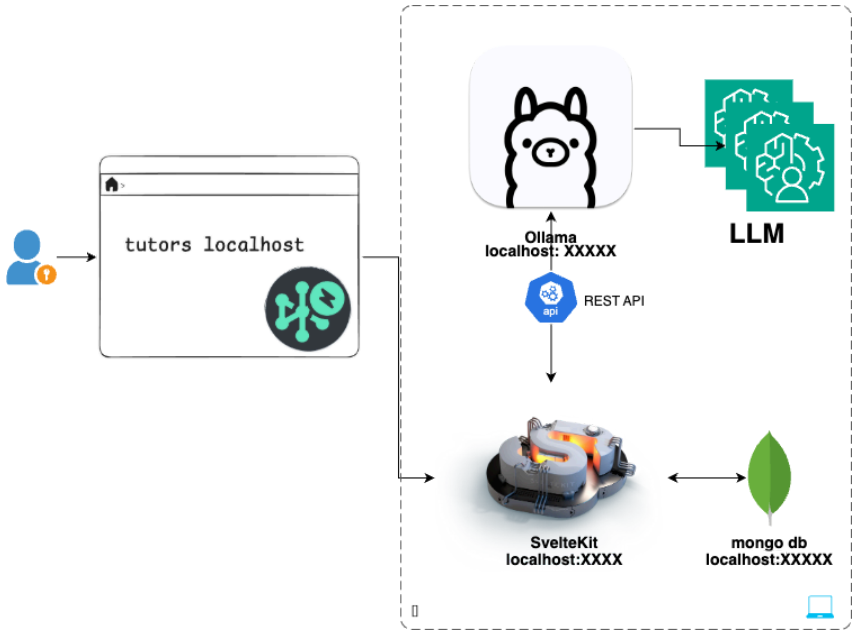


Figure 7: System Architecture Diagram for the system run locally

Options for System Architecture Diagram for Deployed Systems

These are just to show cloud deployment opportunities. One of these diagrams will be used in case I will be deploying the product.

Deployed System using Amazon EC2

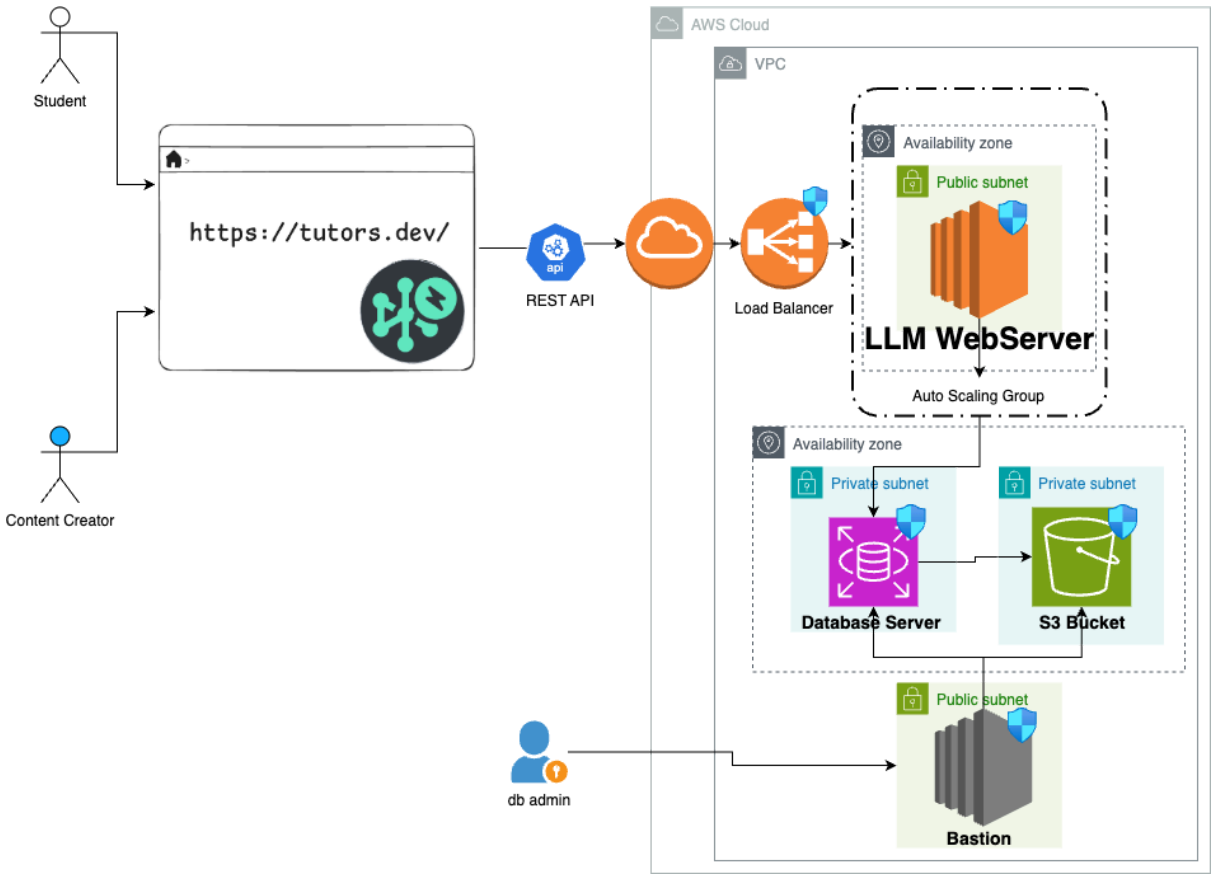


Figure 8: Deployed System using Amazon EC2

Deployed System using Amazon SageMaker

Based on Rizzo (2020), Bhatia (2023) and AI Anytime (2023) the solution can be deployed as per graph below.

