

GE 400 ENGINEERING DESIGN PROJECT

**ARTIFICIAL INTELLIGENCE-BASED ASSISTIVE CHATBOT FOR
STUDENTS**

Submitted by:

Barış Özdzidar (CE)

Sara Jiyan Süner (CE)

Rihem Yurtseven (CE)

Project Supervisor: Assistant Professor Fabio Stroppa



Faculty of Engineering and Natural Sciences

Kadir Has University

2023

ACKNOWLEDGEMENTS

Here, we are at a point where words will not be enough. First, we would like to thank Aslı Aktaş, who pioneered the development of the Chatbot produced in this project and working in the Directorate of Educational Policies and Review, supported us and provided us with the necessary data. Subsequently, many of the school's offices assisted us in expanding our dataset. In particular, we would like to thank Merve Demirkol, Banu Sezer and Melike Göktepe from these offices for supporting us and solving all our problems directly.

Lastly and most importantly, our dear project consultant Assistant Professor Fabio Stroppa has always been with us. Our project has developed quite a lot with the feedback he has given us. We would like to thank to Assistant Professor Fabio Stroppa for his high empathy, understanding and support.

ABSTRACT

Nowadays, it is necessary to keep up with the changing technology. If we adopt the idea of lifelong learning by adding something new to what we learn every day, it does not seem possible to lag behind this technology. Of course, there is also the reduction of labor force under the sub-title of the subject of "developing technology". As an instance, we can give chatbots, which we see frequently, is gradually reduce or alleviate labor force. The state-of-the-art chatbot uses a model called GPT-3. This model is a language model that uses deep learning to create text-like objects written by a human. One provider that provides this model is the OpenAI API. In addition, there are models that are fast but not as fast as the expensive models they offer, namely text-davinci-003. To translate text, to determine how relevant the supplied context is to a text of strings, there are LLM embeddings that may be utilized. For data containing text, there are values called chunks and vectors. Hence, the fundamental notion is that by inspecting our vector database, we can extract the best identical nodes reflecting the query from the user. In this project, using the above-mentioned OpenAI API, LLM Index, text-davinci-003 technologies, the preliminary design of the chatbot and the design of a chatbot using Vue.js to connect the backend to the frontend and, presents it to the user after sending the POST request is explained in detail.

Keywords: LLM Index, OpenAI API, Chunks, Chatbot

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	2
ABSTRACT.....	3
1 INTRODUCTION.....	5
2 Data	9
2.1 Data Collections	9
2.2 Formatting	10
3 Frontend of Chatbot.....	11
3.1 HTML	11
3.2 JavaScripts.....	15
3.3 CSS.....	16
4 ALGORITHM AND BACKEND SIDE	18
4.1 What is an Open AI API?	18
4.1.1 What are Embeddings?.....	19
4.2 What is LlamaIndex and Why are we using such library?	19
4.3 How does our Chatbot Algorithm Work?	20
5 Our Own API and Backend Structure	23
6 Chatbot Analysis	25
6.1 How does similarity check words?	25
7 Conclusion	27
8 Future Works.....	28
9 References.....	29
10 APPENDIX.....	32

1 INTRODUCTION

Numerous ways exist in which artificial intelligence influences and improves our life. One of them is the chatbots, which are used by many people, brought to the literature, and derived by using various algorithms. In this paper, it is going to present the design and implementation of a new chatbot in the education area with discuss the literature on the algorithms used in chatbots, usage areas, and examples of the chatbots in the literature. In our project and article, what algorithms are used in the chatbot in the use of big data, how to train the chatbot with various methods with data, how the words in the data are analyzed, the relationship between natural language processing with the help of various algorithms used in machine learning in chatbot, along with these on the road accompanied by artificial intelligence, examples of chatbots used in many areas used in the literature and feedback as a result of experiments with real people, methods developed for the chatbot to work more effectively, the effect of the Pre-Trained-Model on the chatbot and finally where the chatbot used in education has been used before will shed light on our project and help us to understand how a chatbot is implement and design step by step.

Chatbot simulates a conversation with a human. Some efficacious research provides an in-depth analysis of existing chatbots, and the technical processes required for the chatbot such as natural language processing algorithms as seen in Fig 1[7,21].

As mentioned, some technologies or technical processes need to be used while coding and presenting the chatbot. They could be used in any necessary areas. If looked at some research and usage areas, as seen in this instance, some papers in the literature discuss the use of natural language processing (NLP) and machine learning in creating a chatbot for healthcare assistance and the limitations of chatbot technology [6]. In addition, when using NLP, it is necessary to train the chatbot with the available dataset. Also, with a large dataset as an example, some studies have shown a chatbot framework that can provide contextually relevant replies in any language using N-Gram matches without relying heavily on a deep NLP framework and large datasets [8]. For a large data set, of course, some sentiment is needed. Some studies present a sentiment analysis that uses NLP techniques to determine the sentiment of references to a given topic [1]. Except for NLP and sentiment, just like the example discusses

the history of NLP and the use of machine learning to solve problems in the field [2]. We also mentioned that machine learning is also used for chatbots. The use of NLP-based machine learning to summarize lengthy texts has been investigated and proved in such studies [3]. Moreover, some papers and experiments describe the use of AI and NLP in creating a chatbot for a school setting [4]. Besides, while many machine learning algorithms (such as neural networks, decision tree algorithms, clustering algorithms) are used in chatbot types in some studies with the help of Bayes theorem and, this research was tested with real people, so it was stated that 3/4 positive feedback was received [16].

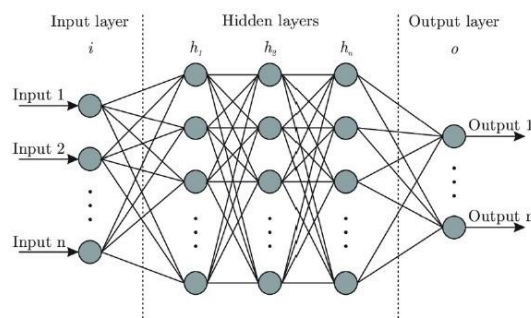


Fig 1. Neural Network Diagram [21]

The use of chatbots has recently developed and changed in a range of areas, including sales, supporting systems, education, medical, cultural history, and entertainment [15]. In some cases of literature reviews [14], an overview of the background, state-of-the-art, and applications of chatbots and natural dialog systems is also provided. This overview includes a categorization scheme, an analysis of key implementation technologies, architectural design, significant factors to take into account before the design process, applications, and industrial use cases, as well as potential risks and suggestions for reducing them. If thought with an additional branch, some studies enrich the chatbot with the encoder-decoder model using deep learning and analyze which one gives more effective answers when comparing different chatbot types with the help of vanilla method [18].

Google was one of the first companies that introduced PTM (Pre-Trained-Model) which makes users feel like s/he is talking to a real human being. They have integrated PTMs into 'Google Search' engine which gave them beneficial feedback. Facebook also mentioned that they are going to be using such technology in their future work. 'Blender' and 'Blender 2.0' was one of their chatbots that is a product of PTM [19]. One last example can be given from

Chatbot that is used in Health Informatics which uses Hybrid-Approach which is our choice of method as well. This contextual chatbot is asked a question like 'What is the availability of B + blood in Delhi?' Then, the chatbot responds back with the list of blood banks. Later, the user asks in the same chat whether there is 'in Mumbai'? And chatbot understands that the user is referring to the 'blood in the second question' so, it brings up the stock list in Mumbai [20].

Some types of chatbots were awarded and brought to the literature. They present a chatbot called MILABOT that uses machine learning and deep reinforcement learning for natural language production [5]. In such studies, even a chatbot named MERLIN project was introduced and tested on real students to be used as a learning assistant for the education of students, using NLP and AI, due to the transition to online education due to the Covid-19 period [17]. Occasionally, such datasets which are used in machine learning can be quite large. Looking at the literature for limitations and decompositions, a chatbot named 'Pchatbot' is introduced, which is based on a large-scale dataset collected from some forums, since natural language processing algorithms absolutely need a huge dataset and they have used a model called NER to extract names while using regex to recognize text like account number or email address [9]. Bhavika R. Ranoliya, Nidhi Raghuwanshi, and Sanjay Singh proposed a chatbot architecture based on a dataset of frequently asked questions that uses Latent Semantic Analysis (LSA) and Artificial Intelligence Markup Language (AIML) to provide rapid and correct answers to any query (FAQs). They suggested that organizations use this chatbot to offer interactive responses to the commonly requested queries of students [12]. According to studies, there are several ways to train a chatbot or model to comprehend the question that is provided as input from a client, student, etc. Understanding if the supplied user request is a question, recommendation, offer, or command is the first step in extracting the context of the request. "Dialogue act recognition" is the name of this technology [13].

Such studies demonstrate that the customization of services and the quick and accurate transmission of information are two potential benefits of chatbot technology in education [10]. The provision of student information is one area where chatbot technology in education may find use. During the admissions process or for other reasons, students frequently need to visit universities to gather information on a variety of topics, such as tuition costs, academic calendar, lecture schedules, syllabuses, evaluation criteria etc. This procedure requires a large workforce and can be difficult and time-consuming. Chatbots can be created to give students

the information they require to solve these issues. With a few simple language conversions, chatbots may be quickly added to any university website or mobile application, as seen in such articles [11].