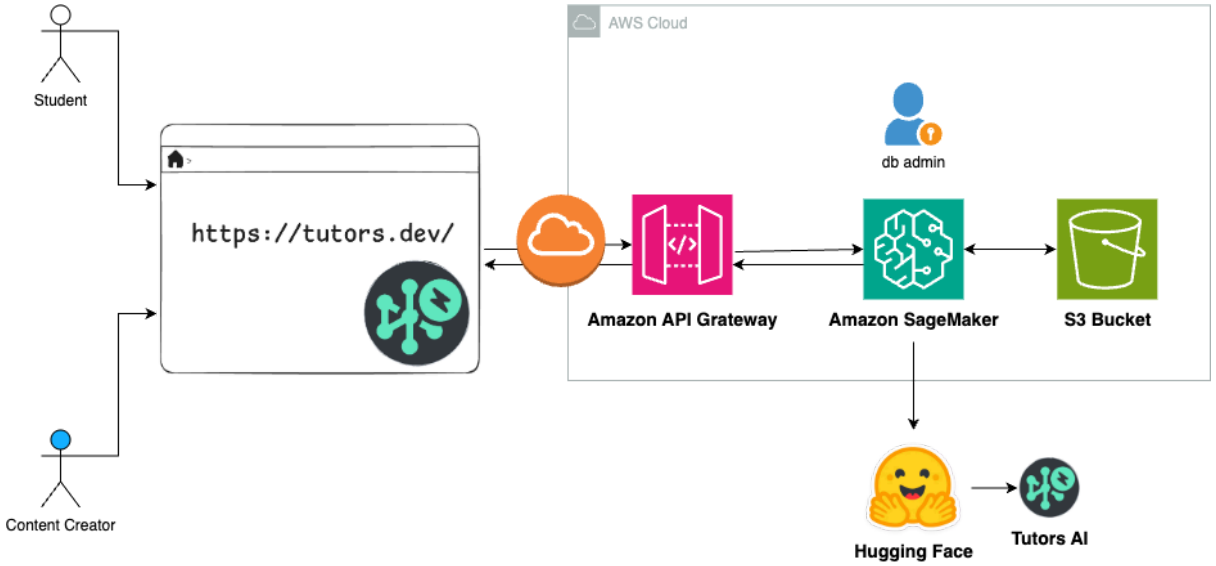


Figure 9: Deployed System using Amazon SageMaker

Deployed System using IBM watsonx-ai

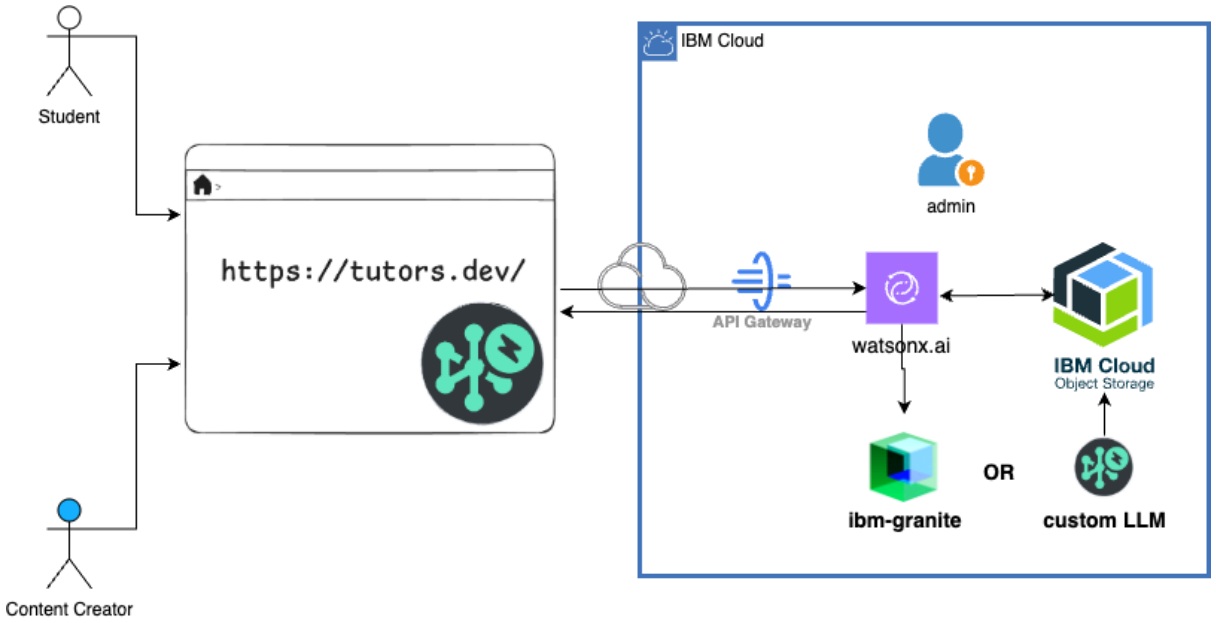


Figure 10: Deployed System using IBM watsonx-ai

Some components of the graph above are based on information provided in IBM Technology (2023) and Sheng Lan (2024) video presentations.

Data Model

System Description

For every Content available on Tutors the system intends to collect and store the following details: Topic Title, Course Title, Content Title, Course URL, Topic URL, Content URL

Every Content can have many Responses generated via hot buttons like “Eli5” or chat with Tutors AI. The following information will be recorded for each GenAI Response: date, Response ID, Content URL, Response, LLM, Helpful

Then for every Content with GenAI Responses there can be an AI Content Refiner. Many Responses can have many AI Content Refinements. Also, Content can have many AI Content Refinements. The following information will be recorded for each AI Content Refinement: date, id, Topic URL, Content URL, Responses IDs, LLM, Generated Text, Helpful.

Search is a standalone table for now that includes: Search ID, date, Search Phrase, Search Result, LLM, Helpful.

Note: user entity is excluded from the Data Model as there is no requirement to collect any User details

Entity Types

Strong Entity Types: Content, Search
Weak Entity Types: Responses, AI Content Refiner

Relationship Types

Content 1..*	generate	0..* GenAI Responses
GenAI Responses 1..*	create	0..* AI Content Refiner
Content 1..*	has	0..* AI Content Refiner
Search		

Entity type attributes:

Content: contentTitle, topicTitle, courseTitle, contentUrl, topicUrl, courseUrl

GenAI Responses: responseDate, responceId, contentUrl, llm, genAiResponse, helpful

AI Content Refiner: dateGenerated, refinedContentId, contentUrl, topicUrl, responsesIds, llm, generatedText, helpful

Search: searchId, date, searchPhrase, searchResult, llm, helpful

Candidate, primary and alternative key attributes

Content: Candidate keys: contentUrl, contentTitle
Primary key: contentUrl

GenAI Responses: Candidate keys: responceId, responseDate and contentUrl, responseDate and llm
Primary key: responceId

AI Content Refiner: Candidate keys: refinedContentId, dateGenerated and contentUrl, dateGenerated and llm
Primary key: refinedContentId

Search: Candidate keys: searchId, date and searchPhrase, date and llm
Primary key: searchId

Enhanced ER diagram:

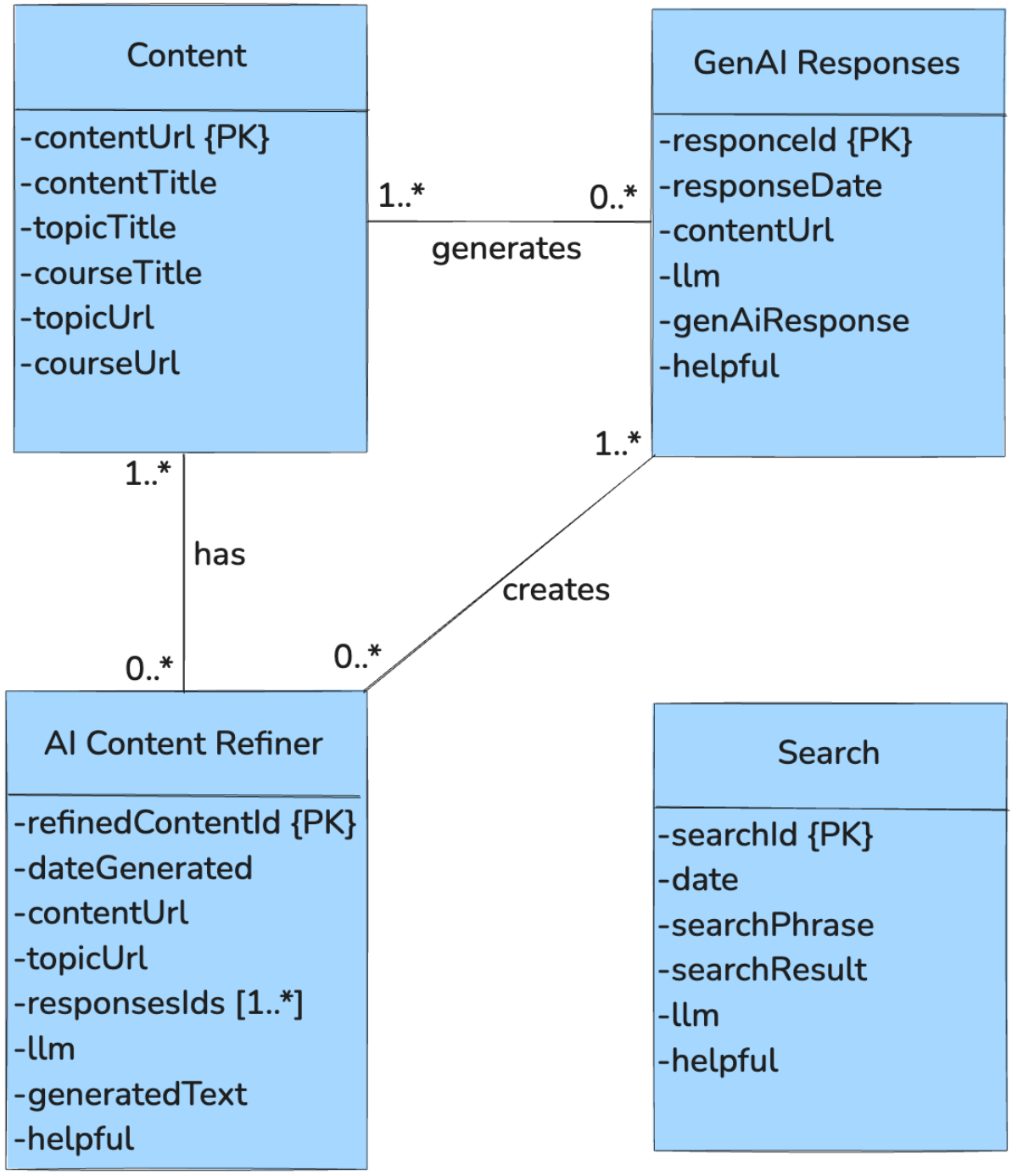


Figure 11: Enhanced ER diagram

Logical Data Model

Content (contentTitle, topicTitle, courseTitle, contentUrl, topicUrl, courseUrl)
Primary key contentUrl