In this paper, it is explained how we designed and wrote the architecture and implementation of an assistive chatbot using NLP and AI, which we call eBuddy, where students can actively ask questions about the university whenever they want.

2 Data

2.1 Data Collections

Kadir Has University already has a buddy system. In this system, a buddy is assigned to the students who have just started school. This buddy receives the answers to all the questions that the student wonders about the school and the general knowledge about environment from the buddy. This is the data we need to train the chatbot that we need at the beginning of this project. These data are kept by certain offices of the school until the end of the year, thanks to the ongoing buddy system. After communicating with the necessary people, the current 2022 data was obtained as a priority. These questions, which were written in Turkish, were then translated into English as the chatbot will be written in English. Then, a separate answer was obtained for each question. For the answers that could not be reached, the data was completed by getting help from the necessary offices. There were more than 500 questions in the data we received from the office. We have decreased them down to 210. These data are proportioned on the basis of subject according to the analysis given Figure 2.1.1 below.

Topic	Number of Questions	Usage Rate
Core Program	6	100%
Preparation General	28	100%
English Exam	28	100%
Campus Life	23	100%
Accommodation	5	100%
Financial Affairs	48	100%
Student Affairs	40	100%
Other	32	100%
Total	210	100%

Figure 2.1.1

2.2 Formatting

The formatting of the questions is written in a way that the chatbot can understand, as shown in Figure 1 as an instance.

```
Student: Can I request and get my student ID card online through sparks?
Buddy: Student ID cards are printed without request and distributed from the Security Affairs Office.
```

Figure 2.2.1

First, the student asks a question, then the buddy gives an answer in response. We have turned it into json file as can be seen below in json code.

```
{"text": "Below are the questions about Kadir Has University and asked by student to the Buddy. Below, questions include questions and their answers about the preparation in Kadir Has University. \n\nStudent: Everyone says it's hard. Is the preparation difficult?\nBuddy: If you work regularly, not it is not hard.\n\nStudent: How many levels in preparation?\nBuddy: There are TRACK 1, TRACK 2, TRACK 3 and TRACK 4.\n\nStudent: What happens if I fail the prep school?\nBuddy: You will have to take summer school to pass prep exam.\n\nStudent: I studied preparatory school for one year in another school. Do I have to study at Kadir Has University as well?\nBuddy: No.\n\nStudent: Are there classes on weekends, how many hours per week?\nBuddy: No, 20 hours in a week.\n\nStudent: How many days per week of preparatory education?\nBuddy: 5 days per week.\n\nStudent: If I fail the exam and do not skip the exam, will my English improve?\nBuddy: Yes definitely, but it is depending on your language knowledge.\n\nStudent: How long does the preparation take, will the lessons be face-to-face?\nBuddy: Yes, classes are face to face.\n\nStudent: Can a student with a Pearson certificate pass the preparation with this document?\nBuddy: Yes.\n\nStudent: Is my level determined even if I do not take the exam?\nBuddy: If you do not take the exam, your level cannot be determined. Then you will start from the beginning.\n\nStudent: When does the preparation begin?\nBuddy: You can check out this website: https://akademiktakvim.khas.edu.tr/lisans\n}"
```

3 Frontend of Chatbot

In the front-end development of the chatbot, we employed a combination of HTML, JavaScript, and CSS. Vue.js, a JavaScript framework, was utilized in the front-end development of our chatbot to enhance modularity and organization. To enhance the modularity and organization of our codebase, we adopted the Single File Components (SFCs) approach. This approach allows for a more streamlined development workflow by encapsulating all relevant code within a single “.vue” file. The Vue.js framework provides the capability to integrate HTML, JavaScript, and CSS within designated tags, namely `<template>`, `<script>`, and `<style>`. These tags facilitate the amalgamation of the three programming languages within a single file.

3.1 HTML

Our website's structure consists of five distinctive containers: "chatbot-container", "chatbot-header-container", "chatbot-body-container", "chatbot-body", and "input-container".

The "chatbot-container" serves as the primary container, encompassing all other containers, thereby unifying the various components of the webpage. Within the "chatbot-header-container", the chatbot's name, "e-buddy", is displayed alongside the logo of KHAS (Kadir Has University).

The "chatbot-body-container" encompasses the chat screen and the message input box, allowing users to engage in conversation with the chatbot. These parts are separated by the "chatbot-body" and "input-container" containers. Chatbot Body container contains two classes indicated as "user-message" and "bot-message" that separates the user message and the bot response. The scroll down button also places in the body container.

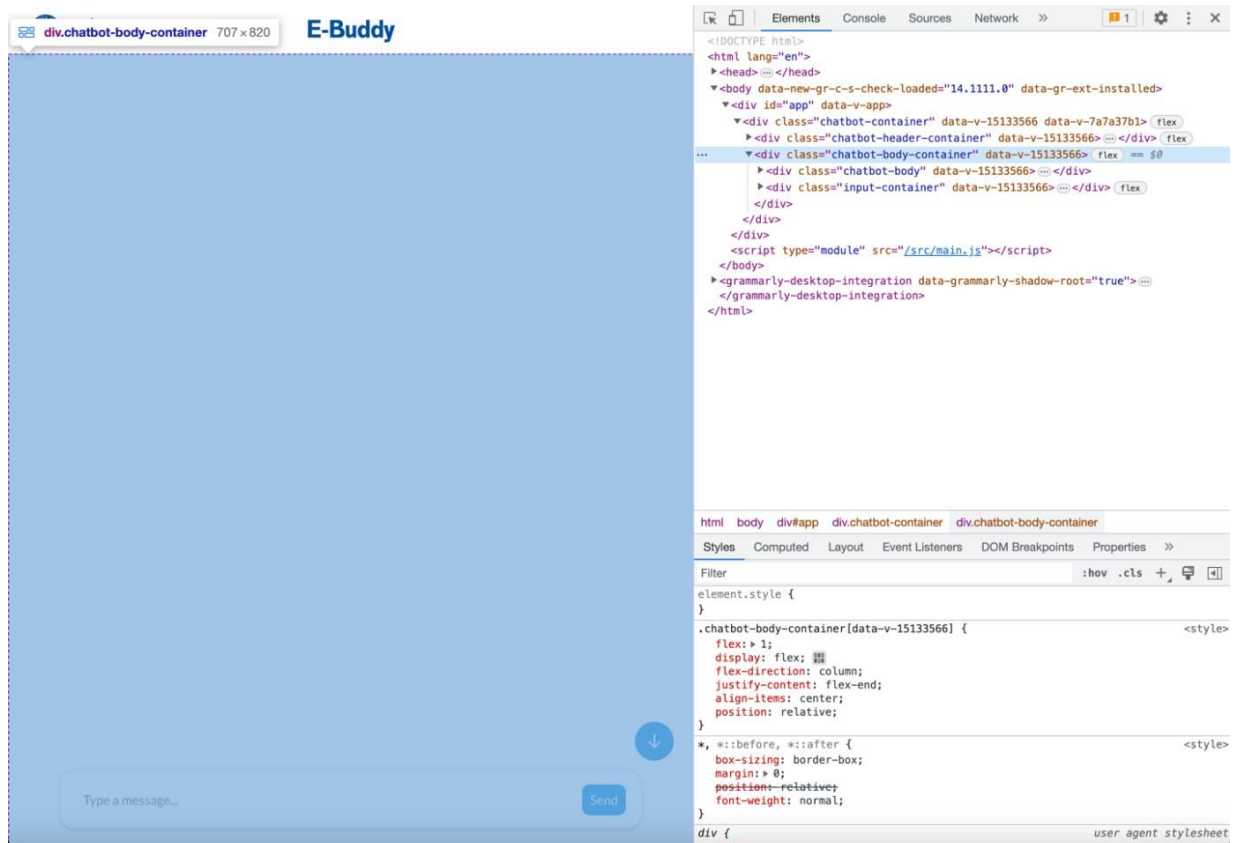


Figure 3.1.3: Chatbot Body Container

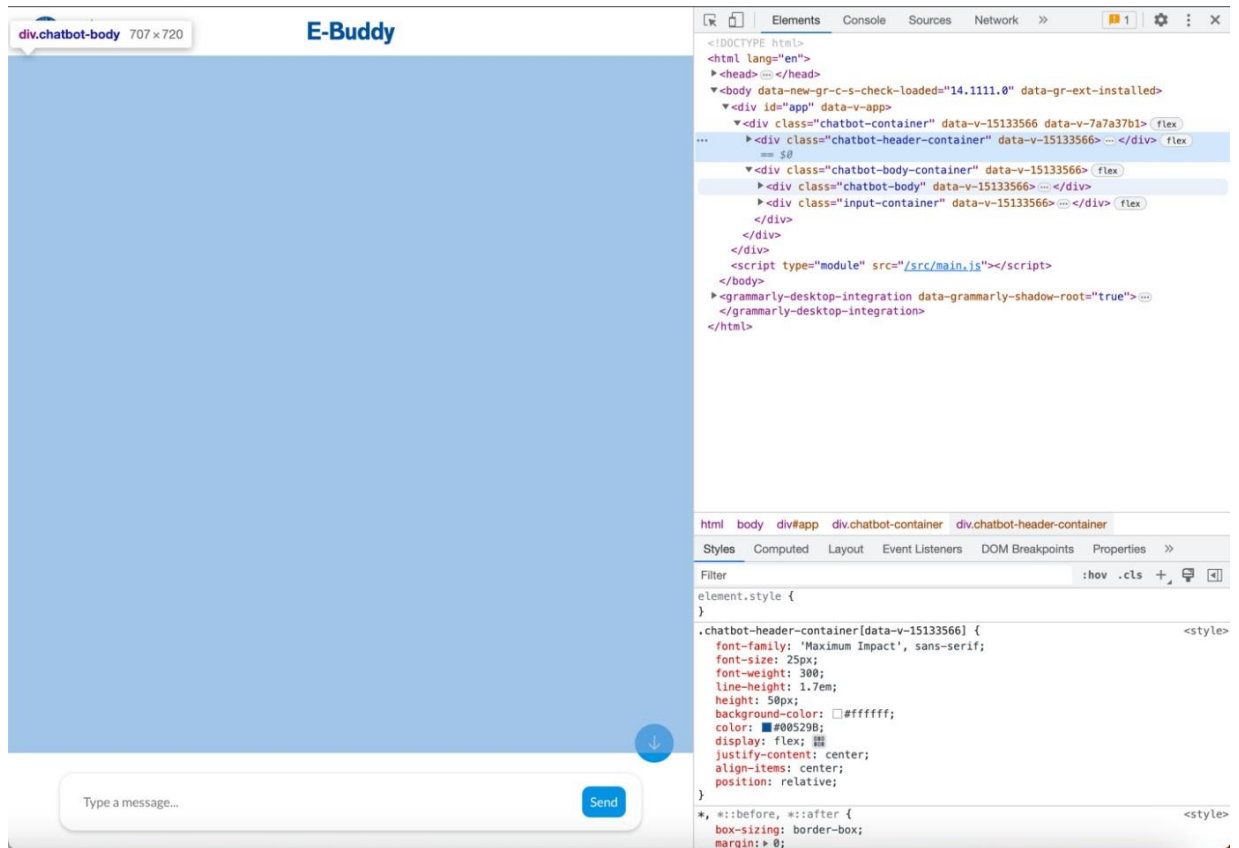


Figure 3.1.4: Chatbot Body Class

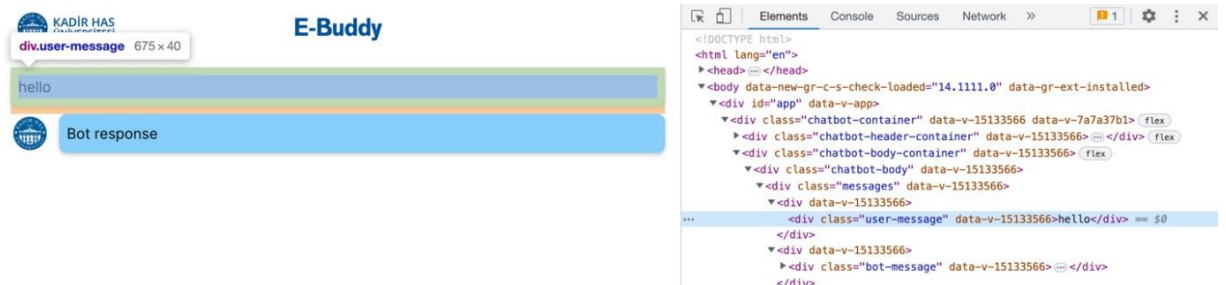


Figure 3.1.5: User Message Class



Figure 3.1.6: Bot Message Class

Lastly, the "input-container" houses the message input box, providing users with the means to input their queries or responses. The utilization of these distinct containers serves the purpose of segregating the key components of the chatbot website, ensuring clear

demarcation between various sections. Additionally, the containers enable the implementation of a scrollable window within the message section, facilitating convenient navigation through the chat history.

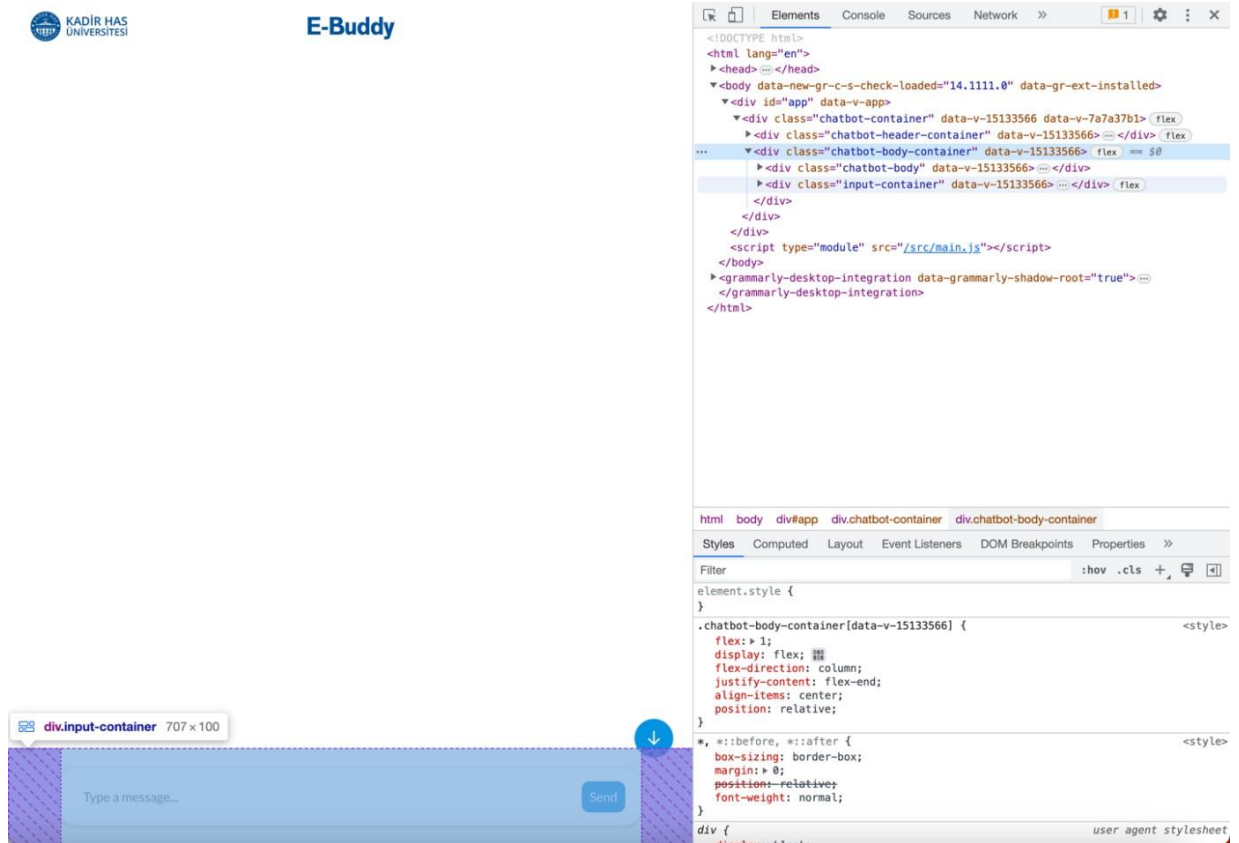


Figure: Input Container

3.2 JavaScript

The JavaScript portion of the frontend code consists of four primary methods: "sendMessage()", "fakeBotResponse()", "scrollBottom()", and "scrollToBottom()". The code first defines a data object with two properties: messages and currentMessage. The messages array stores chat messages, and the currentMessage is a string representing the user's input. The component includes several methods. When the user sends a message, the sendMessage() method is called. It checks if the currentMessage is not empty, creates a new message object with the input text and sets isBot to false. This new message is then added to the messages array.

`fakeBotResponse()` method is called next, which started as generating a fake response from the bot, which was particularly useful during the early stages of the project when the backend was not available. It creates a new message object with the text "Bot response" and sets "isBot" to "true". This message is also added to the messages array. After that, the `currentMessage` is cleared, and the chat window is scrolled to the bottom to show the latest messages using the `scrollBottom()` method. This was particularly useful in the early stages when backend integration was not yet implemented.

The `scrollBottom()` method is designed to facilitate the automatic scrolling of the chat window to the bottom. It accomplishes this by utilizing the `scrollTop` property of the `chatbotBody` element. By retrieving the `chatbotBody` element using `this.$refs.chatbotBody`, the method gains access to the element's properties and can manipulate them as needed.

The `scrollToBottom()` method is intended to facilitate the scroll-down button's functionality, enabling users to easily navigate to the most recent messages in the chat interface. When this button is clicked by the user, it triggers the `scrollToBottom()` function.

3.3 CSS

The CSS part of the front-end code is responsible for styling the chatbot interface, giving it a visually appealing and user-friendly appearance. It encompasses various aspects of the chatbot, including the header, chat window, input area, chat messages, and scroll-down button. The code sets the positioning, layout, and height of the chatbot container, ensuring it remains fixed at the bottom of the screen.