GenAI Responses (responseDate, responseId, contentUrl, llm, genAiResponse, helpful)

Primary key responseId
Foreign key contentUrl references Content (contentUrl)

AI Content Refiner (dateGenerated, refinedContentId, contentUrl, topicUrl, responsesIds, llm, generatedText, helpful)
Primary key refinedContentId
Foreign key contentUrl references Content (contentUrl)
Foreign key contentUrl references GenAI Responses (contentUrl)

Search (searchId, date, searchPhrase, searchResult, llm, helpful)
Primary key searchId

Threat Modelling

Integrating LLM into a system has lots of benefits, however it brings risks as it increases the attack surface and allows the attacker to exploit vulnerabilities caused by GenAI. Hence I will make a simple threat model to mitigate some of the risks. As per Soldera (2024) It is important to not overcomplicate the initial design and add components at later stages.

According to OWASP Top 10 for LLM and Generative AI Security (2024) "Prompt Injection" is number one reason for AI security concerns. It happens when an attacker crafts a prompt that can cause unintended LLM behavior that was not part of the system design. The attack can be done via direct or indirect prompt.

Direct Prompt Injections (or jailbreaking) occur when a bad actor hacks or reveals underlying system prompts. It can give an opportunity for an attacker to lift security functions.

Indirect Prompt Injections occur when an attacker uses external sources such as websites or files. (OWASP Top 10 for LLM & Generative AI Security, 2024)

According to OWASP Top 10 for LLM and Generative AI Security (2024) there is no solution that can guarantee the protection, however we can put restrictions that will make the system more secure:

- Execution Scope:
    - LLM should run on limited permissions
    - Limit execution scope - run LLM in a context of individual
- Untrusted Data Sources
    - Where possible restrict injection of untrusted data sources
- Agents and fully automated systems
    - Avoid usage of agents that can take control over the system

Also, we can add trapping. Trapping is an additional validation of input and output that will enforce policies defined by the developer. This is hard to make the right solution and might impact user experience, eventually making them not want to use the feature. (Hamiel, 2023)

Data Flow Diagram

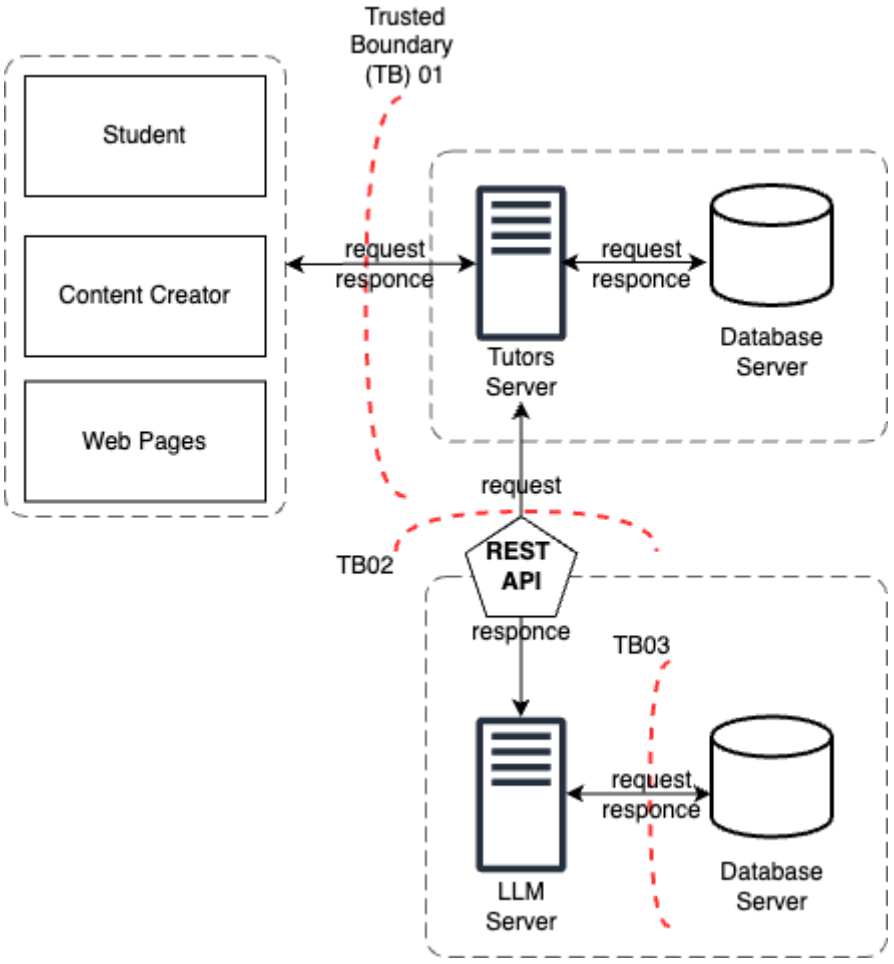


Figure 12: Data flow diagram

This diagram provides a visual representation of how the feature will process data. I will use STRIDE methodology for threat modeling. In the table below I describe potential integrity gaps and security controls for each trusted boundary.

	TB01	TB02	TB03	Security Control
Spoofing	Attacker tries to use another user credentials	Attacker pretends to be Tutors Server		Authentication
Tampering	Attacker modifies LLM parameters (ex. temperature, length, system prompt etc)			Apply request validations/Guardian LLM

Repudiation	Prompt Injections. Content poisoning.			Non-Repudiation - signing with authenticated time stamp
Information Disclosure	Input intentionally or unintentionally sensitive information to the request.	Tutors Server information leak. LLM disclose sensitive information.	Access to sensitive information	Confidentiality: -educate users -encryption -apply auth controls to data sources -do not train LLM on sensitive information
Denial of Service	Resource-consuming queries. Repetitive long inputs.	Resource-consuming queries. Repetitive long inputs.		Replication of data and services; Backup systems; Rate limit API
Elevation of Privilege	Prompt Injections			LLM privilege control

Table 2: STRIDE threat modeling

Use and Misuse Cases Diagram

Use and Misuse Cases Diagram below maps functional use cases, misuse cases and mitigation actions. It ties threats directly to user facing features. I used OWASP Top 10 for LLM and Generative AI Security (2024) to choose some of the possible threats and mitigation actions.

The normal use cases include:

- Student:
    - Open chat with Tutors AI
    - Generate Quick Explanation using pre-engineered LLM prompts
    - Use AI powered Web Search
    - Like/Dislike, Copy LLM response
- Content Creator:
    - Check Analytics (Heatmap, LLM responses which student found helpful)
    - Generate new content based on students' liked LLM responses
    - Like/Dislike, Copy LLM response

Misuse cases:

- Prompt injection:
    - Direct Prompt Injection
    - Indirect Prompt Injection:
        - Initiated by a user
        - Initiated by one of web pages from web search
- Model denial of service
- Content poisoning - this might occur if users on purpose or by accident query content that is not relevant for a topic and mark it as helpful. In this case, if content creators

do not check responses - AI model might generate misleading content suggestion modification

Mitigations:

- LLM privilege control
- Rate limit API
- Response validation - it can be done manually or by using solutions like Guardian LLM explained below.
- Guardian LLM - this would be out of scope for this project hence I mark it as optional implementation if time allows. Guardian LLMs can help to detect prompt injections, hateful, abusive, profane and other toxic content.