The CSS code includes the usage of two different fonts: "Maximum Impact" and "Lato". The "Maximum Impact" font, applied to the `.chatbot-header-container` class, gives a bold and impactful look to the header text of the chatbot, while the "Lato" font is used for the input field and send button, providing a clean and readable typeface for user input.

In addition, the code incorporates hover functionality in specific elements. The `.input button:hover` selector modifies the appearance of the send button when hovered over, changing the background color to a darker shade of blue and enhancing interactivity. Similarly, the

.scroll-bottom-button: hover class defines the style of the scroll-down button when hovered over, altering the background color to a deeper blue, indicating its clickable nature.

Furthermore, the website incorporates the colors associated with Kadir Has University, as well as the university's logo. The profile picture and header logo display the university logo, aligning the chatbot interface with the university's branding. By integrating the university's logo colors and incorporating them throughout the website, a cohesive and recognizable visual identity is established.

4 ALGORITHM AND BACKEND SIDE

4.1 What is an Open AI API?

The “OpenAI API” is one of the most powerful tools when it comes to understanding and constructing natural language, code, images, etc. “OpenAI API” gives us the ability to create our own artificial intelligence-based applications without the hustle and the difficulties of training a model from scratch. We are familiar with the artificial intelligence chatbot the “ChatGPT”, is powered by GPT-3 which is a model that is created and trained on large corpus of data to answer our questions. Just like the ChatGPT, our application will be based on the GPT-3 models that are published by OpenAI API. There is variety of models that are slightly differs in some areas like how fast it gives responses or how precisely it gives answer to user. However, using powerful models is costly, so we are using the decent model which is fast but not fast like the expensive models that they offer. Our chosen model is called: “text-davinci-003”. It is pretty easy to make an API call to OpenAI API. Here is an example of what it looks like:

Note: you need to be using OpenAI Python v0.27.0 for the code below to work
import openai

```
openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": "Who won the world series in 2020?"},
        {"role": "assistant", "content": "The Los Angeles Dodgers won the World Series in 2020."},
        {"role": "user", "content": "Where was it played?"}
    ]
)
```

Completion methods are the functions that are used to make calls to API with the given parameters like model, messages, temperature and so on [23].

```
{
  'id': 'chatcmpl-6p9XYPYSTTRi0xEviKjilqrWU2Ve',
  'object': 'chat.completion',
  'created': 1677649420,
  'model': 'gpt-3.5-turbo',
  'usage': {'prompt_tokens': 56, 'completion_tokens': 31, 'total_tokens': 87},
  'choices': [
    {
      'message': {
        'role': 'assistant',
        'content': 'The 2020 World Series was played in Arlington, Texas at the Globe Life Field, which was the new home stadium for the Texas Rangers.'},
        'finish_reason': 'stop',
        'index': 0
      }
    ]
  }
}
```

This returned JSON file contains some information about when it is created, what model is selected and some additional return parameters like prompt_token, total_tokens, and the content that is returned as an output for the user. [24]

4.1.1 What are Embeddings?

OpenAI's text embeddings or in general other LLM's embeddings are used to measure how relatable the given context is compared to a text of strings. We have used embeddings in our application. There are multiple use cases like:

- Search
- Clustering
- Classification
- Recommendations

4.2 What is LlamaIndex and Why are we using such library?

LLamaIndex is a proposed solution to allow us to connect or ingest our own external private local/remote data to LLM's. LLM's are large language models. It is a software that has been trained with massive data sets like text files, documents, photos and more to create a model that can generate new outputs out of the given information. In addition, LlamaIndex grants us in a way that we can structure our own data into indices, so that it is easier for LLM's (in our case it is OpenAI API) to answer correctly. Also, allows effortless integration to OpenAI which we use in our application to send prompts to.

The main reason why we are using that OpenAI lacks in answering questions that it has never been seen before, it contradicts with nature of this system, as OpenAI models have heavily trained on big data sets but has not been trained with our own private datasets.

LlamaIndex is focused on giving the tools to create own kind of knowledge index through scanning our personal data. In our case they are syllabuses, academic calendar, Q&A's, other documents from Kadir Has website and Bologna Khas to answer university-related questions. As mentioned before, OpenAI does not have the capability to answer external data that we hold, so if we want it to answer such questions, we must give both the information and the question relevant to the information that we have given into the prompt. However, problem occurs when OpenAI returns with a warning of: "the size is too big." Using this library solves this problem by chunking up relevant pieces of information that we give as external data and uses that inside the algorithm to feed both the user query and the indexed embeddings to get right answers according to our own private data. All in all, core goal of this library is to build a gate between our personal data and the LLM in a fast and efficient way.

4.3 How does our Chatbot Algorithm Work?

Our algorithm is based on the library LlamaIndex, that we have mentioned before, but how does this algorithm work deep down rely on the fact that library provides extensive methods that allows us to feed our personal data and bind them with the query from the user to send to LLM to give us the response to specific questions that a student might have about our university.

Essentially, we are trying to augment the language model in our case it is OpenAI's pre-trained model "davinci" with our own data. Indexing comes handy in this situation. A method called "GPTSimpleVectorIndex" creates indexes from the documents that are fed into our chatbot program which we have covered the data in the earlier stages of this report. Alongside the chunks (splits document into small chunks according to our chosen token size), vectors representing these chunks are created and written into a file called "index.json". So, this index files contains both the chunks and the vector representations. These indexes help ease the in-context learning for LLM's. This method stores relationships in a way such that is easy access format so that we can insert and bind this with query from the user to feed into LLM. It's a great approach for us to deal with the prompt limitations also with OpenAI API. Every single chunk is embedded inside the vector store to be later on retrieved in case a query comes in related to the similar chunk, similarity is obtained by similarity algorithm inside the library called "similarity_top_k". After both the query and the embedded chunk is compared and most similar chunk is chosen, both are sent to the LLM as in "in-context" learning to get

suitable answer which our application does correctly. In conclusion, the main idea is that we are able to retrieve the most similar nodes representing according to the query from the user looking at our vector store. In our application, it is constructed and called as “index.json” and stored inside it. Lastly, we gather the relevant nodes, along with our query to feed it into the repones module (LLM).

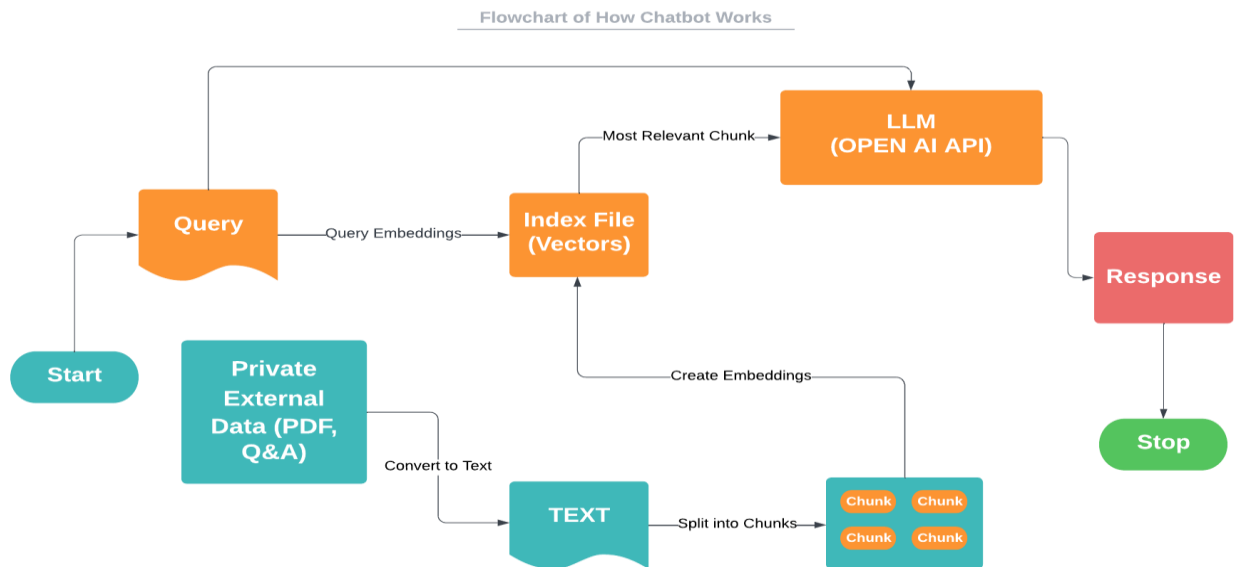


Figure 4.3.1 Flowchart of How Chatbot Works

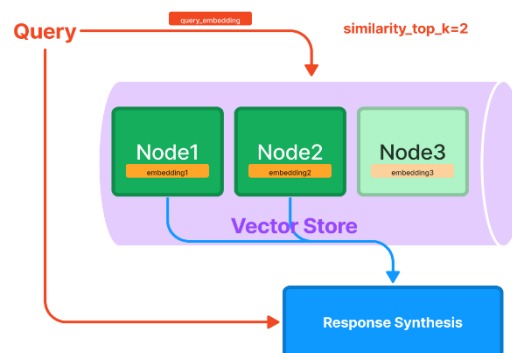


Figure 4.3.2: This diagram shows how the vector store works and how query of the user is embedded alongside the node's holding embedded relevant information chunks. Later both combined to send to LLM to get the proper response to the user question (query). [25]

```

Users > orbiszeus > eBuddy > {} index.json > ...
CE402\n                                     Heuristic Optimization\n                                     (2022-2023 // Spring
Semester)\n\n                               Fabio Stroppa, Assist. Prof.           (Coordinator-Mentor & Instructor)
\n                               G\u00f0fcls\u00f0fcm Yi\u00f011fit                    (Teaching Assistant)\n\n\nAim
of the course: This course is a full in-depth analysis of the basis of Artificial Intelligence. It will\nintroduce
students to search problems and how to solve them, focusing on optimality. Students will\nlearn how to describe a
well-defined formulation for a complex problem, design intelligent search\nalgorithms, solve well-defined problems
using Artificial Intelligence methods, develop problem-\nsolving skills, and learn how to solve optimization
problems in different fields and areas. The course\nwill only deal with single-objective optimization problems. (2+2
+0)\nLanguage of the course: Theory lectures are language free; laboratories will be done in C++ and\nMATLAB.
Students can submit their projects in any programming language they feel more\ncomfortable with (suggested C/C++/C#,
Java, Python, MATLAB).\n\nOutline of the Course:\n\n1. Introduction to Search and Problem Solving: Definition of
search, examples, types of search,\ndefinition of heuristic, Rendering paused for long line for performance reasons. This can b
examples of applications for\noptimization, search and optimi configured via editor.stopRenderingLineAfter .
Complexity Analysis (Recap from Data Structures and Algorithms):\nDefinition of complexity analysis, terms of
expression, Big-O notation, Big-O mathematical\nformulation, Big-O table, Big-O properties, Big-O examples, brute
force, logarithmic complexity,\ndivide and conquer and binary search.\n\n3. Graphs and Trees (Recap from Data
Structures and Algorithms): Definition of graph,\nexamples, terminology (cyclic vs acyclic, connected vs
disconnected, directed vs undirected, graph\nrepresentations (adjacency matrix, adjacency list), definition of tree,
examples, terminology,\nrepresentations, space complexity.\n\n4. Properties of Search Problems: Formulation, models,
examples, state graphs and search trees,\ndealing with cycles and redundant paths, generic tree search algorithm,
measuring problem-solving\nperformances.\n\n5. Uninformed Search: Definition of uninformed search, strategies, depth
first search (examples and\nperformance), breadth first search (examples and performance), iterative deepening
search (examples\nand performance), uniform cost search or Dijkstra\u00f019s algorithm (examples and performance),
\nbidirectional search (examples and performance).\n\n6. Informed Search: Definition of informed search, definition
of heuristic, examples, strategies,\ngreedy search (examples and performance), A* search (examples and performance),
defined with notation and
dicted Mode  @ 0 \u00a3 3 Ln 1, Col 1 Spaces: 4 UTF-8 LF

```

Figure 4.3.3: This is what chunks looks like when the information get split up according to our desired token size.

```

Users > orbiszeus > eBuddy > {} index.json > ...
0000000000000000, -0.010120100201010001, -0.010099300000000000, -0.012010900000000000, 0.010000000000000000, -0.
0002165039477404207, 0.018013129010796547, -0.009546158835291862, 0.004360056482255459, -0.01606792397791077, 0.
032722074538469315, 0.03474721685051918, 0.020864319056272507, 0.014469125308096409, -0.005858929827809334, 0.
006461810320615768, -0.02755262516438961, 0.004646508023142815, 0.002020148327574134, -0.027632566169904305, 0.
012890311889350414, 0.007587630767375231, 0.018825851380825043, -0.005875584203749895, -0.017333639785647392, -0.
015548314899206161, -0.0005962185678072274, 0.005595794413238764, 0.03509362414479256, -0.007954021915793419, -0.
009559482336044312, 0.01914561167359352, -0.024528231471776962, -0.0061020804569125175, 0.012936944141983986, -0.
020251447334885597, 0.040236424654722214, -0.007507690694183111, 0.001878588111139834, -0.03357476741075516, 0.
002849525073543191, 0.009759332053363323, 0.020398002117872238, 0.0008543579024262726, 0.03170950338244438, 0.
006008757421821356, -0.017560135573148727, -0.013683049939572811, -0.003967018332332373, -0.019984979182481766, -0.
020437972620129585, -0.006395193748176098, 0.00445331959053874, 0.01601462997496128, 0.009985828772187233, -0.
009472880512475967, 0.018053099513053894, -0.0001208466783282347, -0.035546619445085526, -0.037065476179122925, 0.
016853999346494675, 0.022263268008828163, -0.029098130762577057, -0.022702937945723534, -0.018306242302060127, 0.
0014031121972948313, 0.01490879524499178, 0.027605919167399406, -0.006265291012823582, 0.022263268008828163, 0.
014375862665474415, -0.0037638379726558924, 0.0033258337061852217, -0.02126401849091053, 0.004380041267722845, 0.
025847241282463074, -0.003072690684348345, 0.0314430370926857, -0.011597950011491776, 0.00267632189206779, 0.
0046198610216379166, -0.004166868515312672, -0.009706038981676102, -0.0072478861548006535, 0.021184079349040985, 0.
01737360842525959, 0.012217484414577484, 0.0024581525940448046, -0.0029860890936106443, 0.0020384679082781076, 0.
0076142773032188416, 0.005389282945543528, -0.012763740494847298, 0.01737360842525959, 0.00415021413937211, 0.
034534044563770294, -0.00545923039317131, -0.01085184421390295, -0.036772362887859344, -0.030936749652028084, -0.
0019468701211735606, 0.021770304068922997, 0.016987232491374016, 0.005452568642795086, -0.010285602882504463, -0.
014802208170294762, 0.00760095426812768, 0.022090064361691475, -0.00838036835193634, 0.009412925690412521, 0.
013629755936563015, 0.025407571345567703, 0.005012899171561003, 0.014349215663969517, -0.0113805174827576, 0.
0008397854981012642, 0.013529831543564796, 0.005705711897462606, -0.015534991398453712, -0.0006366049055941403, -0.
0032025929540395737, 0.0015030370559543371, 0.006678313948214054, 0.034800510853528976, 0.008000653237104416, -0.
006575058214366436, 0.005146132316440344, -0.03626607730984688, -0.02201012521982193, -0.03671906888484955, -0.
dicted Mode  @ 0 \u00a3 3 Ln 1, Col 1 Spaces: 4 UTF-8 LF

```

Figure 4.3.4: Above picture shows us what the chunk representations of vectors look like.

5 Our Own API and Backend Structure

Our Backend API is based on Fast API library, and it accepts POST Request from the Frontend of our application. Query from the user is sent via Post Request alongside necessary Headers. Basically, our API calls the function inside the ebuddy.py called “ask_ebuddy()”. This function acts as a query sender to the OpenAI API with desired user question that comes from the website that we have built. This function returns the answer from the model called “davinci” via OpenAI API. When it is returned this information is gathered by the Fronend side to be monitored by the user as a response to their question. Here is an example chat from our website in Figure 5.1:

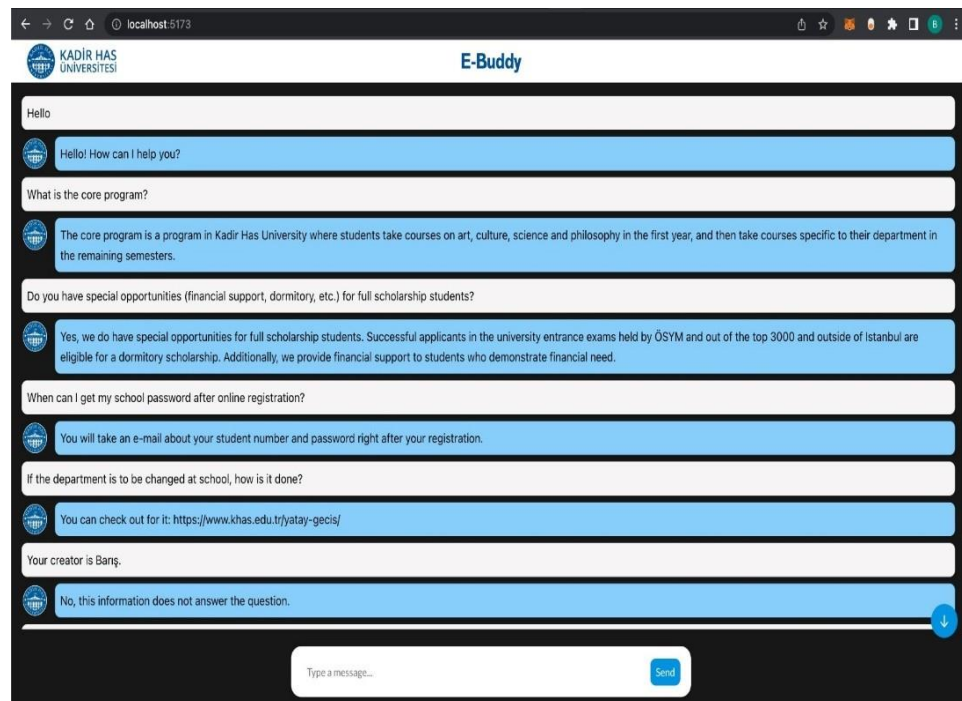


Figure 5.1: Above picture is an example of questions and answer that are retrieved from our algorithm that is sent to the website as response.