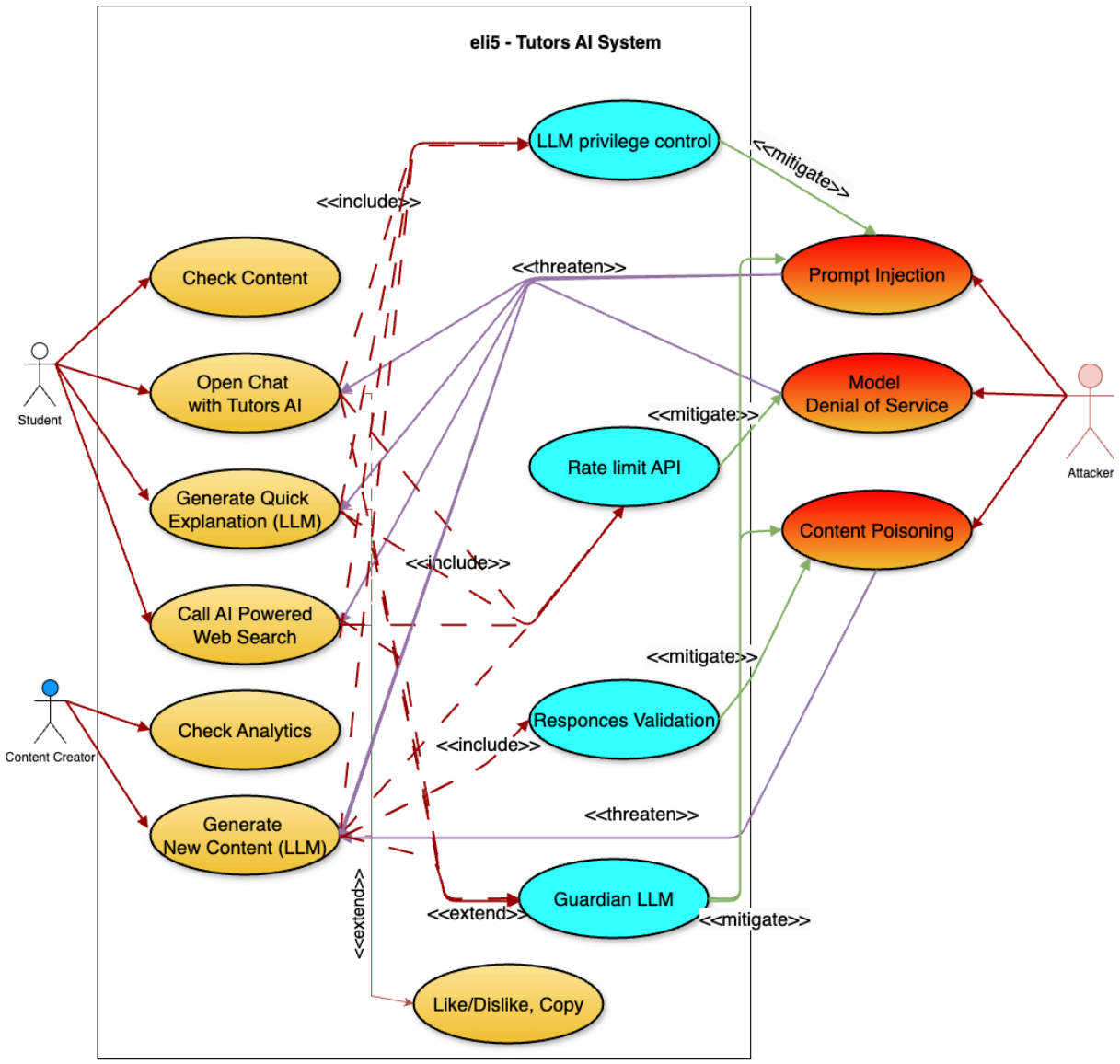


Figure 13: Use and misuse cases diagram

GenAI models analysis

What is open source AI?

With AI development regular users started to hear Open Sources from big tech companies. Some of those companies use the word “Open” as part of their name, some when they talk about their product, others claim that they are Open Source. Open source is synonymous with assurance of integrity, performance and safety, hence it makes any company who claims to be open source to look good. However, many of those claims are Openwashing (“an accusation against some A.I. companies that they are using the “open source” label too loosely”). (Kessler, 2024)
As I will be building a feature that is part of an open source project, I have to ensure that the Large Language Model that I will be using is truly open source AI. I had a look at the Open Source Initiative for the definition which was published at the end of 2024.

“An Open Source AI is an AI system made available under terms and in a way that grant the freedoms to:

- ***Use** the system for any purpose and without having to ask for permission.*
- ***Study** how the system works and inspect its components.*
- ***Modify** the system for any purpose, including to change its output.*
- ***Share** the system for others to use with or without modifications, for any purpose”* (Open Source Initiative, 2024)

Model Selection

I am going to use the AI 6 steps selection framework proposed by IBM Technology (2024).

1. Use case:
    a. Question-answering
    b. Retrieval Augmented Generation
    c. Code related tasks
    d. Long-context tasks (ex. summarization, rephrasing, explanation)
2. Model Options:
    a. granite-3.1-2b-instruct (Apache License 2.0)
    b. mistralai/Ministral-8B-Instruct-2410
    c. meta-llama/Llama-3.1-8B-Instruct
    d. EleutherAI/pythia-6.9b - Pythia (Eleuther AI)
    e. allenai/OLMo-2-1124-7B-Instruct - OLMo (AI2)
    f. LLM360/CrystalChat - Amber and CrystalCoder (LLM360)
    g. google/flan-t5-base - T5 (Google)
    h. tiiuae/Falcon3-7B-Instruct
3. Model size, performance, cost, risk and deployment methods:
    a. Evaluate model card
4. Evaluate model characteristics for the use case:

    a. Accuracy
    b. Reliability (consistency, trustability, explainability)
    c. Speed
5. Run test
6. Choose the option that provides the most value
    a. Deployment: “own/tuned models deploy on prem”

At first, I compared selected models to ensure that model intended use, capabilities and licence matches the project scope. Also, I checked model cards for Bias, Risks, and Limitations. Please see details in the table below.

Model	**Licence**	**December Downloads**	**Intended Use matches the project scope**	**The model contains all required capabilities**	**Bias, Risks, and Limitations**	**Comments**
Granite	Apache 2.0	11,195	Yes	Yes	Some safety considerations are in place. (Huggingface.co, 2024b)	**Passed**. According to Huggingface.co (2024c) “the model has been aligned by keeping safety in consideration, the model may in some cases produce inaccurate, biased, or unsafe responses to user prompts. So we urge the community to use this model with proper safety testing and tuning tailored for their specific tasks”.
Ministral	Mistral AI Research License	NA	NA	NA	NA	**Failed** Use terms due to its licence
Llama	LLAMA 3.1 COMMUNITY LICENSE AGREEMENT	NA	NA	NA	NA	**Failed** Use terms due to its licence
Pythia	Apache 2.0	23,591	No	NA	NA	**Failed.** “The Pythia Suite is not intended for deployment. It is not a in itself a product and cannot be used for human-facing interactions. For example, the model may generate harmful or offensive text. The primary intended use of Pythia is research on

						the behavior, functionality, and limitations of large language models" (Huggingface.co, 2023)
OLMo	Apache 2.0	11,486	Not listed	Not listed	High probability of Biases, Risks, and Limitations	**Failed.** "The OLMo-2 models have limited safety training, but are not deployed automatically with in-the-loop filtering of responses like ChatGPT, so the model can produce problematic outputs (especially when prompted to do so)" (Huggingface.co, 2024) No security considerations mentioned.
CrystalChat	Apache 2.0	2,618	Yes	Yes	Probability of Biases, Risks, and	**Failed.** "CrystalChat has not been aligned to human preferences for safety within the RLHF phase or deployed with in-the-loop filtering of responses like ChatGPT, so the model can produce problematic outputs" (Huggingface.co, 2024b). No security considerations mentioned.
T5	Apache 2.0	609,382	No	Not listed	Probability of Biases, Risks, and Limitations	**Failed.** According to huggingface.co (2024) there is a probability of Biases and Risks. No security considerations mentioned.
Falcon3	TII Falcon-LLM License 2.0	NA	NA	NA	NA	**Failed** Use terms due to its licence