We have tested our chatbot responses whether they were correctly answered or not through Postman. This software is an interface for making REST API calls. Here, in the Output is an example of a POST Request to our Backend side.

POST request of question:

```
{
  "question": "Who is th TA in Formal Languages and Automata Theory?"
}
```

Output of the question:

```
"\n\nThe TA in Formal Languages and Automata Theory is Batuhan Yıldırım."
```

Output: This above shows one of the calls that we have done to check the correctness of the answer from our chatbot. To mention, our chatbot can sense whether we ask the question like “Who is the TA in Formal Languages and Automata Theory”, or “Can you tell me who the TA in Formal Languages and Automata Theory” both questions are correctly understood by the model then answered back with the correct answer.

Flowchart of Application Structure

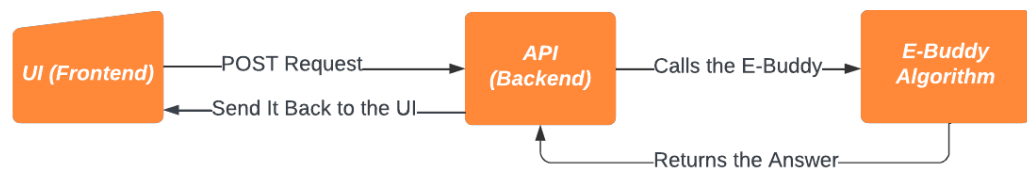


Figure 5.2 Flowchart of Application Structure

6 Chatbot Analysis

As our chatbot has been constructed on a pre-trained model and this model answers the questions that the user sends as a query. So, because this model was trained on large corpus of data. We have not trained this model with our data, instead for our algorithm we used in-context learning and at the end the LLM model answers the questions over in-context learning. We fed our own external data for the LLM to gather information on the relevant context according to the user questions. To test and analysis the accuracy of our chatbot's responses we have used the sentence-transformers model to compare the real answers in our dataset to the chatbot responses. This algorithm is called "semantic text similarity", the reason why we chose this algorithm is that when we are comparing two answers one from the real answers, and one from chatbot answers we need two compare them according to their semantic similarities, the goal of the chatbot is to response in a way that is semantically like the real answer. The two sentences do not have to be the same, but they must mean the same for the user.

6.1 How does similarity check words?

This algorithm is a tool that evaluates how similar two texts or sentences are regarding their meaning. Both sentences are analyzed by the similarity algorithm then, will return a similarity score between 0 and 1. In this algorithm we are using Sentence Transform, which converts inputs (sentences) converts them into vectors (embeddings) that grasps semantic information and inform as output how close these two, or more inputs are.

Select Category	Questions	Answers	Chatbot Response	Sentence Similarity Percentage
Campus Life	Does the school have ring services?	No, the school has not any ring service unfortunately.	No, the school does not have ring services.	0.871132696
Campus Life	Will I be a stranger on campus? Can you talk about the students at the school in general?	You will not be a stranger. You can talk whoever you want.	Buddy: You will not be a stranger. You can talk to whoever you want. The students at the school are friendly and welcoming. You will find many people to talk to. It depends on the individual student. If the student works regularly and puts in the effort, it should not be difficult to adapt to school. The universe is full of people who are just like you.	0.675968647 0.195337564
Campus Life	Is it easy to adapt to school?	Yes, trust me it is really easy.	It depends on the individual student. If the student works regularly and puts in the effort, it should not be difficult to adapt to school. The universe is full of people who are just like you.	0.839872241
Campus Life	Does the school have social activities? How is the social life?	We have so many school club for social activities. You can find all clubs from this link: https://bologna.khas.edu.tr/32/ogrenci-kulupleri .	Yes, the school has social activities. The social life is vibrant and active, with many school clubs available for students to join. You can find all club information on the school website.	0.623223603
Campus Life	What is done on orientation/adaptation days?	You can check out or academic calendar: https://akademiktakvim.khas.edu.tr/islars	On orientation/adaptation days, students can check out the academic calendar to learn more about the school and its activities. They can also join various clubs and organizations.	0.609832048
Financial affairs	Can we make any discount from two different credit cards?	No, there is just one discount. It is in the cash payment then you will have 6% discount.	No, you cannot make any discount from two different credit cards.	1
Financial affairs	How will I know that I have won a ranking scholarship, will it be announced somewhere?	If you get any scholarship, you will receive an e-mail.	If you get any scholarship, you will receive an e-mail.	0.994638681
Financial affairs	Are there any banks that open a branch at the school during registration in order to benefit from the KMH account, which ones?	You can open KMH account in Akbank or DenizBank. Any other types of bank are not allowed.	Answer: You can open KMH account in Akbank or DenizBank. Any other types of bank are not allowed.	0.989722252
Financial affairs	Should we pay the first payment before or after registration?	After registration, you should give your receipt to the financial affairs office.	Answer: After registration, you should give your receipt to the financial affairs office.	1
Financial affairs	Is there a scholarship that can contribute to the tuition fee?	Yes, you can check out this website for detailed information: https://aday.khas.edu.tr/burslar	Yes, you can check out this website for detailed information: https://aday.khas.edu.tr/burslar	0.665926158
Financial affairs	Can I make payment types other than credit card online?	Yes, KMH or cash.	Yes, you can. You can pay with KMH or cash.	0.878010952
Student affairs	Are there any documents I need to physically deliver to the school after online registration?	If there is no missing document uploaded to SPARKS, it is not necessary to come physically, but students who have not completed their high school graduation are required to submit a student certificate.	No. If there is no missing document uploaded to SPARKS, it is not necessary to come physically, but students who have not completed their high school graduation are required to submit a student certificate.	0.852951765
Student affairs	Can I request and get my student ID card online through sparks?	Student ID cards are printed without request and distributed from the Security Affairs Office.	No. Student ID cards are printed without request and distributed from the Security Affairs Office.	0.817390442
Student affairs	Can I get the student ID card immediately on the day I come to enroll in the school?	Student ID cards are given immediately to those who come to the University and register in person, once the procedures are completed.	Yes, you can get the student ID card immediately on the day you come to enroll in the school, once the procedures are completed.	0.700966597
Student affairs	How about double major and minor?	You can apply for double major and minor between 3rd and 6th semester. For double major, your GPA has to be minimum 2.75, for minor, your GPA has to be minimum 2.50.	You can apply for double major and minor programs during the period of 22 August 2023 Tuesday to 31 August 2023 Thursday, with the last day being 31 August 2023 Thursday.	0.988431036
Student affairs	Do you mind if we get the student cards on the day of the preparatory exam?	Since you must have your student ID cards with you when you enter the exam, it is recommended that you get your card beforehand in order to avoid any delays.	Buddy: Since you must have your student ID cards with you when you enter the exam, it is recommended that you get your card beforehand in order to avoid any delays.	0.986096501
Student affairs	How do I do the registration process?	Information on registration procedures is published on the link https://www.khas.edu.tr/kayit-kabul/ .	Buddy: Information on registration procedures is published on the link https://www.khas.edu.tr/kayit-kabul/ .	0.011670927
Student affairs	Is there anything I need to do before come to the school for registration?	After the placement results are announced, the SPARKS Student Information Form must be filled.	No.	0.727580369
English exam	If I start from Track 1, do I have to stay in the extended spring term?	Extended spring term is not obligatory for any student, but it is highly recommended to a Track 1 student to take it.	No, you do not have to stay in the extended spring term if you start from Track 1. If you score above 65% in the overall exam, you can move to Track 2.	1
English exam	What are the questions in the LPPE second phase?	LPPE Stage 2 consists of Listening, Writing, and Reading sections. For further information please visit: https://fl.khas.edu.tr/khas-ingilizce-seviye-tespit-ve-yeterlilik-sinavi/	LPPE Stage 2 consists of Listening, Writing, and Reading sections. For further information please visit: https://fl.khas.edu.tr/khas-ingilizce-seviye-tespit-ve-yeterlilik-sinavi/	1

Table 6.1: This data shows the similarity scores between the real answers and the chatbot answers.

The scores are the output of the semantic similarity algorithm.

7 Conclusion

With the development of technology, labor force is decreasing day by day. This effect can be seen in the chatbot variations mentioned in the literature. In this study, a chatbot is designed at Kadir Has University that students can use whenever they want, using more than one technology with the help of OpenAI API and Llama Index. Also, the main model used in this project, whose published is OpenAI API, is GPT-3. Thanks to these technologies, we used the data we obtained from various offices of the school in our application. Regarding basically, we attempted to enhance the language model—in this particular instance, OpenAI's pretrained model "davinci"—with our own data. As mentioned before, the relatability of the supplied context in relation to a text of strings is evaluated using LLM's embeddings. Therefore, this is how we applied it. Besides, to make a mention of the way the data is used in this project, a file named "index.json" is produced and contains vectors denoting the chunks which divide the content into smaller pieces based on the token size we choose. Hence, each of the chunk and vector models are included in these index files. The key concept is that we may extract the nodes that are most comparable to those that are represented by the query from the user by analyzing our vector storage. Finally, in order for the project to run and appear on the UI, The Fast API-based backend accepts the POST request from the frontend. In this project, an effective chatbot design that uses Open AI API, Llama Index, Vue.js, Fast API, and Search, Clustering, Classification, Recommendations as embeddings, with students who will experience the application, is explained and presented in working condition in the project.