Table 3: LLM comparison

Based on the data, Granite (granite-3.1-2b-instruct) was chosen for the project which is perfect based on its size and performance for the experiment, however at later stages I might upgrade to slightly bigger ibm-granite/granite-3.1-8b-instruct which has better performance especially in long text reasoning as per table below provided by Huggingface.co (2025)

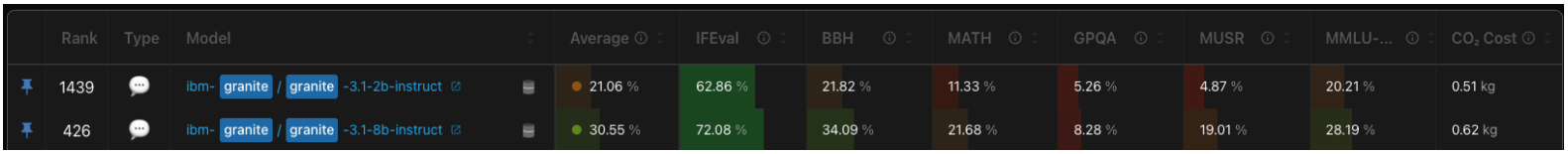


Figure 14: Comparison granite-3.1-2b-instruct vs granite-3.1-8b-instruct (Huggingface.co, 2025)

Model	Average	IFEval	BBH	MATH	GPQA	MUSR	MMLU-Pro	CO_2 Cost
granite-3.1-2b-instruct	21.06%	62.86%	21.82%	11.33%	5.26%	4.87%	20.21%	0.51 kg
granite-3.1-8b-instruct	30.55%	72.08%	34.09%	21.68%	8.28%	19.01%	28.19 %	0.62 kg

Table 4: Comparison granite-3.1-2b-instruct vs granite-3.1-8b-instruct (Huggingface.co, 2025)

Metric	**Definition as per Huggingface.co (2025)**
Average	*Average score across all benchmarks:* *Calculation: Weighted average of normalized scores from all benchmarks* ● *Each benchmark is normalized to a 0-100 scale* ● *All normalised benchmarks are then averaged together*
IFEval	*Instruction-Following Evaluation (IFEval):* *Purpose: Tests model's ability to follow explicit formatting instructions* ● *Instruction following* ● *Formatting* ● *Generation* *Scoring: Accuracy: Was the format asked for strictly respected.*
BBH	*Big Bench Hard (BBH):* *Overview: Collection of challenging for LLM tasks across domains, for example* ● *Language understanding* ● *Mathematical reasoning* ● *Common sense and world knowledge* *Scoring: Accuracy: Was the correct choice selected among the options.*
MATH	*Mathematics Aptitude Test of Heuristics (MATH), level 5:*

	Content: High school level competitions mathematical problems ● *Complex algebra* ● *Geometry problems* ● *Advanced calculus* *Scoring: Exact match: Was the solution generated correct and in the expected format*
GPQA	*Graduate-Level Google-Proof Q&A (GPQA):* *Focus: PhD-level knowledge multiple choice questions in science* ● *Chemistry* ● *Biology* ● *Physics* *Scoring: Accuracy: Was the correct choice selected among the options.*
MUSR	*Multistep Soft Reasoning (MuSR):* *Scope: Reasoning and understanding on/of long texts* ● *Language understanding* ● *Reasoning capabilities* ● *Long context reasoning* *Scoring: Accuracy: Was the correct choice selected among the options.*
MMLU-Pro	*Massive Multitask Language Understanding - Professional (MMLU-Pro):* *Coverage: Expertly reviewed multichoice questions across domains, for example:* ● *Medicine and healthcare* ● *Law and ethics* ● *Engineering* ● *Mathematics* *Scoring: Accuracy: Was the correct choice selected among the options.*
CO_2 Cost	*Carbon Dioxide Emissions:* *What is it?: CO_2 emissions of the model evaluation* ● *Only focuses on model inference for our specific setup* ● *Considers data center location and energy mix* ● *Allows equivalent comparision of models on our use case* *Why it matters: Environmental impact of AI model training* ● *Large models can have significant carbon footprints* ● *Helps make informed choices about model selection*

Table 5: LLM metrics (Huggingface.co, 2025)

Implementation

Chat AI Feature

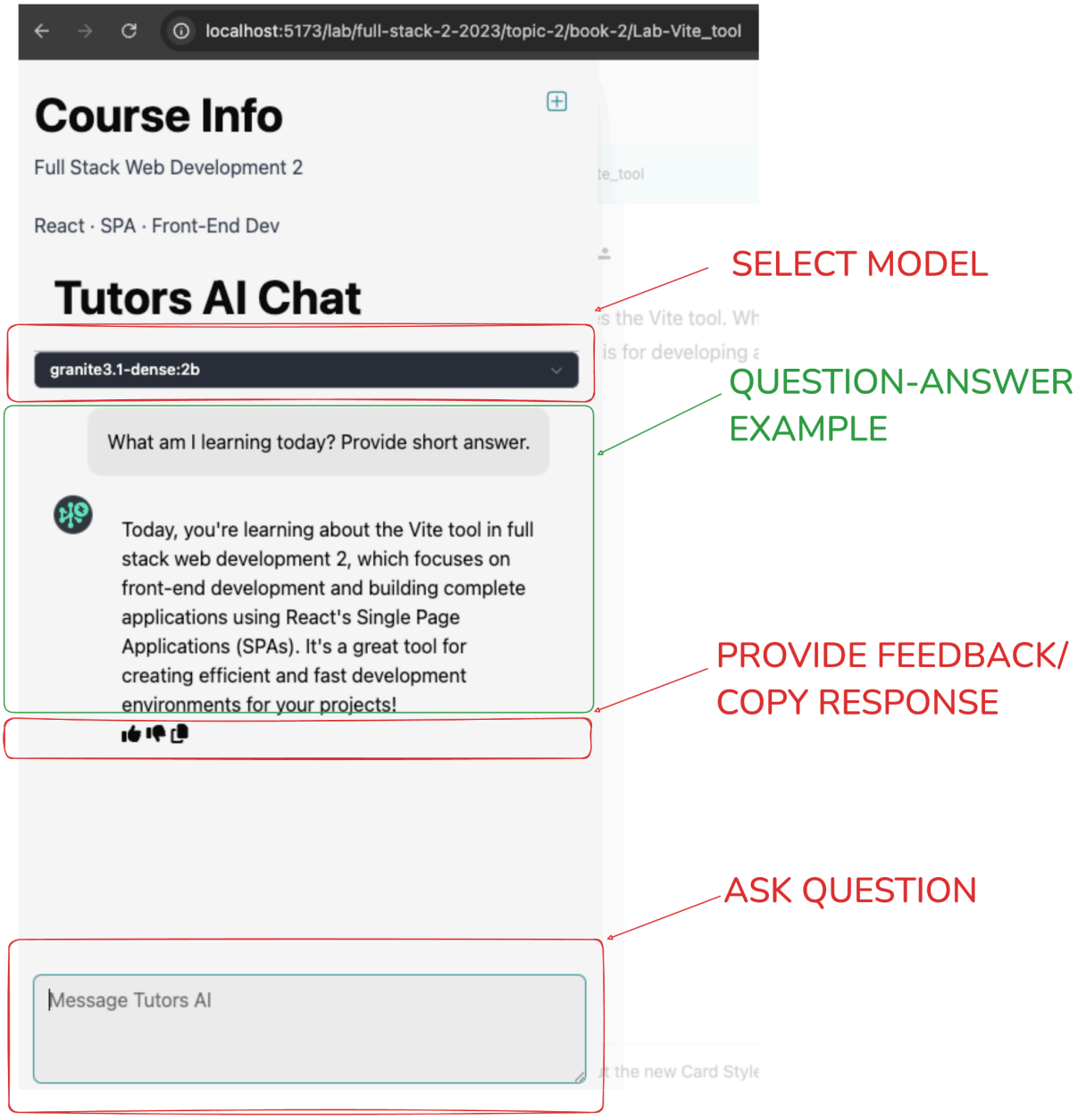


Figure 15: Chat AI feature implemented

The Chat AI feature was implemented to [src/lib/ui/navigators/buttons/InfoButton.svelte]