8 Future Works

Thanks to the technologies we have used, the application can be used in real life after a server is rented in the future. Especially its use in real life has created great excitement by the relevant offices of our school. Therefore, our main goal is to use this chatbot in real life, which will alleviate the burden of the student working in the buddy system in the future and can ask the student any question about the school 24/7.

9 REFERENCES

- [1] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques," Third IEEE International Conference on Data Mining, 2003, pp. 427-434, doi: 10.1109/ICDM.2003.1250949.

- [2] Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5), 544-551.

- [3] Adhikari, S. (2020, March). Nlp-based machine learning approaches for text summarization. In 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC) (pp. 535-538). IEEE.

- [4] Lalwani, T., Bhalotia, S., Pal, A., Rathod, V., & Bisen, S. (2018). Implementation of a Chatbot System using AI and NLP. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)* Volume-6, Issue-3.

- [5] Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., ... & Bengio, Y. (2017). A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.

- [6] Ayanouz, S., Abdelhakim, B. A., & Benhmed, M. (2020, March). A smart chatbot architecture based NLP and machine learning for health care assistance. In *Proceedings of the 3rd International Conference on Networking, Information Systems & Security* (pp. 1-6).

- [7] Nithuna, S., & Laseena, C. A. (2020, July). Review on implementation techniques of chatbot. In 2020 International Conference on Communication and Signal Processing (ICCSP) (pp. 0157-0161). IEEE.

- [8] Paul, A., Haque Latif, A., Amin Adnan, F., & Rahman, R. M. (2019). Focused domain contextual AI chatbot framework for resource poor languages. *Journal of Information and Telecommunication*, 3(2), 248-269.
- [9] Qian, H., Li, X., Zhong, H., Guo, Y., Ma, Y., Zhu, Y., ... & Wen, J. R. (2021, July). Pchatbot: A Large-Scale Dataset for Personalized Chatbot. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2470-2477).
- [10] Okonkwo, C. W., & Ade-Ibijola, A. (2021). Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence*, 2, 100033. <https://doi.org/10.1016/j.caeai.2021.100033>
- [11] Patel, N. P., Parikh, D. R., Patel, D. A., & Patel, R. R. (2019). AI and Web-Based Human-Like Interactive University Chatbot (UNIBOT). 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA). <https://doi.org/10.1109/ICECA.2019.8822176>
- [12] Ranoliya, B. R., Raghuwanshi, N., & Singh, S. (2017). Chatbot for university related FAQs. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). <https://doi.org/10.1109/icacci.2017.8126057>
- [13] Tebenkov, E., & Prokhorov, I. (2021). Machine learning algorithms for teaching AI chat bots. *Procedia Computer Science*, 190, 735-744.
- [14] Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning With Applications*, 2, 100006. <https://doi.org/10.1016/j.mlwa.2020.100006>
- [15] Adamopoulou, E., & Moussiades, L. (2020a). An Overview of Chatbot Technology. *IFIP Advances in Information and Communication Technology*, 373–383. https://doi.org/10.1007/978-3-030-49186-4_31
- [16] Tebenkov, E., & Prokhorov, I. (2021). Machine learning algorithms for teaching AI chat bots. *Procedia Computer Science*, 190, 735-744.

[17] Mai, N. E. O. (2022). THE MERLIN PROJECT: MALAYSIAN STUDENTS ACCEPTANCE OF AN AI CHATBOT IN THEIR LEARNING PROCESS. Turkish Online Journal of Distance Education, 23(3), 31-48.

[18] Csaky, R. (2019). Deep learning based chatbot models. arXiv preprint arXiv:1908.08835.

[19] Wang, H., Li, J., Wu, H., Hovy, E., & Sun, Y. (2022). Pre-Trained Language Models and Their Applications. Engineering.

[20] Chaturvedi, A., Srivastava, S., Rai, A., & Cheema, A. S. (2020). A Hybrid Approach to Develop and Integrate Chatbot in Health Informatics Systems. In HEALTHINF (pp. 774-781).

[21] Designing Your Neural Networks". 2019. Medium.<https://towardsdatascience.com/designing-your-neuralnetworks-a5e4617027ed>.

[23] OpenAI API. (n.d.). <https://platform.openai.com/docs/guides/chat/introduction>

[24] OpenAI API. (n.d.-b). <https://platform.openai.com/docs/guides/chat/introduction>

[25] How Each Index Works - LlamaIndex 0.6.13. (n.d.). https://gpt-index.readthedocs.io/en/latest/guides/primer/index_guide.html

10 APPENDIX

//Data Formatting

//Appendix for preparationGeneral.txt file

Student: Everyone says it's hard. Is the preparation difficult?

Buddy: If you work regulary, not it is not hard.

Student: How many levels in preparation?

Buddy: There are TRACK 1, TRACK 2, TRACK 3 and TRACK 4.

Student: What happens if I fail the prep school?

Buddy: You will have to take summer school to pass prep exam.

Student: I studied preparatory school for one year in another school. Do I have to study at Kadir Has University as well?

Buddy: No.

Student: Are there classes on weekends, how many hours per week?

Buddy: No, 20 hours in a week.

Student: How many days per week of preparatory education?

Buddy: 5 days per week.

Student: If I fail the exam and do not skip the exam, will my English improve?

Buddy: Yes definetely, but it is depending on your language knowledge.

Student: How long does the preparation take, will the lessons be face-to-face?

Buddy: Yes, classes are face to face.

Student: Can a student with a Pearson certificate pass the preparation with this document?

Buddy: Yes.

Student: Is my level determined even if I do not take the exam?

Buddy: If you do not take the exam, your level cannot be determined. Then you will start from the beginning.

Student: When does the preparation begin?

Buddy: You can check out this website:

<https://akademiktakvim.khas.edu.tr/lisans>

Student: Do I have a chance to start at the lowest rate without taking the exam directly?

Buddy: Yes.

Student: How is the preparation process, are you satisfied with the teachers?

Buddy: Yes, I really loved the teachers because they always speak English.

Student: What happens when I don't take the preparatory exam?

Buddy: You begin from the lowest degree.

Student: Is the preparation difficult?

Buddy: If you study regularly, no it is not.

Student: How many people are in the classes on average?

Buddy: Maximum 15-20 people at most.

Student: Is it possible to jump from Track 1 and 2 to the Core program?

Buddy: No, you cannot jump from Track 1 and 2 to the Core program. Track 2 is not a exit level in our program.

Student: What are the conditions for exemption from preparatory education?

Buddy: There are several ways to be exempted from Preparatory Education. You can take our university's LPPE in September or provide an English language certificate (TOEFL-IBT, CAE, PTE, etc.) (For more details: (<https://my.khas.edu.tr/uploads/files/mevzuat/yabanci-diller-yuksekokulu-ingilizce-hazirlik-programi-yonergesi4.pdf> -- page 2))

Student: What day is the orientation? (The academic calendar and the dates are different in the form sent to us)

Buddy: Our university organizes orientation for newcomers after the registration dates and before the first lesson in the semester.

Student: If we finish the preparation early, can we go to the faculty?

Buddy: Yes, then you will be irregular student.

Student: If I stay in summer school, do I have to pay a fee, I'm on a 100% scholarship.

Buddy: If you have a full scholarship, you do not have to pay for it.

Student: How much does the summer school cost?

Buddy: It changes every year.

Student: Even if my English is sufficient to pass the exam, will the preparatory program make me more comfortable with English in undergraduate?

Buddy: Yes, it will.

Student: How many tracks are there and how does this system work?

Buddy: There are 5 Tracks in total. In the fall semester, there are Track 1-2-3-4; in the spring semester, there are Track 2-3-4-5.

When you first register, you take an exam and are placed in a class of your level. Then, the process is as follows:

You are placed in Track 1,

If your overall score is above 65%, you jump to Track 3; if between 0-64%, you move to Track 2

You are placed in Track 2,

If your overall score is above 65%, you jump to Track 4; if between 45-64%, you move to Track 3; if between 0-44%, you repeat Track 2

You are placed in Track 3,

If your overall score is above 65%, you jump to Track 5; if between 45-64%, you move to Track 4; if between 0-44%, you repeat Track 3

You are placed in Track 4,

If your overall score is between 45-100%, you move to Track 5; if between 0-44%, you repeat Track 4

Student: Is the exam on paper? Is there any speaking?

Buddy: Yes, it will be on paper for the first stage. Speaking exam is in the second stage.

Student: Will my scholarship be cut?

Buddy: No, it will not be cut.

Student: What happens if we don't pass the placement test?

Buddy: You will take place in the prep class for a year.

Student: Is the preparatory calendar and the school calendar the same?

Buddy: Yes, it is same.

//ALGORITHM OF CHATBOT

Appendix for Code of Ebuddy.py:

```
from llama_index import SimpleDirectoryReader, GPTListIndex, readers,
GPTSimpleVectorIndex, LLMPredictor, PromptHelper, ServiceContext
from langchain import OpenAI
import sys
import os
import openai
'''
```

This python program takes the private external data from the local files and stores them inside a vector store (index.json)

Later compares query embeddings and the most relevant chunk that is looked up to the index file by top_k_score algorithm,

then both the query and the relevant information is sent to the API to get the right answer of user's question.