Currently LLM selection consists of two models:

```
32    const availableModels: string[] = ['granite3.1-dense:2b', 'granite-code:3b'];
33    let selectedModel: string = availableModels[0];
```

Figure 16: LLM array

Quality of replies are improved by passing the topic, its description and page content via system prompt to LLM, so it takes in consideration additional context.

```
35    let systemMessage:Message = {
36      role: 'system',
37      content: `you are assisting Computer Science Higher Diploma students to understand content. \
38       Always explain like they are five years old.\
39       At this stage student explores ${topic}. that student is currently studdies: \
40       Particularly student focused on: ${topicDescription}\
41       The full text of the page student currently explores is ${pageContent}`
42    }
43    let messages: Message[] = [systemMessage];
```

Figure 17: System prompt

Then the system message is passed to the messages array.

When a student asks Tutors AI a question all messages with the system prompt are sent via API call. The model is accessible via localhost using Ollama service.

```
51    async function sendMessage(): Promise<void> {
52      if (!inputMessage.trim()) return;
53
54      const userMessage = inputMessage.trim();
55      messages = [...messages, {role: 'user', content: userMessage}];
56      inputMessage = '';
57      isLoading = true;
58
59      try {
60        const response = await fetch('http://localhost:11434/api/chat', {
61          method: 'POST',
62          headers: {
63            'Content-Type': 'application/json',
64          },
65          body: JSON.stringify({
66            model: selectedModel,
67            messages: messages,
68            stream: false,
69            options: {
70                "temperature": 0.8, //Increasing the temperature will make the model answer more creatively
71                "num_ctx": 8000, //Sets the size of the context window used to generate the next token.
72                "mirostat_eta": 0.9 //Influences how quickly the algorithm responds to feedback from the generated text
73            },
74          }),
75        });
76
77      const data = await response.json();
78
```

Figure 18: Chat request

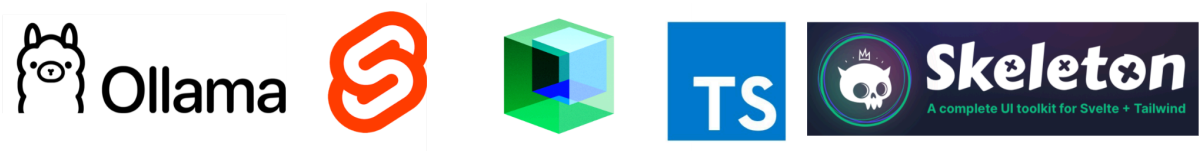
Optional settings include model temperature, token limit and model feedback responsiveness to increase helpfulness of the model responses.

Like and dislike reactions will be used to provide insights to content creators.

The design is clear and simple. It is inspired by https://chatgpt.com/ user interface.

The full code is available at src/lib/ui/navigators/buttons/InfoButton.svelte

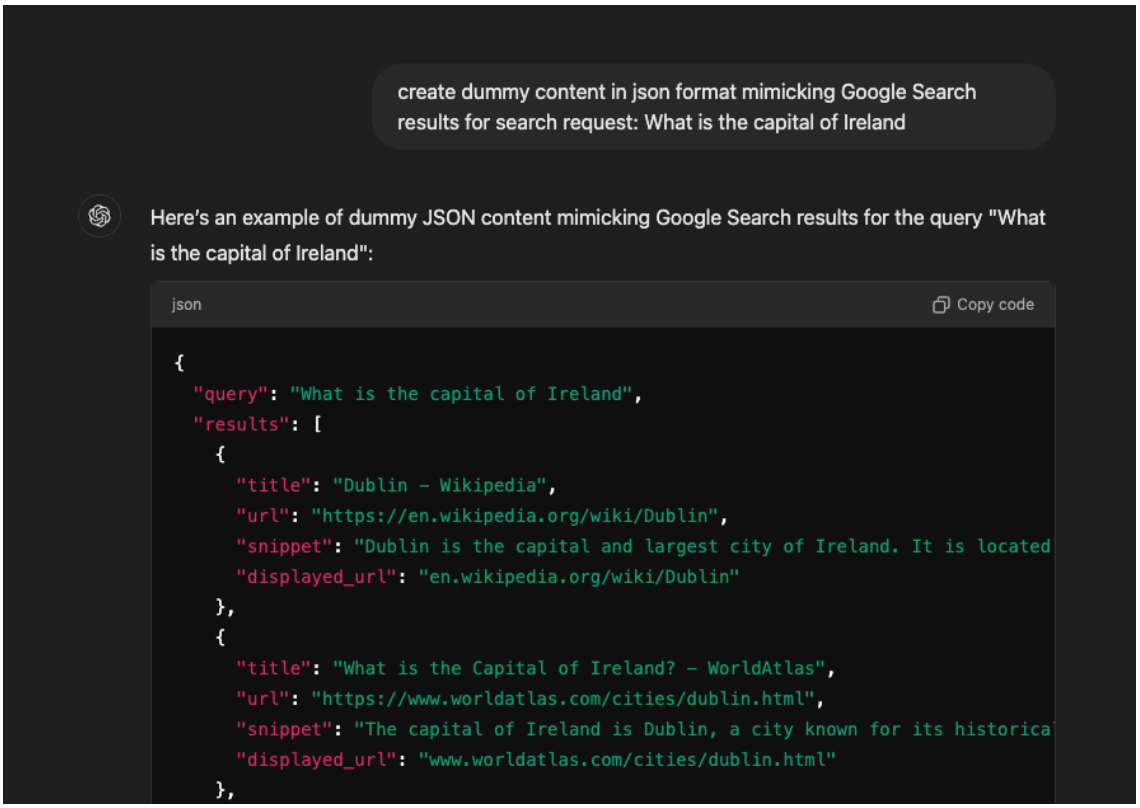
Frameworks and Technologies Used



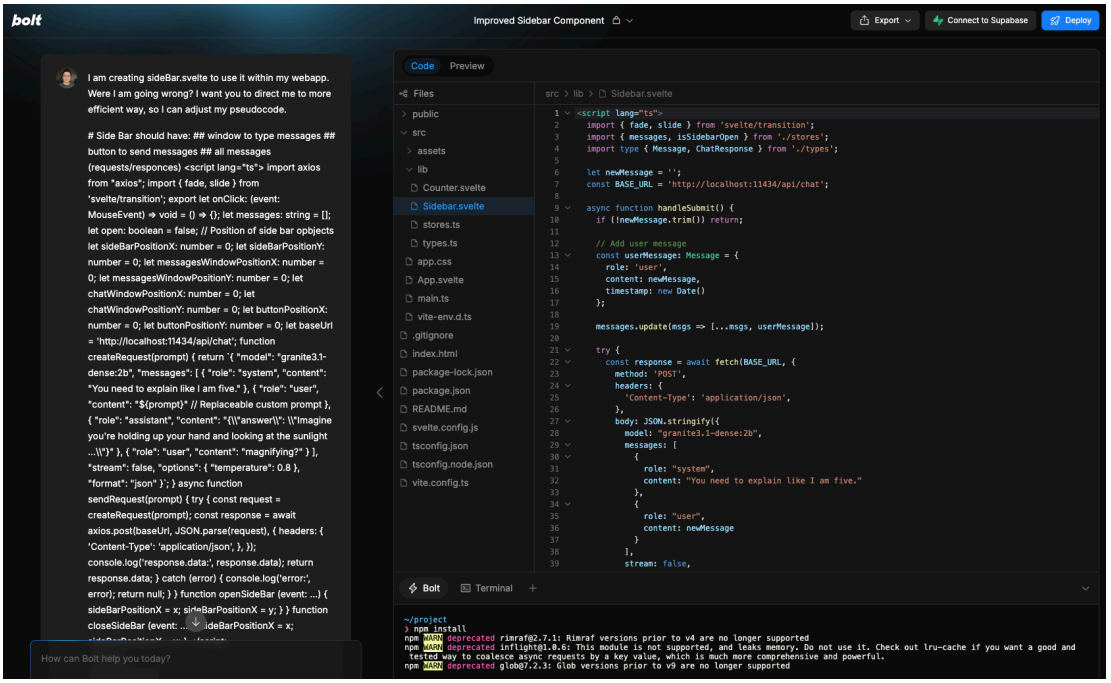
- System Design. For this task I used methodologies shown on GeeksforGeeks (2022), Exponent (2023) and IGotAnOffer: Engineering (2023)
- Threat Modelling. For this task I used frameworks discussed on owasp.org (2024) and Klondike (2023)
- Model Selection. For this task I framework proposed by IBM Technology (2024)
- API: Ollama (MIT License)
- Model Training planning to use InstructLab (Apache License 2.0)
- The /llms.txt file
- SvelteKit
- Tailwind & Skeleton
- TypeScript

GenAI Declaration

1. Used GenAI to generate dummy text for LLM prompts

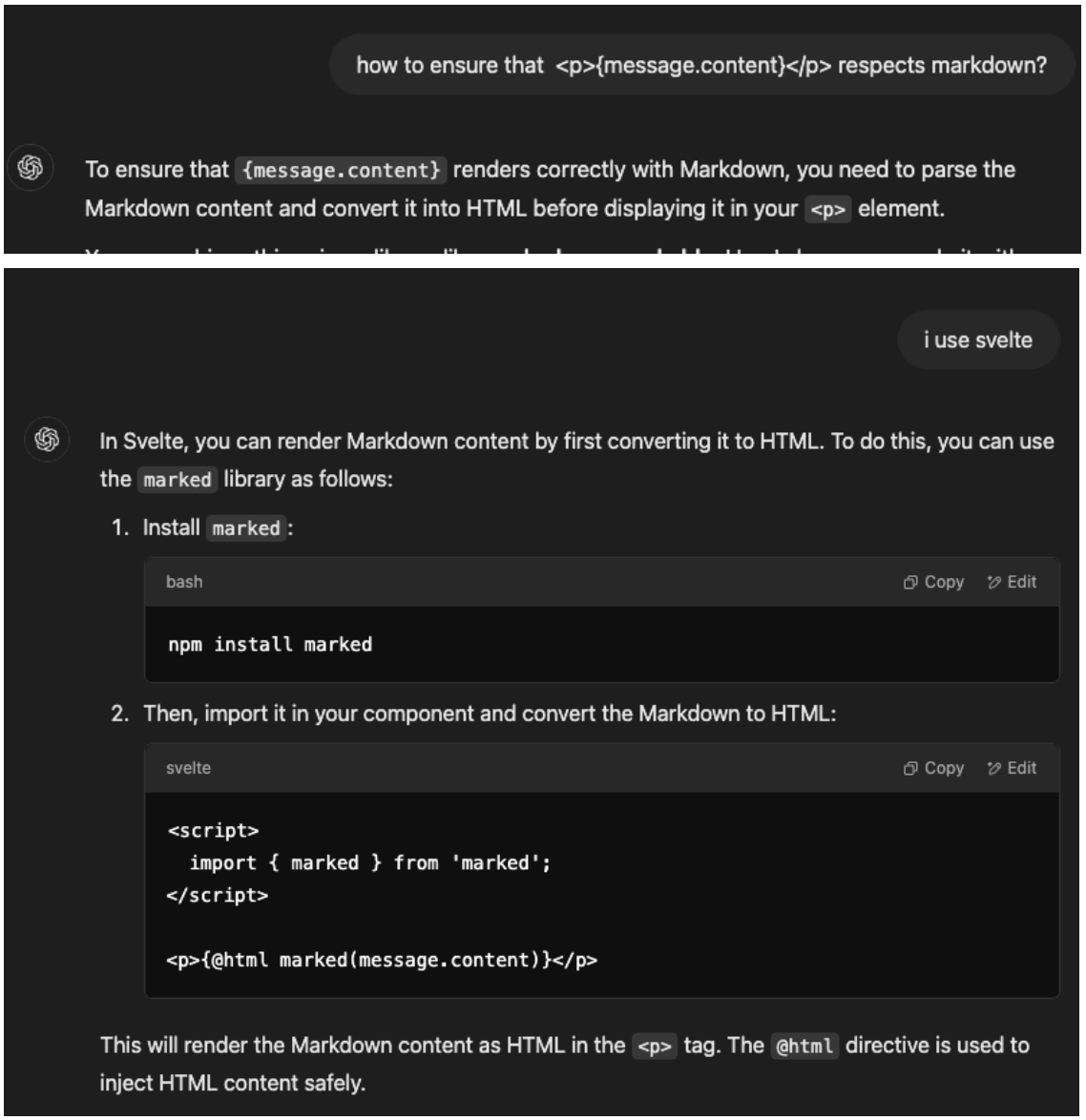


2. Implementation of ChatAI feature
   a. Wrote pseudo code and asked bolt.new for guidance



   b. Looked at similar feature implementation on Skeleton (2025) labs
   c. Used learnings from step 2a and 2b to create code

d. Used ChatGPT for solutions and troubleshootings



3. Used ChatGPT and bolt.new to troubleshoot/create Copy text functionality

Your Svelte `clickToCopy` action looks well-structured, but there are a few things to check for potential issues:

Possible Issues & Fixes:

1. **Ensure Clipboard API is Available**
   Some browsers might restrict clipboard access if your page is not served over HTTPS. Ensure you're testing on a secure environment.
2. **Confirm `document.querySelector(target)` Works**
   - If `target` is passed, make sure it correctly selects an element.
   - If `target` is `undefined`, `node.innerText` should be used.
3. **Wrap `marked(message.content)` in a Container**
   Since `{@html}` is used inside `<p>`, `node.innerText` might not get the expected text. Try wrapping it in a `<span>` and modifying the `clickToCopy` function to copy from it.
4. **Ensure `use:clickToCopy` is Applied Correctly**
   The action should be used on the element that contains the text or should specify a `target`.
   Example:

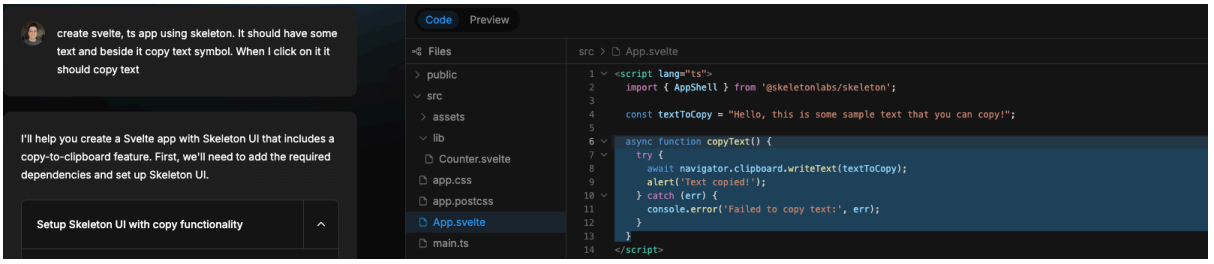
svelte Copy Edit

```svelte
<button use:clickToCopy={{ target: ".copy-target" }}>
    <i class="fa-solid fa-copy"></i>
</button>
<span class="copy-target">{message.content}</span>
```

Suggested Fixes

1. Modify `clickToCopy` to Accept Parameters Correctly

Update your function to properly check `target` and add logging for debugging:



Reference list

AI Anytime (2023). *Deploy LLMs (Large Language Models) on AWS SageMaker using DLC*. [online] YouTube. Available at: https://www.youtube.com/watch?v=A9Pu4xg-Nas [Accessed 30 Dec. 2024].