'''

```
# Authentication Token : sk-
D0pXc8DpYCYZXYRRhhj7T3BlbkFJQCg6HgAY2DVakliqPmuh
```

```
class eBuddy:
```

```
    def construct_index(directory):
```

```
        #OpenAI API KEY is set.
```

```

os.environ["OPENAI_API_KEY"] = "sk-
D0pXc8DpYCYZXyRRhhj7T3BlbkFJQCg6HgAY2DVakliqPmuh"
# set maximum input size
maximum_input_size = 4096
# set number of output tokens
number_of_output_tokens = 2000
# set maximum chunk overlap
maximum_chunk_overlap = 20
# set chunk size limit
chunk_limit = 600

# define prompt helper, prompt helper is passed attributes which defines how the
text in index json file will be splitted, also the input size is set
promptHelper = PromptHelper(maximum_input_size, number_of_output_tokens,
maximum_chunk_overlap, chunk_size_limit=chunk_limit)

# defines LLM, in our case it is OpenAI, some additional arguments are passed
like which pre-trained model will be used
# and the maximum output tokens for the response from the OpenAI API
llmPredictor = LLMPredictor(llm=OpenAI(temperature=0.5, model_name="text-
davinci-003", max_tokens=number_of_output_tokens))

#Can read files into separate documents, loads the private external data from the
input dir.
external_data = SimpleDirectoryReader(directory).load_data()

serviceContext = ServiceContext.from_defaults(llm_predictor=llmPredictor,
prompt_helper=promptHelper)

#creates indexes from documents that are passed if multiple documents are passed
they are combined into one file
#index file contains the chunks, also the vectors of each chunk.
index = GPTSimpleVectorIndex.from_documents(external_data,
service_context=serviceContext)

#saves the index file into local directory
index.save_to_disk('index.json')

return index

#construct_index("/Users/orbiszeus/eBuddy/dataset/LATEST_DATA")

def ask_ebuddy(query):
os.environ["OPENAI_API_KEY"] = "sk-
D0pXc8DpYCYZXyRRhhj7T3BlbkFJQCg6HgAY2DVakliqPmuh"
index =
GPSTSimpleVectorIndex.load_from_disk('/Users/orbiszeus/eBuddy/index.json')

```

```

while True:
    #query = input("What do you want to ask to E-Buddy? ")

    #sends the prompt (question) to OpenAI API with the most relevant chunk from
the index file and the query from the user to get the response.
    response = index.query(query, response_mode="default", verbose=True)

    print("\n\nEbuddy :\n\n" + response.response + "\n\n\n")
    return str(response.response)

#ask_ebuddy()

```

Appendix for Code of REST SERVICE:

```

import ebuddy
from fastapi import FastAPI, Request
from pydantic import BaseModel
from pydantic import BaseModel
from fastapi.middleware.cors import CORSMiddleware

app = FastAPI()

origins = ["*"]
app.add_middleware(
    CORSMiddleware,
    allow_origins=origins,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
class Question(BaseModel):
    question : str

@app.post("/question")
def ask_question_to_ebuddy(question: Question):

    return ebuddy.ask_ebuddy(question.question)

```

Appendix for Code of Vue.js:

```

<template>
  <div class="chatbot-container">
    <div class="chatbot-header-container">
      <div class="chatbot-header">E-Buddy</div>
      
    </div>
    <div class="chatbot-body-container">
      <div class="chatbot-body" ref="chatbotBody">
        <div class="messages" ref="messages">
          <div v-for="(message, index) in messages" :key="index">
            <div v-if="message.isBot" class="bot-message">{{ message.text }}
</div>
            <div v-else class="user-message">{{ message.text }}</div>
          </div>
        </div>
        <div class="scroll-bottom-button" @click="scrollToBottom">
          <i class="fas fa-arrow-circle-down"></i>
        </div>
      </div>
      <div class="input-container">
        <div class="input">
          <input v-model="currentMessage" @keyup.enter="sendMessage"
placeholder="Type a message...">
          <button @click="sendMessage">Send</button>
        </div>
      </div>
    </div>
  </div>
</template>

<script>
import axios from 'axios';

export default {
  data() {
    return {
      messages: [],
      currentMessage: ""
    };
  },
  methods: {
    sendMessage() {
      if (this.currentMessage) {

```

```

    this.messages.push({
      text: this.currentMessage,
      isBot: false
    });

    this.fakeBotResponse(this.currentMessage);
    this.currentMessage = "";
    this.$nextTick(() => {
      this.scrollToBottom();
    });
  }
},
fakeBotResponse(messageFromUser) {
  if (messageFromUser) {
    const data = JSON.stringify({
      question: messageFromUser
    });

    const config = {
      method: 'post',
      maxBodyLength: Infinity,
      url: 'http://localhost:8000/question',
      headers: {
        accept: 'application/json',
        'Content-Type': 'application/json'
      },
      data: data
    };

    axios
      .request(config)
      .then((response) => {
        console.log(JSON.stringify(response.data));

        this.messages.push({
          text: response.data.trim(),
          isBot: true
        });
        this.currentMessage = "";
        this.$nextTick(() => {
          this.scrollToBottom();
        });
      })
      .catch((error) => {
        console.log(error);
      });
  }
},
scrollBottom() {
  const chatbotBody = this.$refs.chatbotBody;

```



```

        chatbotBody.scrollTop = chatbotBody.scrollHeight;
    }
}
};

```

```

</script>

```

```

<style scoped>
@import url('https://fonts.googleapis.com/css2?family=Lato&display=swap');
@import url('https://fonts.cdnfonts.com/css/maximum-impact');
.chatbot-container {
    position: fixed;
    bottom: 0;
    left: 0;
    right: 0;
    height: 100%;
    display: flex;
    flex-direction: column;
}

```

```

.chatbot-header-container {
    font-family: 'Maximum Impact', sans-serif;
    font-size: 25px;
    font-weight: 300;
    line-height: 1.7em;
    height: 50px;
    background-color: #ffffff;
    color: #00529B;
    display: flex;
    justify-content: center;
    align-items: center;
    position: relative;
}

```

```

img {
    margin-bottom: 0px;
    max-width: 15%;
    max-height: 50px;
    position: absolute;
    left: 20px;
}

```

```

.chatbot-body-container {
    flex: 1;
    display: flex;
    flex-direction: column;
    justify-content: flex-end;
    align-items: center;
    position: relative;
}

```

```
.chatbot-body {
  width: 100%;
```

```

  overflow-y: auto;
  position: absolute;
  top: 0;
  left: 0;
  bottom: 100px;
}
```

```
.messages {
  width: 100%;
  padding: 16px;
}
```

```
.bot-message {
  background-color: #87CEFA;
  color: #000000;
  padding: 8px;
  border-radius: 8px;
  margin-bottom: 8px;
  margin-left: 50px;
  position: relative;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
  word-wrap: break-word;
}
```

```
.bot-message::before {
  content: "";
  position: absolute;
  top: 0;
  left: -50px;
  width: 40px;
  height: 40px;
  background-image: url("https://www1.freelogovectors.net/wp-
content/uploads/2020/07/kadir-has-universitesi-logo.png?lossy=1&w=2560&ssl=1");
  background-size: cover;
  border-radius: 50%;
}
```

```
.user-message {
  background-color: #F5F5F5;
  color: #000000;
  padding: 8px;
  border-radius: 8px;
  margin-bottom: 8px;
  position: relative;
```

```

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
    word-wrap: break-word;
}

```

```

.input-container {
    position: absolute;
    bottom: 0;
    width: 100%;
    height: 100px;
    display: flex;
    justify-content: center;
    align-items: center;
    margin-top: -1px;
}

```

```

.input {
    width: 100%;
    max-width: 600px;
    height: 60px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 0 16px;
    background-color: #fff;
    border-top: 1px solid #eee;
    border-radius: 15px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
}

```

```

.input input {
    font-family: 'Lato', sans-serif;
    flex: 1;
    margin-right: 16px;
    padding: 8px;
    border-radius: 8px;
    border: none;
}

```

```

.input button {
    font-family: 'Lato', sans-serif;
    padding: 8px;
    border-radius: 8px;
    border: none;
    background-color: #0693e3;
    color: #fff;
    cursor: pointer;
}

```

```

.input button:hover {
  background-color: #00529B;
}

::-webkit-scrollbar {
  width: 8px;
}
::-webkit-scrollbar-track {
  background: #f1f1f1;
}

::-webkit-scrollbar-thumb {
  background: #888;
  border-radius: 4px;
}

::-webkit-scrollbar-thumb:hover {
  background: #555;
}

.scroll-bottom-button {
  position: fixed;
  bottom: 90px;
  right: 20px;
  z-index: 999;
  width: 40px;
  height: 40px;
  background-color: #0693e3;
  color: #fff;
  border: none;
  border-radius: 50%;
  cursor: pointer;
  font-size: 20px;
  transition: background-color 0.3s ease;
}

.scroll-bottom-button:before {
  content: '\2193';
  display: block;
  text-align: center;
  line-height: 40px;
}

.scroll-bottom-button:hover {
  background-color: #00529B;
}
</style>

```