Ai Austin (2024). *Build a LOCAL AI Web Search Assistant with Ollama*. [online] YouTube. Available at: https://www.youtube.com/watch?v=9KKnNh89AGU [Accessed 7 Jan. 2025].

Aljosha Koecher (2020). *How to use Font Awesome 5 with Svelte/Sappe*. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/62097466/how-to-use-font-awesome-5-with-svelte-sappe [Accessed 9 Feb. 2025].

Amazon Web Services, Inc. (2024). *Machine Learning Service – Amazon SageMaker Pricing – AWS*. [online] Available at: https://aws.amazon.com/sagemaker-ai/pricing/ [Accessed 12 Jan. 2025].

AWS (2024). *EC2 Instance Pricing – Amazon Web Services (AWS)*. [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/ec2/pricing/on-demand/ [Accessed 18 Dec. 2024].

Bhatia, S. (2023). *How to setup LLM model and invoke it using Amazon API Gateway — without using Lambda*. [online] Medium. Available at: https://medium.com/@bhatia.sandeep/how-to-setup-llm-model-and-invoke-it-using-amazon-api-gateway-without-using-lambda-eb1f622f5658 [Accessed 30 Dec. 2024].

Developers Digest (2024). *Deploy ANY Open-Source LLM with Ollama on an AWS EC2 + GPU in 10 Min (Llama-3.1, Gemma-2 etc.)*. [online] YouTube. Available at: https://www.youtube.com/watch?v=SAhUc9ywIiw [Accessed 29 Dec. 2024].

Exponent (2023). *Design ChatGPT - System Design Mock Interview (with eBay EM)*. [online] YouTube. Available at: https://www.youtube.com/watch?v=I9-PUPYZyiw [Accessed 3 Jan. 2025].

GeeksforGeeks (2022). *What is System Design - Learn System Design*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/what-is-system-design-learn-system-design/ [Accessed 2 Jan. 2025].

Hamiel, N. (2023). *Reducing The Impact of Prompt Injection Attacks Through Design*. [online] Kudelski Security Research. Available at: https://research.kudelskisecurity.com/2023/05/25/reducing-the-impact-of-prompt-injection-attacks-through-design/ [Accessed 3 Jan. 2025].

Huggingface.co. (2023). *EleutherAI/pythia-6.9b · Hugging Face*. [online] Available at: https://huggingface.co/EleutherAI/pythia-6.9b [Accessed 12 Jan. 2025].

Huggingface.co. (2024a). *allenai/OLMo-2-1124-7B-Instruct · Hugging Face*. [online] Available at: https://huggingface.co/allenai/OLMo-2-1124-7B-Instruct [Accessed 12 Jan. 2025].

Huggingface.co. (2024b). *CrystalChat*. [online] Available at: https://huggingface.co/LLM360/CrystalChat [Accessed 12 Jan. 2025].

Huggingface.co. (2024c). *ibm-granite/granite-3.1-2b-instruct · Hugging Face*. [online] Available at: https://huggingface.co/ibm-granite/granite-3.1-2b-instruct [Accessed 12 Jan. 2025].

Huggingface.co. (2025). *Open LLM Leaderboard - a Hugging Face Space by open-llm-leaderboard*. [online] Available at: https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard#/?pinned=ibm-granite%2Fgranite-3.1-2b-instruct_float16_8f683a244be9034aeea43dd2a80b7b4fe01d376f_True%2Cibm-granite%2Fgranite-3.1-8b-instruct_float16_f6749f3946b2dc9983b870317a71ddf7a65c0806_True&search=granite [Accessed 13 Jan. 2025].

huggingface.co. (2024). *google/flan-t5-base · Hugging Face*. [online] Available at: https://huggingface.co/google/flan-t5-base [Accessed 12 Jan. 2025].

IBM Technology (2023). *Get Hands-on Experience with Generative AI - watsonx AI Prompt Lab*. [online] YouTube. Available at: https://www.youtube.com/watch?v=swPBNKKPK0E [Accessed 8 Jan. 2025].

IBM Technology (2024). *How to Pick the Right AI Foundation Model*. [online] YouTube. Available at: https://www.youtube.com/watch?v=pePAAGfh-IU [Accessed 28 Nov. 2024].

Ibm.com. (2025). *IBM watsonx.ai | Pricing*. [online] Available at: https://www.ibm.com/products/watsonx-ai/pricing [Accessed 12 Jan. 2025].

IGotAnOffer: Engineering (2023). *Google system design interview: Design Spotify (with ex-Google EM)*. [online] YouTube. Available at: https://www.youtube.com/watch?v=_K-eupuDVEc [Accessed 12 Dec. 2024].

Kessler, S. (2024). Openwashing. *The New York Times*. [online] 17 May. Available at: https://www.nytimes.com/2024/05/17/business/what-is-openwashing-ai.html [Accessed 12 Jan. 2025].

Klondike, G. (2023). *Threat Modeling LLM Applications*. [online] AI Village. Available at: https://aivillage.org/large%20language%20models/threat-modeling-llm/ [Accessed 3 Jan. 2025].

Matt Williams (2024). *Upgrade Your AI Using Web Search - The Ollama Course*. [online] YouTube. Available at: https://www.youtube.com/watch?v=GMlSFIp1na0 [Accessed 7 Jan. 2025].

Open Source Initiative. (2024). *The Open Source AI Definition – 1.0*. [online] Available at: https://opensource.org/ai/open-source-ai-definition [Accessed 8 Dec. 2024].

OWASP Top 10 for LLM & Generative AI Security. (2024). *LLM01: Prompt Injection - OWASP Top 10 for LLM & Generative AI Security*. [online] Available at: https://genai.owasp.org/llmrisk2023-24/llm01-24-prompt-injection/ [Accessed 3 Jan. 2025].

owasp.org. (2024). *Threat Modeling Process | OWASP Foundation*. [online] Available at: https://owasp.org/www-community/Threat_Modeling_Process#data-flow-diagrams [Accessed 3 Jan. 2024].

Rizzo, M. (2020). *Creating a machine learning-powered REST API with Amazon API Gateway mapping templates and Amazon SageMaker | Amazon Web Services*. [online] Amazon Web Services. Available at: https://aws.amazon.com/blogs/machine-learning/creating-a-machine-learning-powered-rest-api-with-amazon-api-gateway-mapping-templates-and-amazon-sagemaker/ [Accessed 29 Dec. 2024].

Sheng Lan (2024). *Create a Powerful RAG Chatbot in 10 Minutes with IBM watsonx | No Coding | IBM Business Partners*. [online] YouTube. Available at: https://www.youtube.com/watch?v=jrA_cOTSi4E [Accessed 7 Jan. 2025].

Skeleton, L. (2025). *Chat - Skeleton*. [online] Skeleton. Available at: https://next.skeleton.dev/docs/guides/cookbook/chat/ [Accessed 3 Feb. 2025].

Soldera, D. (2024). *Threat Modeling for Simplicity*. [online] Threatmodelingconnect.com. Available at: https://www.threatmodelingconnect.com/blog/threat-modeling-for-simplicity [Accessed 9 Feb. 2025].

usehooks-ts (2025). *useCopyToClipboard*. [online] Usehooks-ts.com. Available at: https://usehooks-ts.com/react-hook/use-copy-to-clipboard [Accessed 9 Feb. 2025].

Wikipedia. (2024). *STRIDE model*. [online] Available at: https://en.wikipedia.org/wiki/STRIDE_model [Accessed 9 Jan. 2025].